Multiple-Operation Image Anti-Forensics with WGAN-GP Framework

Jianyuan Wu*, Zheng Wang*, Hui Zeng† and Xiangui Kang*
* Guangdong Key Lab of Information Security, School of Data and Computer science, Sun Yat-sen University, China
E-mail: isskxg@mail.sysu.edu.cn
† School of Computer Science & Tech., Southwest University of Science and Tech., Mianyang, China
E-mail: zengh5@mail2.sysu.edu.cn

***Abstract*— A challenging task in the field of multimedia security involves concealing or eliminating the traces left by a chain of multiple manipulating operations, *i.e.*, multiple-operation anti-forensics in short. However, the existing anti-forensic works concentrate on one specific manipulation, referred as single-operation anti-forensics. In this work, we propose using the improved Wasserstein generative adversarial networks with gradient penalty (WGAN-GP) to model image anti-forensics as an image-to-image translation problem and obtain the optimized anti-forensic models of multiple-operation. The experimental results demonstrate that our multiple-operation anti-forensic scheme successfully deceives the state-of-the-art forensic algorithms without significantly degrading the quality of the image, and even enhancing quality in most cases. To our best knowledge, this is the first attempt to explore the problem of multiple-operation anti-forensics.**

I. INTRODUCTION

Image forensics and anti-forensics are techniques serving opposed purposes in the field of multimedia security. One major goal of image forensic algorithms is to reveal the traces left by manipulations such as resampling [1], JPEG compression [2], [3], median filtering [4], [5], contrast enhancement [6], [7], unsharp masking sharpening [8], etc. In response, various anti-forensic methods [9]–[16] have concentrated on hiding or removing these traces, with the aim of fooling the forensic algorithms and thus pushing them to become safer and more reliable. However, artifacts that are inevitably left by an anti-forensic algorithm while concealing the traces of manipulations are often found. Counter-anti-forensic techniques [17]–[19] have also been explored to identify these artifacts left by anti-forensic algorithms.

In reality, an attacker often makes use of a variety of operations such as gamma correction, median filtering, Gaussian blurring and JPEG compression, to cover up the visual traces left by image copy-move or splicing, e.g., inconsistencies in the background and artificiality of the boundaries. The existing anti-forensic researches focused on single-operation anti-forensics. Little work has been done on anti-forensics of multiple operations. How to hiding or erasing the traces left by multiple operations remains an open problem.

This work was supported by NSFC (Grant Nos. U1536204, 61772571, 61702429). (Corresponding author: Xiangui Kang)

Considering the gradient vanishing in traditional GAN [20]–[23], we propose using WGAN-GP [24] to model image anti-forensics as an image-to-image translation problem and obtain the optimized anti-forensic models, which can remove the traces left by multiple-manipulation chain. The experimental results show that our multiple-operation anti-forensic scheme successfully deceives state-of-the-art forensic algorithms without significantly degrading the image quality, and even enhancing the quality in most cases.

II. NETWORK ARCHITECTURE AND LOSS FUNCTIONS

The architecture of WGAN-GP framework involves a generator network (G) and a critic network (C) as illustrated in Fig. 1.

A. Generator Network

The generator network G, as shown in Fig. 1a, generates anti-forensically modified images. It takes a manipulated image, the pixel values of which are normalized to [−1, 1], as the input. The input image is first convolved with sixty four 9×9 convolutional kernels, followed by a rectified linear unit (ReLU). Next, 16 identical residual blocks are used, each of which is composed of identical components, namely, two convolutional layers with 3×3 kernels and 64 feature maps, followed by a batch normalization (BN) layer and a ReLU as the activation function. We apply a skip connection to the output of each block, which is added to the output of the second BN layer of the next block.

Following these residual blocks, there are three other convolutional layers with a stride of 1×1. The size of the feature maps remains consistent in every convolutional layer since we adopt "same" mode padding. After the last convolutional layer, we use a hyperbolic tangent (TanH) function for scaling the output pixel values within [−1, 1].

B. Critic Network

The critic network C, as shown in Fig. 1b, is the opponent of G in GAN. It is trained to maximize the Wasserstein distance between the distributions of generated images and the original ones.

