Received: 14 October 2020 | Revised: 22 December 2020 | Accepted: 27 January 2021

IET Biometrics

DOI: 10.1049/bme2.12025

The Institution of Engineering and Technology WILEY

ORIGINAL RESEARCH PAPER

# Extended IMD2020: a large-scale annotated dataset tailored for detecting manipulated images

Adam Novozámský | Babak Mahdian | Stanislav Saic

The Czech Academy of Sciences, Institute of Information Theory and Automation, Prague, Czechia

**Correspondence**
Adam Novozámský, The Czech Academy of Sciences, Institute of Information Theory and Automation, Prague, Czechia.
Email: novozamsky@utia.cas.cz

**Abstract**

Image forensic datasets need to accommodate a complex diversity of systematic noise and intrinsic image artefacts to prevent any overfitting of learning methods to a small set of camera types or manipulation techniques. Such artefacts are created during the image acquisition as well as the manipulating process itself (e.g., noise due to sensors, interpolation artefacts, etc.). Here, the authors introduce three datasets. First, we identified the majority of camera models on the market. Then, we collected a dataset of 35,000 real images captured by these cameras. We also created the same number of digitally manipulated images. Additionally, we also collected a dataset of 2,000 'real-life' (uncontrolled) manipulated images. They are made by unknown people and downloaded from the Internet. The real versions of these images are also provided. We also manually created binary masks localising the exact manipulated areas of these images. Moreover, we captured a set of 2,759 real images formed by 32 unique cameras (19 different camera models) in a controlled way by ourselves. Here, the processing history of all images is guaranteed. This set includes categorised images of uniform areas as well as natural images that can be used effectively for analysis of the sensor noise.

## 1 | INTRODUCTION

The histories of visual content manipulation and photography run practically in parallel [1]. In modern times, we face a plethora of manipulated images that create significant problems in our society. Advanced image editing techniques have become increasingly accessible in the form of user-friendly editing software and have resulted in manipulated visual content that appears convincingly realistic. Both classic image editing programs and an abundance of apps and software tools now use the latest advances in computer vision, for example generative adversarial networks (GAN) [2]. GAN methods can be used to create a fake but realistic visual content in no time at all. Deepfakes (artificial intelligence generated videos of people doing and saying fictional) are a popular form of such manipulations.

Clearly, we require technologies that permit us to assess the integrity of digital visual media to a reliable degree, yet the limitations of our current forensic technology result in low accuracy in real-life situations. The image forensic community seeks to apply the successes of deep nets in computer vision problems to the difficult problem of detecting manipulated imagery. But, we face a few obstacles while achieving this objective.

A major obstacle is that deep nets require large-scale datasets for training. For image classification, the ImageNet [2] released in 2009, yielded a large-scale annotated dataset containing 1,000 distinct object categories. Fei-Fei Li et al. [3] employed Google Image Search to pre-filter large candidate sets for each category, and Amazon Mechanical Turk crowdsourcing pipeline [4] to manually validate that each image belonged to its assigned category. This large dataset has advanced computer vision and machine learning research and improved the performance accuracy of classification models in relation to earlier methods. Today, the computer vision community benefits from several such publicly available datasets like: UCID [5] and ImageCLEF [6] for image retrieval; PASCAL [7], ImageNet [3], and Microsoft COCO [8] for tasks such as object detection, segmentation and recognition.

The above-mentioned datasets cannot serve the purposes of the forensic image community directly because they were not gathered with forensic research in mind and therefore

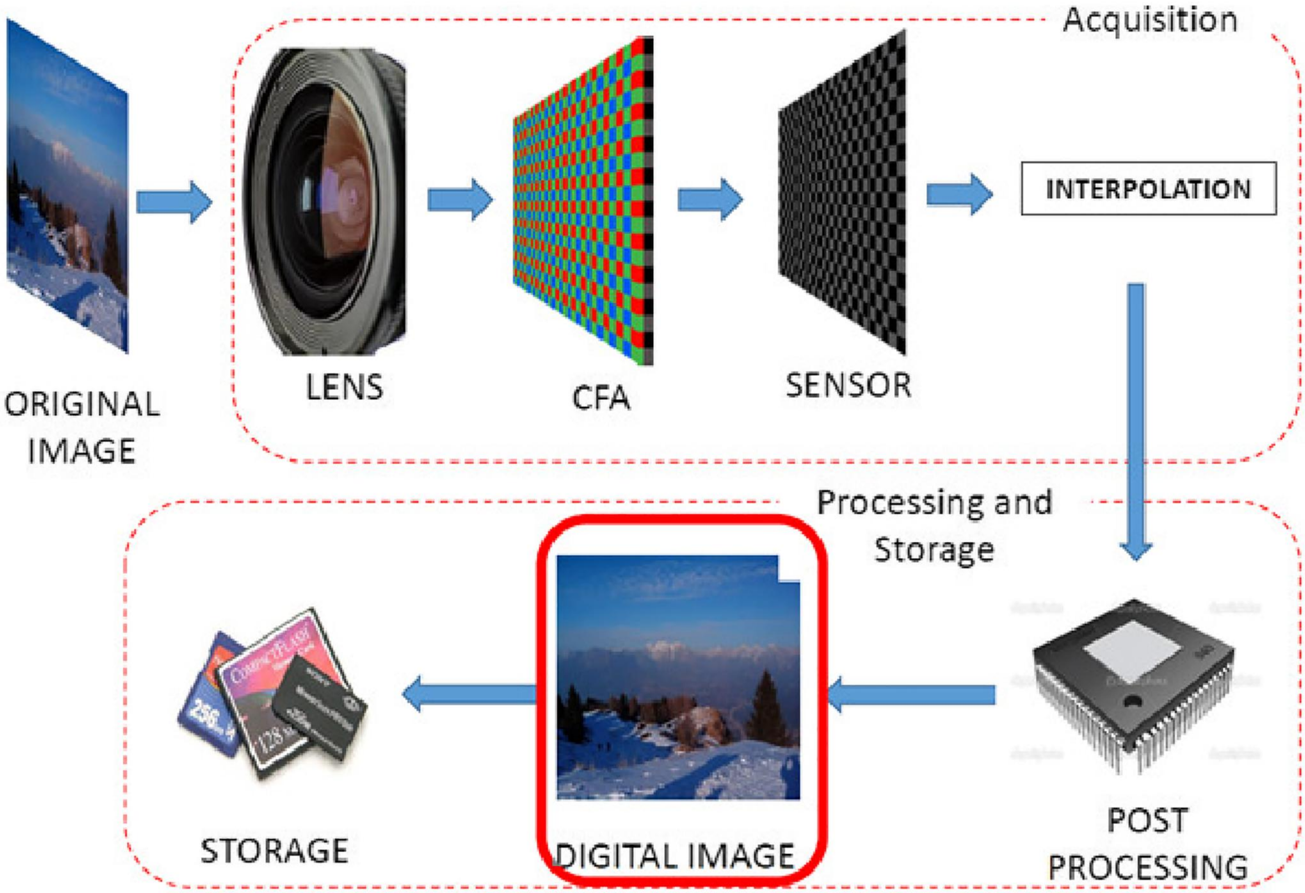

**FIGURE 1** Individual steps and components forming a typical digital image [11]

lack the desired diversity and annotations. To date, most image forensic authors have worked with small datasets that failed to capture the wide, complex image artefacts that appear in the lifecycle of real-life images. As a result, these methods fail in cross-data testing and generalisation. Some authors have tried to solve this problem by training their methods using only real images (e.g. [9]); others have tried to build internal limited datasets (e.g. [10]) and focus on domain adaptation.

The aim of the authors is to introduce a large, annotated dataset for detecting manipulated visual content. Inspired by the semi-automatic way that ImageNet has been built, we will build in a semi-automatic way a dataset that captures a large diversity of image and manipulation artefacts. This is a challenging task. Each camera brings into the image different kinds of artefacts. Some artefacts are unique to particular camera device and some are unique to camera model. A range of compression levels brings a range of quantization noise into the visual content. Different manipulation techniques yield different editing traces. In general, we can categorise intrinsic artefacts in visual content into three groups: (i) acquisition artefacts, see Figure 1 (e.g. sensor noise, demosaicking algorithms or gamma correction); (ii) format artefacts (e.g. JPEG and quantization noise); and (iii) manipulation artefacts (e.g. artefacts left by GAN in the image).

The artefacts mentioned above are essential to create image/video forensic methods. In fact, forensic methods that are based on high-pass filters and their resulting noise residuals, often seek to eliminate the image content to emphasise these intrinsic artefacts and so expose traces of image manipulation. Although the above-mentioned artefacts are often invisible by naked eye, dataset with lack of a high variety in them might result in overfitting of learning methods to a narrow set of cameras or types of manipulations causing those methods to perform poorly on new and unseen manipulations (e.g. [10]).

## 1.1 | Contribution

Extended IMD2020 introduces three datasets. The first dataset consists of 35,000 real images captured by 2,322 different camera models. These camera models form the majority of existing cameras in the market. The dataset provides a rich and diverse set of sensor noise—artefacts that various imaging software embedded in cameras bring into images—and compression artefacts. Moreover, we also synthetically created a set of manipulated images using a large variety of manipulation operations including core image processing techniques as well as advanced methods based on GAN or Inpanting. This resulted in 70,000 images in total. In addition to this dataset, we also downloaded 2,000 'real-life' (uncontrolled) manipulated images created by random people from Internet. Real versions of these images also are also provided. Binary masks localising the manipulated areas have been created manually. The last part of the datasets consists of 2,759 real images formed by 19 camera models in a controlled way by ourselves. To this end, we used 32 different cameras. The processing history of all images is guaranteed. This set also includes images of uniform areas that can be used for analysis of sensor noise as well as other camera-dependent artefacts.

The dataset contributes to facilitating future research in: (i) classification of the manipulated image and localisation of the manipulated area; (ii) source camera identification and sensor noise (e.g. PRNU (photo response non-uniformity) analysis; and (iii) reverse search of visual content (the dataset includes tens of thousands of near-duplicates in the form of real and manipulated versions of the same image that can serve for train and test needs of image search engine).

In addition to the dataset, the authors study intrinsic artefacts in images and empirically demonstrate their presence. Also it provides a comprehensive review of existing image forensic datasets. Moreover, the authors bring a survey of existing CNN (convolutional neural network)-based methods for detecting image manipulation.

