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
{novozamsky,mahdian,ssaic}@utia.cas.cz

Abstract

Witnessing impressive results of deep nets in a number of computer vision problems, the image forensic community has begun to utilize them in the challenging domain of detecting manipulated visual content. One of the obstacles to replicate the success of deep nets here is absence of diverse datasets tailored for training and testing of image forensic methods. Such datasets need to be designed to capture wide and complex types of systematic noise and intrinsic artifacts of images in order to avoid overfitting of learning methods to just a narrow set of camera types or types of manipulations. These artifacts are brought into visual content by various components of the image acquisition process as well as the manipulating process.

In this paper, we introduce two novel datasets. First, we identified the majority of camera brands and models on the market, which resulted in 2,322 camera models. Then, we collected a dataset of 35,000 real images captured by these camera models. Moreover, we also created the same number of digitally manipulated images by using a large variety of core image manipulation methods as well we advanced ones such as GAN or Inpainting resulting in a dataset of 70,000 images. In addition to this dataset, we also created a dataset of 2,000 "real-life" (uncontrolled) manipulated images. They are made by unknown people and downloaded from Internet. The real versions of these images also have been found and are provided. We also manually created binary masks localizing the exact manipulated areas of these images. Both datasets are publicly available for the research community at http://staff.utia.cas.cz/novozada/db.

1. Introduction

Today, manipulated visual content has become a serious problem that is negatively impacting many aspects of our life. Advances in image editing techniques and user-friendly editing software have made possible the creation of realistic looking manipulated visual content. In addition to classic image editors, we are also facing a growing popularity of novel apps and software tools using recent advances in computer vision such as Generative Adversarial Networks (GAN) [29].

It is obvious that there is a fundamental need to have technologies that make possible to reliably assess the integrity of digital images and videos. However, today's methods of image/video forensics suffer from serious limitations, resulting in their low accuracy when applied in real-life conditions. Witnessing impressive results of deep nets in a number of computer vision problems, the image forensic community has begun to utilize deep nets in the challenging domain of detecting manipulated visual content. However, there are a few obstacles in order to replicate the success of deep nets here.

One of the major obstacles is the demand of deep nets for large-scale datasets during training. In 2009, the ImageNet dataset [15] was released. It provided researchers in the area of images classification with a large scale annotated dataset. In order to build this dataset Fei-Fei Li et al. [15] leveraged Google Image Search to pre-filter large candidate sets for each category. Additionally, they used the Amazon Mechanical Turk crowdsourcing pipeline [55] to manually validate each image if it belonged to the associated category. This large dataset significantly pushed computer vision and machine learning research forward and helped to develop classification models performing at much better accuracies than academic methods previously published. Today, the computer vision community benefits from several such publicly available datasets like: UCID [51] and ImageCLEF [28] for image retrieval; PASCAL [18], ImageNet [15], and Microsoft COCO [34] for tasks such as object detection, segmentation, and recognition.

None of mentioned datasets can directly serve the image forensic community since they have not been intentionally collected digitally manipulated data. They also lack

978-1-7281-7162-3/20/$31.00 ©2020 IEEE

diversity and annotations required from the forensic perspective. So far, most of the image forensic authors relied on small datasets which typically cannot capture wide and complex image artifacts that are brought into real-life images throughout their lifecycle. This also has caused that existing methods often fail in cross-dataset tests and generalization. Some of the authors tried to overcome the problem by training their methods only using real images (e.g., [26]). Some others tried to overcome the problem by building their internal limited datasets (e.g., [12]) and rather focus on domain adaptation.

In this work, our aim 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 artifacts. Creating such a dataset is a challenging task. Each camera brings into the image different kinds of artifacts. Some artifacts are unique to particular camera device and some are unique to camera model. Various compressions levels bring different quantization noise into the visual content. Each type of manipulation brings different traces of editing into images, etc. Overall, we can categorize intrinsic artifacts existing in the visual content into three main groups: (i) acquisition artifacts, see Fig. 1 (e.g., sensor noise, demosaicking algorithms or gamma correction); (ii) format artifacts (e.g., JPEG and quantization noise); (iii) manipulation artifacts (e.g., artifacts left by GAN in the image).

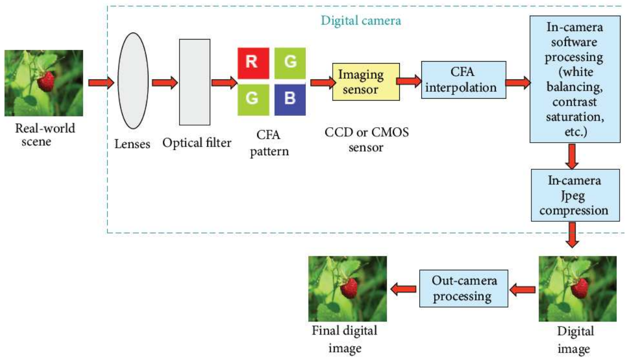


Figure 1: Individial steps and components forming a typical digital image [45].

