Joren *et al.*
EURASIP Journal on Image and Video Processing 2022, **2022**(1):2
https://doi.org/10.1186/s13640-022-00579-5

EURASIP Journal on Image and Video Processing

RESEARCH Open Access

Printing and scanning investigation for image counter forensics

Hailey Joren[1*], Otkrist Gupta[1] and Dan Raviv[1]

*Correspondence:
hailey.joren@lendbuzz.com

[1] Lendbuzz, 100 Summer St Suite 3150, Boston, MA 02110, USA

Abstract

Examining the authenticity of images has become increasingly important as manipulation tools become more accessible and advanced. Recent work has shown that while CNN-based image manipulation detectors can successfully identify manipulations, they are also vulnerable to adversarial attacks, ranging from simple double JPEG compression to advanced pixel-based perturbation. In this paper we explore another method of highly plausible attack: printing and scanning. We demonstrate the vulnerability of two state-of-the-art models to this type of attack. We also propose a new machine learning model that performs comparably to these state-of-the-art models when trained and validated on printed and scanned images. Of the three models, our proposed model outperforms the others when trained and validated on images from a single printer. To facilitate this exploration, we create a data set of over 6000 printed and scanned image blocks. Further analysis suggests that variation between images produced from different printers is significant, large enough that good validation accuracy on images from one printer does not imply similar validation accuracy on identical images from a different printer.

Keywords: Computer vision, Adversarial attack, Image forensics

1 Introduction

Determining the authenticity of an image is becoming increasingly important for legal proceedings, criminal investigations, and verifying identity-supporting documents. In recent years, convolutional neural networks (CNNs) have been employed to detect image manipulations, ranging from identifying splicing and copy–move forgeries [1] to manipulations such as contrast enhancement [2, 3], resampling [4], JPEG compression [5], Gaussian blurring [6, 7], median filtering [8], and Additive White Gaussian Noise [9]. Of this latter group, some may be either innocuously applied or maliciously included [8].[1]

More recently, research in image forensics has included the presence of an adversary, a situation in which the vulnerability of CNNs has been well-studied [10]. In regards to image forensics, consideration of these adversarial attacks have been primarily limited to pixel-based adversarial examples and JPEG double compression

[1] Global manipulations such as these can be used on their own to obscure features of an image or used in tandem with another type of manipulation such as copy–move or splicing. We adopt the approach used other researchers in this area in investigating these types of global image manipulations alone (see Sect. 2).

[5, 11, 12]. In pixel-based attacks, an adversary with knowledge of the CNN model in deployment can craft an "attacked" image which appears visually identical to the original image, but is mislabeled by the CNN [13]. This problem is well known in computer vision and has been at the forefront of recent work in field. However, this type of attack demands a certain level of expertise by the adversary, and is unlikely to be employed in a majority of cases in image forensics. Even for skilled adversaries, constructing pixel-based adversarial attacks is often labor-intensive, and recent work has cast doubt on the transferability of adversarial attacks in image forensics applications [14]. While pixel-based adversarial attacks require at least some knowledge of the model, a low-level adversarial manipulation such as double JPEG compression requires no such knowledge [11]. In this type of attack, the images are simply JPEG compressed after the manipulation has been applied, hampering the model's ability to correctly identify post-processing methods such as Additive White Gaussian Noise or median filtering [11]. For this reason, building models robust to low-level, simple adversarial manipulations such as JPEG double compression, to which several manipulation detection models have been found to be vulnerable [5, 11, 12], is particularly important. The goal of this paper is to investigate the vulnerability of state of the art models to another kind of low-level adversarial manipulation: printing and scanning. To our knowledge this is the first investigation into adversarial attack in digital image manipulations through printing and scanning.

In physical forgery, repeated printing and scanning can be used to obscure manipulations or watermarks. A document may be modified, usually non-digitally, and then repeatedly printed and scanned to disguise the manipulation artifacts. While scanning a printed document is not always related to forgery, it is reasonable to expect that state-of-the-art models be impervious to this type of post-processing, as is noted in related work in double JPEG compression [5, 11]. In addition, unlike complex pixel-based adversarial attacks, simply printing and scanning an image is both low-cost and requires little expertise, similar to JPEG compression.

In this paper, we limit our investigation to globally-applied manipulations, such as Gaussian Blurring (GB), Additive White Gaussian Noise (AWGN), and median filtering (MF), rather than local manipulations such as copy–move or splicing, as in related work [9]. We construct printed and scanned data sets from three different printers and experiment with two state-of-the-art models, as well as our own model. Related to our work is research involved in identifying camera models [15]—we additionally report results for identifying printer model. Our main contributions include the following:

- We conduct the first analysis into the vulnerability of image manipulation detectors to printing and scanning, demonstrating that at least two state-of-the-art models are vulnerable to this type of highly plausible and inexpensive attack
- We propose a model architecture which performs comparably than the state-of-the-art models when trained and evaluated on printed and scanned images, including performing 5% better when trained on images from a single printer
- We conduct an in-depth analysis on the relationship between CNN-based image manipulation detectors, including training on composite data sets, and plan to

share our data set of over 6000 printed and scanned images with the community to facilitate further investigation

The rest of the paper is organized as follows. In Sect. 2, we give context and background through related work. In Sect. 3.1, we describe our model architecture, as well as those of the models we used for comparison. In Sect. 3.5, we describe the data sets used for training and validation. In Sect. 4, we explain the experiments conducted, and in Sect. 5, we discuss the results of these experiments. The paper ends in Sect. 6, where we summarize our conclusions and suggest areas of future research.