The work of the authors is organised as follows. Section 2 breaks down digital manipulation into different categories. The subsequent sections summarises artefacts brought into the image during their lifecycle. After this, we introduce previously published datasets and papers related to the topic discussed here. In Section 5, the dataset is introduced in details. The following section after Section 5 includes experimental results and the last section summarises the work that has been performed by the authors.

## 2 | TYPES OF MANIPULATION

Any kind of operation applied to an image or video that cause the visual content differs from its authentic version is a digital manipulation. However, there are types of image processing methods, such as rotation, down-sizing, application of global filters on images that manipulate the information represented by visual content in a very limited way. Therefore, today image forensic methods are rather interested in detection of visual contents manipulated in a malicious way.

There are three major types of malicious manipulation of digital images: (i) copy-paste (copying an area from the same image and pasting it to a different area of the same image); (ii) splicing (the manipulated image is created by combination of two or more images.) and (iii) and re-touching (locally editing an area of the image). Different types of malicious manipulations that can be applied to an image are shown in Figure 2. Such manipulations can be achieved by using basic image processing techniques, as well as advanced methods based, for instance, on GAN.

## 3 | ARTEFACTS BROUGHT INTO IMAGES IN THEIR LIFECYCLE

The journey of a digital image can be represented as a composition of several steps: (i) acquisition; (ii) coding and digital editing [11]. For the sake of simplicity, we model the image acquisition process in the following way:

$$I_{i,j} = I^o_{i,j} + I^o_{i,j} \cdot \Gamma_{i,j} + \Upsilon_{i,j} \tag{1}$$

Here, $I_{i,j}$ denotes the image pixel at position $(i, j)$ produced by the camera, $I^o_{i,j}$ denotes the noise-free image (perfect image of the scene), $\Gamma_{i,j}$ is the multiplicative noise, such as PRNU and $\Upsilon_{i,j}$ stands for all additive noise components.

The following sections briefly describe the major types of artefacts brought into images during the acquisition process and in their later stages of the lifecycle.

### 3.1 | Artefacts associated with acquisition devices

Digital image acquisition devices introduce intrinsic artefacts or fingerprints in the final visual content output through their various components.

When an image is captured, the light from the actual scene is focused through the camera's optical system onto its sensor (usually CCD or CMOS). The sensor's pixels collect photons and convert these into voltages that are then sampled by a digital signal in an A/D converter. Before reaching the sensor, however, the light is usually filtered by the colour filter array (CFA). The CFA is a mosaic of tiny colour filters placed over pixels of the image sensor to capture particular colour information. The CFA is necessary because typical consumer cameras have a single sensor which is not capable of separating colour information. Each pixel captures only one main colour (red, green, or blue). During the demosaicking process, the sensor output is interpolated to produce the digital colour image [11]. The subsequent signal is then processed again for colour correction and white balance adjustment. Additional processing includes gamma correction to adjust for the linear response of the imaging sensor, noise reduction and filtering operations to visually enhance the image.

Among the artefacts we have mentioned, some are unique to the specific camera sensor, and others are common to all cameras sharing a model number or brand by virtue of the embedded software they share. For example, a specific image sensor will produce a unique pattern noise. As stated in [12], taking a photo of a uniform scene will still produce a digital image that exhibits variations in the intensity of the individual pixels, which is partly due to the pattern, readout or shot noise. Authors have used sensor pattern noise to identify the exact camera that captured an image [13]. To this end, typically, PRNU , a unique part of the sensor pattern noise has been used (the multiplicative component of Equation (1)): Figure 3 shows sensor pattern noise of two different cameras capturing the same scene, as apparent sensor

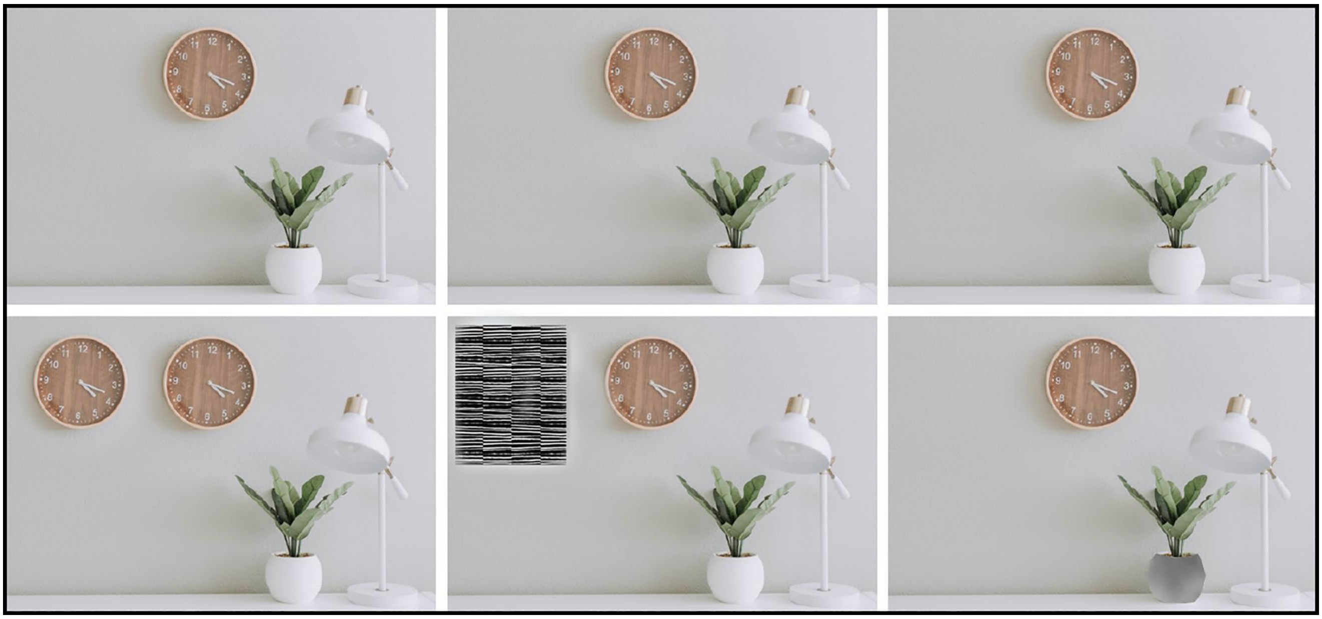

**FIGURE 2** Types of image manipulation. On the left copy-paste is shown, in the middle splicing is shown and on the right an example of a re-touching operation is demonstrated

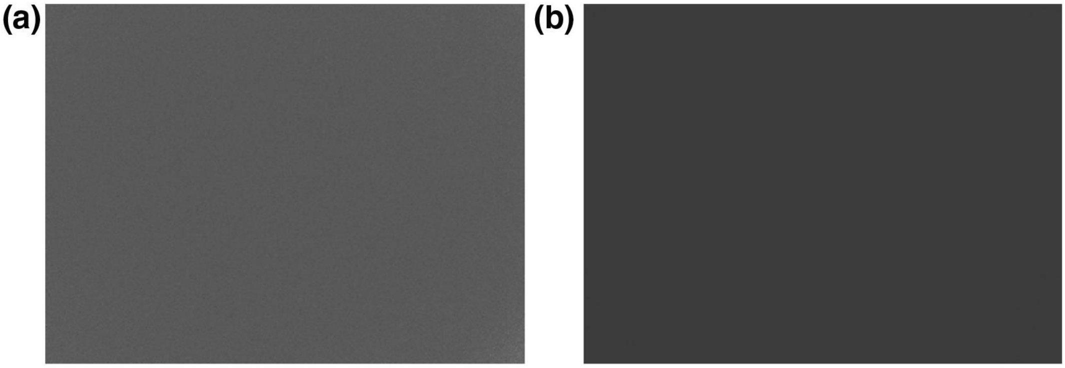

**FIGURE 3** (a) The extracted sensor pattern noise of a Nikon Coolpix L23 device is shown and (b) shows the same for Canon Powershot A495. Note that the apparent sensor noise of these two cameras differ

noise of these two cameras differ. A light uniform scene with minimal number of edges that enables a more accurate extraction and modelling of the sensor noise have been used [13].

If we examine the demosaicking process on the other hand, we will find it is typically identical for all cameras belonging to the same model (since they share common embedded software and the same demosaicking algorithm). For example, Mahdian et al [14] shows that these interpolation techniques often bring into the image invisible periodic artefacts.

### 3.2 | Artefacts associated with lossy compression

The output of the camera is typically compressed and stored in JPEG which is the most commonly used image format. In JPEG, the image is first converted from RGB to YCbCr, consisting of one luminance component ($Y$) and two chrominance components (Cb and Cr). Mostly, the resolution of the chroma components is reduced (usually by a factor of two). Then each component is split into adjacent blocks of 8 × 8 pixels. Each block of each of the Y, Cb and Cr components undergoes a discrete cosine transform ($DCT$). Let $f(x, y)$ denote a pixel $(x, y)$ of an 8 × 8 block. Its $DCT$ is:

$$F(u,v) = \frac{1}{4}C(u)C(v) \sum_{x=0}^{7}\sum_{y=0}^{7} f(x,y)\cos\frac{(2x+1)u\pi}{16}\cos\frac{(2y+1)v\pi}{16},$$

where $u, v \in \{0 \cdots 7\}$; $C(u), C(v) = 1/\sqrt{2}$ for $u, v = 0$; otherwise $C(u)$, $C(v) = 1$.