C receives an input image, either a generated image or an original image, whose pixel values are within [−1, 1]. Then the high-frequency residual feature maps are extracted and

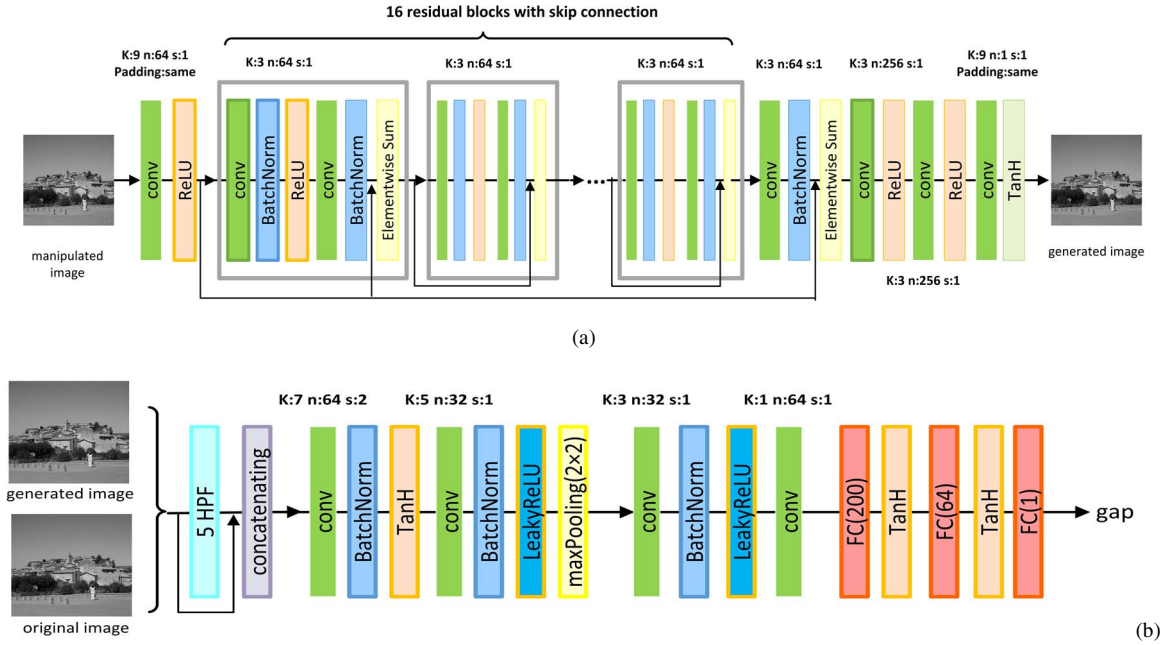


Fig. 1 The architecture of the WGAN-GP framework for multiple-operation anti-forensics. (a) generator network. (b) critic network. The parameters *K*, *n* and *s* refer to the kernel size, the number of kernels and the stride of each convolutional layer respectively.

concatenated using five 5×5 high-pass filters, the same filters as that in [25]. Three convolutional layers with 64 filters of size 7×7×6 and 2×2 stride, 32 filters of size 5×5×64 and 1×1 stride, and 32 filters of size 3×3×32 and 1×1 stride respectively are then deployed. Each of these convolutional layers is followed by a BN layer, a TanH or LeakyReLU as the activation function, and a 2×2 max-pooling layer with a stride of 2×2. Next, to achieve cross-channel information interaction and integration, we apply a convolutional layer with sixty four 1×1 kernels and a stride of 1×1. Finally, three fully connected (FC) layers are used, followed by the TanH activation function, to obtain a gap between the distributions of generated images and the original ones.

C. *The Loss Function of the Generator Network*

From the perspective of the generator, we want the generated images are similar with the original images, statistically and perceptually. Hence, the loss function for G are:

$$L_G = \mathbb{E}_{\mathbf{x}'}\left[\alpha L_G^{pixel} + \beta L_G^{vgg} + \gamma L_G^{adv}\right], \tag{1}$$

where $\mathbf{x}'$ refers to the manipulated images; L_G^{pixel}, L_G^{vgg}, and L_G^{adv} denote the pixel-wise loss, the perceptual loss [28], and the adversarial loss, respectively; α, β and γ represent the pre-defined weights of each loss. The purpose of the training process for G is to seek optimal parameters for minimizing L_G.

