Task-Guided Denoising Network for Adversarial Defense of Remote Sensing Scene Classification

Yonghao Xu[1], Weikang Yu[2] and Pedram Ghamisi[1,3]

[1]Institute of Advanced Research in Artificial Intelligence (IARAI), 1030 Vienna, Austria
[2]The Chinese University of Hong Kong in Shenzhen, School of Science and Engineering, 518172 Shenzhen, China
[3]Helmholtz-Zentrum Dresden-Rossendorf, Helmholtz Institute Freiberg for Resource Technology, Machine Learning Group, 09599 Freiberg, Germany

Abstract

Deep learning models have achieved state-of-the-art performance in the interpretation of geoscience and remote sensing data. However, their vulnerability to adversarial attacks should not be neglected. To address this challenge, we propose a task-guided denoising network to conduct adversarial defense for the remote sensing scene classification task in this study. Specifically, given an adversarial remote sensing image, we use a denoising network to transform it as close to its corresponding clean image as possible with the constraint of the appearance loss. Besides, to further correct the predicted logits, the perceptual loss and the classification loss are adopted with the aid of a pre-trained classification network with fixed weights. Despite its simplicity, extensive experiments on the UAE-RS (universal adversarial examples in remote sensing) dataset demonstrate that the proposed method can significantly improve the resistibility of different deep learning models against the adversarial examples.

Keywords

Adversarial defense, adversarial attack, adversarial example, remote sensing, scene classification, deep learning

1. Introduction

Recent advances in deep learning algorithms have significantly boosted the interpretation of geoscience and remote sensing data [1, 2]. Nevertheless, the vulnerability of deep learning models to adversarial examples should not be neglected. Szegedy et al. first discovered that deep neural networks are very fragile to specific perturbations generated by adversarial attack methods [3]. Simply adding these mild perturbations to the clean images, the adversarial examples are generated, which may possess imperceptible differences from the original images for human observers but could mislead the deep neural networks to make wrong predictions with high confidence. In fact, this phenomenon is not limited to computer vision tasks. Researchers have found that adversarial examples do exist in the geoscience and remote sensing field and can be generated based on optical data [4], LiDAR point cloud [5], or even synthetic aperture radar (SAR) data [6]. Since most geoscience and remote sensing tasks are highly safety-critical, it is vitally important to develop adversarial defense methods and improve the resistibility of the deployed deep learning model against adversarial examples.

One possible way to conduct adversarial defense is adversarial training, where both the original clean samples and the generated adversarial examples are combined to train the model [7, 8]. Nevertheless, adversarial training can hardly improve the inherent robustness of deep neural networks. Thus, the trained model can be attacked again by newly generated adversarial examples [9]. Another type of defense is to design novel architectures or modules that are more robust against adversarial examples. For example, the self-attention mechanism and the context encoding module are utilized in [10] to improve the inherent resistibility of deep neural networks. Despite its effectiveness, this method requires retraining the deployed models since it changes the architecture of the target models. Considering that retraining the deployed models may be infeasible in practical applications, it would be significant to develop adversarial defense methods that could directly decrease the harmfulness of the input adversarial examples.

To this end, transformation-based methods are developed which aim to remove or weaken the perturbations that exist in the adversarial examples. In [11], Tabacof et al. explored how different levels of the Gaussian noise could influence the classification performance on adversarial examples. Raff et al. further conducted more complex transformations like Gaussian blur, gray scale, and color jitter [12]. However, since these transformation-based methods may also cause new noises (e.g., Gaussian noise) or style differences (e.g., color jitter) to the transformed image, their defense performance is limited.

