A CycleGAN Adversarial Attack Method Based on Output Diversification Initialization

Peiyuan Liu*, Lei Sun, XiuQin Mao, Leyu Dai, Song Guo, Yu Yang
Information Engineering University, Zheng Zhou, China
*E-mail: ycxxtgcjzl@163.com

Abstract. The powerful image generation capabilities of generative adversarial networks (GAN) bring great threats to applications related to images. Style transfer networks realize the style transform between image domains through which we can easily modify images like portraits and calligraphy. To eliminate the negative impact caused by the forged images, there emerged technical methods to detect forged images, which might trigger remedial actions afterwards but cannot prevent maliciously tampered content from spreading over network media. Therefore, some scholars put forward the idea of protecting images from hostile generative networks with adversarial attack. However, the initial random noise of adversarial perturbation cannot be effectively mapped to the output space. In order to improve the visual effect of adversarial attacks, this paper proposes an adversarial attack algorithm based on output diversification initialization (ODI) for CycleGAN. We firstly utilize output diversification initialization to find an effective starting point for the adversarial attack, and then we use Project Gradient Descent (PGD) to iteratively attack the style transfer network by modifying the adversarial loss function. Experimental results demonstrate that the introduction of ODI can effectively enlarge the distance between the adversarial output and the original output. It achieves better results in identifying the forged images generated by the targeted model, and does not significantly increase the disturbance, which can guarantee the normal use of original images.

1. Introduction
Artificial intelligence has brought many breakthrough outcomes, but the abuse of technology easily causes serious security problems. Generative Adversarial Network [1] is currently a popular model in the field of image generation. The game between generator and discriminator enables generator to produce very realistic images, which makes it applicable to many novel scenes, such as virtual fitting [2], style transfer [3]-[4], face image generation [5]-[6]. In the field of style transfer, a trained style transfer network can convert an ordinary photo into an art work like Van Gogh’s paintings, and can also manipulate on human faces. In other words, GAN has the ability to forge images, and might cause security and privacy issues when abused under some special conditions, exerting extremely bad impact on the society like Deepfakes [7].

In face of the threats caused by GAN forged images, the academic community is also trying to find various methods to limit the proliferation of deepfakes. Mccloskey et al. [8] distinguished fake images from real images by analysing the difference in colour processing between GAN and real cameras. However, the method of detecting fake images and videos through detection cannot directly prevent the harm to personal privacy caused by the forged content for the detected images or videos are most likely to have been spread on social platforms. Other studies have begun to consider how to affect the
GAN, such as using methods of poisoning attacks or adversarial attacks to make GAN cannot normally produce forged images. This type of method has a more direct effect and can protect personal images from malicious use by forged algorithms.

Adversarial attack uses adversarial examples to attack deep learning model and is mainly applied to the deep classification model. By adding adversarial perturbations that are imperceptible to the naked eye in the image, the deep classification model will produce incorrect classifications with high confidence. Accordingly, with regard to the abuse of GAN in deep forgery, it seems natural to consider extending adversarial example from deep classification models to GANs, which is currently a relatively new and less researched area. The main goal of adversarial attack is to construct adversarial examples to make GAN produce false forgery results. Aiming at controlling the distance between output image and original image, current researches mainly focus on adversarial example generation methods, such as gradient-based ones like FGSM [9], PGD [10] and other classic ones. However, the iterative starting point of the initialization commonly used in the generation is random, which sometimes produces a negative attack effect. [11] proposed an initialization method called Output Diversification Initialization (ODI), which initializes adversarial examples according to the distance in the output space. In the classification model, ODI can improve attack performance and provide a more diverse and effective iterative starting point. Consequently, this paper introduces this initialization method to improve the PGD-based adversarial example generation with CycleGAN as the target. We have conducted experiments on two measurement indicators to verify the effectiveness of our method. Results show that it can improve the success rate of the attack and produce outputs that is farther away from the original image.