2 Related work

As this paper primarily investigates manipulation detectors based on convolutional neural networks (CNNs), we provide background on CNN-based manipulation detectors. Similarly, we provide context on adversarial attacks on CNNs generally as well as specifically on CNN-based image manipulation detectors.

Related to this work is work on detecting manipulations through inconsistencies in lighting [16] and despite various compression qualities [17]. Additionally, [4] contributes significantly to this problem area, though without examining models that leverage deep learning. [17] explores a similar problem, but without addressing specifically the problem of printing and scanning in relation to CNN-based detectors, and is thus complementary to this work.

2.1 Deep learning for image forensics

Recent methods in image forensics techniques leveraging deep learning have reached impressive performance. In 2015, a CNN-based classifier was proposed for detecting median filtering in images [18]. Building on this work, [9] proposed CNN-based model with the addition of a "constrained convolutional layer", or a layer constrained to learn the high-pass features of an image by attempting to predict a central pixel based on its neighbors. This serves to suppress the image content while learning the manipulation fingerprint, drawing inspiration from Steganalysis Rich Model (SRM) filters in steganalysis [19]. In recent years, procedural similarities between SRM filters and learned CNN layers have been noted [20] and SRM filters have been used as a foundation for additional steganalytic and forensic methods. Accordingly, CNN architectures have been specially designed to account for SRM-like features, including methods that leverage absolute-value functions and TanH activation to learn steganalysis relevant features [21]. Additional methods include designing networks specifically for cases in which SRM filters yield weak signal [22], as well as methods that alter pool and stride hyperparameters in these cases [23].

Additionally, recent work has shown that third order subtractive pixel analysis matrix (S3SPAM) features can be learned by a simple shallow CNN, and can employ transfer learning to achieve good performance on little training data [24]. In addition to directly detecting manipulations, a deep learning method for analyzing the image processing history as an important component for image forensics has been proposed, as the processing history pipeline can affect the accuracy of other forensic tools [25].

The performance achieved by constrained convolutional layers and particularly deep networks is particularly impressive. These techniques serve as inspiration for our proposed model, and we thus compare our proposed model with models that leverage these modifications.

2.2 Adversarial attacks on CNNs

The vulnerability of CNNs to adversarial attacks has been well documented [13, 26]. Adversarial noise can be designed in such a way that, when added to the image, can retain visual quality while misleading the classifier. For example, Fast Gradient Sign Method (FGSM) [13] leverages the differentiability of the loss function, assumed to be known to the adversary. The method proposes altering each pixel based on the gradient of the loss with respect to the original pixels in the input image. These changes small are enough such that the resulting image is visually nearly identical to the original, but are large enough cumulatively to increase the loss such as to impair the classification. Similarly, Projected Gradient Descent (PGD) [27] seeks a perturbation that maximises the loss on a specific input while keeping the perturbation size smaller than a given epsilon. DeepFool [28] uses a local linearization of the classifier to approximate the decision boundary and alter the images accordingly. The Jacobian-based Saliency Map Attack (JSMA) [26] uses a greedy iterative procedure, altering only the pixels which contribute most to the correct classification as identified by a saliency map. Each of these pixel-based adversarial attacks, while effective, requires at least partial knowledge of the network used for image manipulation detection. In contrast, low-level adversarial attacks such as JPEG compression or printing and scanning, the subject of this paper, require no such knowledge.

2.3 Adversarial attacks in image forensics

While CNN-based classifiers have achieved high performance on benchmark image forensic tasks, recent research in computer vision has demonstrated that CNN-based manipulation detectors, like CNNs more broadly, are highly vulnerable to adversarial attacks. For example, in [29], the authors demonstrate that a GAN-based architecture can conceal 3x3 median filtering manipulation, one of the manipulations we explore in this paper. This type of adversarial attack causes a detector to label the image as non-manipulated, including for the CNN-based detectors proposed in [9] and [18]. Additionally, a method of adversarial attack based on small pixel-based distortions has been proposed for fooling global image manipulation detectors [30]. However, [31] notes that unlike in most pattern recognition tasks, pixel-based adversarial attacks such as Fast Gradient Sign Method (FGSM) [13] and Jacobian-based Saliency Map Attack (JSMA) [26], are not for the most part transferable between manipulation detection models.

Recent work has explored the vulnerability of image manipulation detectors to low-resolution median filtering [32] and JPEG compression [11, 12, 33]. To our knowledge, ours is the first paper to examine model vulnerability to printing and scanning.

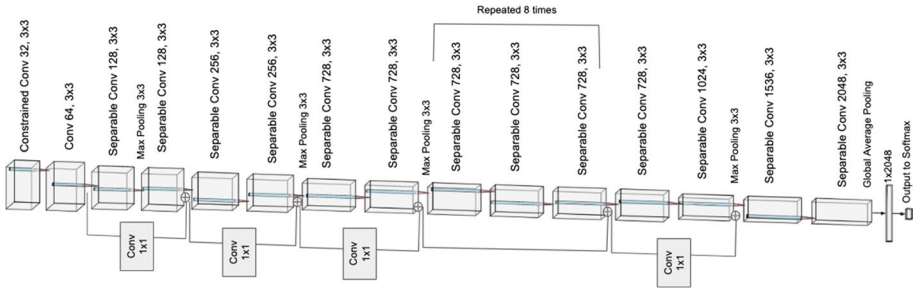

Fig. 1 Proposed network architecture. The first layer is a constrained convolutional layer to extract SRM features, followed by an deep architecture with separable convolutional layers to improve generalization. The last layer is either a 4×1 vector (for four classes) or a 6×1 vector (for all six classes)

3 Methods

3.1 Models

Here we describe our proposed model architecture for improved performance on printed and scanned images. We compare our model's performance with the model proposed in [9], the inspiration for the constrained convolutional layer. We additionally compare our model with XceptionNet (Xception) [34], as it and our proposed model have nearly identical number of parameters and similar architecture, so the difference in performance cannot be attributed to increased network capacity.