Different from the aforementioned methods, this study

CDCEO 2022: 2nd Workshop on Complex Data Challenges in Earth Observation, July 25, 2022, Vienna, Austria
yonghao.xu@iarai.ac.at (Y. Xu); weikangyu@link.cuhk.edu.cn (W. Yu); pedram.ghamisi@iarai.ac.at (P. Ghamisi)
0000-0002-6857-0152 (Y. Xu); 0000-0003-1111-572X (W. Yu); 0000-0003-1203-741X (P. Ghamisi)

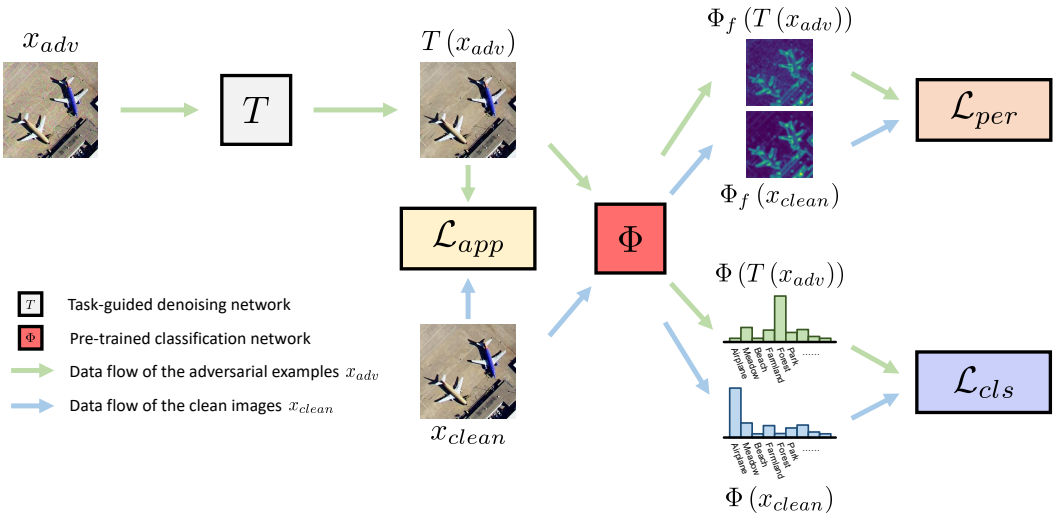

Figure 1: An illustration of the proposed adversarial defense framework with the task-guided denoising network (TGDN).

addresses the adversarial defense problem from the perspective of denoising. Specifically, we propose a novel task-guided denoising network (TGDN) for the adversarial defense of remote sensing scene classification. The main idea of the proposed method is to train a denoising network using clean remote sensing images and the corresponding adversarial examples. Since it is usually infeasible to know which attack method the adversary would use in practice, we adopt the iterative fast gradient sign method (I-FGSM) [13] to generate the adversarial examples for the simulation purpose in the training phase. Once the training is finished, the denoising network is expected to possess the defense ability against unknown adversarial attacks. Despite its simplicity, extensive experiments on the UAE-RS (universal adversarial examples in remote sensing) dataset [14] demonstrate that the proposed TGDN can significantly improve the resistibility of different deep learning models against the adversarial examples.

The rest of this paper is organized as follows. Section 2 describes the proposed TGDN in detail. Section 3 presents the experiments in this study. Conclusions and other discussions are made in Section 4.

2. Methodology

2.1. Overview of the Proposed TGDN

As shown in Figure 1, there are two main components in the proposed adversarial defense framework, including a task-guided denoising network T and a pre-trained classification network Φ (with fixed weights). Given a clean image x_{clean} from the training set, we first use I-FGSM [13] to generate the corresponding adversarial example x_{adv} (note that the use of I-FGSM is only to simulate the adversarial examples that may exist in the test set since we have no access to the real adversarial attack method adopted by the adversary in practice). Then, we use T to denoise x_{adv} and get the transformed image $T(x_{adv})$. Specifically, T aims to alleviate the difference between x_{adv} and x_{clean} from three aspects: the visual appearance difference, the feature representation difference, and the probability distribution difference. Accordingly, the training of T is constrained by the appearance loss $\mathscr{L}_{app}$, perceptual loss $\mathscr{L}_{per}$, and classification loss $\mathscr{L}_{cls}$ with the aid of Φ. Once the training is finished, we then use T to denoise samples in the adversarial test set.