Above mentioned artifacts play a key role in methods of image/video forensics. To detect traces of image manipulation, we can notice that often image forensic methods actually attempt to eliminate the image content and emphasize these kinds of intrinsic artifacts. This is often achieved by employing high pass filters and resulting noise residuals. These noise residuals are then employed (instead of the original image) to detect traces of manipulation. Although the above-mentioned artifacts 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 that methods perform poorly on new and unseen manipulations (e.g., [12]).

1.1. Contribution

This paper studies intrinsic artifacts in images, and provides a comprehensive review of existing image forensic datasets. Moreover, it also brings a survey of existing CNN-based methods for detecting image manipulation. The main contribution of the paper is creation of two novel datasets. The first dataset comprises 35,000 real images captured by 2,322 different camera models. These camera models form the majority of existing cameras on the market. The dataset provides a rich and diverse set of sensor noise, artifacts that various imaging software embedded in cameras bring into images, and compression artifacts. Moreover, we also synthetically created a set of manipulated images by 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 have been found and are provided. Binary masks localizing the manipulated areas have been created manually.

We hope that collected datasets will contribute to facilitating future research on detection of manipulated visual content.

2. Artifacts 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 [45]. For the sake of simplicity, we model the image acquisition process in the following way:

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

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 multiplicative noise, such as PRNU (photo response non-uniformity) and $\Upsilon_{i,j}$ stands for all additive noise components.

In this section, we briefly describe the major types of artifacts brought into images during the acquisition process and in their later stages of the lifecycle.

2.1. Artifacts associated with acquisition devices

Each component of the digital image acquisition device brings into the image some intrinsic artifacts (fingerprints) that are present in the final visual content output.

During acquisition, the light of the real scene is focused through the optical system of the camera on its sensor (typically CCD or CMOS). The sensor consists of small elements called pixels that collect photons and convert them into voltages that are subsequently sampled by a digital signal in an A/D converter. Before reaching the sensor, however, the light is usually filtered by the Color Filter Array (CFA). The CFA is a mosaic of tiny color filters placed over the pixel of an image sensor to capture particular color information. Color filters are used because typical consumer cameras only have one sensor and hence cannot separate color information. In practice, each pixel collects only one particular main color (red, green, or blue). The sensor output is successively interpolated to obtain all the three main colors for each pixel, through the so-called demosaicking process, in order to obtain the digital color image [45]. The resulting signal is then further processed using color 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.

Some of the above-mentioned artifacts are unique to particular camera (per sensor) and some are in common for all cameras of the same model or brand (i.e., cameras having the same embedded software). For instance, pattern noise associated with the image sensor is typically unique. As pointed out in [16], if we take a picture of an absolutely evenly light scene, the resulting digital image will still exhibit small changes in intensity among individual pixels which is partly because of pattern noise, readout noise or shot noise. Sensor pattern noise has been widely used by authors to identify the exact camera that captured the image [37]. To this end, authors typically use PRNU which is a of the sensor pattern (the multiplicative component of Eq. 1). Figure 2 shows two different cameras capturing the same scene and their corresponding sensor pattern noise. A light scene as use since light scene with minimal number of edges enable an easier extraction and modeling of the sensor noise [37].

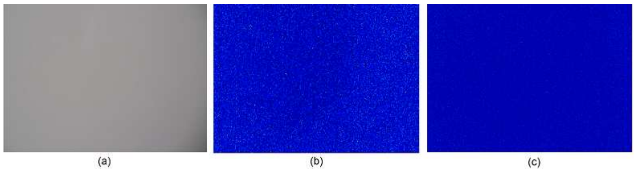


Figure 2: In (a) the captured scene is shown. (b) shows the extracted sensor pattern noise of a Nikon Coolpix S3000 device. (c) shows the same for Samsung Pl51. As apparent sensor noise of these two cameras differ.

.

On the other hand, if we look at, for example, the demosaicking process, it is typically identical for all cameras of the same model (if they use the same embedded software and demosaicking algorithm). Different demosaicking algorithms bring different interpolation related artifacts. For example, Mahdian et al [39] shows that these interpolation techniques often bring into the image invisible periodic artifacts.

2.2. Artifacts 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 are 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 $C(u), C(v) = 1$.

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 artifacts 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 artifacts and compression levels.

