MISLGAN: AN ANTI-FORENSIC CAMERA MODEL FALSIFICATION FRAMEWORK USING A GENERATIVE ADVERSARIAL NETWORK

Chen Chen[*]*, Xinwei Zhao*[*] *and Matthew C. Stamm*

Dept. of Electrical and Computer Engineering, Drexel University, Philadelphia, PA 19104, USA

ABSTRACT

Deep learning techniques have become popular for performing camera model identification. To expose weaknesses in these methods, we propose a new anti-forensic framework that utilizes a generative adversarial network (GAN) to falsify an image's source camera model. Our proposed attack uses the generator trained in the GAN to produce an image that can fool a CNN-based camera model identification classifier. Moreover, our proposed attack will only introduce a minimal amount of distortion to the falsified image that is not perceptible to human eyes. By conducting experiments on a large amount of data, we show that the proposed attack can successfully fool a state-of-art camera model identification CNN classifier with 98% probability and maintain high image quality.

Index Terms— Generative adversarial network, Anti-forensics, Camera model identification, Convolutional neural networks

1. INTRODUCTION

Determining the model and manufacturer of an image's source camera is an important task in multimedia forensics [1]. In many situations such as criminal investigations and news reporting, the integrity and origin of images may be put into question. Many forensic techniques have been developed to perform camera model identification using traces left by an image's source camera model, such as JPEG header information [2], sensor noise statistics [3, 4], and demosaicing strategy [5, 6, 7, 8, 9]. In recent years, data-driven methods, particularly deep learning, have been gaining its popularity for forensic tasks [10, 11, 12, 13, 14, 15, 16].

Research has shown, however, that a malicious attacker may launch an anti-forensic attack against existing forensic detectors [17, 18, 19, 20, 21, 22]. Therefore, studying anti-forensics can help researchers and investigators be aware of weaknesses in existing forensic detectors [23, 24, 25, 26, 27, 28].

One approach to anti-forensically falsify an image's source camera model is to use forensic algorithms to obtain demosaicing estimates of the target camera model, then use the estimates to redemosaic the image captured by another camera model. However, this attack can be detected by forensic algorithms that utilize more generic and sophisticated information [29]. Recently, forensic researchers have started to explore more complex approaches to counter deep learning based detectors [30, 31].

Generative adversarial networks (GANs) are a deep learning framework first used in the machine learning and computer vision communities to generate data that can statistically mimic the distribution of training data [32]. The methodology of GANs is to alternatively train two individual deep neural networks, a generator and a discriminator, in a competing fashion. While the discriminator gets better at distinguishing the real data from the generated data, the generator aims to fool the discriminator by minimizing the difference between the real and generated data. GANs have been used to produce visually realistic images in many computer vision techniques [33, 34, 35]. This raises many important forensic questions. If GANs can be used to generate visually realistic images, can they be used to generated forensically realistic images? A GAN has recently been proposed that can anti-forensically hide traces left by median filtering [31]. Can new anti-forensic GANs be built to attack other forensic tasks such as camera model identification?

In this paper, we propose a new anti-forensic framework aimed to fool CNN-based camera model identification classifiers. Our proposed attack utilizes a GAN to falsify the forensic information of an image's source camera model and only introduces visually imperceptible distortion to the falsified images. The proposed framework, MISLGAN, is named after our group, the Multimedia Information and Security Laboratory (MISL). To construct our attack, we design and train our GAN architecture specially for forensic purposes, and then we use the trained generator to falsify the forensic traces in an input image. The GAN's loss function is formulated to ensure visual quality of the attacked image and force the generator to learn camera model specific traces by incorporating feedback from a camera model identification classifier. We conduct a series of experiments to demonstrate the effectiveness of our proposed attack. The falsified images can successfully fool a CNN-based camera model identification classifier with approximately 98% probability. The mean SSIM and PSNR between the original images and the falsified images are above 0.97 and 43dB.

2. PROBLEM FORMULATION

The overall goal of our anti-forensic attack for camera model identification is to falsify the forensic traces left by an image's source camera model, such that the attacked image can fool the investigator's camera model identification classifier. To build a successful attack, the attacked images should look realistic and contain no visible artifacts, i.e. the anti-forensic attack should not change image contents and can only introduce an acceptable amount of distortion to images. Since CNN-based camera model identification techniques

This material is based upon work supported by the National Science Foundation under Grant No. 1553610. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

This material is based on research sponsored by DARPA and Air Force Research Laboratory (AFRL) under agreement number PGSC-SC-111346-03. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of DARPA and Air Force Research Laboratory (AFRL) or the U.S. Government.