2.2. Optimization

Since the adversarial perturbation also belongs a special type of noise, an intuitive idea is to conduct a transformation on the input adversarial example and remove the existing adversarial perturbation. To this end, we first adopt the ℓ_1 norm to define the appearance loss $\mathscr{L}_{app}$:

$$\mathscr{L}_{app} = \frac{1}{n_{ir}\, n_{ic}} \sum_{r=1}^{n_{ir}} \sum_{c=1}^{n_{ic}} |T(x_{adv})^{(r,c)} - x_{clean}^{(r,c)}|, \qquad (1)$$

where n_{ir} and n_{ic} denote the numbers of row and column in the image, respectively. The constraint in (1) will encourage the transformed image $T(x_{adv})$ to possess similar appearance to the original clean image x_{clean}.

Considering that the adversarial perturbation would also influence the intermediate feature representation of

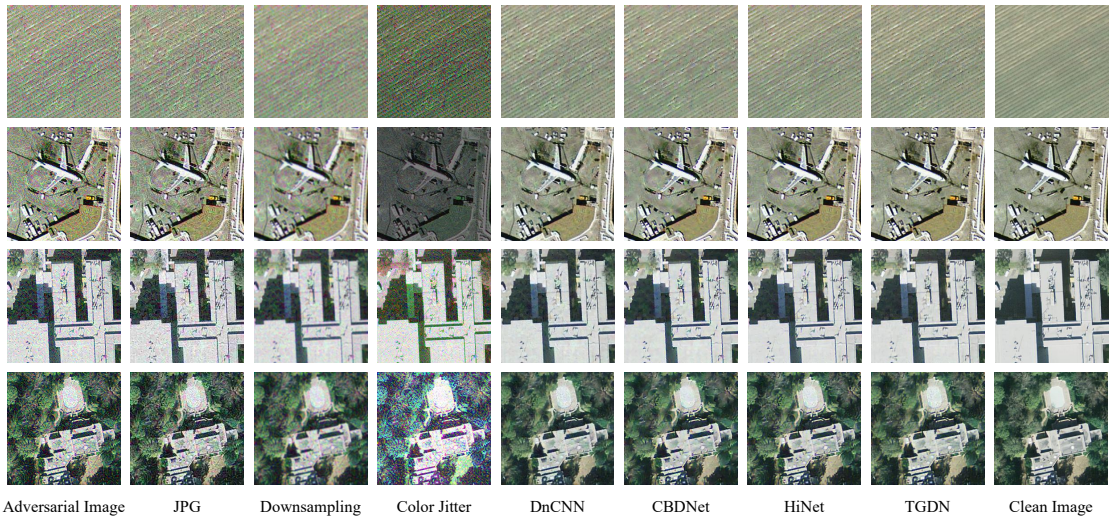

Figure 2: Example adversarial images in the UAE-RS UCM dataset and the corresponding transformed images using different methods.

Table 1
The overall accuracy (%) of different deep models on the UAE-RS UCM adversarial test set with different transforms.

	No Transform	JPG	Downsampling	Color Jitter	DnCNN	CBDNet	HiNet	TGDN (ours)
AlexNet	30.86	33.90	32.29	30.57	65.62	62.57	67.05	**73.90**
VGG11	26.57	28.57	32.00	26.48	52.38	50.00	57.24	**65.24**
VGG16	19.52	38.10	39.24	22.48	58.57	49.81	55.43	**68.67**
VGG19	29.62	32.00	42.86	32.00	51.81	46.00	59.62	**69.71**
Inception-v3	30.19	49.71	52.29	34.10	60.48	58.48	64.76	**74.48**
ResNet18	2.95	7.05	7.14	4.86	**11.52**	5.62	4.10	9.90
ResNet50	25.52	37.62	39.71	26.57	53.05	47.05	52.38	**65.81**
ResNet101	28.10	39.52	45.33	28.38	53.24	50.67	56.48	**69.43**
ResNeXt50	26.76	40.10	41.90	28.10	47.81	41.52	49.33	**63.24**
ResNeXt101	33.52	40.67	48.48	30.67	59.05	56.67	62.10	**74.67**
DenseNet121	17.14	35.90	31.90	24.86	48.29	43.71	45.81	**61.52**
DenseNet169	25.90	37.14	40.48	28.86	47.24	41.43	46.67	**59.81**
DenseNet201	26.38	40.67	48.67	32.29	52.57	43.81	51.33	**64.95**
RegNetX-400MF	27.33	32.29	40.29	27.05	51.81	49.81	56.67	**66.38**
RegNetX-8GF	40.76	41.52	48.38	34.57	56.76	53.43	63.71	**73.33**
RegNetX-16GF	34.86	54.67	55.14	34.95	68.19	64.19	69.05	**78.67**