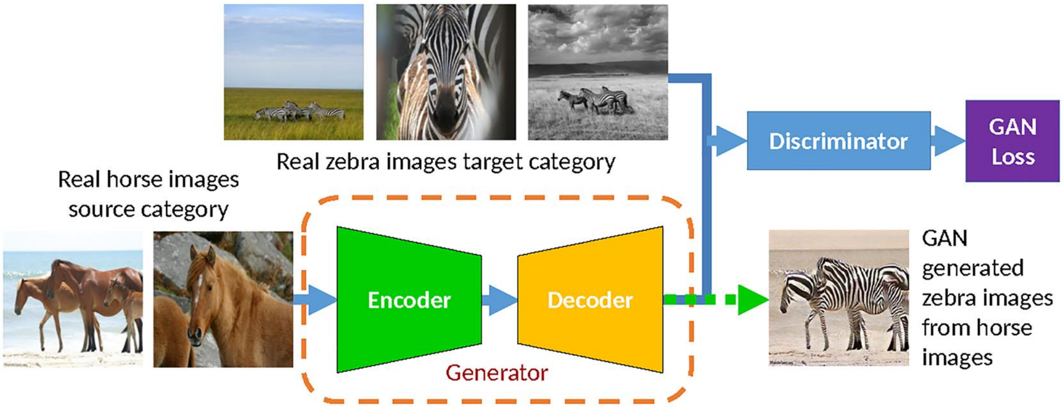
Real zebra images target category
Discriminator
GAN Loss
Real horse images source category
Encoder
Decoder
Generator
GAN generated zebra images from horse images

**FIGURE 4** Typical pipeline of image2image translation [19]

In the next step, all 64 $F(u, v)$ coefficients are quantized. The quantization step is given by a 64-element quantization table ($QT$):

$$F^{QT}(u,v) = \text{round}\left(\frac{F(u,v)}{QT(u,v)}\right), \quad u,v \in \{0 \cdots 7\}$$

where $QT(u, v)$ defines the quantization step for each $DCT$ frequency $u$ and $v$. Commonly, there is one $QT$ for $Y$ and another single $QT$ for both $Cb$ and $Cr$.

Quantization tables determine the quantization rate (compression rate). They bring into the image quantization noise and blocking artefacts that are typical for JPEG compressed images. Therefore an image forensic dataset should ideally cover a wide range of quantization tables (compression rates) to avoid overfitting of learning methods to specific kinds of JPEG artefacts and compression levels.

### 3.3 | Artefacts associated with various types of manipulation

Different image editing can be applied to an image during its life. This includes simple operations such as geometric transformation (rotation, scaling etc.), blurring, sharpening or more advanced and possibly malicious changes such as image splicing or cloning (copy-move), inpainting operations (e.g. [15, 16]), or GAN (e.g. Cycle-GAN [17] or Style-GAN [18]). Obviously, image forensic community is mainly focused on detecting malicious types of manipulations. There are three major types such manipulation: (i) copy-paste (copying an area from the same image and pasting it to a different area of the same image); (ii) splicing (the manipulated image is created by combination of two or more images.); and (iii) and re-touching (locally editing an area of the image).

All such manipulations leave characteristic traces in the image. For instance, authors have noticed that GAN-based methods also leave distinct invisible artefacts in the image (e.g. [19]). There are two main components in GAN: discriminator and generator. The discriminator tries to distinguish real images of the target category from those generated by the generator. On the other hand, the generator takes an image of the source category as input and tries to generate an image similar to images of the target category and making them indistinguishable by the discriminator. Looking ar more details to the GAN pipeline (e.g. Figure 4) we can notice that typically generator contains two components: encoder and decoder.

The encoder contains a few down-sampling layers which aim to extract high-level information from the input image and generate a low-resolution feature tensor. The decoder, on the other hand contains a few up-sampling layers which take the low resolution feature tensor as the input and a high-resolution image as the output. According to Zhang et al. [19], although the structures of the GAN models are quite diverse, the up-sampling modules used in different GAN models are consistent. The up-sampling bring into the image specific artefacts (e.g. interpolation based [14]). Zhang et al. [19] addressed these up-sampling related artefacts and used them to detect GAN-based images. They showed that they are present in most of the commonly used GAN methods.

To summarise the work performed till now, we can say that a well-designed forensic dataset should capture changes brought into images by variety of acquisition devices, compression levels and types of manipulations. As pointed out some of these artefacts are unique per each particular camera (i.e. sensor), and some of them are unique per camera brand or model or software editor (e.g. demosaicking algorithm or JPEG compression parameters).

## 4 | RELATED WORKS

This section focuses on reviewing existing datasets as well as CNN-based methods dealing with detection of image and video manipulation.

### 4.1 | Related datasets

The work performed by authors here is an extended version of [20]. In addition to [20], the authors are introducing an innovative part into the dataset consisting of 2,759 real images formed by 32 unique cameras (19 different camera models). They have been captured manually in a controlled way by ourselves and so their processing history is guaranteed. Both images of uniform areas as well as natural images have been captured. This enables effective analysis of sensor noise as well as other camera-dependent artefacts. In the experimental part of our work, this resourceful part of the dataset is used to demonstrate the presence of hidden camera-dependent artefacts in images.

The CoMoFoD dataset [21] has been designed for copy-move forgery detection. It consists of 260 forged images in two categories of small (512×512 pixels), and large (3000 × 2000 pixels). Each set includes a forged image, mask of the manipulated area and its original image. Images are divided into five groups according to applied manipulation: translation, rotation, scaling, combination and distortion etc. The MICC-F220 and MICC-F2000 [22] are another dataset focused on copy-paste. MICC-F220 is formed by 220 images: 110 are tampered images and 110 are originals. The resolution varies from 722 × 480 to 800 × 600 pixels. The Columbia spliced image database [23] has two parts. First, a grayscale image dataset with 933 authentic and 912 spliced grayscale image blocks, and a colour image dataset with 183 authentic uncompressed colour block images and 180 spliced uncompressed colour block images.

CASIA Image Tampering Detection Evaluation Database [24] is an image forensics dataset that focused on splicing. CASIA v1.0 has 800 authentic and 921 spliced 384×256 images. CASIA v2.0 contains 7,491 authentic and 5,123 tampered images. The First Image Forensics Challenge [25] collected thousands of images of various scenes, both indoors and outdoors. The dataset served for an international competition organised by the IEEE Information Forensics and Security Technical Committee and comprises of a total of 1,176 forged images. Wen et al. [26] introduced a small dataset called Coverage designed for copy-paste detection. The REWIND (REVerse engineering of audio-VIsual coNtent Data) [27] dataset contains 142 handmade manipulated images for the evaluation of image tampering detectors. Half of the images are original; the other half is a set of hand-made forgeries. There are also 4800 automatically manipulated images. Barni et al. [28] created a small dataset for detecting cut and paste splicing (ISCAS). Zhou et al. created a dataset of manipulated faces [29] by using FaceSwap [30] and SwapMe [31]. There are 1005 tampered images for each tampering technique (2010 tampered images in total) and 1400 authentic images for each subset. Realistic Tampering Dataset [32] proposes a dataset of realistic forgeries created manually by using editors such a GIMP and Affinity Photo. The National Institute of Standards and Technology (NIST) was presented with a large benchmark dataset—Nimble Challenge 2017 [33]. This dataset contains a total of 2,520 manipulated images. Moreover, NIST also has published additional datasets MFC2018 and MFC2019 [33] in subsequent years.

Most of the currently published datasets (see Table 1) are limited in size, acquisition device variety, content, attacks type and compression/post processing variety. Typically, they are created in a controlled environment.

## 4.2 | State-of-the-art methods

Early image forensic methods used hand-crafted features to detect individual types of manipulation. These traditional methods typically aim to detect some targeted inconsistencies among pixels. For example, Farid et al. [34] designed a method to detect composites created from JPEG images of varying quality. This method determines whether a section of the image was initially compressed more, to produce a lower quality than the rest of the image. In [35], Hany Farid described the specific correlations brought by the CFA interpolation into the image and proposed a method capable of detecting their inconsistency across the image.

Mahdian et al. [36] used estimates of local noise variance using wavelet transform to detect local image noise inconsistencies. Weiqi Luo et al. [37] used JPEG blocking artefact characteristics to detect recompressed image blocks. Wei Wang et al. [38] utilised grey level co-occurrence matrix (GLCM) of thresholded edge image of image chroma as an image splicing detection method. Sevinc Bayram et al. [39] used Fourier-Mellin transform to propose a clone detector. The Fourier-Mellin transform does not vary with respect to scale and rotation which permits stronger performance of the method when confronted with cloned areas that have been resized and rotated. In [40] a range of classic image forensic methods can be viewed.

### 4.2.1 | CNN-based image forensic methods

Deep neural networks have shown to be very effective in various image processing tasks and computer vision so there is no surprise that the image forensic community also has shifted

**TABLE 1** Examples of datasets designed for image manipulation detection

| Dataset | Size | Binary mask |
|---|---|---|
| CoMoFoD dataset [21] | 260 | Yes |
| MICC-F220, MICC-F2000 [22] | 2,200 | No |
| Columbia [23] | 1,845 | No |
| CASIA [24] | 1,721 | No |
| CASIA v2.0 [24] | 12,323 | No |
| REWIND Real [27] | 142 | Yes |
| Zhou et al. [29] | 3,410 | No |
| Nimble Challenge 2017 (manipulated) [33] | 2,520 | Yes |
| ISCAS [28] | 20 | No |
| Realistic Tampering [32] | 440 | Yes |
| Coverage [26] | 100 | Yes |
| IMD2020 Synthetically Created (proposed) | 70,000 | Yes |
| IMD2020 Manually Created (proposed) | 2,000 | Yes |
| IMD2020 Guaranteed dataset | 2,759 | Yes |

its direction to utilise achievements of deep learning. In [41], Ghosh et al., assume that the spliced and host regions come from different camera-models and segment these regions using a Gaussian-mixture model. They learn high pass rich filters using constrained CNNs that compute residuals, highlighting low-level information over the semantics of the image. In [42], Bunk et al. used resampling features computed on overlapping image patches that are passed through a long short-term memory (LSTM) based network for classification and localisation of manipulation. In [43], Wu et al. introduced a novel deep neural architecture for image copy-move forgery detection. The method is based on a two-branch architecture followed by a fusion module. The two branches localise potential manipulation areas using visual discontinuities and copy-move regions via visual similarities, respectively.

In [44], Zhang et al. used information of chrominance and saturation channels to develop a shallow convolutional neural network (SCNN) that learned to detect doctored areas in in low-resolution images. To this end, boundaries of modified areas have been used. In [10], Cozzolino et al. demonstrate limited generalisation capability of underlying CNN. They showed that CNN learn features that are highly discriminatory for the given dataset but lack of generalisation resulting in inaccurate results of today's CNN-based methods when performed in cross-dataset test scenarios. To avoid the underlying CNN to overfit to manipulation-specific, they introduced forensic-transfer (FT). They learn a forensic embedding based on an auto-encoder based architecture [45] that can be used to distinguish between real and fake imagery. An unseen manipulated image will be detected as fake if it gets mapped sufficiently far away from the cluster of real images. The authors show that only a few training samples of the target domain of tampering enable to finetune their model to achieve high accuracies.