3.2 Proposed model

Our proposed architecture consists of one constrained convolutional layer [9], 1 convolutional layer, 34 separable convolutional layers, 5 pooling layers (4 max pooling, 1 global average pooling), and a final fully connected layer (see Fig. 1). Each convolutional layer was followed by ReLU activation, and max pooling layers were performed with a stride of 2×2.

In the constrained convolutional layer, a 5×5 filter is employed in which the sum of all the weights is constrained to be zero [9]. Specifically, the center pixel is predicted by the rest of the pixels in the field, and the output of the filter can be interpreted as the prediction error, as suggested by research in steganalysis [25]. Specifically, the weights in the filter are constrained such that:

$$\begin{cases} w(0,0) & = -1 \\ \sum_{l,m \neq 0} w(l,m) & = 1 \end{cases}$$

where w refers to the weight, and l and m refer to the coordinates in the filter, where 0, 0 is the central weight.

The purpose of the constrained convolutional layer is to constrain the model to learn image manipulation fingerprints, rather than image content and higher order features, such as those useful for object recognition and classification tasks. The prediction error fields are then used as low-level forensic trace features by the rest of the network to assist in classifying global image manipulation detection.

For the separable convolutional layers, a spatial convolution is performed independently for each channel and is followed by a point-wise or 1×1 convolution, as

proposed in [34]. These components decrease the number of free parameters allowing the deep network to learn effectively even with a small training set, making it particularly appropriate for our investigation.

In this approach, we hope to leverage both the SRM-like features produced by the convolutional layer as well as the improved generalization ability provided by the added depth and separable layers.

3.3 Bayar2016

Proposed in 2016, the constrained convolution method of image manipulation detection, hereafter referred to at Bayar2016, proposes a three-layer CNN, with two max-pooling layers and three fully-connected layers (including the initial constrained convolutional layer) [9]. This model demonstrates impressive results in discerning between the six manipulations investigated in this paper using the data set described in the next section, achieving 99.9% validation accuracy.

3.4 Xception

In addition to a the Bayar2016 shallow network, recent work has demonstrated that increasing network depth can dramatically improve model generalization. To compare with a model of similar depth that also uses separable convolutional layers, we experiment with XceptionNet, a deep network comprising of 42 layers, including separable convolutional layers [34]. The network design is built upon Inception architecture [35], with the innovation of separable filters. Similar to Bayar2016, this model also achieves near 99% accuracy on the data set described in [9] before printing and scanning. While a variety of popular deep learning models could be appropriate for comparison, we compare with Xception due to (1) its comparable architecture and number of parameters and (2) its demonstrated image classification performance, performing in the top 1% accuracy on ImageNet [34, 36].

3.5 Data sets

For accurate comparison, we follow the procedure described in [9], using images from the first IEEE IFSTC Image Forensics Challenge as described by [37]. The portion of the data set used consists of 3334 images of size 1024×768, which was further split into training, validation and testing data. The images are captured from several different digital cameras of both indoor and outdoor scenes.

3.6 Printing and scanning

We used three different printers and one scanner to create a data set of printed and scanned images: one Dell S3845CDN Laser Multifunction Printer, one Xerox Altalink C8070 Multifunction Printer, and one Xerox WorkCentre 7970 Multifunction Printer, which we refer to as Dell, Xerox1 and Xerox2 respectively hereafter. We printed 50 images of each manipulation type on each printer and used the Dell scanner to scan each image (see Fig. 2). After scanning and extracting the images from the resulting

Fig. 2 Pristine image before (left) and after (right) printing and scanning on Xerox1 (Xerox Altalink C8070 Multifunction Printer). We note that there is significant variation between the two images, similar to that introduced by the global manipulation methods with which we experimented

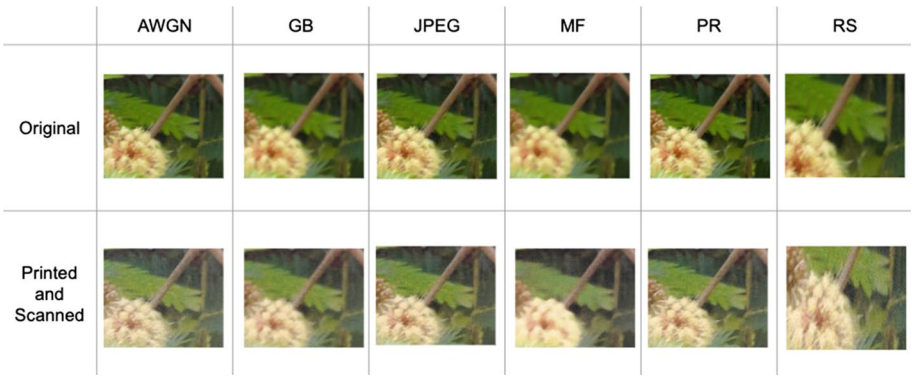

Fig. 3 Examples of manipulations before and after printing and scanning. The six manipulations refer to Additive White Gaussian Noise (AWGN), Gaussian blurring (GB), JPEG compression (JPEG), median filtering (MF), Pristine or no manipulation (PR) and bilinear resampling (RS). We note that due to the algorithms employed, JPEG compression and resampling might be reasonably similar to the printing and scanning process. For this reason, we additionally train and evaluate the models on a restricted set of four classes only, excluding JPEG and bilinear resampling. See Table 1 for details on the parameters used for each manipulation

pdfs, the image sizes were 1700 x 2200 pixels, which was then center-cropped to 1536 $\times$ 1792 to remove the white border added by the scanning process. Each image was then split into 42 299 $\times$ 299 blocks (or 256 $\times$ 256 blocks for Bayar2016), resulting in 2142 image blocks of each class from each printer (see Fig. 3). We limited our data creation to 900 full-page color images both for budget constraints and environmental concerns; creating a synthetic data set through printing and scanning simulation may be an avenue of future work.