2.3. Artifacts 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., [27], [61]), or GAN (e.g., Cycle-GAN [67] or Style-GAN [30]). 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.); (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 artifacts in the image (e.g., [63]). 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 on more details to the GAN pipeline (e.g., Fig. 3) 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 contains a few up-sampling layers which take the low resolution feature tensor as input and output a high-resolution image. According to Zhang et al. [63], although the structures of GAN models are quite diverse the up-sampling modules used in different GAN models are consistent. The up-sampling bring into the image specific artifacts (e.g., interpolation based [39]). Zhang et al. [63] addressed these up-sampling related artifcats and used them to detect GAN based images. They showed that they are present in most of the commonly used GAN methods.

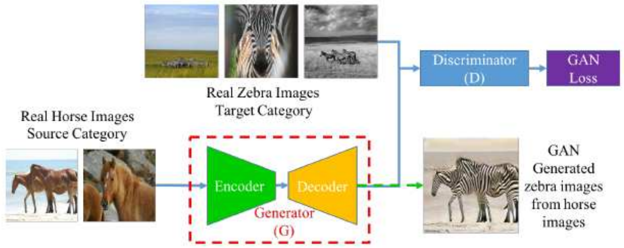


Figure 3: Typical pipeline of image2image translation [63].

To summarize this section, 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 artifacts 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).

3. Related Work

In this section we review existing datasets as well as CNN-based methods dealing with detection of image and video manipulation.

3.1. Related Datasets

The CoMoFoD dataset [53] has been designed for copy-move forgery detection. It consists of 260 forged images in two categories of small (512x512 pixels), and large (3000x2000 pixels). Each set includes a forged image, mask of the manipulated area, and its original image. Images are divided into 5 groups according to applied manipulation: translation, rotation, scaling, combination and distortion, etc. The MICC-F220, MICC-F2000 [2] is 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 [44] has two parts. First, a grayscale image dataset with 933 authentic and 912 spliced grayscale image blocks, and a color image dataset with 183 authentic uncompressed color block images and 180 spliced uncompressed color block images.

CASIA Image Tampering Detection Evaluation Database [17] 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 [1] collected thousands of images of various scenes, both indoors and outdoors. The dataset served for an international competition organized by the IEEE Information Forensics and Security Technical Committee and comprises of a total of 1176 forged images. Wen et al. [57] introduced a small dataset called Coverage designed for copy-paste detection. The REWIND (REVerse engineering of audio-VIsual coNtent Data) [48] dataset contains 142 hand-made manipulated images for the evaluation of image tampering detectors. Half of the images is original; the other half is a set of hand-made forgeries. There are also 4800 automatically manipulated images. Barni et al. [5] created a small dataset for detecting cut and paste splicing (ISCAS). Zhou et al. created a dataset of manipulated faces [66] by using FaceSwap [20] and SwapMe [52]. There are 1005 tampered images for each tampering technique (2010 tampered images in total) and 1400 authentic images for each subset. Realistic Tampering Dataset [31] 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) presented with a large benchmark dataset - Nimble Challenge 2017 [24]. This dataset contains a total of 2,520 manipulated images. In following years, NIST also has published additional datasets MFC2018 and MFC2019 [24].

Most of the currently published datasets (see Tab 1) are

Table 1: Examples of datasets designed for image manipulation detection.

Dataset	Size	Binary mask
CoMoFoD dataset [53]	260	Yes
MICC-F220, MICC-F2000 [2]	2,200	No
Columbia [44]	1,845	No
CASIA [17]	1,721	No
CASIA v2.0 [17]	12,323	No
REWIND Real [48]	142	Yes
Zhou et al. [66]	3,410	No
Nimble Challenge 2017 (manipulated) [24]	2,520	Yes
ISCAS [5]	20	No
Realistic Tampering [31]	440	Yes
Coverage [57]	100	Yes
IMD2020 Synthetically Created (proposed)	70,000	Yes
IMD2020 Manually Created (proposed)	2,000	Yes

limited in size, acquisition device variety, content, attacks type, and compression/post processing variety. Typically, they are created in a controlled environment.

3.2. State-of-the-Art Methods

Early methods of image forensics have focused on detecting individual types of manipulations using handcrafted features. These traditional methods typically aim to detect some targeted inconsistencies among pixels. Here, we mention a few examples of such methods. Farid et al. [21] proposed a method for detecting composites created by JPEG images of different qualities. The method detects whether a part of an image was initially compressed at a lower quality than the rest of the image. In [46], 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. [40] proposed a method for detecting local image noise inconsistencies based on estimating local noise variance using wavelet transform. Weiqi Luo et al. [38] proposed a method for detecting recompressed image blocks based on JPEG blocking artifact characteristics. In [56], Wei Wang et al. proposed an image splicing detection method based on gray level co-occurrence matrix (GLCM) of thresholded edge image of image chroma. In [8], Sevinc Bayram et al. proposed a clone detector based on Fourier–Mellin transform of the image's blocks. The Fourier–Mellin transform is invariant with respect to scale and rotation. This allows a better behavior of the method when dealing with slightly resized and rotated cloned regions. A survey of classic image forensic methods is provided in [49].