1) *Pixel-Wise Loss*

Adopting the pixel-wise loss helps to obtain high PSNR in the training process. Given an original image $\mathbf{x}$ and its corresponding manipulated image $\mathbf{x}'$, both with the size of $W \times H$, the pixel-wise loss L_G^{pixel} is [27]:

$$L_G^{pixel} = \frac{1}{N}\sum_{i=1}^{N}|\mathbf{x}_i - G(\mathbf{x}')_i|, \tag{2}$$

where the subscript i represents the pixel index of an image, $N = W \times H$, and $G(\mathbf{x}')$ refers to a generated image.

2) *Perceptual Loss*

Perceptual loss represents the loss in image texture details, and it contributes to the generation of visually realistic images. It is defined as a l_2 loss with respect to the differences in the CNN feature maps between the output generated image and its corresponding original image [28]:

$$L_G^{vgg} = \frac{1}{N}\sum_{i=1}^{N}\left\|\emptyset(\mathbf{x})_i - \emptyset\left(G(\mathbf{x}')\right)_i\right\|_2^2, \tag{3}$$

where $\emptyset(\cdot)$ indicates the feature map acquired from the output of the 12th convolutional layer within the VGG-19 network, pre-trained on ImageNet [28]; i and N represent the element index and the total number of elements in the feature maps, respectively.

3) *Adversarial Loss*

From the perspective of the generator, we want the images generated by G to narrow the gap in the distributions of the original images and the generated images. Thus, we can define the adversarial loss L_G^{adv} as

$$L_G^{adv} = -C\left(G(\mathbf{x}')\right), \tag{4}$$

where $C(\cdot)$ refers to the output of the critic network.

D. *The Loss Function of the Critic Network*

G and C are trained iteratively. G is fixed when C is trained, and vice versa. C is trained to maximize the gap between the distributions of original images and generated ones, *i.e.*, the loss function of C:

$$L_C = \mathbb{E}_{\boldsymbol{x}}[C(\boldsymbol{x})] - \mathbb{E}_{\boldsymbol{x}'}\left[C\left(G(\boldsymbol{x}')\right)\right] + \lambda\mathbb{E}_{\hat{\boldsymbol{x}}}[(\|\nabla_{\hat{\boldsymbol{x}}}C(\hat{\boldsymbol{x}})\|_2 - 1)^2] \tag{5}$$

TABLE I
OPERATIONS, PARAMETERS AND DESCRIPTIONS USED TO BUILD OUR 20 EXPERIMENTAL IMAGE DATASETS WITH RANDOM PARAMETERS WITHIN THE GIVEN RANGE

Operation	Parameters and descriptions
Additive White Gaussian Noise (AWGN)	variance: 1.0, 1.44, ..,4.0
Median Filtering (MF)	window size: 3, 5, 7
JPEG compression (JPG)	QF: 55, 56, ... , 90
Gamma Correction (GC)	Gamma: 0.5, 0.6, ... ,2.0
S Mapping (SM) [7]	$m(x) = round(255(arcsin(2x/255 - 1)/\pi + 1/2))$
Unsharp Masking Sharpening (UMS) with σ= 0.5, 0.6, ... , 1.5. [8]	λ = 0.5, 0.6, ... , 1.5. [8]
Gaussian Blurring (GB) with σ = 0.5, 0.6, ... , 1, 1.2, 1.5, 1.8, 2.[8]	window size: 3, 5, 7

where $\hat{\mathbf{x}} = \epsilon\mathbf{x} + (1-\epsilon)G(\mathbf{x}')$ means random samples which is sampled uniformly from $\mathbf{x}$ and $G(\mathbf{x}')$; $\epsilon \sim U[0, 1]$ is a uniform distributed random number; $\|\cdot\|_2$ means the l_2 norm; λ is the gradient penalty coefficient constant and is set to 10 in this work.