2. Related Work

2.1. GAN-based style transfer algorithm

Generative adversarial networks are proposed by GoodFellow et al. [1] in 2014 and mainly used in the field of image generation. The discriminator $D$ and generator $G$ is used for adversarial training. The target loss functions of GAN are as follows:

$$
\min_G \max_D V(D,G) = E_{x \sim P_{data}(x)}[\log D(x)] + E_{z \sim P_{z}(z)}[\log(1 - D(G(z)))]
$$

The basic idea of GAN originates from the adversarial goals of generators and discriminators. The ultimate goal is to produce realistic images to deceive while the goal of $D$ is to distinguish real images from false images, and that of $G$ is to improve the quality of the generated images during adversarial training.

The target network CycleGAN [3] for adversarial attack in this paper belongs to the field of style transfer, which uses two styles of images for training, such as men and women, winter and summer, etc. The final trained model can complete the transfer between two image domains. Different from the classic GAN, CycleGAN uses a dual structure composed of two discriminators and two generators, and introduces a cycle consistency loss function, which ensures that the image is roughly equivalent to the original image during the style transfer process. The main innovation of CycleGAN is that it can complete the style transfer from the source domain to the target domain without establishing a paired mapping between training data.

On the basis of CycleGAN, several other corresponding networks have been proposed for style transfer tasks. StarGAN [12] proposed by Choi et al. uses only one generator to achieve style transfer between domains. CoGAN [13] shares the parameters of the last few layers of the two decoders, making the facial feature structures extracted by the network similar. XGAN [14] adopts the concept of Semantic Consistency, that is, to retain the image features during the style transfer process instead of retaining the image content, which avoids feature information loss during the style transfer process. In view that most style transfer networks adopt CycleGAN as their basic network structure, this paper chooses CycleGAN as the target model of the adversarial attack.
2.2. Adversarial Example

Adversarial examples refer to injecting artificially constructed perturbations into the input of the deep learning model, so that the deep learning model gives wrong output. It is firstly introduced by Szegedy et al. [15], who proposes the L-BFGS method to construct adversarial examples. For a particular classifier \( f \), the formula to solve the least disturbance \( \eta \) that makes the model classified as \( l' \) is as follows:

\[
\min c\|\eta\|_p + loss_f(x + \eta, l)
\]  

(2)

Where \( l \) is the original classification result, \( c \) is the hyperparameter, and the constraint conditions that the equation (2) must satisfy is:

\[
x + \eta \in [0,1]^m; f(x + \eta) = l'
\]  

(3)

The adversarial example constructed by the L-BFGS method is \( x' = x + \eta \). On this basis, the Fast Gradient Sign Method (FGSM) [9] is proposed, which reduces the complexity of the L-BFGS calculation process and can generate a large number of adversarial examples in a short time. Later, by constructing a clip function, Kurakin of Google Brain proposes the Basic Iterative Method [16] (BIM), which reduces the part of FGSM that has large changes to the input and achieves a good attack effect. The Projected Gradient Descent (PGD) method is a variant of BIM and is considered to be the most powerful first-order attack method. Therefore, we choose PGD as the basic attack method.

2.3. GAN-oriented adversarial attack

In response to the problems caused by the abuse of GAN in deepfakes, how to extend the adversarial attack from a classification model to a generative one has captured lots of attention. Dario et al. [19] discovered the extra-domain samples of the deep generative model for the first time, which proved that it is statistically indistinguishable from the set of real inputs, but it can make the generative model to produce completely different outputs. Shan et al. [20] started from the dataset required for model training, and added a cloak to the user's image to generate an adversarial dataset. [21] and [22] applied adversarial example generation methods to implement attacks on style transfer networks such as StarGAN and CycleGAN. Especially, [21] utilized PGD to achieve the adversarial attack on the CycleGAN model, which successfully prevented the normal style transfer of the model and produced a visually distorted output image. At present, this work is the first one to implement a complete adversarial attack against GAN, provide a concrete and feasible solution for future generations, and propose corresponding measurement indicators for qualitative comparisons of the attack results. Therefore, this paper attempts to improve the PGD method on the basis of the literature [21] to improve the attack effect of CycleGAN.