3.2.1 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 its direction to utilize achievements of deep learning. In [23], 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 [10], 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 localization of manipulation. In [59], 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 localize potential manipulation areas using visual discontinuities and copy-move regions via visual similarities, respectively.

In [64] Zhang et al., introduced a Shallow Convolutional Neural Network (SCNN) capable of distinguishing the boundaries of forged regions from original edges in low-resolution images. It uses information of chrominance and saturation channels. In [12], Cozzolino et. al, address the problem of inaccurate results of today's CNN based methods when performed in cross-dataset test scenarios. The underlying CNN quickly overfit to manipulation-specific artifacts resulting in learning features that are highly discriminatory for the given dataset but lack of generalization. To address this limitation in transferability, they introduced Forensic-Transfer (FT). They learn a forensic embedding based on am auto-encoder based architecture [54] 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. Authors show that only a few training samples of the target domain of tampering enable to finetune their model to achieve high accuracies.

In [43], Mazaheri et al. assume that most of the manipulated images leave some traces near boundaries of manipulated regions including blurred edges. They proposed an encoder-decoder based network where they fuse representations from early layers in the encoder. In [3], Bappy et al. employed manipulation localization architecture which utilizes resampling features, Long-Short Term Memory (LSTM) cells, and encoder-decoder network to segment manipulated areas of the image. Resampling features are used to capture artifacts like JPEG quality loss, upsampling, down-sampling, rotation, and shearing. In [62] Yu et al. analyzed learning GAN 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 differencein initialization) can leave a distinct fingerprint that commonly exists over all its generated images. In [4], Bappy et al. presented a unified framework for joint patch classification and segmentation to localize manipulated regions from an image. They assume that a key property of manipulated regions is that they exhibit discriminative features in boundaries shared with neighboring non-manipulated pixels. Their method learns the boundary discrepancy, i.e., the spatial structure, between manipulated and non-manipulated regions with the combination of LSTM and convolution layers. In [11], Kim et al. employed a deep learning approach that utilizes a high pass filter to acquire hidden features in the image rather than semantic information in the image.

In [47], Rao et al. proposed a customized CNN based method for detecting manipulation. The weights at the first layer of their network are initialized with the 30 basic high-pass filters used in spatial rich model for image steganalysis, which helps to efficiently suppress the effect of complex image contents and accelerate the convergence of the network. In [14], Cun et al. instead of classifying the spliced region by a local patch, they leveraged the features from whole image and local patch together, calling this structure Semi-Global Network. In [65], Zhou et al, proposed a novel network using both an RGB stream and a noise stream to learn rich features for image manipulation detection. The authors observed that the fusion of the two streams leads to improved performance. In [13], Cozzolino et al. proposed a deep learning method to extract a noise residual, called noiseprint, where the scene content is largely suppressed and model-related artifacts are enhanced. In the paper they demonstrate promising forgery localization results.

In [9], 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 [7], 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 [36], Liu et al. proposed to utilize Convolutional Neural Networks and the segmentation-based multi-scale analysis to locate tampered areas in digital images. Authors observed that exploiting the benefits of both the small scale and large-scale analyses, the segmentation-based multiscale analysis can lead to a performance leap in forgery localization of CNNs.

In [50], Salloum et al. proposed a technique that utilizes a fully convolutional network (FCN) to localize image-splicing attacks. The utilized FCN is based on the FCN VGG-16 architecture with skip connections, and authors incorporated several modifications, such as batch normalization layers and class weighting. They show significant improvement in comparison to state of the art methods. In [26], Huh et al. proposed an algorithm that uses the automatically recorded photo EXIF metadata as supervisory signal for training 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 [32], Le-Tien et al. proposed a low computational-cost and fully connected neural network to address the problem of image forgery detection. In [6], Bayar et al. tried to prevent the CNN from learning features that represent an image's content. They proposed a new form of convolutional specifically designed to suppress an image's content and learn manipulation detection features. In [58], Wu et al. showed that both image splicing detection as well as localization can be jointly solved using a multitask network in an end-to-end manner. In [42], Marra et al. attempts to avoid downsizing of images before analyzing 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 [63], Zhang et al. proposed a GAN simulator, which can simulate the artifacts produced by the common pipeline shared by several popular GAN models. They identified a unique artifact 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 state-of-the-art performances on detecting fake images generated by popular GAN models. In [41], Marra et al. studied the performance of several 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. They showed that particularly XceptionNet can achieve high accuracies in detection of GAN-generated fake images published on social networks. In [60], Wu et al. introduced a network called ManTra-Net. They formulated the forgery localization problem as a local anomaly detection problem, designed a Z-score feature to capture local anomaly, and propose a novel long short-term memory solution to assess local anomalies. 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.