In order to detect GAN generated images, in [46], Yu et al. used GAN-based fingerprints in order to use them to classify an image as real or GAN-generated. Their experiments show that even a small difference in GAN training (e.g. the difference in initialisation) can leave a distinct fingerprint that commonly exists over all its generated images. To avoid learning the semantic information in the image, in [47], Kim et al. used a deep learning approach that utilises a high-pass filter to acquire hidden features in the image. In [48], Mazaheri et al. developed an encoder-decoder based network. They assume that manipulated images commonly leave some traces near boundaries of manipulated areas such as blurred edges. In order to detect forgeries, they use representations from early layers in the encoder. In [49], Bappy et al. used manipulation localisation architecture which utilises resampling features, LSTM cells, and encoder-decoder network to segment manipulated areas of the image. Resampling features are used to capture artefacts like JPEG quality loss, up-sampling, downsampling, rotation, and shearing. In another work [50], Bappy et al. assumed manipulated areas often exhibit discriminative features in boundaries shared with neighbouring non-manipulated pixels. They focused on these characteristics and developed a unified framework for joint patch classification and segmentation to localise manipulated regions from an image. The proposed method learns the boundary discrepancy, that is, the spatial structure, between manipulated and non-manipulated regions with the combination of LSTM and convolution layers.

In [51], Zhou et al. realised that they can use multiple modalities as the input to their CNN to increase the accuracy. They proposed a network using the RGB information. In addition to this, they also added a noise stream to the architecture. Authors observed that the fusion of the two streams leads to learning effective and rich features and higher accuracy. In [52], Rao et al. focused on eliminating the complex image content to detect manipulation. This enables them to achieve a faster time to accuracy when training the underlying CNN. Specifically, weights at the first layer of their network are initialised with the 30 basic high-pass filters used in spatial rich model for image steganalysis. The results obtained are promising. In [53], Cun et al. instead of classifying the spliced region by a local patch, authors leveraged the features from whole image and local patch together, calling this structure a semi-global network. Furthernore, the work of Cozzolino et al. focused on eliminating the image content as proposed in [54]. Here, authors proposed a deep learning method to extract a noise residual, called noiseprint, where the image content is removed. Results shown in the paper signify this direction is promising in forgery localisation.

In [55], Bondi et al. proposed a method leveraging characteristic footprints left on images by different camera models. The rationale behind the method is that all pixels of pristine images should be detected as being shot with a single device. By contrast to such images, if a picture is obtained through image composition, traces of multiple devices can be detected. In [56], Bayar et al. have developed a new type of CNN layer called a constrained convolutional layer that is able to jointly suppress an image's content and adaptively learn manipulation detection features. Through a series of experiments, they show that the proposed constrained CNN is able to learn manipulation detection features directly from data and outperforms the existing state-of-the-art general purpose manipulation detectors. In [57], Liu et al. proposed to utilise CNNs and the segmentation-based multi-scale analysis to locate tampered areas in digital images. The authors observed that exploiting the benefits of the small scale and large-scale analyses, the segmentation-based multiscale analysis can lead to a performance leap in forgery localisation of CNNs.

In [58], Salloum et al. proposed a method based on VGG-16 which is a fully convolutional network (FCN). The authors introduced several modifications such as batch normalisation layers and class weighting to train VGG-19 to localise image-splicing attacks. They demonstrate improvement in comparison to the state-of-the-art methods. In [9], Huh et al. proposed an algorithm that uses the automatically recorded photo EXIF metadata. EXIF stands for Exchangeable Image File Format and typically it is embedded into JPEG files by the camera. EXIF can include date, time, camera settings etc. The authors used EXF to train a model to determine whether an image is self-consistent. In other words, whether its content could have been produced by a single imaging pipeline. The method demonstrated superior results in comparison to other existing ones.

In [59], Le-Tien et al. proposed a low computational-cost and fully connected neural network to address the problem of image forgery detection. In [60], Bayar et al. tried to prevent the CNN from learning features that represent an image's content. They proposed a new covolutuional form specifically designed to suppress an image's content and learn manipulation detection features. In [61], Wu et al. showed that both image splicing detection as well as localisation can be jointly solved using a multitask network in an end-to-end manner. In [62], Marra et al. attempt to avoid downsizing of images before analysing them by CNNs. They propose a CNN-based image forgery detection framework which makes decisions based on full-resolution information gathered from the whole image.

In [19], Zhang et al. proposed a GAN simulator, which can simulate the artefacts produced by the common pipeline shared by several popular GAN models. They identified a unique artefact caused by the up-sampling component included in the common GAN pipeline. Without seeing the fake images produced by the targeted GAN models during training, the approach achieves a state-of-the-art performances on detecting fake images generated by the popular GAN models. In [63], Marra et al. observed that XceptionNet is capable to achieve superior accuracy in detecting image manipulation. For instance, authors demonstrate that this network accurately detects GAN-generated fake images that are published on social networks. To achieve this conclusion, authors studied the performance of various image forgery detectors against image-to-image translation, both in ideal conditions, and in the presence of high compression, routinely performed upon uploading on social networks. The winning architecture was XceptionNet. Another promising image manipulation detectors based on CNN was proposed in [64] by Wu et al., called ManTra-Net. ManTra-Net performs both detection and localisation. The network handles images of arbitrary sizes and various types of manipulation such as splicing, copy-move, removal, enhancement etc. (they learn robust image manipulation traces from 385 image manipulation types). In [64], authors formulated the forgery localisation problem as a local anomaly detection problem. The method extracts image manipulation trace features for a testing image, and identifies anomalous regions by assessing how different a local feature is from its reference features. They demonstrated a good improvement over the existing methods.

## 5 | THE EXTENDED IMD2020 DATASET

Image forensic methods often eliminate the image content and analyse the underlying (hidden) noise/artefacts component of the image to find inconsistencies. As pointed out earlier, some of the intrinsic artefacts are unique to sensor/camera and some others shared by images captured by cameras of the same brand/model.

### 5.1 | Flickr-based images

To prevent possible overfitting to a narrow range of camera models, we collected a list of the majority of camera models existing in the market. Subsequently, we searched for images captured by these devices on Flickr (Flickr enables a search based on camera information included in metadata). If available, 30 real images per camera model have been downloaded.

Most Flickr users are unlikely to publish maliciously manipulated visual content, but Flickr itself cannot guarantee and exactly identify the source of its images. The processing history of these images remains unknown. So, to reduce potential risk, we manually reviewed ('cleaned') all the images and eliminated those with obvious signs of digital manipulation. We were left with a set of 35,000 real images, some of which are shown in Figure 5. The top ten popular camera brands represented in Flickr were Apple (iPhone 7 etc.), Canon (EOS 5D Mark III etc.), Nikon (D750 etc.), Sony (ILCE-7M3 etc.), Fujifilm (X-T2 etc.), Samsung (Galaxy S etc.), Olympus (E-M1MiarkII etc.), Panasonic (DMC-FZ1000 etc.), Google (Pixel 3 etc.), and Leica (Camera AG Q etc.).

We also generated a same number of synthetically manipulated images using various methods. As pointed our earlier, advanced techniques such as GAN often bring characteristic artefacts into images [46]. Such kinds of artefacts might lead to overfitting of learning methods. This has also been empirically confirmed by Cozzolino et al. [10] where authors experimentally demonstrated CNN-based approaches for image forgery detection tend to overfit to the source training data and perform poorly on new and unseen manipulations. Therefore, to manipulate images we also used a high variety of core image processing techniques.

Specifically, a random area of a random shape of images has been manipulated, using one of the following types of manipulations: copy-paste, splicing and re-touching. Size of the manipulated area has been randomly selected to be from 5% to 30% of the image. Additionally, a random combination of image processing operations has been applied on the manipulated area. These operations are based on JPEG (random compression level), blurring (various kernels), contrast manipulation, various types of noise and resampling and interpolation using bilinear and bicubic kernels. About half of the images have been manipulated in this way. Some examples of such manipulated images are shown in Figure 6.

To synthetically manipulate the second half, we used advanced methods such as GAN or Inpainting. Specifically, the following methods have been used to manipulate images: built-in OpenCV inpainting function, inpainting method proposed in [16], and FaceApp [65] which is currently one of the most popular face manipulation mobile applications based on GAN in iOS and Android. Some examples of such manipulated images are shown in Figures 7 and 8.

To summarise, this dataset is formed by 70,000 images. Half of them are real and the second half has been manipulated in a controlled manner. Binary masks of all manipulated images localising the manipulated areas are also provided.

**FIGURE 5** Some examples of real pictures in our dataset

## 5.2 | Real-life manipulated images

We also collected a large set of real-life (uncontrolled) manipulated images from the Internet (for example, see Figure 9). Specifically, 2000 manipulated images created by random people have been downloaded (URL of most images were obtained from [66]). For all of the manipulated images, we also downloaded their real versions. Binary masks localizing the manipulated areas for all manipulated images have been created manually. Some examples of this dataset are shown in Figures 9 and 10.

## 5.3 | Guaranteed set of real images

In addition to above-mentioned data, we also created a set of real images captured by ourselves so their processing history is guaranteed. To collect this set, we used 32 unique cameras (19 different camera models). Table 2 shows cameras used and corresponding number of images acquired by each camera.

Using each camera, we captured images of natural scenes (for example, see Figure 11 (a)) as well as images of a uniform light scene with minimal number of edges (for example, see Figure 11 (b)). Images of uniform scenes enable an easier and more accurate estimation of the sensor noise and PRNU [13].

### 5.3.1 | Estimating camera sensor noise

As pointed out earlier, cameras bring into images different kinds of artefacts. Some artefacts are unique to the particular camera device and some are unique to camera model. For example, the demosaicking process which brings into the image specific hidden changes [14] is typically identical for all cameras of the same model (assuming these cameras use the same embedded software and demosaicking algorithm). On the other hand, the sensor pattern noise has that been widely studied by authors to identify the exact camera that captured the image is unique per camera.

To design an experiment that will demonstrate the presence of artefacts unique per camera as well as unique per camera model, let us to briefly point out the typical procedure of examining whether a digital image under investigation has been captured by an exact camera.