the image in the deep neural network, we further define the perceptual loss $\mathscr{L}_{per}$:

$$\mathscr{L}_{per} = \frac{1}{n_{fr}\, n_{fc}} \sum_{r=1}^{n_{fr}} \sum_{c=1}^{n_{fc}} \| \Phi_f(T(x_{adv}))^{(r,c)} - \Phi_f(x_{clean})^{(r,c)} \|^2, \tag{2}$$

where n_{fr} and n_{fc} denote the numbers of row and column in the intermediate feature map, respectively. $\Phi_f(\cdot)$ denotes the output of the intermediate feature extraction layer in the pre-trained classification network Φ (with fixed weights). With the constraint in (2), $T(x_{adv})$ will tend to possess identical high-level feature representation to the original clean image [15].

Finally, we define the classification loss $\mathscr{L}_{cls}$ to clean the wrong logits in the output space of Φ:

$$\mathscr{L}_{cls} = \| \sigma(\Phi(T(x_{adv}))) - \sigma(\Phi(x_{clean})) \|^2, \tag{3}$$

where $\sigma(\cdot)$ denotes the softmax function, and $\Phi(\cdot)$ is the predicted logits of Φ. With the constraint in (3), $T(x_{adv})$ will tend to possess similar probability distribution to the original clean image on the pre-trained network Φ. The complete loss function $\mathscr{L}$ for training the proposed framework is formulated as:

$$\mathscr{L} = \mathscr{L}_{app} + \lambda_{per}\mathscr{L}_{per} + \mathscr{L}_{cls}, \tag{4}$$

where λ_{per} is a weighting factor.

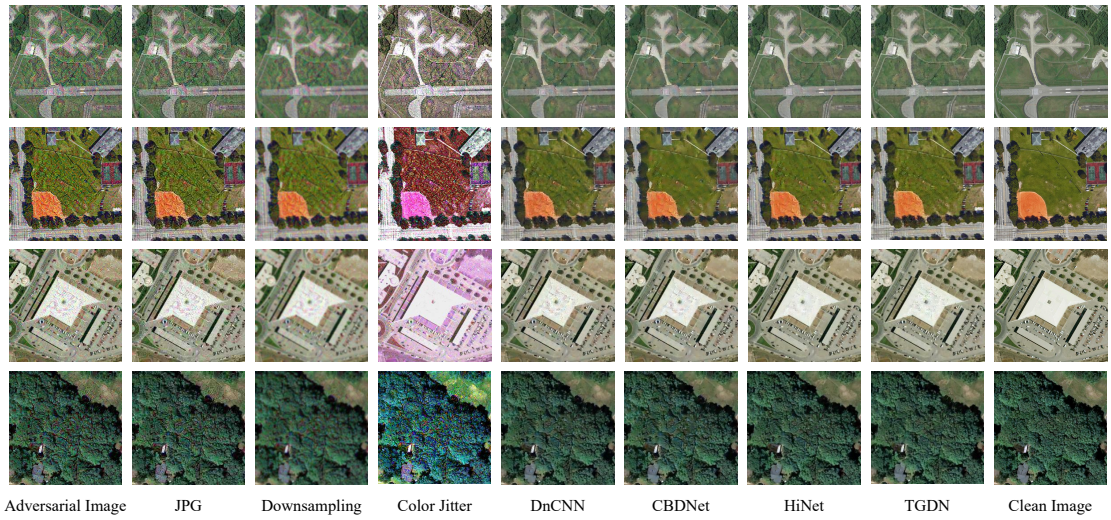

Figure 3: Example adversarial images in the UAE-RS AID dataset and the corresponding transformed images using different methods.