Email: {cc3359, xz355}@drexel.edu, mstamm@drexel.edu

* These authors contributed equally to this manuscript.

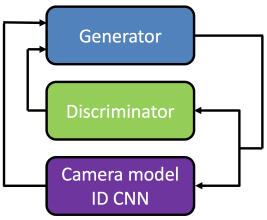

Fig. 1: Proposed Framework: MISLGAN

have achieved highly reliable and stable performance on large scale databases [10, 14, 36] recently, we focus on building targeted attack for CNN-based camera model identification classifiers. Our anti-forensic attack takes in a target camera model and an image originally captured by any source camera model as inputs, and produces an output image by modifying its forensic traces to match the target camera model. The output image of our attack has the same contents as original image but will be classified by the investigator's CNN classifier as originating from the target camera model.

We assume that the forensic investigator will first build and train a CNN-based camera model identification classifier, then use the trained CNN classifier to perform camera model identification. The investigator will only be presented with an image itself and will not know a priori whether the image is attacked or not. If the image is attacked, the attacker will keep the original image private and only show the final attacked image to the investigator. To launch the camera model attack, we assume the attacker has access to the investigator's classifier or can build an equivalent copy of the investigator's classifier. In case a data-driven attack is needed for the investigator's CNN-based classifier, the attacker can also collect a large image set to build and train the anti-forensic attack. We introduce how we construct our proposed attack in the next section.

3. PROPOSED FRAMEWORK

3.1. Overview

To construct our attack, we propose a new framework that utilizes a generative adversarial network to build a generator $G(\cdot)$ to produce falsified images which can mimic the forensic statistics of a target camera model, and our attack will leave no artifacts that are perceptible to human eyes. The proposed framework, MISLGAN, is named after our group, Multimedia Information and Security Laboratory.

Generative adversarial networks (GANs) are a deep learning framework that have been widely used in computer vision to produce visually realistic images. A GAN consists of two major components, discriminator D and generator G. Assuming real images I have distribution $I \sim p_r(I)$ and generated images I' have distribution $I' \sim p_g(I')$, the two deep networks are trained in a competing fashion using (1):

$$\min_G \max_D \mathbb{E}_{I\sim p_r(I)}[\log D(I)] + \mathbb{E}_{I'\sim p_g(I')}[\log(1 - D(I'))] \quad (1)$$

where $\mathbb{E}$ represents the operation of calculating expected value, until they reach an equilibrium such that the generated data can mimic statistical distribution of the real data [32].

The overview of MISLGAN is shown in Fig. 1. It consists of three components: generator, discriminator, and a pre-trained CNN-based camera model classifier. By incorporating the pre-trained CNN-based camera model identification classifier into the GAN, the generator is forced to reproduce the forensic traces of the target camera model. Moreover, we formulate the loss function fully described in Sec. 3.3 to ensure that the falsified images have acceptable visual quality and fool investigator's camera model identification CNN classifier.

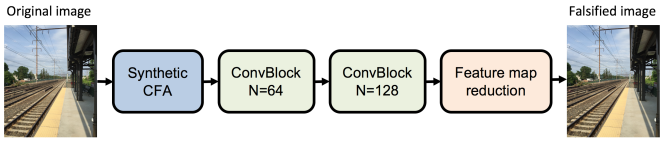

Fig. 2: Architecture of the generator

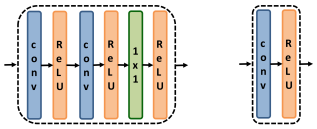

Fig. 3: Left: ConvBlock, Right: Feature map reduction

3.2. Architecture of proposed framework

We now describe the details of the three networks in the proposed MISLGAN.

Generator: The architecture of the generator is shown in Fig. 2. It consists of one synthetic color filter array (CFA) block, two ConvBlocks made of several convolutional layers, activation functions and 1×1 convolutional layers arranged in a common structure, and one feature map reduction block made of convolutional layers and activation function to combine high dimensional feature map into one color image or image patch.