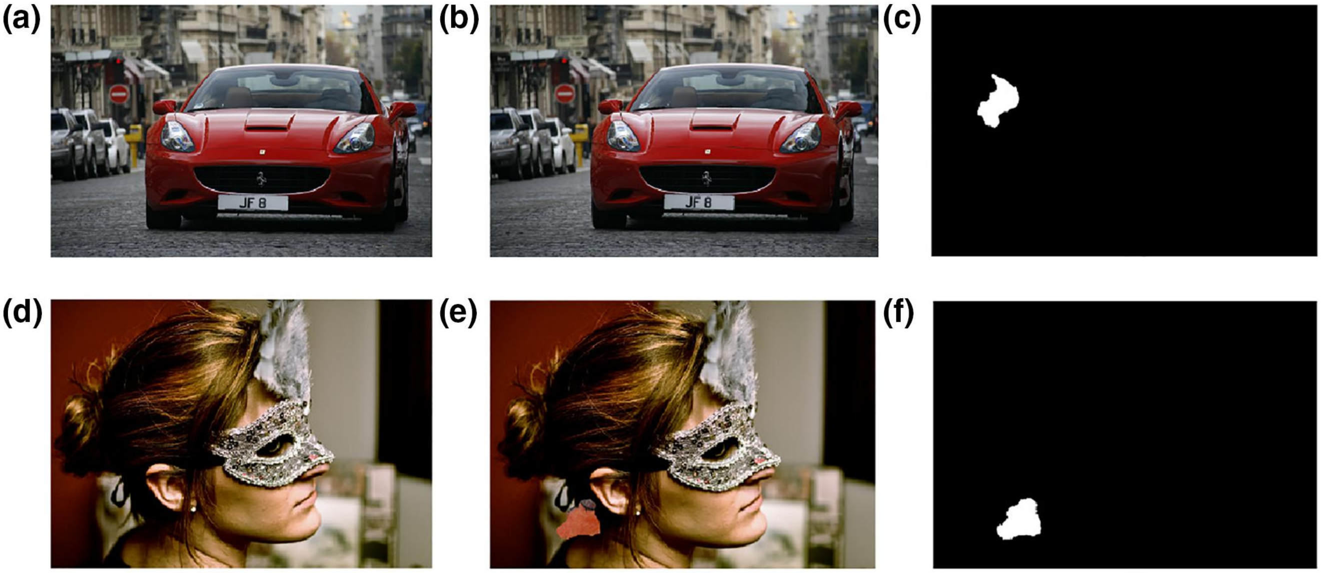

**FIGURE 6** A few samples from the synthetically generated dataset. On left is shown the real image, in middle the manipulated image (JPEG and noise used) and on right the binary mask localising the manipulated area. Sometimes the manipulated area is not visible by naked eye (e.g. (b))

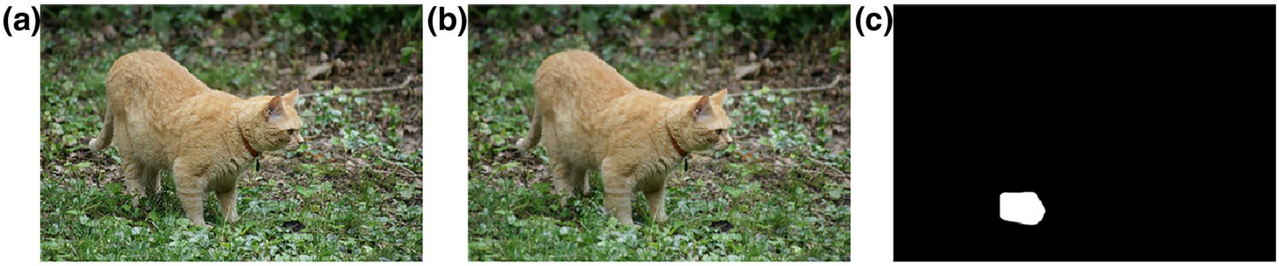

**FIGURE 7** On the left the real image is shown, in the middle the manipulated image (using an inpainting method [16]) and on the right the binary mask localising the manipulated area

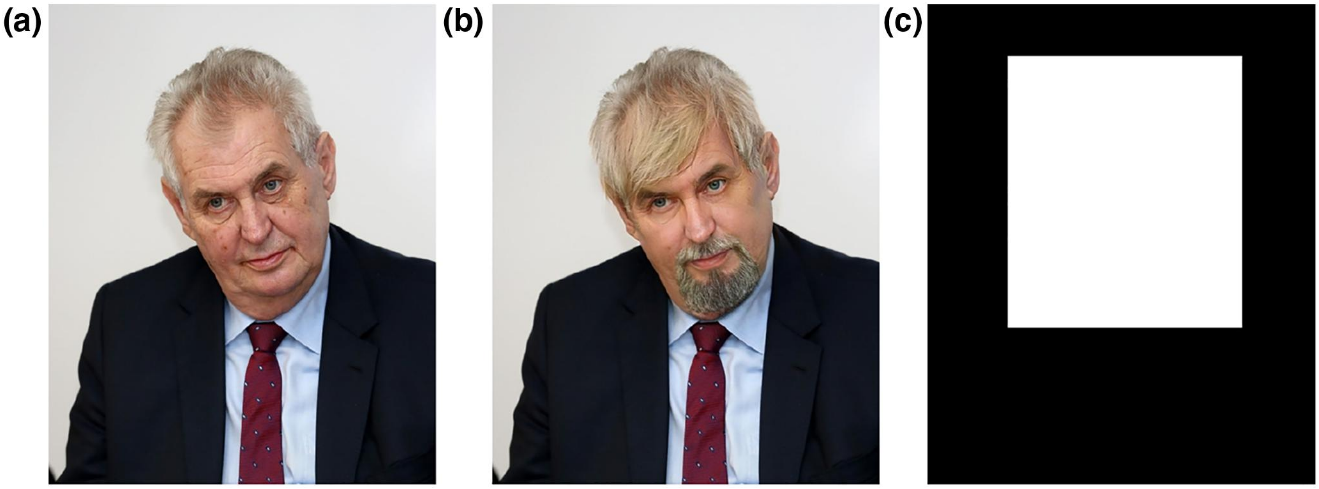

**FIGURE 8** On the left he real image is shown, in the middle the manipulated image (using FaceApp [65]) and on the right the binary mask localizing the manipulated area. It is interesting to note that although the visible area of manipulation of FaceApp is typically inside the face area, pixels of a larger rectangular area around the face gets modified as a result of face transform

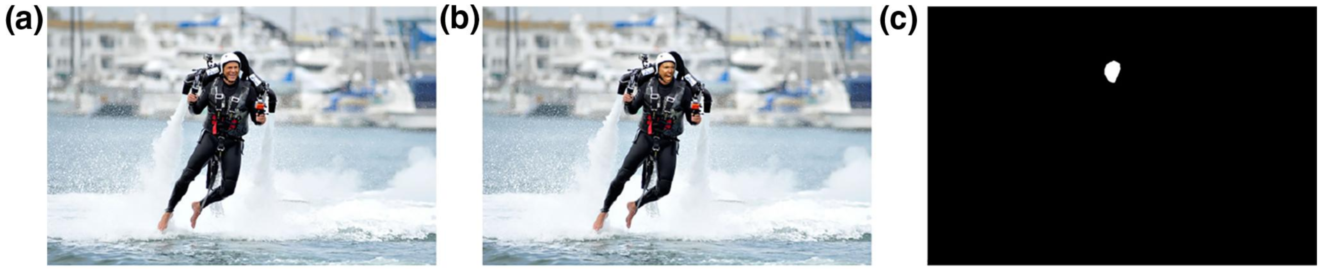

**FIGURE 9** A real-life manipulated image. On the left the real image is shown, in the middle the manipulated image, and on the right the binary mask localising the manipulated area, are displayed

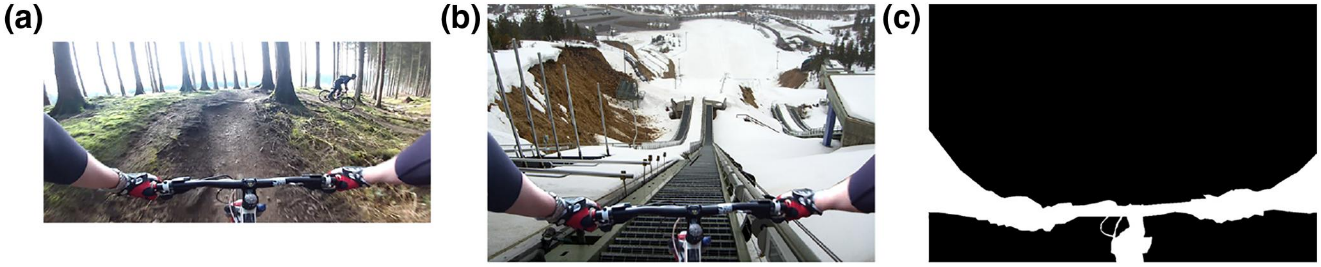

**FIGURE 10** A real-life manipulated image. On left the real image is shown , in the middle the manipulated image and on the right the binary mask localising the manipulated area, are observed. Binary masks of real-life manipulated images have been created manually

To link a digital image to an exact camera, first the camera sensor fingerprint is needed to be constructed. Specifically, for a given camera, the corresponding sensor noise fingerprint is estimated by averaging multiple camera reference images $I_k$, $k = 1, \ldots, N$. Camera reference images are photos captured by the camera under examination. It is recommended to use photos of an uniformly illuminated surface.

The process is often sped up by suppressing the scene content from the image prior to averaging. This is achieved by using a denoising filter $\mathscr{F}$ and averaging the noise residuals instead. $I^o$ is approximated by denoising $I$ that results in mentioned residuals as stated here:

$$I^o \approx I - \mathscr{F}(I) \tag{2}$$

In the above equation, we omitted pixel indexes $(i, j)$ in our denotations. Now, $\Gamma$ can be approximated in the following way:

$$\Gamma_N = \frac{1}{N} \sum_{k=1}^{N} I_k - \mathscr{F}(I_k) \tag{3}$$

Testing if an image has been captured by a particular camera is typically carried out by performing a similarity measure of two sensor fingerprints, $\Gamma_{s_1}, \Gamma_{s_2}$. Here, $\Gamma_{s_1}$ is obtained from the image under investigation and $\Gamma_{s_2}$ corresponds to the camera and obtained by using the set of camera reference images.