III. EXPERIMENTAL RESULTS

We used public image dataset BossBase V1.01 (BossBase) to create twenty multiple-operation manipulated image datasets for training, and used public image dataset BOWS2-Original (BOWS) to create twenty multiple-operation manipulated image datasets for testing. Each manipulated image has been subjected to a chain of two or three operations. Each manipulation is with random parameters which are listed in Table I. In Table I, SM operation is a popular contrast enhancement operation [7]; UMS operation [8] is a sharpening method in popular softwares such as Adobe Photoshop, and is implemented with the MATLAB command imsharpen(·) in this work, σ denotes the standard deviation.

For each multiple-operation chain, we applied the proposed chain anti-forensic scheme as follows:

(1) Train WGAN-GP networks using the training image pairs which includes an original image and a multiple-operation manipulated one;

(2) Generate 10000 anti-forensically modified images with the trained G in Step 1 using the test images;

(3) Evaluate performance of the proposed anti-forensic scheme with state-of-the-art forensic methods, and compare the visual quality before and after anti-forensics.

A. Training of WGAN-GP

The proposed WGAN-GP framework was implemented in Tensorflow and Tensorlayer and were trained on a workstation equipped with GPU of Nvidia GTX TITAN XP. For data augmentation, we cropped four non-overlapping middle parts with size of 128×128 from each image in BossBase and thus obtained 80,000 training image samples for each multi-manipulation chains, half of which were original and half manipulated images.

In the training, we first trained G with a batch size of 16 manipulated images for two epochs. The learning rate is 5.0×10^{-4}. The weight terms $\alpha = 1$, $\beta = 0$ and $\gamma = 0$. Then we trained the whole WGAN-GP networks for 60 epochs. In each epoch, C was trained for two iterations with a batch of 32 images which contains 16 generated images and the original ones, while G was trained one iteration with the weight terms $\alpha = 1.0$, $\beta = 1.0 \times 10^{-6}$ and $\gamma = 1.0 \times 10^{-2}$. We employed Adam as an optimizer and $\beta_1 = 0$, $\beta_2 = 0.9$ and $\varepsilon = 10^{-8}$ for G and C. In the first 20 epochs, the learning rates of G and C

TABLE II
DETECTION ACCURACIES (%) OF FORENSIC DETECTORS AND AVERAGE PSNR (dB) AND SSIM VALUES FOR 12 KINDS OF TWO-OPERATION CHAINS AND OUR ANTI-FORENSICS OF THESE CHAINS

Manipulation	Bayar [29]	Chen [19]	PSNR	SSIM	Manipulation	Bayar [29]	Chen [19]	PSNR	SSIM
UMS_JPG	99.72	99.99	31.334	0.910	JPG_UMS	99.93	99.99	30.703	0.893
Anti_UMS_JPG	50.08	50.01	**37.301**	**0.964**	Anti_JPG_UMS	50.23	50.04	**35.799**	**0.948**
SM_JPG	99.96	100.00	23.588	0.888	JPG_SM	99.80	99.99	23.657	0.895
Anti_SM_JPG	50.37	50.00	**36.108**	**0.944**	Anti_JPG_SM	50.10	50.00	**37.352**	**0.957**
GC_JPG	99.89	99.99	19.657	0.852	JPG_GC	99.95	99.99	19.644	0.853
Anti_GC_JPG	50.00	49.99	**19.924**	**0.865**	Anti_JPG_GC	50.56	50.08	**27.681**	**0.925**
MF_JPG	99.94	100.00	29.724	0.794	JPG_MF	99.92	99.96	29.867	0.798
Anti_MF_JPG	49.98	50.01	29.425	**0.821**	Anti_JPG_MF	50.01	50.21	**31.401**	**0.881**
AWGN_JPG	99.87	99.99	36.970	0.953	JPG_AWGN	99.93	99.97	36.515	0.945
Anti_AWGN_JPG	50.03	50.00	36.833	0.952	Anti_JPG_AWGN	51.45	52.77	**36.752**	**0.951**
GB_JPG	99.97	100.00	28.415	0.823	JPG_GB	99.95	99.99	28.727	0.837
Anti_GB_JPG	50.01	50.00	**31.948**	**0.882**	Anti_JPG_GB	49.97	50.03	**34.583**	**0.931**

are set to 5.0×10^{-4} and 5.0×10^{-6} respectively, and was reduced to 1/10 times after each 20 epochs.