3.7 Manipulations

Again following the procedure described in [9], we manipulated each image with each of six manipulation types: Additive White Gaussian Noise (AWGN), Gaussian blurring (GB), JPEG compression (JPEG), median filtering (MF), re-sampling (RS) and retaining the Pristine image (PR).

- Additive white Gaussian noise constructs a noise matrix of the same shape as the image according to a normal distribution with a given sigma value and adds this matrix to the original image. The result is then normalized to values between 0 and 255.

- Gaussian blurring blurs the image using a Gaussian filter by convolving the input image using a given kernal.
- JPEG compression is a lossy compression method which compresses the image through converting the color map, down-sampling and Discrete Cosine Transform (DCT).
- Median filtering replaces each pixel with the median value of the neighboring pixels using a given kernal area.
- Bilinear resampling works similarly, resizing the image using the distance-weighted average of the neighboring pixels to estimate the new pixel value.

See Table 1 for manipulation parameter details.

Table 1 Parameter specifications for each manipulation type

Manipulation	Hyperparameters
Additive White Gaussian Noise (AWGN)	$\sigma = 2.0$
Gaussian blurring (GB)	Kernal size = (5,5) $\sigma = 1.1$
JPEG compression (JPEG)	Quality = 70
Median filtering (MF)	Kernal size = (5,5)
Pristine (PR)	None
Bilinear resampling (RS)	Ratio = 1.5

We used the same parameters as in [9] for fair comparison. See Sect. 3.7 for details on the manipulations

Table 2 Descriptions and sizes of each data set used for training and validation

Data set name	Description	Size
Original	IFSTC data set after six manipulations	198,624
Xerox1	Images from IFSTC data set with manipulations after being printed and scanned on Xerox1	2142
Composite Printers	Combined set of images from each printer (balanced)	6426
Composite Full	Combined set of images from each printer plus original IFSTC images (balanced)	8568
Printer Identification	Printer identification and pristine images after being printed and scanned by all three printers	3213
JPEG Compression	IFSTC data set with JPEG compression (QF=80) on all images	198,624

Size refers to the number of 299 × 299 or 256 × 256 image blocks in each data set, which is then split in 75% training and 25% validation. X1 and X2 refer to Xerox1 and Xerox2 printers (Sect. 3.6), respectively. The labels refer to the labels used when training and evaluating on each data set

4 Experiments

We trained each model (our proposed model, Bayar2016, and Xception) on a variety of training sets and evaluated each trained model on multiple validation data sets (see Table 2).

We first investigated the extent to which our selected models can correctly classify the validation images after printing and scanning. We trained each model on the original data set (before printing and scanning) with all six classes: Additive White Gaussian Noise (AWGN), gaussian blurring (GB), JPEG compression (JPEG), Median Filtering (MF), Bilinear Resampling (RS) and Pristine or no manipulation (PR). For a more complete analysis, we removed the Bilinear Resampling (RS) and JPEG compression (JPEG) classes from the training and validations sets and retrained the models, as these two classes could intuitively be considered similar to changes introduced during the printing and scanning process (see Table 3).

Second, we explored countering this vulnerability by training on the printed and scanned image blocks [13]. We trained each model on the printed and scanned image blocks from a single printer. The data set (see Table 2, Xerox1) consists of 50 full images (1700×2200 pixels), which were then divided into 299×299 for our proposed model and Xception, and 256×256 for the Bayar2016 model. This resulted in 2142 image blocks for each data set, which was divided into training and validation sets of size 1722 and 420 respectively, using only the central images to avoid including border artifacts from the scanning process.

Third, we created composite data sets, one consisting of all printed and scanned image blocks (from all three printers), and the other consisting of all printed and scanned image blocks as well as a number of image blocks from the original data set (before printing and scanning), at a size equivalent to those from one of the three printers. The first composite data set, which we refer to as Composite Printers, consists of 6426 image blocks (printed and scanned only), while the second consists of 8568 image blocks (75% printed and scanned, 25% original). The goal of this experiment was to evaluate if the poor accuracy fitting the printed and scanned data could be mitigated by dramatically increasing the size of the training set.

Table 3 Validation accuracy for various validation sets after training on IFSTC data set

	Bayar2016	Xception	Proposed model
Original (6c)	0.9979	0.9916	0.993
Dell (6c)	0.1643	0.1632	0.1673
Xerox1 (6c)	0.1976	0.201	0.1827
Xerox2 (6c)	0.1972	0.202	0.1953
Original (4c)	0.9948	0.9954	0.997
Dell (4c)	0.2571	0.223	0.2347
Xerox1 (4c)	0.2411	0.246	0.2367
Xerox2 (4c)	0.2387	0.255	0.2393
JPEG (4c)	0.4255	0.5126	0.4825

We note that although all three models perform exceptionally well on the original IFSTC data set, each performs little better than random when evaluated on images from any of the three printers. Because JPEG compression and Bilinear Resampling (RS) could be reasonably inferred to be similar to printing and scanning, we remove these classes and train and evaluate on a restricted set of four classes (4c) (see Sect. 4). Despite this restricted set of manipulations, however, the models perform no better than random

Finally, we evaluated the performance of each of the models on identifying the printer of printed and scanned images (see Table 2, Printer Identification).