Typically, a normalised correlation (a black-box method) is used to compare two estimated sensor fingerprints. Having available $\Gamma_{s_1}$ and $\Gamma_{s_2}$, we measure their similarity by employing a normalised correlation:

$$\frac{corr(\Gamma_{s_1}, \Gamma_{s_2}) = (\Gamma_{s_1} - \overline{\Gamma_{s_1}}) \odot (\Gamma_{s_2} - \overline{\Gamma_{s_2}})}{(\|\Gamma_{s_1} - \overline{\Gamma_{s_1}}\|) \cdot (\|\Gamma_{s_2} - \overline{\Gamma_{s_2}}\|)} \tag{4}$$

where $\overline{X}$ denotes mean of the vector $X$, $\odot$ stands for dot product of vectors defined as $X \odot Y = \sum_{k=1}^{N} X(k)X(k)$ and $\|X\|$ denotes $L_2$ norm of $X$ defined as $\|X\| = \sqrt{X \odot X}$.

The estimated $\Gamma_N$ is the basic version of the camera sensor fingerprint and is not usable in practice for identifying the exact source camera. The reason is a strong presence of components non-unique to sensor in the estimated $\Gamma_N$. They are caused by operations performed by embedded software in cameras such as gamma correction, CFA interpolation, colour enhancement, geometric deformation corrections, JPEG compression, invisible watermarks etc.

To minimise this problem, sensor fingerprint can be, for example, enhanced by Wiener filtering in the frequency domain to remove traces of periodic artefacts [13]. This is not used in the experiments carries out in the next section.

## 6 | EXPERIMENTS

Here, we demonstrate results of a few popular image forensic methods on the collected real-life dataset. Moreover, we perform an experiment using the guaranteed set of images to demonstrate the presence of camera-dependent artefacts.

**TABLE 2** Cameras forming the guaranteed set of real images

| Camera ID | Camera make | Camera model | Sensor width | Sensor height | Images of uniform areas | Images of natural scenes |
|---|---|---|---|---|---|---|
| 1-1 | Apple | iphone 5 | 3264 | 2448 | 40 | 40 |
| 2-1 | Canon | 550d | 2592 | 1728 | 30 | 60 |
| 2-2 | Canon | 550d | 2592 | 1728 | 30 | 60 |
| 2-3 | Canon | 550d | 2592 | 1728 | 30 | 60 |
| 3-1 | Canon | 5d | 2496 | 1664 | 30 | 60 |
| 4-1 | Canon | ixus 145 | 2048 | 1536 | 30 | 40 |
| 4-2 | Canon | ixus 145 | 2048 | 1536 | 30 | 40 |
| 5-1 | Canon | powershot a495 | 3648 | 2736 | 50 | 100 |
| 6-1 | Canon | powershot g11 | 1600 | 1200 | 30 | 35 |
| 7-1 | Lg | l70 | 2240 | 1344 | 40 | 40 |
| 8-1 | Nikon | coolpix l23 | 3648 | 2736 | 50 | 100 |
| 8-2 | nikon | coolpix l23 | 3648 | 2736 | 50 | 100 |
| 8-3 | nikon | coolpix l23 | 3648 | 2736 | 50 | 100 |
| 9-1 | nikon | coolpix s2800 | 1600 | 1200 | 30 | 30 |
| 9-2 | nikon | coolpix s2800 | 1600 | 1200 | 30 | 40 |
| 9-3 | nikon | coolpix s2800 | 1600 | 1200 | 30 | 35 |
| 9-4 | nikon | coolpix s2800 | 1600 | 1200 | 25 | 40 |
| 10-1 | nikon | coolpix s3500 | 1600 | 1200 | 30 | 40 |
| 10-2 | nikon | coolpix s3500 | 1600 | 1200 | 30 | 40 |
| 11-1 | nikon | coolpix s4300 | 1600 | 1200 | 30 | 34 |
| 11-2 | nikon | coolpix s4300 | 1600 | 1200 | 30 | 40 |
| 12-1 | nikon | d40 | 1504 | 1000 | 30 | 60 |
| 13-1 | panasonic | lumix dmc zs3 | 2048 | 1536 | 30 | 30 |
| 14-1 | ricoh | cx5 | 3648 | 2736 | 30 | 30 |
| 14-2 | ricoh | cx5 | 3648 | 2736 | 30 | 40 |
| 15-1 | samsung | galaxy s4 mini | 3264 | 1836 | 40 | 50 |
| 16-1 | samsung | pl51 | 3648 | 2736 | 50 | 100 |
| 17-1 | samsung | galaxy tablet s 10.5 | 3264 | 1836 | 30 | 50 |
| 18-1 | sony | cybershot dsc wx80 | 4608 | 3456 | 30 | 40 |
| 18-2 | sony | cybershot dsc wx80 | 4608 | 3456 | 30 | 40 |
| 18-3 | sony | cybershot dsc wx80 | 4608 | 3456 | 30 | 40 |
| 19-1 | sony | xperia z ultra | 3104 | 1746 | 30 | 60 |

## 6.1 | Methods detecting manipulation

We applied the following methods on our dataset: NOI1 [36], CFA1 [67], BLK [68], ADQ1 [69] and ManTraNet [64]. To evaluate methods, all images have been first resized to 480 × 480 pixels. We computed false and true positive rates (FPR and TPR) as a function of the detection threshold, going from 0 to 1 and obtained the corresponding receiver operating characteristic (ROC) curve. Moreover, we calculated the area under the receiver operating characteristic curve (AUC) [58]. Results are shown in Figure 12 and Table 3.

As suggested by results, current methods have considerable limitations in their accuracy when applied on real-life (unseen) image forgery. Typical undetected types of manipulations are small manipulated areas, heavily compressed images, images degraded with correlated noise, images with multiple areas manipulated differently etc.

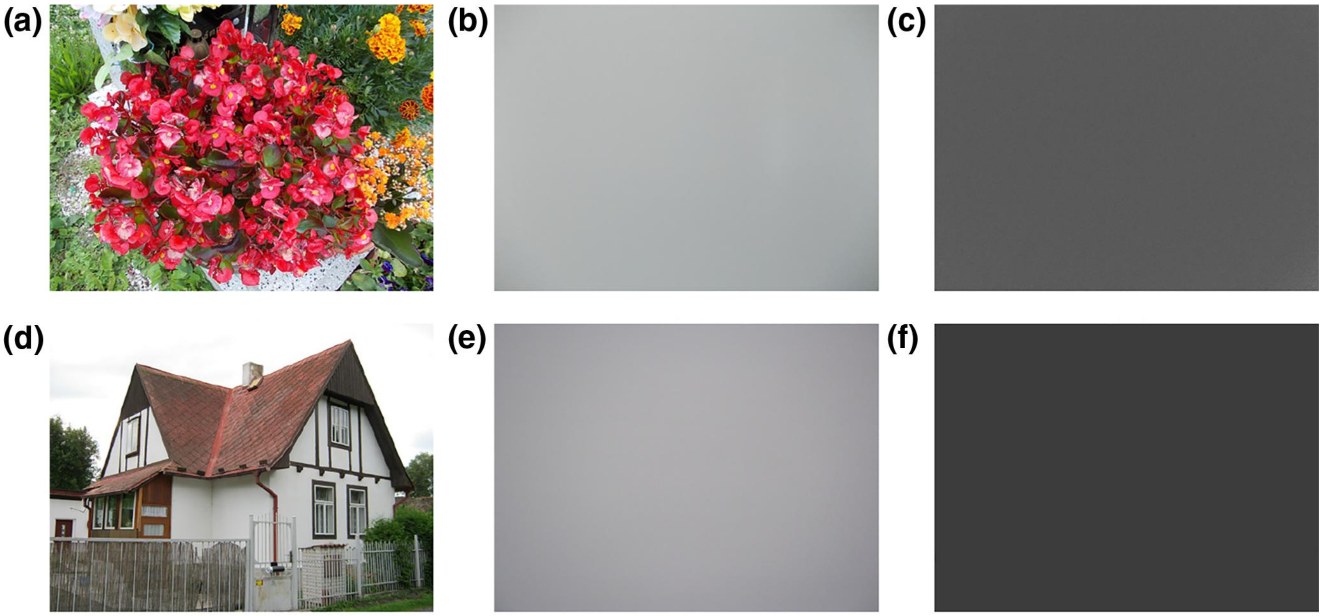

**FIGURE 11** (a) and (d) show two real-life images captured by Nikon Coolpix L23 and Canon Powershot A495, respectively; In (b) and (e) two images of a uniform scene captured by these cameras; and (c) and (f) show visualisation of sensor noise of these two cameras extracted from images shown in (b) and (e), respectively. Note that the apparent sensor noise of these two cameras differ

## 6.2 | Camera-dependent artefacts

Here, we will experimentally validate the presence of artefacts unique to particular cameras and unique to each camera model. To this end, camera sensor noise fingerprint, $\Gamma$, of all cameras pointed out in Table 2 was estimated.

For the sake of simplicity, $\Gamma$ was constructed by using residuals based on a simple median de-noise filter of size $3 \times 3$. Only the central part of images of size $976 \times 976$ has been used. For each camera, two different fingerprints have been constructed: (i) by using images of uniform areas, $\Gamma_{uniform}$ and (ii) by using images of natural scenes captured by the camera, $\Gamma_{natural}$. Next, camera fingerprints have been compared to each other using Equation (4). Specifically, for each camera, we first measured the similarity of the fingerprint formed by images of uniform areas and the fingerprint of the same camera formed by images of the natural scenes, $corr(\Gamma_{uniform}, \Gamma_{natural})$. Then, we calculated similarity of all uniform area fingerprints of all cameras with each other. Results are shown in Table 4. Figure 13 provides another view (a high-level view) on results obtained.

As it is apparent, the highest correlation values are obtained when comparing fingerprints of the same camera estimated using two different sets of images, $\Gamma_{uniform}$ and $\Gamma_{natural}$. This signifies a strong presence of artefacts unique to each camera sensor in images. On the other hand, the lowest values correspond to comparing fingerprints of totally different camera makes and models. Also, it is interesting to note that comparing fingerprints of different cameras of the same model results in higher correlation values than comparing the same for cameras of different models. This signifies the presence of artefacts unique to camera model. Analogically, we can see that correlation values obtained by comparing cameras of same manufacturer (without considering camera models) are still slightly higher than comparing cameras produced by different manufacturers.