Demosaicing trace are one very important forensic trace that many algorithms used to perform camera model identification [6, 5, 6, 7, 8, 9]. While CNN-based camera model identification classifier can learn more generic forensic traces of a camera model, anecdotally these traces still contain demosaicing traces [14]. Therefore, by using the synthetic CFA block, we can first remove the demosaicing traces left by the image's original capturing model, and force the generator to redemosaic the image such that demosaicing traces are ensured to be falsified. In this paper, we used the Bayer pattern.

Besides demosaicing traces, the generator should also be able to induce more sophisticated and complex forensic traces that the deep learning based classifier used for performing camera model identification. The ConvBlocks are design to redemosaic an image and reconstruct other forensic statistics of the target camera model. The architecture of a ConvBlock is shown in Fig. 3. It consists of a convolutional layer with N, 3×3 filters and stride 1 followed by ReLU activation, then another convolutional layer with N, 3×3 filters and stride 1 followed by ReLU activation, then a 1×1 convolutional layer followed by ReLU activation. By using the 1×1 convolutional layer, our generator is capable of learning the correlations between feature maps and thus produce more forensically realistic traces. In practice, our generator uses $N = 64$ ConvBlock followed by a $N = 128$ ConvBlock.

Since the second ConvBlock outputs a large number of feature maps, to combine 128 feature maps to a 3 color-layer image, we design the feature map reduction block shown as in Fig. 3. It consists of a convolutional layer with 3, 3×3 filters and stride 1 followed by ReLU activation function. We use ReLU as the activation function for the generator is because experimentally we found that it yields the best performance.

Discriminator: The discriminator is designed to differentiate between real and generated images (i.e. attacked images). It is built into the GAN to strengthen the quality of the generator's output.

The discriminator architecture in the proposed framework is a variant of the camera model identification CNN proposed by Bayar and Stamm in [14, 37]. This CNN architecture is specifically designed for forensic tasks. It consists of a constrained convolutional layer to learn low level forensic feature extractors, followed by 4 standard convolutional layers with batch normalization and hyperbolic tangent activation, then followed by 3 fully connected layers. Due to space limitations, we refer the readers to [14, 37] for a complete description of the CNN architecture.

The key modification made to the CNN architecture for our discriminator is we replace the last fully connected layer with a single neuron followed by sigmoid activation. When training the discriminator, the last neuron activation corresponds to the probability that the input image is real (i.e 1 if the image is real, 0 if the image is generated).

Pre-trained CNN-based camera model identification classifier: This can be any CNN-based camera model identification classifier. In this paper, we use the CNN architecture proposed by Bayar and Stamm [14, 37], due to its stable and successful performance for camera model identification. We modified the depth of filters in the input layer of their CNN architecture to accommodate color instead of grayscale images.

3.3. Generator loss

For our generator to be successful, it must both fool a camera model identification CNN and introduce minimum distortion into the image. As a result, we define the generator's loss function $\mathcal{L}_G$ as

$$\mathcal{L}_G = \alpha\mathcal{L}_p + \beta\mathcal{L}_c + \gamma\mathcal{L}_a, \tag{2}$$

where $\mathcal{L}_p$ represents the perceptional loss between the original image and its falsified copy, $\mathcal{L}_c$ represents the classification loss due to fooling the camera model identification CNN classifier, $\mathcal{L}_a$ represents the adversarial loss due to fooling the discriminator, and α, β, γ are the weights for each loss term.

Since the attack should leave no perceptual artifacts, we model the perceptual loss using the mean absolute difference between the original image and its falsified copy. For an original image I of size $w \times h$, the absolute difference between I and its corresponding falsified copy is computed as

$$\mathcal{L}_p = \frac{1}{w \times h} \sum_{i=1}^{w} \sum_{j=1}^{h} \mid I_{i,j} - G(I)_{i,j} \mid, \tag{3}$$

where $G(\cdot)$ denotes output of the generator and the subscript i, j represents the pixel location in the image.

The classification loss is designed to measure the difference between the camera model identification classifier's output for the falsified image and the ideal output for the target camera model. Let $C(\cdot)$ represent the softmax output of the classifier. For image I, the classifier's output for its falsified copy is $C(G(I))$. For a particular target camera model, the ideal softmax output t is a vector with a 1 at the location of target class and 0's elsewhere. The classification loss $\mathcal{L}_c$ is quantified as the cross entropy between t and $C(G(I))$, and it can be calculated by

$$\mathcal{L}_c = -\sum_{k=1}^{m} t_k \log\left(C(G(I))_k\right), \tag{4}$$

where m is the number of camera models that the camera model identification classifier is train to differentiate.