4.1 Hyperparameters

For Bayar2016, we used a batch size of 64, an initial learning rate of 0.01, stochastic gradient descent (SGD) with momentum 0.95, weight decay 0.0005, gamma 0.7, and step size 6.

We used similar hyperparameters for Xception and our proposed model. Specifically, for both models, we use the pre-trained weights from the network as trained on ImageNet. We again used SGD, and inferred the batch size and learning rate at training time based on the number of GPUs, using

$$batch\_size = 4 \times num\_gpus$$

for the batch size and 0.01 for the initial learning rate. We use momentum 0.9 and weight decay 0.0005. For learning rate decay, we use polynomial decay as described in [38]. For each model, we trained until the validation accuracy plateaued or began to fall.

Following the original methodology for Bayar2016, we retain only the green color layer of each image and divide into 256×256 non-overlapping blocks, retaining nine central blocks. For our proposed model and for Xception, we retain all three color channels and split the images into 299×299 non-overlapping blocks, according to the input size of the original architecture (Figs. 4, 5).

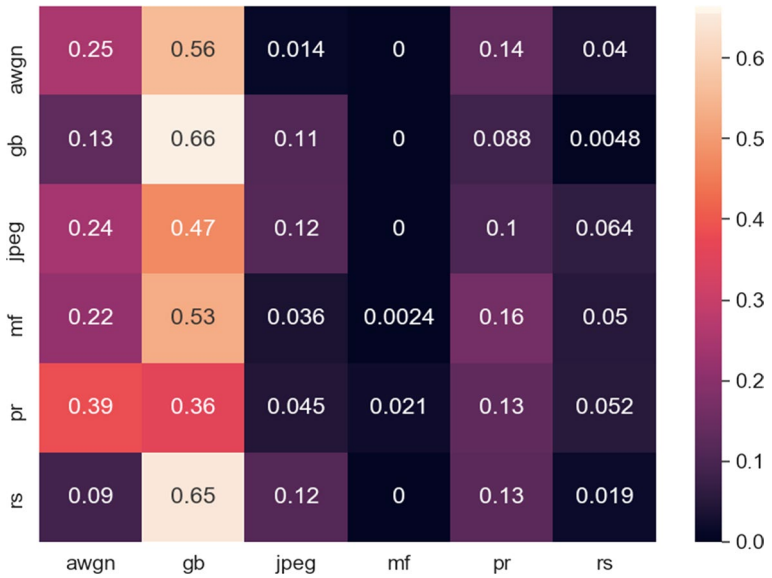

Fig. 4 Confusion matrix for Bayar2016 trained on original IFSTC, evaluated on Xerox1 (see Table 2, Xerox1). We note that despite the high reported validation accuracy on the original data set, the model struggles to distinguish between the classes after printing and scanning

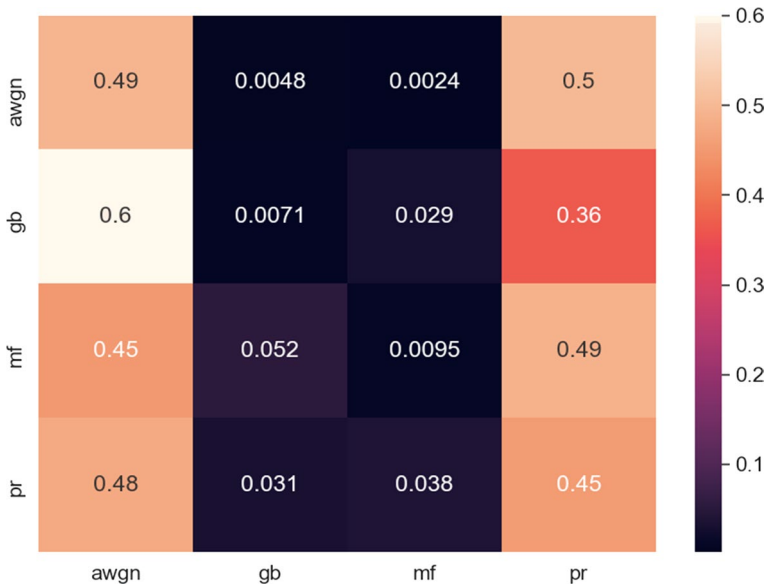

Fig. 5 Confusion matrix for Bayar2016 trained on Original IFSTC (without RS and JPEG), evaluated on Xerox1 (see Table 2). We investigate the model's performance after removing bilinear resampling (RS) and JPEG compression, but find that it still performs little better than random

5 Results and discussion

5.1 Print-scan manipulation

To evaluate the general vulnerability of each of the models to printed and scanned images, we trained on the original IFSTC data set (before printing and scanning) and evaluated each model on validation sets from each of the three printers. When we evaluated the models on the printed and scanned validation sets, we found that each model performed only slightly better than random.

We additionally removed the bilinear resampling (RS) and JPEG compression classes, and found that the resulting models are similarly unable to correctly classify the remaining four manipulations, still performing at or below random. We additionally note that the models perform worse on the printed and scanned validation images than on the validation images after JPEG compression, a known vulnerability of these types of models, indicating that printing and scanning may be more effective at masking the manipulations [17] (see Table 3).