Table 2
The overall accuracy (%) of different deep models on the UAE-RS AID adversarial test set with different transforms.

	No Transform	JPG	Downsampling	Color Jitter	DnCNN	CBDNet	HiNet	TGDN (ours)
AlexNet	21.54	22.30	31.76	19.78	53.56	54.92	60.54	**63.40**
VGG11	15.40	16.26	21.46	14.08	39.60	41.14	48.14	**49.78**
VGG16	11.88	13.76	16.42	11.46	40.92	43.04	47.78	**51.34**
VGG19	12.64	18.78	20.44	15.66	35.10	37.88	43.00	**54.02**
Inception-v3	21.54	36.12	39.62	22.46	49.06	47.46	53.16	**65.14**
ResNet18	1.28	7.72	4.96	2.28	10.50	7.60	4.60	**24.22**
ResNet50	10.74	18.40	19.84	8.06	40.26	37.58	41.10	**60.66**
ResNet101	11.56	24.30	24.02	13.86	43.74	42.14	45.04	**66.90**
ResNeXt50	8.12	26.50	26.84	12.00	40.10	38.56	44.66	**62.94**
ResNeXt101	7.86	20.64	29.18	9.10	45.84	43.14	47.20	**63.26**
DenseNet121	10.30	16.52	21.74	14.70	36.82	35.48	39.28	**58.16**
DenseNet169	10.78	17.90	20.82	12.94	36.48	32.16	38.18	**56.62**
DenseNet201	14.22	24.28	29.14	16.00	42.90	38.90	44.12	**65.60**
RegNetX-400MF	23.20	27.26	28.92	18.24	45.10	41.64	51.28	**64.78**
RegNetX-8GF	18.92	30.22	31.26	16.28	47.76	47.86	55.02	**66.88**
RegNetX-16GF	21.00	33.82	32.06	20.00	49.44	49.52	55.20	**65.26**

3. Experiments

3.1. Dataset

The UAE-RS (universal adversarial examples in remote sensing) dataset[1] is utilized to evaluate the performance of the proposed method.

UAE-RS provides high-resolution remote sensing adversarial examples for both scene classification and semantic segmentation tasks [14]. For the scene classification task, UAE-RS contains 1050 adversarial test samples from the UCM dataset and 5000 adversarial test samples from the AID dataset generated by the Mixcut-Attack method. Some example adversarial images are shown in the first columns of Figures 2 and 3.

3.2. Implementation Details

We adopt the JPG [16], Downsampling [17], Color Jitter [12], DnCNN (Denoising CNN) [18], CBDNet (Convolutional Blind Denoising Network) [19], HiNet (Half Instance Normalization Network) [20], along with the proposed TGDN to conduct adversarial defenses. For the JPG method, we compress the adversarial examples

[1] https://github.com/YonghaoXu/UAE-RS

with a quality of 25 [21]. The Downsampling method is implemented with a sampling rate of 0.5 by the bilinear interpolation. For the Color Jitter method, we randomly change the brightness, contrast, saturation, and hue of the adversarial examples using the uniform distribution from 0.5 to 1.5.

The I-FGSM [13] with the ℓ_∞ norm is adopted to generate adversarial examples for training the denoising networks used in this study. The perturbation level in I-FGSM is fixed to 1 and the number of total iterations is set to 5. We use the same transform network used in the HiNet to implement the task-guided denoising network T. The ResNet18 [22] pre-trained on the UCM dataset or the AID dataset is adopted as the classification network Φ. The weighting factor λ_{per} in (4) is set to $1e-3$. We use the Adam optimizer [23] with a learning rate of $1e-3$ and a weight decay of $5e-5$ to train the denoising networks used in this study. The batch size is set to 32, and numbers of training epochs are set to 100, and 30 for UCM and AID datasets, respectively. All experiments in this study are implemented with the PyTorch platform [24] using two NVIDIA Tesla A100 (40GB) GPUs.