While the perceptual loss and the classification loss result in visually and forensically plausible falsified images, they may have some limitation in reconstructing complex statistics between real and generated data. Hence, we incorporate adversarial loss for the purpose of fooling a discriminator. It is expressed as

$$\mathcal{L}_a = \log(1 - D(G(I))), \tag{5}$$

where $D(\cdot)$ denotes the output of the discriminator.

3.4. Deploying MISLGAN

To falsify an image that was originally captured by camera model A and make it look as if it was captured by camera model B, we first train the proposed framework using target camera model B. After training, we discard the camera model identification CNN and the discriminator. Next we divide the full-size color image into smaller patches and use the generator to attack each patch individually. The patches are then grouped together to form the full-size attacked image. We note that if the attacker possesses enough computational resources, the entire image can be attacked at once.

4. EXPERIMENTAL RESULTS

4.1. Experimental Setup

To train and evaluate our proposed GAN, we created two different databases of images. Database I was built from images in the publicly available Dresden Image Database [38]. We randomly generated 932,400 non-overlapping color image patches of size 256×256 from 18 camera models shown in Table 1. We then divided all patches into 900,000 patches for training, 16,200 patches for validation and 16,200 patches for testing. We ensured that training, validation and testing patches came from separate full-size images and each model contributed equally in this database. For Database II, we first created 318,000 non-overlapping patches of size 256×256 from 10 camera models shown in Table 2. These patches were created from full-size images that we captured using the camera models listed in Table 2. We divided the patches into 300,000 patches for training, 9,000 for validation and 9,000 for testing. Again, no training, testing or validation patches came from the same full-size image and each model contributed equally.

Table 1: Camera Model and Identification Accuracy of Database I

ID	Camera Model	Acc. (%)	ID	Camera Model	Acc. (%)
1	Kodak M1063	99.56	10	Sony DSC-H50	81.11
2	Canon Ixus70	99.44	11	Rolli RCP-7325XS	97.67
3	Casio EX-Z150	98.89	12	Samsung NV15	99.44
4	Fujifilm FinePixJ50	99.67	13	Panasonic DMC-FZ50	99.67
5	Praktica DCZ5.9	99.56	14	Sony DSC-W170	80.22
6	Ricoh GX100	98.89	15	Sony DSC-T77	99.67
7	Nikon CoolPixS710	100	16	Pentax OptioA40	99.67
8	Nikon D200	100	17	Olympus mju-1050SW	99.44
9	Samsung L74wide	99.67	18	Nikon D70	99.11

Table 2: Camera Model and Identification Accuracy of Database II

ID	Camera Model	Acc. (%)	ID	Camera Model	Acc. (%)
1	Canon EOS SL1	96.89	6	Motorola Droid Maxx	96.00
2	Canon Powershot S100	99.11	7	Nikon D7100	97.56
3	iPhone 4S	98.89	8	Samsung Galaxy S4	99.89
4	iPhone 6S	95.11	9	Sony A6000	98.56
5	LG G3	97.22	10	Sony NEX-5TL	95.44

Next, we trained the CNN classifier proposed by Bayar & Stamm [14] with the same parameters used in [14] to perform camera model identification on each database. The camera model identification accuracies achieved on each database are reported in the last column of Table 1 and Table 2. The CNN classifier achieved an average camera model identification of 97.31% on Database I and 97.47% on Database II, which is consistent with results reported in the original publication [14].

Given the trained CNN classifier, we used our proposed MISLGAN framework to create generators targeted at 5 randomly chosen camera models in each database. For Database I, we created generators to target camera models 1, 5, 9, 13 and 17. For Database II, we created generators to target models 1, 4, 6, 8 and 10. To build all 10 generators with different targets, we trained MISLGAN using Tensorflow's default Adam optimizer with a learning rate of 10^{-4}