5.2 Cross-training on printed and scanned examples

We additionally trained each model on printed and scanned images from an individual printer (Xerox1) (see Sect. 3.6).

We note that Bayar2016 and Xception achieve accuracies 66.6% and 70.4% respectively, while our proposed model is able to achieve an accuracy of 75.3%. It also appears that training on one printer does not lend itself to similar validation accuracy on examples from another printer, even of the same make (see Table 4).

Table 4 Validation accuracy for various validation sets after training on Xerox1 data set (see Table 2, Xerox1)

	Bayar2016	Xception	Proposed model
Xerox1 (4c)	0.7036	0.666	**0.753**
Dell (4c)	0.3018	0.482	0.456
Original (4c)	0.2342	0.3873	0.3649
Xeros2 (4c)	0.4738	0.611	0.572
JPEG (4c)	0.2418	0.3848	0.364

Bold value refer to models that perform better than the rest - to highlight the model performance - its a common practice to do this and usually helps improve readability

We trained each model on images from only the Xerox1 data set, or images after being printed and scanned on the first Xerox printer. We find that while no model is able to perfectly fit the printed and scanned data set, our proposed models significantly outperforms the current state-of-the-art models. We also note that transferability to other printers remains weak, indicating significant variance between the printers. Here 4c indicates that we used the restricted set of manipulations (AWGN, GB, MF, and PR) (see Sect. 4)

5.3 Composite training

To compensate for the small size of the data set for each printer alone, we created a composite data set, consisting of all of the printed and scanned examples (total size 6426 blocks), which we refer to as Composite Printers. However, we found that training on this composite data set did not improve validation performance on any single printer compared with training on images from that printer alone. While this is possibly due to a still insufficiently large training data set, it also likely provides further evidence that the difference between printers and scanners may be significant enough to preclude fitting a general printed and scanned data set (see Table 5).

For completion, we additionally created another composite data set, which we refer to as Composite Full, which consists of the same composition as Composite Printers plus an equivalent number of examples from the original data set (total size 8568), and found similar results (see Table 6).

Table 5 Validation accuracy for various validation sets after training on the composite printers data set

	Bayar2016	Xception	Proposed model
Dell (4c)	0.6506	0.649	**0.713**
Xerox1 (4c)	**0.7001**	0.626	**0.696**
Xeros2 (4c)	0.5381	0.623	**0.663**
JPEG (4c)	0.2643	0.2902	0.2601
Original (4c)	0.2617	0.2847	0.2449

Bold values refer to models that perform better than the rest - to highlight the model performance - its a common practice to do this and usually helps improve readability

One possible explanation for the poor validation accuracy on a single printer could be the small size of the data set. To investigate this, we combine the images from all three printers for training, but note that performance on a single printer does not improve. Here 4c indicates that we used the restricted set of manipulations (AWGN, GB, MF, and PR) (see Sect. 4)

Table 6 Validation accuracy for various validation sets after training on the Composite Printers data set

	Bayar2016	Xception	Proposed model
Dell (4c)	0.6339	**0.662**	**0.661**
Xerox1 (4c)	0.6982	0.632	0.674
Xerox2 (4c)	0.5637	0.602	0.696
JPEG (4c)	0.519	0.6374	0.4972
Original (4c)	0.8063	0.9259	0.9629

Bold values refer to models that perform better than the rest - to highlight the model performance - its a common practice to do this and usually helps improve readability

For a complete analysis, we add additional image blocks (blocks before printing and scanning) to the composite data set, but again find that performance does not improve. Here 4c indicates that we used the restricted set of manipulations (AWGN, GB, MF, and PR) (see Sect. 4)

Table 7 Validation accuracy for printer identification by model

	Bayar2016	Xception	Proposed model
Printer identification	0.9048	0.956	0.9533

We investigate the variation of the images between printers by training each model to discern between printers. The high accuracy indicates that the images produces by each printer vary significantly

5.4 Printer identification

For comparison with work on camera model identification, we additionally experimented with printer identification on each of the three printers using the discussed models, and found that the models could distinguish between images from the printers with up to 95% accuracy. This is particularly impressive considering the accuracies were achieved using a relatively small set of training data (2410 image blocks) and without any additional meta-data (see Table 7), indicating significant variance between the artifacts introduced by each printer [15].

6 Conclusions

We investigated the robustness of current state-of-the-art image manipulation detection models in the context of printing and scanning, and found that these models perform poorly on printed and scanned image data. We proposed a model architecture which performs 5% better than the state-of-the-art models when trained and evaluated on images from a single printer. We constructed a data set of over 6000 printed and scanned image blocks which we plan to release to the community for further investigation.

That current state-of-the-art models are vulnerable to printing and scanning is an important finding given the availability and ease of printing and scanning images versus constructing complex adversarial examples.

Further analysis suggest that the variability between images produced by each printer is large, significant enough for the models to easily distinguish between printers and for models trained on a single printer to generalize poorly to images from another printer. This conclusion may create additional challenges in designing models robust to printing and scanning, and sets it apart from work on creating models robust to more uniform and predictable JPEG compression. Future work may include developing methods to simulate printing and scanning in order to create a larger data sets for training the models.