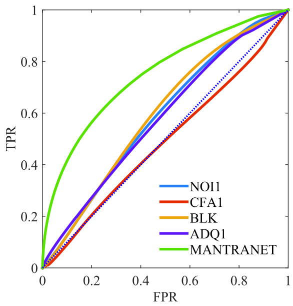

**FIGURE 12** Obtained ROC and AUC

**TABLE 3** Obtained ROC and AUC

| Method | AUC (%) |
|---|---|
| NOI1 [36] | 58.6 |
| CFA1 [67] | 48.7 |
| BLK [68] | 59.6 |
| ADQ1 [69] | 57.9 |
| ManTraNet [64] | 74.8 |

## 7 | CONCLUSION

In order to make possible deep nets to learn discriminatory features that well generalise to the unseen data, we need to

**TABLE 4** Similarity of camera fingerprints obtained by using Equation 4. Shown are (a) results of comparison of fingerprints obtained by using images captured by the same camera ($corr(\Gamma_{uniform}, \Gamma_{natural})$); (b) comparison of uniform fingerprints ($\Gamma_{uniform}$) of cameras of the same make and model; (c) comparison of uniform fingerprints of cameras of same make; and (d) and fingerprints of cameras of different make and model

| Camera ID | Make | Camera model | (a) Same camera | (b) Same make and model | (c) Same make and different model | (d) Different make and model |
|---|---|---|---|---|---|---|
| 1-1 | apple | iphone 5 | 0.12205 | — | — | −0.00042 |
| 2-1 | canon | 550d | 0.08801 | 0.01741 | 0.01575 | −0.00206 |
| 2-2 | canon | 550d | 0.32634 | 0.01248 | 0.02209 | −0.00191 |
| 2-3 | canon | 550d | 0.05191 | 0.00732 | 0.00403 | 0.00057 |
| 3-1 | canon | 5d | 0.20971 | — | 0.04415 | −0.00547 |
| 4-1 | canon | ixus 145 | 0.10322 | 0.04802 | 0.01619 | −0.00580 |
| 4-2 | canon | ixus 145 | 0.17284 | 0.04802 | 0.01517 | −0.00584 |
| 5-1 | canon | powershot a495 | 0.05238 | — | 0.00042 | 0.00012 |
| 6-1 | canon | powershot g11 | 0.02130 | — | 0.01673 | −0.00269 |
| 7-1 | lg | l70 | 0.08707 | — | — | −0.00228 |
| 8-1 | nikon | coolpix l23 | 0.09164 | 0.00270 | 0.00069 | −0.00010 |
| 8-2 | nikon | coolpix l23 | 0.09770 | 0.00283 | 0.00101 | −0.00051 |
| 8-3 | nikon | coolpix l23 | 0.15717 | 0.00212 | 0.00113 | −0.00070 |
| 9-1 | nikon | coolpix s2800 | 0.01935 | 0.00789 | 0.00089 | −0.00209 |
| 9-2 | nikon | coolpix s2800 | 0.02303 | 0.00737 | 0.00205 | −0.00294 |
| 9-3 | nikon | coolpix s2800 | 0.02147 | 0.00789 | 0.00170 | −0.00329 |
| 9-4 | nikon | coolpix s2800 | 0.01546 | 0.00915 | 0.00178 | −0.00197 |
| 10-1 | nikon | coolpix s3500 | 0.05165 | 0.01127 | 0.00175 | −0.00333 |
| 10-2 | nikon | coolpix s3500 | 0.03884 | 0.01127 | 0.00189 | −0.00389 |
| 11-1 | nikon | coolpix s4300 | 0.02582 | 0.00729 | 0.00159 | −0.00343 |
| 11-2 | nikon | coolpix s4300 | 0.02983 | 0.00729 | 0.00278 | −0.00466 |
| 12-1 | nikon | d40 | 0.14646 | — | −0.00515 | 0.00985 |
| 13-1 | panasonic | lumix dmc zs3 | 0.25236 | — | — | −0.00110 |
| 14-1 | ricoh | cx5 | 0.09227 | 0.10994 | — | 0.00046 |
| 14-2 | ricoh | cx5 | 0.10039 | 0.10994 | — | 0.00042 |
| 15-1 | samsung | galaxy s4 mini | 0.15203 | — | 0.00386 | −0.00256 |
| 16-1 | samsung | pl51 | 0.09657 | — | 0.00187 | −0.00031 |
| 17-1 | samsung | galaxy tablet s 10.5 | 0.09081 | — | 0.00361 | −0.00022 |
| 18-1 | sony | cybershot dsc wx80 | 0.08886 | 0.18171 | 0.00005 | −0.00120 |
| 18-2 | sony | cybershot dsc wx80 | 0.08709 | 0.17243 | −0.00047 | −0.00098 |
| 18-3 | sony | cybershot dsc wx80 | 0.15547 | 0.17715 | −0.00083 | −0.00049 |
| 19-1 | sony | xperia z ultra | 0.09899 | — | −0.00047 | −0.00020 |

have large and diverse datasets available. Such datasets need to be designed to capture wide and complex types of systematic noise and intrinsic artefacts of images in order to avoid overfitting of learning methods to just a narrow set of camera types or types of manipulations. These artefacts are brought into visual content by various components of the image acquisition process as well as the manipulating process (e.g. sensor noise, JPEG quantization noise, demosaicking and interpolation-related artefacts, image enhancement etc.). In the proposed and performed work, we collected three large-scale and diverse datasets with a high variety of artefacts. We have demonstrated results of a few popular methods of image forensics.

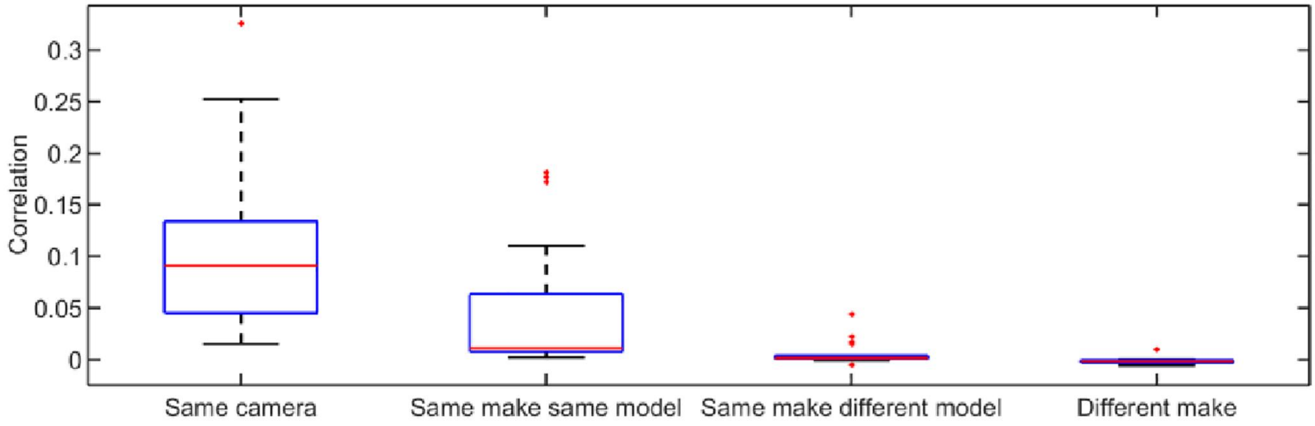

**FIGURE 13** Shown is a high-level view on results pointed out in Table 4. As it is apparent, the highest correlation values are obtained when comparing fingerprints of the same exact camera estimated by using two different sets of images, $\Gamma_{uniform}$ and $\Gamma_{natural}$. Comparing fingerprints of different cameras of the same model still results in higher correlation values than comparing the same for cameras of different models (but same make). Analogically, we can see that correlation values obtained by comparing cameras of the same make but different models are still slightly higher than comparing cameras produced by different manufacturers

Moreover, we empirically demonstrated the existence of different types of artefacts in the dataset.

We hope that the dataset will contribute to facilitating future research on training and testing methods for detecting of manipulated visual content as well as source camera identification (PRNU and sensor noise analysis).

## ACKNOWLEDGEMENTS

This work was supported by the European Union's Horizon 2020 research and innovation programme under grant agreement No 825227.

## ORCID

*Adam Novozámský* https://orcid.org/0000-0002-2470-9642
*Stanislav Saic* https://orcid.org/0000-0002-8043-1841