4. The IMD2020 Dataset

Image forensic methods often eliminate the image content and analyze the underlying (hidden) noise/artifacts component of the image to find inconsistencies. As pointed out earlier, some of the intrinsic artifacts are unique to sensor/camera and some others shared by images captured by cameras of the same brand/model. To avoid potential overfitting to a narrow set 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.

However, when downloading images from unknown and non-guaranteed environments such as Flickr, the processing history of the images is typically unknown. Although we can assume that most of the users of Flickr have no practical reason to publish a maliciously manipulated visual content, to minimize the quantity of such content, we manually reviewed all of the images and discarded those having obvious traces of digital manipulation. This has been resulted in 35,000 manually reviewed (cleaned) set of real images. Some examples of pictured in this set are shown in Fig. 4. 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.).

Figure 4: Some examples of real pictures in our dataset.

We also generated a same number of synthetically manipulated images using high variety of methods. As pointed our earlier, advanced techniques such as GAN often bring characteristic artifacts into images [62]. Such kinds of artifacts might lead to overfittling of learning methods. This has also been empirically confirmed by Cozzolino et al. [12] 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 percent to 30 percent of the image. Additionally, a random combination of image processing operations have 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 Fig. 5.

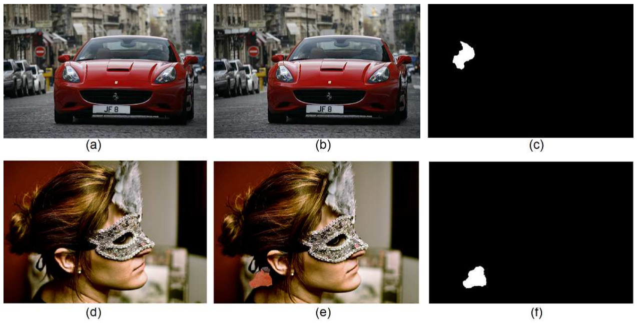


Figure 5: A few samples from the synthetically generated dataset. On left is shown the real image, in middle the manipulated image (JPEG and noise used), and on right the binary mask localizing the manipulated area. Sometimes the manipulated area is not visible by naked eye (e.g., (b)).

To synthetically manipulate the second half, we employed advanced methods such as GAN or Inpainting. Specifically, the following methods have been used to manipulate images: built-in OpenCV inpainting function, inpaining method proposed in [61], and FaceApp [19] 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 Fig. 6 and Fig. 7.

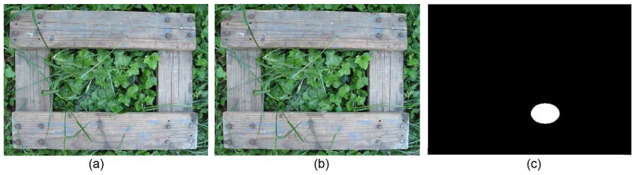


Figure 6: On left is shown the real image, in middle the manipulated image (using an Inpainting method [61]), and on right the binary mask localizing the manipulated area.

To summarize, this dataset is formed by 70,000 images. Half of them are real and the second half have been manipulated in a controlled manner. Binary masks of all ma-

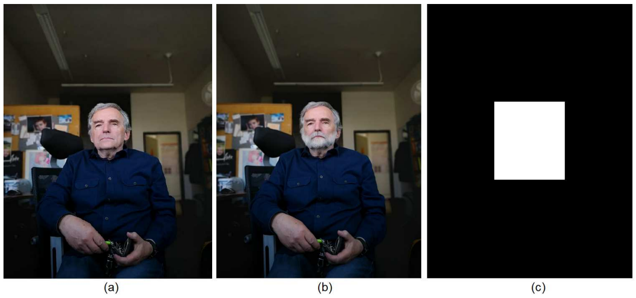


Figure 7: On left is shown the real image, in middle the manipulated image (using FaceApp [19]), and on right the binary mask localizing the manipulated area. It is interesting to note that although the visible area of manipulation of FaceApp is typically inside the face area, pixels of a larger rectangular area around the face gets modified as a result of face transform.

nipulated images localizing the manipulated areas are also provided.