Abbreviations

AWGN	Additive White Gaussian Noise
Bayar2016	Model described in [9]
CNN	Convolutional neural network
Dell	Dell S3845CDN laser multi-function printer
FGSM	Fast gradient sign method
GAN	Generative adversarial network
GB	Gaussian blurring
GPU	Graphics processing unit
IFSTC	IEEE IFSTC image forensics challenge data set as described in [37]
JPEG	JPEG compression/JPEG compressed
JSMA	Jacobian-based saliency map attack
MF	Median filtering
PGD	Projected gradient descent
PR	Pristine or no manipulation
RS	Bilinear resampling
SGD	Stochastic gradient descent
SRM	Steganalysis rich model
Xception/XceptionNet	Model described in [34]
Xerox1/X1	Xerox Altalink C8070 multi-function printer
Xerox2/X2	Xerox Work Centre 7970 multi-function printer

Acknowledgements
We thank Lendbuzz for providing the support for this work.

Authors' contributions
H.J. and O.G. conceived of the presented idea. H.J. developed the theory and performed the computations. O.G. verified the analytical methods. D.R. supervised the project. All authors discussed the results and H.J. wrote the manuscript in consultation with O.G and D.R. All authors read and approved the final manuscript.

Author's information
Hailey James received Bachelor's degree in Computer Science from Harvard College in Cambridge, Massachusetts in 2018. She is currently working as a machine learning engineer at Lendbuzz in Boston, Massachusetts, where her research includes image and document forensics. Her interests also include work in fairness, human-computer cooperation, and explainability in artificial intelligence.
Otkrist Gupta is currently Vice President of Data Science at Lendbuzz, focussing on deep learning with applications in finance and computer vision. He completed his Ph.D. at MIT Media Lab from camera culture group. His research is focused on inventing new algorithms for deep learning for health screening and diagnosis, hidden geometry detection, exploiting techniques from optimization, linear algebra and compressive sensing. He also works on designing algorithms for futuristic 3D projective displays. Before joining MIT Media Lab Otkrist worked in Google Now team where he built voice actions such as take a picture and what's on my Chromecast and worked on voice response quality from Google Now. He also worked at LinkedIn where he developed services such as Smart ToDo, Ultra fast auto-complete, Notifications and CheckIn platform. He completed his bachelors from Indian Institute of Technology Delhi (IITD) in Computer Science with emphasis on algorithms and linear algebra. After graduating from IITD, he worked for one year in field of High Frequency Trading at Tower Research Capital.
Dan Raviv is the Chief Technology Officer of Lenbuzz Inc, an AI-based fintech underwriting company, and is faculty in the department of Engineering, Tel Aviv University, Israel. Dan did his post-doc at MIT in the Media Lab, working on various problems in the intersection of Geometry, Computer vision, and Machine learning, and is one of the leading researchers in Geometric Deep Learning. Dan was awarded the 2016 biennial award for Imaging Sciences granted by SIAM and published dozens of academic papers in his research field. Dan earned his Ph.D. and M.Sc. in computer science from the Technion—Israel Institute of Technology, Israel, and holds a bachelor's degree in Mathematics and Computer Science, Summa Cum Lauda, from the Technion as well. Dan was a member of the Technion's Excellence program, which handpicks the best and the brightest.

Funding
The funding for this project was provided by Lendbuzz.

Availability of data and materials
The IEEE IFSTC Image Forensics Challenge can be found as described in [37]. We hope to eventually make the code publicly available, but it is not available at this time.

Declarations

Competing interests
The authors declare that they have no competing interests.