## REFERENCES

1. Farid, H.: Photo Forensics. The MIT Press (Cambridge, United States 2016)
2. Karras, T., et al.: Progressive growing of GANs for improved quality, stability, and variation. arXiv: 1710.10196 (2018)
3. Deng, J., et al.: Imagenet: a large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition,. pp. 248–255. IEEE, Miami (2009)
4. Kevin, C., Turk, A.M.: A Research Tool for Organizations and Information Systems Scholars. In: Shaping the Future of ICT Research. Methods and Approaches, pp. 210–221. Springer, Berlin (2012)
5. Schaefer, G., Stich, M.: UCID: An uncompressed color image database. In Storage and Retrieval Methods and Applications for Multimedia 2004, Vol. 5307, pp. 472–480. SPIE (2004)
6. Kalpathy-Cramer, J., et al.: Evaluating performance of biomedical image retrieval systems-an overview of the medical image retrieval task at imageclef 2004-2013. Comput. Med. Imag. Grap, 01, (55–61 2015)
7. Everingham, M., et al.: The pascal visual object classes challenge: a retrospective. Int. J. Comput. Vis. 111(1), 98–136 (2015)
8. Lin, T.-Y., et al.: Microsoft coco: common objects in context. In: Computer Vision – ECCV 2014, pp. 740–755. Springer International Publishing (2014)
9. Huh, M., et al.: Fighting fake news: Image splice detection via learned self-consistency. In: Proceedings of the European Conference on Computer Vision (ECCV), vol. 5 Springer International Publishing (2018)
10. Cozzolino, D., et al.: Forensictransfer: Weakly-supervised domain adaptation for forgery detection. arXiv:1812.02510 (2018)
11. Piva, A.: An overview on image forensics. ISRN Signal Process. 01 (2013)
12. Dirik, A.E., et al.: New features to identify computer generated images. In: 2007 IEEE International Conference on Image Processing, pp. IV–433 IEEE, San Antonio (2007)
13. Lukás, J., Fridrich, J., Goljan, M.: Digital camera identification from sensor pattern noise. IEEE Trans. Inform. Forens. Secur. 1, 205–214 (2006)
14. Mahdian, B., Saic, S.: Blind authentication using periodic properties of interpolation., IEEE Trans. Inform. Forens. Security, 3:529–538 (2008)
15. Iizuka, S., Simo-Serra, E., Ishikawa, H.: Globally and locally consistent image completion. ACM Trans. Graph., 36, 1–14 (2017)
16. Yu, J., et al.: Generative image inpainting with contextual attention. arXiv: 1801.07892. 5505–5514, 06 (2018)
17. Zhu, J.-Y., et al.: Unpaired image-to-image translation using cycle-consistent adversarial networks. arXiv: 1703.10593, 2242–2251 (2017)
18. Karras, T., Laine, S., Aila, T.: A style-based generator architecture for generative adversarial networks. arXiv: 1812.04948 (2019)
19. Zhang, X., Karaman, S., Chang, S.: Detecting and Simulating Artifacts in GAN Fake Images, arXiv: 1907.06515 (2019)
20. Novozamsky, A., Mahdian, B., Saic, S.: Imd2020: Imd2020: A large-scale annotated dataset tailored for detecting manipulated images. In: Proceedings of the IEEE/CVF Winter Conference on Applications of computer Vision (WACV) Workshops, pp. 71–80. IEEE, Snowmass (2020)
21. Tralic, D., et al.: Comofod-new database for copy-move forgery detection. In: Proceedings ELMAR-2013, pp. 49–54. (2013)
22. Amerini, I., et al.: A sift-based forensic method for copy-move attack detection and transformation recovery. IEEE Trans. Inform. Forens. Security, 6, 1099–1110 (2011)
23. Ng, T.-T., Chang, S.: A data set of authentic and spliced image blocks. Tech. Rep. Columbia University (2004)
24. Dong, J., Wang, W., Tan, T.: Casia image tampering detection evaluation database. In: 2013 IEEE China Summit and International Conference on Signal and Information Processing, pp. 422–426. IEEE, Beijing (2013)
25. Piva, J.H.A., Rocha, A.: The first IFS-TC image forensics challenge. Accessed 22 January 2014. (2014) https://signalprocessingsociety.org/community-involvement/information-forensics-and-security

26. Wen, B., et al.: Coverage - a novel database for copy-move forgery detection. In: 2016 IEEE International Conference on Image Processing (ICIP),. pp. 161–165. (2016)
27. REWIND. Reverse engineering of audio-visual content data
28. Barni, M., Costanzo, A., Sabatini, L.: Identification of cut & paste tampering by means of double-jpeg detection and image segmentation. In: Proceedings of 2010 IEEE International Symposium on Circuits and Systems, pp. 1687–1690. IEEE, Paris, France. (2010)
29. Zhou, P., et al.: Two-stream neural networks for tampered face detection. In: IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 1831–1839 (2017)
30. Kowalski, M.: Faceswap. https://github.com/shaoanlu/faceswap.
31. Swapme. Accessed 1 October 2019 https://itunes.apple.com/us/app/swapme-by-faciometrics/.acquiredbyfacebook. No longer available in app-store.
32. Korus, P., Huang, J.: Evaluation of random field models in multi-modal unsupervised tampering localization. In: 2016 IEEE International Workshop on Information Forensics and Security (WIFS), pp. 1–6. (2016)
33. Guan, H., et al.: Mfc datasets: Large-scale benchmark datasets for media forensic challenge evaluation. In: 2019 IEEE Winter Applications of Computer Vision Workshops (WACVW), pp. 63–72 IEEE, Waikoloa (2019)
34. Farid, H.: Exposing digital forgeries from jpeg ghosts. IEEE Trans. Inform. Forens. Secur. 4, 154–160 (2009)
35. Popescu, A., Farid, H.: Exposing digital forgeries in color filter array interpolated images. IEEE Trans. Signal Process. 53, 3948–3959 (2008)
36. Mahdian, B., Saic, S.: Using noise inconsistencies for blind image forensics. Image Vis. Comput. 27, 1497–1503 (2009)
37. Luo, W., et al.: A novel method for detecting cropped and recompressed image block. In: 2007 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP '07. pp. 217–2020, IEEE, Honolulu (2007)
38. Wang, W., Dong, J., Tan, T.: Effective image splicing detection based on image chroma. In: 2009 16th IEEE International Conference on Image Processing (ICIP),. pp. 1257–1260 (2009)
39. Bayram, S., Sencar, T., Memon, N.: An efficient and robust method for detecting copy-move forgery. In: 2009 IEEE International Conference on Acoustics, Speech and Signal Processing, IEEE, Taiwan pp. 1053–1056. (2009)
40. Rössler, A., et al.: Faceforensics++: Learning to detect manipulated facial images. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 1–6 IEEE, Seoul (2019).
41. Ghosh, A., et al.: A learned method for blind image forensics. CoRR, abs/1906, 11663 (2019)
42. Bunk, J., et al.: Detection and localization of image forgeries using resampling features and deep learning. CoRR, abs/170,00433 (2017)
43. Wu, Y., Abd-Almageed, W., Natarajan, P.: BusterNet: Detecting Copy-Move Image Forgery with Source/Target Localization. In: 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part VI, pp. 170–186.(2018)
44. Zhang, Z., et al.: Boundary-based image forgery detection by fast shallow cnn. 08, 2658–2663, arXiv: 1801.06732 (2018)
45. Tschannen, M., Bachem, O., Lucic, M.: Recent advances in autoencoder-based representation learning. CoRR, abs/1812, 05069 (2018)
46. Yu, N., Davis, L., Fritz, M.: Attributing fake images to gans: Analyzing fingerprints in generated images. CoRR, abs/1811, 08180 (2018)
47. Choi, H.-Y., et al.: Detecting composite image manipulation based on deep neural networks. In: 2017 International Conference on Systems, Signals and Image Processing (IWSSIP), IEEE, Poznan pp. 1–5. (2017)
48. Mazaheri, G.: A skip connection architecture for localization of image manipulations. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops.. (2019)
49. Bappy, J. H., et al.: Hybrid LSTM and encoder-decoder architecture for detection of image forgeries. CoRR abs/1903, 02495 (2019)
50. Bappy, M. J., et al.: Exploiting spatial structure for localizing manipulated image regions. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV), pp. 4970–4979. (2017). IEEE, Cambridge, MA
51. Zhou, P., et al.: Learning rich features for image manipulation detection. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition,. pp. 1053–1061 IEEE, Salt Lake City (2018)
52. Rao, Y., Ni, J.: A deep learning approach to detection of splicing and copy-move forgeries in images. In: 2016 IEEE International Workshop on Information Forensics and Security (WIFS). pp. 1–6 IEEE, Abu Dhabi (2016)
53. Cun, X., Pun, C.-M.: Image splicing localization via semi-global network and fully connected conditional random fields. In: Lecture Notes in Computer Science,. pp. 252–266. Springer International Publishing (2019)
54. Cozzolino, D., Noiseprint, L.V.: A CNN-based camera model fingerprint. IEEE Trans. Inform. Forens. Secur. 05 (2019)
55. Bondi, L.: et al.: Tampering detection and localization through clustering of camera-based CNN features. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), IEEE, Honolulu pp. 1855–1864. (2017)
56. Bayar, B., Stamm, M.: Constrained convolutional neural networks: A new approach towards general purpose image manipulation detection. IEEE Trans. Inform. Forens. Secur. 04 (2018)
57. Liu, Y., et al.: Image forgery localization based on multi-scale convolutional neural networks. arXiv: 1706.07842. 85–90 (2018)
58. Salloum, R., Ren, Y., Kuo, C.: Image splicing localization using a multi-task fully convolutional network (mfcn). J. Visual Commun. Image Represent. 51 (2017)
59. Le-Tien, T., et al.: Image forgery detection: a low computational-cost and effective data-driven model. Intl. J. Mach. Learn. Comput. 9, 181–188 (2019)
60. Bayar, B., Stamm, M.: A deep learning approach to universal image manipulation detection using a new convolutional layer. In: Proceedings of the 4th ACM Workshop on Information Hiding and Multimedia Security (IH&MMSec '16), pp. 5–10. (2016) Association for Computing Machinery, NY.
61. Wu, Y., Abd-Almageed, W., Natarajan, P.: Deep matching and validation network: An end-to-end solution to constrained image splicing localization and detection. In: Proceedings of the 25th ACM International Conference on Multimedia,. pp. 1480–1502. (2017)
62. Marra, F., et al.: A full-image full-resolution end-to-end-trainable CNN framework for image forgery detection. 09 (2019)
63. Marra, F., et al.: Detection of gan-generated fake images over social networks. In: 2018 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR), pp. 384–389. IEEE, Miami (2018)
64. Wu, Y., AbdAlmageed, W., Natarajan, P.: Mantra-net: Manipulation tracing network for detection and localization of image forgeries with anomalous features. In:2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 9535–9544, IEEE, Long Beach (2019)
65. FaceApp. Ai face editor. Accessed 22 January 2020. https://www.faceapp.com/
66. Heller, S., Rossetto, L., Schuldt, H.: The PS-Battles Dataset – an Image Collection for Image Manipulation Detection. CoRR, abs/1804, 04866 (2018)
67. Ferrara, P., et al.: Image forgery localization via fine-grained analysis of cfa artifacts. Trans. Info. For. Secur. 7(5), 1566–1577 (2012)
68. Li, W., Yuan, Y., Yu, N.: Passive detection of doctored jpeg image via block artifact grid extraction. Signal Process. 89(9), 1821–1829 (2009)
69. Lin, Z., et al.: Fast, automatic and fine-grained tampered jpeg image detection via dct coefficient analysis. Pattern Recogn. 42(11), 2492–2501 (2009)

**How to cite this article:** Novozámský A, Mahdian B, Saic S. Extended IMD2020: A large-scale annotated dataset tailored for detecting manipulated images. *IET Biom*. 2021;1–16. https://doi.org/10.1049/bme2.12025