4.1. Real-Life Manipulated Images

We also collected a large set of real-life (uncontrolled) manipulated images from the Internet (for example, see Fig. 8). Specifically, 2,000 manipulated images created by random people have been downloaded (URL of most images were obtained from [25]). 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 8 and 9.

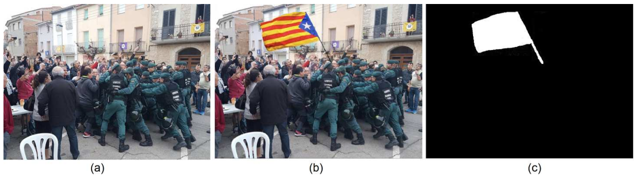


Figure 8: A real-life manipulated image. On left is shown the real image, in middle the manipulated image, and on right the binary mask localizing the manipulated area.

5. Experiments

We now demonstrate quantitative results of a few popular image forensic methods on the collected real-life dataset. We applied the following methods on our dataset: NOI1 [40], CFA1 [22], BLK [33], ADQ1 [35], and ManTraNet [60]. 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) [50]. Results are shown in Fig. 10 and Tab 2.

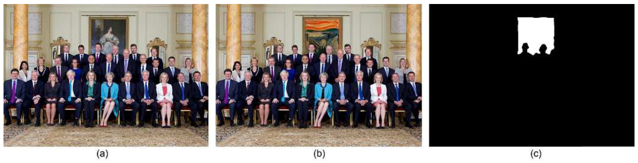
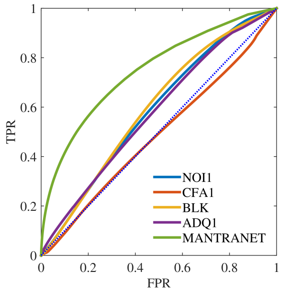


Figure 9: A real-life manipulated image. On left is shown the real image, in middle the manipulated image, and on right the binary mask localizing the manipulated area. Binary masks of real-life manipulated images have been created manually.



Method	AUC
NOI1 [40]	58.6%
CFA1 [22]	48.7%
BLK [33]	59.6%
ADQ1 [35]	57.9%
ManTraNet [60]	74.8%

Figure 10 & Table 2: Obtained ROC and AUC.

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.

6. Conclusion

In this work, we collected two large-scale and diverse datasets with a high variety of artifacts. We hope collected datasets will contribute to facilitating future research on training and testing methods and detection of manipulated visual content. Both datasets are made publicly available for the research community at http://staff.utia.cas.cz/novozada/db.

Acknowledgement

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

References

[1] J. H. A. Piva, A. Rocha. The first ifs-tc image forensics challenge. 5 2013.

[2] I. Amerini, L. Ballan, R. Caldelli, A. Del Bimbo, and G. Serra. A sift-based forensic method for copy-move attack detection and transformation recovery. *Information Forensics and Security, IEEE Transactions on*, 6:1099 – 1110, 10 2011.

[3] J. H. Bappy, C. Simons, L. Nataraj, B. S. Manjunath, and A. K. Roy-Chowdhury. Hybrid LSTM and encoder-decoder architecture for detection of image forgeries. *CoRR*, abs/1903.02495, 2019.

[4] M. J. Bappy, A. Roy-Chowdhury, J. Bunk, L. Nataraj, and B. Manjunath. Exploiting spatial structure for localizing manipulated image regions. 10 2017.

[5] M. Barni, A. Costanzo, and L. Sabatini. Identification of cut & paste tampering by means of double-jpeg detection and image segmentation. pages 1687 – 1690, 07 2010.

[6] B. Bayar and M. Stamm. A deep learning approach to universal image manipulation detection using a new convolutional layer. pages 5–10, 06 2016.

[7] B. Bayar and M. Stamm. Constrained convolutional neural networks: A new approach towards general purpose image manipulation detection. *IEEE Transactions on Information Forensics and Security*, PP:1–1, 04 2018.

[8] S. Bayram, T. Sencar, and N. Memon. An efficient and robust method for detecting copy-move forgery. pages 1053–1056, 04 2009.

[9] L. Bondi, S. Lameri, D. Guera, P. Bestagini, E. Delp, and S. Tubaro. Tampering detection and localization through clustering of camera-based cnn features. pages 1855–1864, 07 2017.

[10] J. Bunk, J. H. Bappy, T. M. Mohammed, L. Nataraj, A. Flenner, B. S. Manjunath, S. Chandrasekaran, A. K. Roy-Chowdhury, and L. Peterson. Detection and localization of image forgeries using resampling features and deep learning. *CoRR*, abs/1707.00433, 2017.