Received: 4 January 2021 Accepted: 9 January 2022
Published: 7 February 2022

References

1. Y. Rao, J. Ni, 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 (2016)
2. M. Stamm, K.R. Liu, Blind forensics of contrast enhancement in digital images. In 2008 15th IEEE International Conference on Image Processing (IEEE, 2008), pp. 3112–3115
3. Y. Pengpeng, N. Rongrong, Z. Yao, C. Gang, W. Haorui, Z. Wei, Robust contrast enhancement forensics using convolutional neural networks. CoRR abs/1803.04749 (2018). 1803.04749
4. A. Popescu, H. Farid, Exposing digital forgeries by detecting traces of re-sampling. IEEE Trans. Signal Process. **53**, 758–767 (2005). https://doi.org/10.1109/TSP.2004.839932
5. H. Farid, Exposing digital forgeries from jpeg ghosts. IEEE Trans. Inf. Forensics Secur. **4**(1), 154–160 (2009)
6. D.-Y. Hsiao, S.-C. Pei, Detecting digital tampering by blur estimation. In First International Workshop on Systematic Approaches to Digital Forensic Engineering (SADFE'05) (IEEE 2005), pp. 264–278
7. G. Cao, Y. Zhao, R. Ni, Edge-based blur metric for tamper detection. J. Inf. Hiding Multimed. Signal Process. **1**(1), 20–27 (2010)
8. G. Cao, Y. Zhao, R. Ni, L. Yu, H. Tian, Forensic detection of median filtering in digital images. In 2010 IEEE International Conference on Multimedia and Expo (IEEE 2010), pp. 89–94
9. B. Bayar, M.C. Stamm, A deep learning approach to universal image manipulation detection using a new convolutional layer. ACM Workshop on Information Hiding and Multimedia Security **5–10**(2016)
10. N. Carlini, D. Wagner, Adversarial examples are not easily detected: bypassing ten detection methods. In Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security, pp. 3–14 (2017)
11. M. Barni, A. Costanzo, E. Nowroozi, B. Tondi, Cnn-based detection of generic contrast adjustment with jpeg post-processing. 2018 25th IEEE International Conference on Image Processing (ICIP) (2018). https://doi.org/10.1109/icip.2018.8451698
12. W. Shan, Y. Yi, J. Qiu, A. Yin, Robust median filtering forensics using image deblocking and filtered residual fusion. IEEE Access PP, 1–1 (2019). https://doi.org/10.1109/ACCESS.2019.2894981
13. I.J. Goodfellow, J. Shlens, C. Szegedy, Explaining and Harnessing Adversarial Examples **1412**, 6572 (2014)
14. D. Gragnaniello, F. Marra, G. Poggi, L. Verdoliva, Analysis of adversarial attacks against cnn-based image forgery detectors. 2018 26th European Signal Processing Conference (EUSIPCO) (2018). https://doi.org/10.23919/eusipco.2018.8553560
15. A. Tuama, F. Comby, M. Chaumont, Camera model identification with the use of deep convolutional neural networks. In: 2016 IEEE International Workshop on Information Forensics and Security (WIFS) (IEEE 2016), pp. 1–6
16. M.K. Johnson, H. Farid, Exposing digital forgeries in complex lighting environments. IEEE Trans. Inf. Forensics Secur. **2**(3), 450–461 (2007)
17. M. Dejean-Servières, K. Desnos, K. Abdelouahab, W. Hamidouche, L. Morin, M. Pelcat, Study of the impact of standard image compression techniques on performance of image classification with a convolutional neural network. (2017)
18. J. Chen, X. Kang, Y. Liu, Z. Wang, Median filtering forensics based on convolutional neural networks. Signal Process. Lett., IEEE **22**, 1849–1853 (2015). https://doi.org/10.1109/LSP.2015.2438008
19. J. Fridrich, J. Kodovsky, Rich models for steganalysis of digital images. IEEE Trans. Inf. Forensics Secur. **7**, 868–882 (2012). https://doi.org/10.1109/TIFS.2012.2190402
20. S. Tan, B. Li, Stacked convolutional auto-encoders for steganalysis of digital images. In: Signal and Information Processing Association Annual Summit and Conference (APSIPA), 2014 Asia-Pacific, pp. 1–4 (2014). https://doi.org/10.1109/APSIPA.2014.7041565
21. G. Xu, H.-Z. Wu, Y.-Q. Shi, Structural design of convolutional neural networks for steganalysis. IEEE Signal Process. Lett. **23**(5), 708–712 (2016). https://doi.org/10.1109/LSP.2016.2548421
22. J. Ye, J. Ni, Y. Yi, Deep learning hierarchical representations for image steganalysis. IEEE Trans. Inf. Forensics Secur. **12**(11), 2545–2557 (2017). https://doi.org/10.1109/TIFS.2017.2710946
23. Y. Yousfi, J. Butora, E. Khvedchenya, J. Fridrich, Imagenet pre-trained cnns for jpeg steganalysis. In: 2020 IEEE International Workshop on Information Forensics and Security (WIFS), pp. 1–6 (2020). https://doi.org/10.1109/WIFS49906.2020.9360897
24. D. Cozzolino, G. Poggi, L. Verdoliva, Recasting residual-based local descriptors as convolutional neural networks. Proceedings of the 5th ACM Workshop on Information Hiding and Multimedia Security - IHMMSec '17 (2017). https://doi.org/10.1145/3082031.3083247
25. M. Boroumand, J.J. Fridrich, Deep learning for detecting processing history of images. In: Media Watermarking, Security, and Forensics (2018)
26. N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z.B. Celik, A. Swami, The limitations of deep learning in adversarial settings. 2016 IEEE European Symposium on Security and Privacy (EuroSP) (2016). https://doi.org/10.1109/eurosp.2016.36
27. A. Madry, A. Makelov, L. Schmidt, D. Tsipras, A. Vladu Towards deep learning models resistant to adversarial attacks. arXiv preprint arXiv:1706.06083 (2017)
28. S.-M. Moosavi-Dezfooli, A. Fawzi, P. Frossard, Deepfool: a simple and accurate method to fool deep neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2574–2582 (2016)
29. D. Kim, H.-U. Jang, S.-M. Mun, S. Choi, H.-K. Lee, Median filtered image restoration and anti-forensics using adversarial networks. IEEE Signal Process. Lett. **25**, 278–282 (2018)

30. B. Tondi, Pixel-domain adversarial examples against cnn-based manipulation detectors. Electron. Lett. (2018). https://doi.org/10.1049/el.2018.6469
31. M.B.K.K.E.N.B. Tondi, On the transferability of adversarial examples against cnn-based image forensics (2020)
32. H. Tang, R. Ni, Y. Zhao, X. Li, Median filtering detection of small-size image based on cnn. J. Vis. Commun. Image Represent. **51**, 162–168 (2018). https://doi.org/10.1016/j.jvcir.2018.01.011
33. Y. Chen, X. Kang, Y.Q. Shi, Z. Wang, A multi-purpose image forensic method using densely connected convolutional neural networks. J. Real-Time Image Process. (2019). https://doi.org/10.1007/s11554-019-00866-x
34. F. Chollet, Xception: Deep learning with depthwise separable convolutions. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017). https://doi.org/10.1109/cvpr.2017.195
35. C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going Deeper with Convolutions **1409**, 4842 (2014)
36. J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255 (2009). https://doi.org/10.1109/CVPR.2009.5206848
37. IEEE IFS-TC Image Forensics Challenge. IEEE (2017). https://signalprocessingsociety.org/newsletter/2014/01/ieee-ifs-tc-image-forensics-challenge-website-new-submissions
38. H. Zhao, J. Shi, X. Qi, X. Wang, J. Jia, Pyramid scene parsing network. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017). https://doi.org/10.1109/cvpr.2017.660

Publisher's Note