3.3. Experimental Results

To qualitatively evaluate how different transformations would influence the input adversarial examples, we first visualize some example adversarial images in the UAE-RS dataset and the corresponding transformed images with different methods in Figures 2 and 3. Compared to the traditional transformation methods like the JPG or Downsampling, the transformed images generated by denoising methods generally possess much more similar appearances to the original clean images. Besides, it can be observed that the visual appearance difference of the denoised images generated by DnCNN, CBDNet, HiNet, and TGDN is very difficult to perceive for human observers.

We further test the overall accuracy (OA) of different deep learning models on the UAE-RS dataset using different transforms to quantitatively evaluate how these defense methods would influence the classification performance. As shown in Tables 1 and 2, due to the threat of adversarial attacks, the existing state-of-the-art deep learning models can hardly achieve satisfactory recognition results if no transform process or defense method is used. On the UAE-RS AID adversarial test set, all models used in this study can only achieve an OA of less than 25% without transform. Besides, the improvements obtained from traditional transform methods like the JPG, Downsampling, and Color Jitter are limited and not stable. In some cases, they would even decrease the accuracy as these methods may bring about new noises or style differences, which are harmful to the deployed models. Compared to traditional transform methods, the performance improvements obtained by denoising methods are more obvious. However, although all the denoising networks used in this study may yield similar results from the perspective of the visual appearance according to Figures 2 and 3, their quantitative defense performance may vary a lot in different scenarios. Take the VGG16 model on the UCM adversarial test set for example. While the proposed TGDN can yield an OA of around 68%, CBDNet can only yield an OA of around 49% in this case. This phenomenon indicates that simply using traditional denoising networks may not defend the adversarial attacks effectively since there is no specific design to tackle the adversarial perturbations in traditional denoising methods. Even though the denoised images are very similar to the original clean images, there may still exist imperceptible adversarial perturbations that are harmful to the recognition performance. By contrast, the proposed TGDN can achieve the highest OA in all defense scenarios except the case of the ResNet18 on the UCM adversarial test set, where TGDN ranks second place. These results demonstrate the effectiveness of the proposed method.

4. Conclusions and Discussions

Although deep learning-based methods have achieved state-of-the-art performance in the interpretation of geoscience and remote sensing data, their vulnerability to adversarial examples can not be ignored in practical applications. To address the threat of adversarial examples for the remote sensing scene classification task, we propose a novel task-guided denoising network (TGDN) to conduct the adversarial defense in this study. Specifically, the proposed TGDN aims to alleviate the difference between the adversarial examples and the original clean images from three aspects: the visual appearance difference, the feature representation difference, and the probability distribution difference. To further evaluate how TGDN would influence the classification results of different deep learning models, the UAE-RS dataset is used in the experiments. Despite the simplicity of the proposed TGDN, extensive experiments demonstrate that TGDN can significantly improve the resistibility of different deep learning models against the adversarial examples.

Since the proposed method only considers a single pre-trained network (ResNet18) when training the task-guided denoising network, whether the ensemble learning with multiple pre-trained networks would improve the defense performance deserves further study. We will try to explore it in our future work.

References

[1] P. Ghamisi, J. Plaza, Y. Chen, J. Li, A. J. Plaza, Advanced spectral classifiers for hyperspectral images: A review, IEEE Geosci. Remote Sens. Mag. 5 (2017) 8–32.

[2] Y. Xu, B. Du, L. Zhang, D. Cerra, M. Pato, E. Carmona, S. Prasad, N. Yokoya, R. Hänsch, B. Le Saux, Advanced multi-sensor optical remote sensing for urban land use and land cover classification: Outcome of the 2018 ieee grss data fusion contest, IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens. 12 (2019) 1709–1724.