[11] H.-Y. Choi, H.-U. Jang, D. Kim, J. Son, S.-M. Mun, S. Choi, and H.-K. Lee. Detecting composite image manipulation based on deep neural networks. pages 1–5, 05 2017.

[12] D. Cozzolino, J. Thies, A. Rössler, C. Riess, M. Nießner, and L. Verdoliva. Forensictransfer: Weakly-supervised domain adaptation for forgery detection. *arXiv*, 2018.

[13] D. Cozzolino and L. Verdoliva. Noiseprint: A cnn-based camera model fingerprint. *IEEE Transactions on Information Forensics and Security*, PP:1–1, 05 2019.

[14] X. Cun and C.-M. Pun. *Image Splicing Localization via Semi-global Network and Fully Connected Conditional Random Fields: Subvolume B*, pages 252–266. 01 2019.

[15] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and F. F. Li. Imagenet: a large-scale hierarchical image database. pages 248–255, 06 2009.

[16] A. E. Dirik, S. Bayram, H. Sencar, and N. Memon. New features to identify computer generated images. volume 4, pages IV – 433, 01 2007.

[17] J. Dong, W. Wang, and T. Tan. Casia image tampering detection evaluation database. pages 422–426, 07 2013.

[18] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision*, 111(1):98–136, Jan. 2015.

[19] FaceApp. `https://play.google.com/store/apps/details?id=io.faceapp`.

[20] Faceswap. https://github.com/ marekkowalski/faceswap.

[21] H. Farid. Exposing digital forgeries from jpeg ghosts. *Information Forensics and Security, IEEE Transactions on*, 4:154 – 160, 04 2009.

[22] P. Ferrara, T. Bianchi, A. De Rosa, and A. Piva. Image forgery localization via fine-grained analysis of cfa artifacts. *Trans. Info. For. Sec.*, 7(5):1566–1577, Oct. 2012.

[23] A. Ghosh, Z. Zhong, T. E. Boult, and M. Singh. Splicer-adar: A learned method for blind image forensics. *CoRR*, abs/1906.11663, 2019.

[24] H. Guan, M. Kozak, E. Robertson, Y. Lee, A. Yates, A. Delgado, D. Zhou, T. Kheyrkhah, J. Smith, and J. Fiscus. Mfc datasets: Large-scale benchmark datasets for media forensic challenge evaluation. pages 63–72, 01 2019.

[25] S. Heller, L. Rossetto, and H. Schuldt. The PS-Battles Dataset – an Image Collection for Image Manipulation Detection. *CoRR*, abs/1804.04866, 2018.

[26] M. Huh, A. Liu, A. Owens, and A. Efros. Fighting fake news: Image splice detection via learned self-consistency. Technical report, 05 2018.

[27] S. Iizuka, E. Simo-Serra, and H. Ishikawa. Globally and locally consistent image completion. *ACM Transactions on Graphics*, 36:1–14, 07 2017.

[28] J. Kalpathy-Cramer, A. García Seco de Herrera, D. Demner-Fushman, S. Antani, S. Bedrick, and H. Müller. Evaluating performance of biomedical image retrieval systems-an overview of the medical image retrieval task at imageclef 2004-2013. *Computerized Medical Imaging and Graphics*, 01 2015.

[29] T. Karras, T. Aila, S. Laine, and J. Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. In *International Conference on Learning Representations*, 2018.

[30] T. Karras, S. Laine, and T. Aila. A style-based generator architecture for generative adversarial networks. pages 4396–4405, 06 2019.

[31] P. Korus and J. Huang. Evaluation of random field models in multi-modal unsupervised tampering localization. 12 2016.

[32] T. Le-Tien, H. Phan-Xuan, T. Nguyen-Chinh, and T. Do-Tieu. Image forgery detection: A low computational-cost and effective data-driven model. *International Journal of Machine Learning and Computing*, 9:181–188, 04 2019.

[33] W. Li, Y. Yuan, and N. Yu. Passive detection of doctored jpeg image via block artifact grid extraction. *Signal Processing*, 89(9):1821 – 1829, 2009.

[34] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. Zitnick. Microsoft coco: Common objects in context. volume 8693, 04 2014.

[35] Z. Lin, J. He, X. Tang, and C.-K. Tang. Fast, automatic and fine-grained tampered jpeg image detection via dct coefficient analysis. *Pattern Recognition*, 42(11):2492 – 2501, 2009.

[36] Y. Liu, Q. Guan, X. Zhao, and Y. Cao. Image forgery localization based on multi-scale convolutional neural networks. pages 85–90, 06 2018.