The trained G is used to generate anti-forensic images.

B. Multiple-Operation Anti-Forensics

We used one of the most common operation, JPEG compression, as a post-processing or pre-processing operation to imitate the physical reality of the acquisition process of the camera, image storage, or digital transmission over the Internet. We generated multiple-operation anti-forensic images denoted by Anti_M_N or Anti_M_N_O with the trained G, each from 10,000 manipulated BOWS images which has undergone a chain of two manipulations (denoted as M_N) or three manipulations from left to right (denoted as M_N_O), as can be seen in Table II and III, respectively.

We used two state-of-art forensic detectors [19], [29] to detect whether an image is original or modified. Table II shows the detection accuracies for 12 kinds of two-manipulation chains and their anti-forensic. Average detection accuracies with the detector in [29] decrease from 99.90% for these manipulation chains to 50.23% for their anti-forensics, and average detection accuracies with the detector in [19] decrease from 99.90% for these manipulation chains to 50.26% for their anti-forensics, thus the anti-forensically modified images can successfully fool the forensic detectors.

Table III shows the detection results for eight kinds of three-manipulation chains. It can be observed that each anti-forensics of three-manipulation chain achieved nearly 50% detection accuracy, *i.e.*, close to random guess, which also proves the effectiveness and generality of our scheme.

C. Comparisons of Image Quality

Fig. 2 shows an original image, the JPG_UMS manipulated image, and the generated anti-forensic image, i.e., Anti_JPG_UMS image. It is observed that the quality of the generated image and the manipulated image are very good. We also compared the image quality before and after anti-forensics in Table II and III in terms of the average PSNR and SSIM values with the original images as references. From Tables II and III, we can observe that the anti-forensic images have higher average PSNR and SSIM values in most cases than the images before anti-forensics, thus the proposed multiple-operation anti-forensic method is effective against forensic detectors without significantly degrading the quality of an image, and even enhancing its quality in most cases.

TABLE III
DETECTION ACCURACIES (%) OF FORENSIC DETECTORS AND AVERAGE PSNR (DB) AND SSIM VALUES FOR 8 KINDS OF THREE-MANIPULATION CHAINS AND OUR ANTI-FORENSICS OF THESE CHAINS

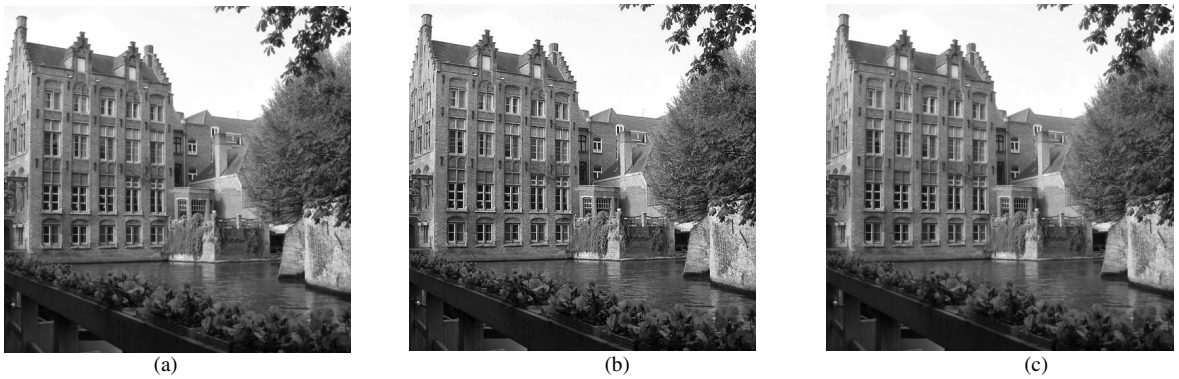
Manipulation	Bayar [29]	Chen [19]	PSNR	SSIM
MF_GB_JPG	99.99	100.00	26.114	0.713
Anti_MF_GB_JPG	49.99	50.00	**27.603**	**0.758**
GB_MF_JPG	99.97	99.99	27.378	0.745
Anti_GB_MF_JPG	50.00	50.01	**28.389**	**0.781**
_MF_UMS_JPG	99.93	99.99	29.286	0.796
Anti_MF_UMS_JPG	50.02	50.05	**29.364**	**0.823**
UMS_MF_JPG	99.96	99.99	29.077	0.789
Anti_UMS_MF_JPG	50.00	49.99	**29.111**	**0.821**
AWGN_MF_JPG	99.95	99.99	29.696	0.793
Anti_AWGN_MF_JPG	50.00	50.00	27.157	0.765
MF_AWGN_JPG	99.99	99.99	29.606	0.790
Anti_MF_AWGN_JPG	50.01	50.00	28.951	0.805
AWGN_UMS_JPG	99.86	99.93	30.937	0.900
Anti_AWGN_UMS_JPG	50.12	50.01	**37.106**	**0.962**
UMS_AWGN_JPG	99.70	99.99	31.150	0.907
Anti_UMS_AWGN_JPG	50.30	50.05	**37.110**	**0.963**