Table 3: MISLGAN evaluation on Database I

	Seen			Unseen		
Target	m-PSNR [1]	m-SSIM [2]	SAR [3](%)	m-PSNR [1]	m-SSIM [2]	SAR [3](%)
1	43.4916	0.9842	99.25	43.0004	0.9847	98.94
5	43.5857	0.9815	99.43	43.0854	0.9827	98.32
9	43.7658	0.9860	99.04	43.4468	0.9870	92.00
13	44.8767	0.9866	99.30	44.5755	0.9884	98.78
17	42.9282	0.9840	97.91	42.7480	0.9860	98.16

Table 4: MISLGAN evaluation on Database II

	Seen			Unseen		
Target	m-PSNR [1]	m-SSIM [2]	SAR [3](%)	m-PSNR [1]	m-SSIM [2]	SAR [3](%)
1	45.9622	0.9904	96.19	44.7714	0.9877	96.12
4	42.4920	0.9831	98.60	45.9622	0.9704	98.52
6	44.3386	0.9878	98.33	43.1272	0.9842	99.20
8	44.4763	0.9876	99.10	43.3912	0.9849	99.20
10	45.9931	0.9924	97.13	44.4994	0.9892	96.98

for both generator and discriminator. The generator was trained with batch size 30 and weights $\alpha = 1, \gamma = 1, \beta = 1$. The batch size for discriminator is 60 (30 generated patches and 30 corresponding original patches). We stopped training when loss on validation patches reached an acceptable level.

4.2. Experiment 1

In this experiment, we evaluated the ability of our generators trained in MISLGAN to launch a camera model falsification attack using images from camera models used to train the GAN, i.e. the generators have seen other images from the same camera models in the training data. For each database, we fed the testing patches into all 5 trained generators to create attacked patches targeted on different camera models. This yielded a total number of 81,000 attacked patches for Database I and 45,000 attacked patches for Database II. We then used the CNN classifier to identify the source camera model of each of the attacked patches. We report the successful attack rate (SAR) as the percentage of attacked patches whose camera models are identified as the target model by the CNN classifier. We also computed the PSNR and SSIM between the attacked patches and original patches to measure the distortion introduced by the attack. We averaged the PSNR and SSIM of all attacked patches generated by each generator, and reported the experimental results on Database I and Database II in the left portion of Table 3 and Table 4 respectively.

For all 5 different target models in each database, our attack can achieve an average successful attack rate of 98.99% on Database I and an average successful attack rate of 97.87% on Database II. The attack rates are comparable with the average camera model identification accuracies achieved by the CNN classifier on Database I and Database II. We can see that our attacked patches have very high PSNR and SSIM compared to their original patches. This shows that our attack introduces very small distortion to attacked patches and maintains high image quality. Fig. 4 shows an example of an attacked image (bottom) and an original image (top). The original image was captured by a Canon EOS SL1. It was attacked patch by patch using a generator with a target model of an iPhone 6S. We stitched all attacked patches together to generate the full-size attacked image. The attacked image looks perceptually realistic and it is difficult to visually differentiate the attacked image from the original one, even in a zoomed view. This experiment demonstrates that our proposed MISLGAN can perform successful targeted camera model falsification with both a high attack rate and high visual quality.

[1,2] m-PSNR and m-SSIM are mean PSNR and SSIM calculated using testing data for this database.

[3] SAR is short for successful attack rate. It is the percentage of attacked images that are classified as the target camera model by the camera model identification CNN.

Fig. 4: An attack example. Top left: original full-size image captured by Canon EOS SL1. Bottom left: attacked full-size image with target model iPhone 6S. Middle: zoomed view of 256×256 patch marked in read box in full-size image. Right: zoomed view of 25×25 patch marked in read box in 256×256 patch.

4.3. Experiment 2

In this experiment, we evaluate the generalizability of our attack using image patches whose camera models have never been used or seen when training MISLGAN. Specifically, we randomly chose 5,000, 256×256 patches from 5 new camera models: Motorola X, HTC One, iPhone 6, Sony NEX-7 and Samsung Galaxy Note 4. Each model contributed 1,000 patches. We then followed the same procedure described in Experiment 1 to attack image patches from this unseen dataset using all trained generators. This yielded another 25,000 attacked patches for each database. The source camera model of each patch was identified using the trained CNN classifier associated with the corresponding database. Again, we report average PSNR, average SSIM and successful attack rate for each target model in the right part of Table 3 and Table 4.