3. CycleGAN Adversarial Attack Method Based on Output Diversification Initialization

We introduce output diversification initializations to find a more reasonable and diverse initial starting point for the iterations of the PGD method, thereby improving the overall performance of adversarial attacks for CycleGAN. In this section, we first introduce how to generate initial iteration points for PGD by ODI, and then further explain how to perform adversarial attacks on target networks.

3.1. Output Diversification Initialization

For adversarial attacks on a classification model, the ODI is put forward to replace the random initialization method used in adversarial example generation which cannot guarantee effective output, and may cause the final output be close to the original input. Different from the classification models, the output of GANs are generated images and have more variability than classification labels, and the mechanism of GAN to generate images is more complicated than that of ordinary deep classification networks. Consequently, the negative impact that random initialization has on the effect of the adversarial attack will be more prominent than classification models. For this reason, we introduce ODI as an initialization method for constructing adversarial examples. It directly generates a starting point as far away as possible from the original input based on the distance of output space, and applies
the gradient-based method for iterations to ensure that the output image is certainly different from the
original image.

Suppose the input image is $x_{org}$, and a trained GAN generator model is $G$. We randomly select an
initialization direction, which is represented by the vector $w_d$. The size of $w_d$ is the same as the output
of $G$, which satisfies the uniform distribution in the interval $[-1, 1]$. According to the output of the
model and the direction vector, the perturbation vector is defined as follows:

$$v_{ODI}(x_{org}, G, w_d) = \left( \nabla_x \left( w_d^T G(x_{org}) \right) \right) \cdot \left( \left\| \nabla_x \left( w_d^T G(x_{org}) \right) \right\|_2 \right)^{-1}$$ (4)

Obviously, the initial perturbation vector contains the original image input and the vector
representing its direction, which can be regarded as the normalized gradient of the original output
projected on the direction vector. Therefore, we can use perturbation vector to find an attack starting
point far away from the initial image through a certain number of iterations. We measure the distance
between this starting point and the original output in the output space of the model, which mean to
maximize the following objective function:

$$w_d^T \left( G(x_k) - G(x_{org}) \right)$$ (5)

According to this objective function and the principle of gradient update, the corresponding
iterative formula is designed to update the original image:

$$x_{k+1} = Proj_{B(x_{org})} \left( x_k + \eta_{ODI}(v_{ODI}(x_k, G, w_d)) \right)$$ (6)

Where $B$ represents the set of perturbations that are allowed to be generated to ensure that the
updated image satisfies the requirements of the adversarial perturbation size, that is, $\| v_{ODI} \|_\infty \leq \varepsilon$,
and $\eta_{ODI}$ represents the step size. After several iterations, we can get an attack point which is further
away from the initial image, and the gradient adversarial example generation method is further used on
this attack point to iterate. In addition, $w_d$ will be resampled every time the experiment is restarted to
ensure the diversification of the attack results.

3.2. Adversarial Attack Method

We choose PGD to complete the attack on the style transfer network, and choose CycleGAN as the
main target network. In light of the basic definition of adversarial examples [15], the corresponding
adversarial example image $x_{adv}$ is constructed for the original image $x$ as follows:

$$x_{adv} = x + \eta$$

$$\max_{\eta} \mathcal{L}(G(x + \eta), r), \quad s.t. \| \eta \|_\infty \leq \varepsilon$$ (7)

Where $\eta$ represents the adversarial perturbation, $r = G(x)$ is the result of the original image being
tampered with, and $\mathcal{L}$ represents the image distance between the adversarial image and the original
image after being tampered with. With the perturbation $\eta$ that can maximize $\mathcal{L}$ to affect the output of
GAN, the output of the adversarial image after being tampered with can be more easily recognized as
a forged image.