[3] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, R. Fergus, Intriguing properties of neural networks, arXiv preprint arXiv:1312.6199 (2013).

[4] L. Chen, Z. Xu, Q. Li, J. Peng, S. Wang, H. Li, An empirical study of adversarial examples on remote sensing image scene classification, IEEE Trans. Geos. Remote Sens. 59 (2021) 7419–7433.

[5] Y. Cao, C. Xiao, B. Cyr, Y. Zhou, W. Park, S. Rampazzi, Q. A. Chen, K. Fu, Z. M. Mao, Adversarial sensor attack on lidar-based perception in autonomous driving, in: Proceedings of the 2019 ACM SIGSAC conference on computer and communications security, 2019, pp. 2267–2281.

[6] H. Li, H. Huang, L. Chen, J. Peng, H. Huang, Z. Cui, X. Mei, G. Wu, Adversarial examples for cnn-based sar image classification: An experience study, IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens. 14 (2020) 1333–1347.

[7] I. J. Goodfellow, J. Shlens, C. Szegedy, Explaining and harnessing adversarial examples, arXiv preprint arXiv:1412.6572 (2014).

[8] Y. Xu, B. Du, L. Zhang, Assessing the threat of adversarial examples on deep neural networks for remote sensing scene classification: Attacks and defenses, IEEE Trans. Geos. Remote Sens. 59 (2021) 1604–1617.

[9] N. Akhtar, A. Mian, N. Kardan, M. Shah, Advances in adversarial attacks and defenses in computer vision: A survey, IEEE Access (2021).

[10] Y. Xu, B. Du, L. Zhang, Self-attention context network: Addressing the threat of adversarial attacks for hyperspectral image classification, IEEE Trans. Image Process. 30 (2021) 8671–8685.

[11] P. Tabacof, E. Valle, Exploring the space of adversarial images, in: International Joint Conference on Neural Networks (IJCNN), IEEE, 2016, pp. 426–433.

[12] E. Raff, J. Sylvester, S. Forsyth, M. McLean, Barrage of random transforms for adversarially robust defense, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2019, pp. 6528–6537.

[13] A. Kurakin, I. Goodfellow, S. Bengio, Adversarial machine learning at scale, arXiv preprint arXiv:1611.01236 (2016).

[14] Y. Xu, P. Ghamisi, Universal adversarial examples in remote sensing: Methodology and benchmark, IEEE Trans. Geos. Remote Sens. 60 (2022) 1–15.

[15] J. Johnson, A. Alahi, L. Fei-Fei, Perceptual losses for real-time style transfer and super-resolution, in: European Conference on Computer Vision, Springer, 2016, pp. 694–711.

[16] G. K. Dziugaite, Z. Ghahramani, D. M. Roy, A study of the effect of jpg compression on adversarial images, arXiv preprint arXiv:1608.00853 (2016).

[17] C. Guo, M. Rana, M. Cisse, L. Van Der Maaten, Countering adversarial images using input transformations, arXiv preprint arXiv:1711.00117 (2017).

[18] K. Zhang, W. Zuo, Y. Chen, D. Meng, L. Zhang, Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising, IEEE Trans. Image Process. 26 (2017) 3142–3155.

[19] S. Guo, Z. Yan, K. Zhang, W. Zuo, L. Zhang, Toward convolutional blind denoising of real photographs, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2019, pp. 1712–1722.

[20] L. Chen, X. Lu, J. Zhang, X. Chu, C. Chen, Hinet: Half instance normalization network for image restoration, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2021, pp. 182–192.

[21] G. K. Wallace, The jpeg still picture compression standard, IEEE Transactions on Consumer Electronics 38 (1992) xviii–xxxiv.

[22] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2016, pp. 770–778.

[23] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980 (2014).

[24] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al., Pytorch: An imperative style, high-performance deep learning library, Advances in Neural Information Processing Systems 32 (2019).