We obtained an average successful attack rate of 97.24% for 5 target models on Database I and an average successful attack rate of 98.00% for 5 target models on Database II. For each of the 10 generators targeted on different models from our two databases, the average PSNR of all generated patches was at least 42.7840 and the average SSIM is above 0.9704. On both databases, the evaluation results on image patches from unseen camera models are consistent with results from seen camera models in Experiment 1. These results show that our proposed MISLGAN trained on a particular database can attack images from camera models outside the database with a high successful attack rate and high image quality. This experiment demonstrates the generalizability of our MISLGAN, which is critical for an attack to be applied in real-world scenarios.

5. CONCLUSION

In this paper, we proposed a new anti-forensic framework against CNN-based camera model identification classifiers. Our proposed attack uses a generative adversarial network to construct a generator that is designed to introduce forensic information from a target camera model into the falsified images. Moreover, we formulated a loss function to ensure the generator can produce attacked images that successfully fool the CNN classifier and maintain high visual quality.

6. REFERENCES

[1] M. C. Stamm, M. Wu, and K. J. R. Liu, "Information forensics: An overview of the first decade," *IEEE Access*, vol. 1, pp. 167–200, 2013.

[2] E. Kee, M. K. Johnson, and H. Farid, "Digital image authentication from jpeg headers," *IEEE Transaction on Information Forensics and Security*, vol. 6, no. 3, pp. 1066–1075, 2011.

[3] M. Chen, J. Fridrich, M. Goljan, and J. Lukás, "Determining image origin and integrity using sensor noise," *IEEE Transactions on Information Forensics and Security*, vol. 3, no. 1, pp. 74–90, 2008.

[4] T. H. Thai, R. Cogranne, and F. Retraint, "Camera model identification based on the heteroscedastic noise model," *IEEE Transactions on Image Processing*, vol. 23, no. 1, pp. 250–263, Jan 2014.

[5] H. Cao and A. C. Kot, "Accurate detection of demosaicing regularity for digital image forensics," *IEEE Transactions on Information Forensics and Security*, vol. 4, no. 4, pp. 899–910, Dec 2009.

[6] A. Swaminathan, M. Wu, and K. Liu, "Nonintrusive component forensics of visual sensors using output images," *IEEE Transaction on Information Forensics and Security*, vol. 2, no. 1, pp. 91–106, 2007.

[7] C. Chen and M. C. Stamm, "Camera model identification framework using an ensemble of demosaicing features," in *International Workshop on Information Forensics and Security*. IEEE, 2015, pp. 1–6.

[8] X. Zhao and M. C. Stamm, "Computationally efficient demosaicing filter estimation for forensic camera model identification," in *International Conference on Image Processing*. IEEE, 2016, pp. 151–155.

[9] S. Bayram, H. Sencar, N. Memon, and I. Avcibas, "Source camera identification based on cfa interpolation," in *International Conference on Image Processing*, vol. 3. IEEE, 2005, pp. III–69.

[10] L. Bondi, L. Baroffio, D. Gera, P. Bestagini, E. J. Delp, and S. Tubaro, "First steps toward camera model identification with convolutional neural networks," *IEEE Signal Processing Letters*, vol. 24, no. 3, pp. 259–263, March 2017.

[11] B. Bayar and M. C. Stamm, "Augmented convolutional feature maps for robust cnn-based camera model identification," in *International Conference on Image Processing*. IEEE, 2017.

[12] L. Bondi, S. Lameri, D. Gera, P. Bestagini, E. J. Delp, and S. Tubaro, "Tampering detection and localization through clustering of camera-based cnn features," in *Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE, July 2017, pp. 1855–1864.

[13] A. Tuama, F. Comby, and M. Chaumont, "Camera model identification with the use of deep convolutional neural networks," in *International Workshop on Information Forensics and Security (WIFS)*. IEEE, 2016, pp. 1–6.

[14] B. Bayar and M. C. Stamm, "Design principles of convolutional neural networks for multimedia forensics," *Electronic Imaging*, vol. 2017, no. 7, pp. 77–86, 2017.

[15] O. Mayer and M. C. Stamm, "Learned forensic source similarity for unknown camera models," in *Acoustics, Speech and Signal Processing (ICASSP), 2018 IEEE International Conference on*. IEEE, 2018, pp. 1–4.

[16] B. Bayar and M. C. Stamm, "Towards open set camera model identification using a deep learning framework," in *Acoustics, Speech and Signal Processing (ICASSP), 2018 IEEE International Conference on*. IEEE, 2018, pp. 1–4.