IV. CONCLUSIONS

In this work, we propose using WGAN-GP framework to model image anti-forensics as an image-to-image translation problem and obtain the optimized anti-forensic models for multiple-operation. The experimental results demonstrate that our multiple-operation anti-forensic scheme successfully deceives the state-of-the-art forensic algorithms without significantly degrading the quality of the image, and even enhancing the quality in most cases.

REFERENCES

[1] A. C. Popescu and H. Farid, "Exposing digital forgeries by detecting traces of resampling," *IEEE Trans. Signal Process.*, vol. 53, no. 2, pp. 758–767, Feb. 2005.



(a) (b) (c)

Fig. 2 A sample of the generated image. (a) original image, (b) JPG_UMS manipulated image, and (c) generated image

[2] Z. Fan and R. de Queiroz, "Identification of bitmap compression history: JPEG detection and quantizer estimation," *IEEE Trans. Image Process.*, vol. 12, no. 2, pp. 230–235, Feb. 2003.

[3] T. Bianchi and A. Piva, "Detection of nonaligned double JPEG compression based on integer periodicity maps," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 2, pp. 842–848, Apr. 2012.

[4] C. Chen and J. Ni, "Median filtering detection using edge based prediction matrix," *in Proc. Int. Workshop Digit. Forensics Watermarking*. Atlantic City, NJ, USA, 2011, pp. 361–375.

[5] X. Kang, M. C. Stamm, A. Peng, and K. J. R. Liu, "Robust median filtering forensics using an autoregressive model," *IEEE Trans. Inf. Forensics Security*, vol. 8, no. 9, pp. 1456–1468, Sep. 2013.

[6] M. C. Stamm and K. J. R. Liu, "Forensic detection of image manipulation using statistical intrinsic fingerprints," *IEEE Trans. Inf. Forensics Security*, vol. 5, no. 3, pp. 492–506, Sep. 2010.

[7] G. Cao, Y. Zhao, R. Ni, and X. Li, "Contrast enhancement based forensics in digital images," *IEEE Trans. Inf. Forensics Security*, vol. 9, no.3, pp. 515–525, Mar. 2014.

[8] G. Cao, Y. Zhao, R. Ni, and A. C. Kot, "Unsharp masking sharpening detection via overshoot artifacts analysis," *IEEE Signal Process. Lett.*, vol. 18, no. 10, pp. 603–606, Oct. 2011.

[9] M. Kirchner and R. Böhme, "Hiding traces of resampling in digital images," *IEEE Trans. Inf. Forensics Security*, vol. 3, no. 4, pp. 582-592, Dec. 2008.

[10] M. C. Stamm and K. J. R. Liu, "Anti-forensics of digital image compression," *IEEE Trans. Inf. Forensics Security*, vol. 6, no. 3, pp. 1050–1065, Sep. 2011.