[37] J. Lukás, J. Fridrich, and M. Goljan. Digital camera identification from sensor pattern noise. *Information Forensics and Security, IEEE Transactions on*, 1:205 – 214, 07 2006.

[38] W. Luo, Z. Qu, J. Huang, and G. Qiu. A novel method for detecting cropped and recompressed image block. volume 2, pages II–217, 05 2007.

[39] B. Mahdian and S. Saic. Blind authentication using periodic properties of interpolation. *Information Forensics and Security, IEEE Transactions on*, 3:529 – 538, 10 2008.

[40] B. Mahdian and S. Saic. Using noise inconsistencies for blind image forensics. *Image and Vision Computing*, 27:1497–1503, 09 2009.

[41] F. Marra, D. Gragnaniello, D. Cozzolino, and L. Verdoliva. Detection of gan-generated fake images over social networks. pages 384–389, 04 2018.

[42] F. Marra, D. Gragnaniello, L. Verdoliva, and G. Poggi. *A Full-Image Full-Resolution End-to-End-Trainable CNN Framework for Image Forgery Detection*, 09 2019.

[43] G. Mazaheri. A skip connection architecture for localization of image manipulations. 06 2019.

[44] T.-T. Ng and S. Chang. A data set of authentic and spliced image blocks. 01 2004.

[45] A. Piva. An overview on image forensics. *ISRN Signal Processing*, 2013, 01 2013.

[46] A. Popescu and H. Farid. Exposing digital forgeries in color filter array interpolated images. *Signal Processing, IEEE Transactions on*, 53:3948 – 3959, 11 2005.

[47] Y. Rao and J. Ni. A deep learning approach to detection of splicing and copy-move forgeries in images. pages 1–6, 12 2016.

[48] REWIND. Reverse engineering of audio-visual content data.

[49] A. Rössler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner. Faceforensics++: Learning to detect manipulated facial images. 01 2019.

[50] R. Salloum, Y. Ren, and C. Kuo. Image splicing localization using a multi-task fully convolutional network (mfcn). *Journal of Visual Communication and Image Representation*, 51, 09 2017.

[51] G. Schaefer and M. Stich. Ucid: An uncompressed color image database. volume 5307, pages 472–480, 01 2004.

[52] Swapme. https://itunes.apple.com/us/app/ swapme-by-faciometrics/. acquired by facebook and no longer available in app-store.

[53] D. Tralic, I. Zupancic, S. Grgic, and M. Grgic. Comofod -new database for copy-move forgery detection. 09 2013.

[54] M. Tschannen, O. Bachem, and M. Lucic. Recent advances in autoencoder-based representation learning. *CoRR*, abs/1812.05069, 2018.

[55] A. M. Turk. `https://www.mturk.com/`.

[56] W. Wang, J. Dong, and T. Tan. Effective image splicing detection based on image chroma. pages 1257–1260, 11 2009.

[57] B. Wen, Y. Zhu, R. Subramanian, T.-T. Ng, X. Shen, and S. Winkler. Coverage - a novel database for copy-move forgery detection. pages 161–165, 09 2016.

[58] Y. Wu, W. Abd-Almageed, and P. Natarajan. Deep matching and validation network: An end-to-end solution to constrained image splicing localization and detection. pages 1480–1502, 10 2017.

[59] Y. Wu, W. Abd-Almageed, and P. Natarajan. *BusterNet: Detecting Copy-Move Image Forgery with Source/Target Localization: 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part VI*, pages 170–186. 09 2018.

[60] Y. Wu, W. AbdAlmageed, and P. Natarajan. Mantra-net: Manipulation tracing network for detection and localization of image forgeries with anomalous features. pages 9535–9544, 06 2019.

[61] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. Huang. Generative image inpainting with contextual attention. pages 5505–5514, 06 2018.

[62] N. Yu, L. Davis, and M. Fritz. Attributing fake images to gans: Analyzing fingerprints in generated images. *CoRR*, abs/1811.08180, 2018.

[63] X. Zhang, S. Karaman, and S. Chang. *Detecting and Simulating Artifacts in GAN Fake Images*, 07 2019.

[64] Z. Zhang, Y. Zhang, Z. Zhou, and J. Luo. Boundary-based image forgery detection by fast shallow cnn. pages 2658–2663, 08 2018.

[65] P. Zhou, X. Han, V. Morariu, and L. Davis. Learning rich features for image manipulation detection. pages 1053–1061, 06 2018.

[66] P. Zhou, X. Han, V. I. Morariu, and L. S. Davis. Two-stream neural networks for tampered face detection. *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1831–1839, 2017.

[67] J.-Y. Zhu, T. Park, P. Isola, and A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. pages 2242–2251, 10 2017.