[17] M. C. Stamm, S. K. Tjoa, W. S. Lin, and K. R. Liu, "Anti-forensics of jpeg compression," in *International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2010, pp. 1694–1697.

[18] M. Kirchner and R. Bohme, "Hiding traces of resampling in digital images," *IEEE Transactions on Information Forensics and Security*, vol. 3, no. 4, pp. 582–592, 2008.

[19] M. Fontani and M. Barni, "Hiding traces of median filtering in digital images," in *Signal Processing Conference (EUSIPCO), Proceedings of the 20th European*. IEEE, 2012, pp. 1239–1243.

[20] Z.-H. Wu, M. C. Stamm, and K. R. Liu, "Anti-forensics of median filtering," in *International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2013, pp. 3043–3047.

[21] M. C. Stamm, W. S. Lin, and K. R. Liu, "Temporal forensics and anti-forensics for motion compensated video," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 4, pp. 1315–1329, 2012.

[22] S. Sharma, A. V. Subramanyam, M. Jain, A. Mehrish, and S. Emmanuel, "Anti-forensic technique for median filtering using l1-l2 tv model," in *2016 IEEE International Workshop on Information Forensics and Security (WIFS)*, Dec 2016, pp. 1–6.

[23] M. Barni, Z. Chen, and B. Tondi, "Adversary-aware, data-driven detection of double jpeg compression: How to make counter-forensics harder," in *International Workshop on Information Forensics and Security*. IEEE, Dec 2016, pp. 1–6.

[24] A. Peng, H. Zeng, X. Lin, and X. Kang, "Countering anti-forensics of image resampling," in *International Conference on Image Processing*. IEEE, Sept 2015, pp. 3595–3599.

[25] W.-H. Chuang and M. Wu, "Robustness of color interpolation identification against anti-forensic operations," in *International Workshop on Information Hiding*. Springer, 2012, pp. 16–30.

[26] M. C. Stamm, W. S. Lin, and K. R. Liu, "Forensics vs. anti-forensics: A decision and game theoretic framework," in *International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2012, pp. 1749–1752.

[27] G. Valenzise, M. Tagliasacchi, and S. Tubaro, "Revealing the traces of jpeg compression anti-forensics," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 2, pp. 335–349, Feb 2013.

[28] H. Zeng, T. Qin, X. Kang, and L. Liu, "Countering anti-forensics of median filtering," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2014, pp. 2704–2708.

[29] C. Chen, X. Zhao, and M. C. Stamm, "Detecting anti-forensic attacks on demosaicing-based camera model identification," in *International Conference on Image Processing*. IEEE, 2017.

[30] D. Gera, Y. Wang, L. Bondi, P. Bestagini, S. Tubaro, and E. J. Delp, "A counter-forensic method for cnn-based camera model identification," in *Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE, July 2017, pp. 1840–1847.

[31] D. Kim, H. U. Jang, S. M. Mun, S. Choi, and H. K. Lee, "Median filtered image restoration and anti-forensics using adversarial networks," *IEEE Signal Processing Letters*, vol. 25, no. 2, pp. 278–282, Feb 2018.

[32] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems 27*.

[33] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi, "Photo-realistic single image super-resolution using a generative adversarial network," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[34] E. L. Denton, S. Chintala, a. szlam, and R. Fergus, "Deep generative image models using a laplacian pyramid of adversarial networks," in *Advances in Neural Information Processing Systems 28*.

[35] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *CoRR*, vol. abs/1511.06434, 2015.

[36] A. Tuama, F. Comby, and M. Chaumont, "Camera model identification with the use of deep convolutional neural networks," in *Information Forensics and Security (WIFS)*. IEEE, 2016, pp. 1–6.

[37] B. Bayar and M. C. Stamm, "Constrained convolutional neural networks: A new approach towards general purpose image manipulation detection," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 11, pp. 2691–2706, Nov 2018.

[38] T. Gloe and R. Böhme, "The 'dresden image database' for benchmarking digital image forensics," in *Proceedings of the 2010 ACM Symposium on Applied Computing*, ser. SAC '10. New York, NY, USA: ACM, 2010, pp. 1584–1590. [Online]. Available: http://doi.acm.org/10.1145/1774088.1774427