[11] Y. Luo, H. Zi, Q. Zhang, and X. Kang, "Anti-forensics of JPEG compression using Generative Adversarial Networks," *in Proc. European Signal Processing Conf.*, Rome, Italy, Sep. 2018, pp. 957–961.

[12] Z. H. Wu, M. C. Stamm, and K. J. R. Liu, "Anti-forensics of median filtering," *in Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Vancouver, Canada, May 2013, pp. 3043–3047.

[13] 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 Process. Lett.*, vol. 25, no. 2, pp. 278–282, Feb. 2018.

[14] C.-W. Kwok, O. C. Au, and S.-H. Chui, "Alternative anti-forensics method for contrast enhancement," *in Proc. Int. Conf. Digital Forensics and Watermarking*, Atlantic City, NJ, USA, 2012, pp. 398–410.

[15] L. Lu, G. Yang, and M. Xia, "Anti-forensics for unsharp masking sharpening in digital images," *Int. J. Digital Crime Forensics*, vol. 5, no. 3, pp. 53–65, 2013.

[16] H. Ravi, A. Subramanyam, and S. Emmanuel, "Ace—An effective antiforensic contrast enhancement technique," *IEEE Signal Process. Lett.*, vol. 23, no. 2, pp. 212–216, Feb. 2016.

[17] G. Valenzise, M. Tagliasacchi, and S. Tubaro, "Revealing the traces of JPEG compression anti-forensics," *IEEE Trans. Inf. Forensics Security*, vol. 8, no. 2, pp. 335–349, Feb. 2013.

[18] H. Zeng, X. Kang, A. Peng, "A multi-purpose countermeasure against image anti-forensics using autoregressive model," *Neurocomputing*, vol. 189, pp. 117–122 , 2016.

[19] Y. Chen, X. Kang, Z. Jane Wang and Qiong Zhang, "Densely connected convolutional neural network for multi-purpose image forensics under anti-forensic attacks," *in Proc. 6th ACM Workshop Inf. Hiding Multimedia Secur.*, Innsbruck, Austria, 2018, pp. 91-96.

[20] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi, "Photo-realistic single image super-resolution using a generative adversarial network," *in Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Hawaii, USA, Jul. 2017, pp. 4681–4690.

[21] 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(NIPS)*, Montreal, Canada, 2014, pp. 2672–2680.

[22] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein Generative Adversarial Networks," *in Proc. Int. Conf. Machine Learning (ICML)*, Sydney, Australia, 2017, pp. 214-223.

[23] L. Yu, X. Long, C. Tong, "Single image super-resolution based on improved WGAN", *in Proc. Int. Conf. Advanced Control, Automation and Artificial Intelligence*, Shenzhen, China, 2018, pp. 101–104.

[24] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville, "Improved training of wasserstein gans," *in Advances in Neural Information Processing Systems (NIPS)*, Long Beach, USA, 2017, pp. 5767-5777.

[25] M. Chen, V. Sedighi, M. Boroumand, and J. Fridrich, "JPEG-phase-aware convolutional neural network for steganalysis of JPEG images," *in Proc. 5th ACM Workshop Inf. Hiding Multimedia Security*, Philadelphia, Pennsylvania, USA, 2017, pp. 75–84.

[26] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, and Li. Fei-Fei, "ImageNet: a large-scale hierarchical image database," *in Proc. IEEE Comput. Vis. Pattern Recognit.*, Miami Beach, FL, USA, 2009, pp. 248–255.

[27] H. Zhao, O. Gallo, I. Frosio, and J. Kautz, "Loss functions for image restoration with neural networks," *IEEE Trans. Comput. Imaging*, vol. 3, no. 1, pp. 47–57, Mar. 2017.

[28] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for realtime style transfer and super-resolution," *in Proc. European Conf. Comput. Vis. (ECCV)*, Springer, Amsterdam, Netherlands, 2016, pp. 694–711.

[29] B. Bayar and M. C. Stamm, "Design principles of convolutional neural networks for multimedia forensics," *in Proc. Int. Symp. Electron. Imag.*, San Francisco, USA, 2017, pp. 77–86.