The basic attack steps of PGD are as follows:

$$x_0^{adv} = x + noise$$

$$x_{n+1}^{adv} = clip \left( x_n^{adv} + \alpha \text{sign} \left( \nabla_x \mathcal{L}_{adv}(x_n^{adv}) \right) \right)$$ (8)

Where $noise$ represents an initial random noise, $n$ is the number of iterations, $\alpha$ is the adaptive step
size, $\nabla_x \mathcal{L}_{adv}$ is the gradient of the adversarial loss function, and the overall iteration result is
controlled by the clip function to a certain range.

In general, PGD obtains initial noise through random sampling, such as sampling in Gaussian
distribution, while the noise in this article is generated by ODI, which can obtain an attack starting
point that makes the attack effect more obvious. In addition, due to the matrix characteristics of the
image data, the $\text{sign}()$ function is removed in the calculation process, and the gradient of the anti-loss
function is directly used for iteration, which is beneficial to improve the diversity of anti-disturbance.
Thus, the basic iterative formula obtained above improvements is as follows:
The adversarial loss function should have the ability to measure the distance of the image. If the output image of the target network can be away from the original result as far as possible, the output will be more visually-distorted and the result of style transfer is more likely to be regarded as a fake image. Therefore, the basic form of the adversarial loss function is designed as follows:

\[
L_{adv}(x) = L(G(x_{adv})) - G(x)
\]

(10)

According to comparative experiments of [21], when performing PGD attacks against style transfer networks, taking \( L(x) = x^2 \) as a function of image distance can achieve better attack effects. Because norms such as \( \ell_2 \) and \( \ell_\infty \) have a weaker effect on the model output, and the higher-order functions will cause greater disturbance to the original image, which is not conducive to the use of the original image.

3.3. Overall Algorithm

The algorithm of adversarial attack is divided into two parts as a whole. First, using ODI to perform a certain number of iterations to adjust the original image to a more effective starting point, and then using PGD to perform an adversarial attack on the target model and to complete further iterations to generate an adversarial example image. The pseudo code is as follows.

**Algorithm 1. Adversarial Attack Method Based on Output Diversification Initialization**

| Input: Target image \( x \), Target network \( G \), Adversarial Loss Function \( L_{adv} \) | Output: Adversarial Image \( x_{adv} \) |
| --- | --- |
| Parameters: Iteration Numbers for ODI \( s_1 \), Iteration Numbers for PGD \( s_2 \), Step Size for ODI \( \eta_{ODI} \), Step Size for PGD \( \alpha \) |

1. Set \( w_d \sim U(-1,1) \), \( x_0 = x \)
2. for \( i \) in \( s_1 \)
3. \( v_{ODI} = v_{ODI}(x_i, G, w_d) \)
4. \( x_{i+1} = \text{Proj}_{B(x)}(x_i + \eta_{ODI}(v_{ODI}(x_i, G, w_d))) \)
5. End for
6. \( j = s_1 \)
7. for \( j \) in \( s_1 + s_2 \)
8. \( x_{j+1} = \text{clip}(x_j + \alpha \cdot \nabla_x L_{adv}(x_j)) \)
9. \( x_{adv} = x_{j+1} \)
10. End for

4. Experiments Analysis and Results

4.1. Dataset

We choose CelebFaces Attribute, abbreviated to CelebA, as the training dataset, which is openly provided by the Chinese University of Hong Kong and widely used in face-related computer vision training tasks such as face attribute identification training, face detection training and landmark marks, etc. It consists of 202,599 face photos from 10,177 celebrities with resolution of 178×218 pixels and provides feature markers including standard face frames, 5 facial feature point coordinates, and 40 attribute markers. Among them, the 40 attribute tags of the face include many face styles features, such as baldness, blond hair, etc., which are helpful for the training of the style transfer model.
4.2. Implementation Details
In pre-processing stage, we convert the image to the resolution of 224×224 pixels, normalize it and send it into the generator of CycleGAN in the form of RGB with three channels. In ODI stage, the direction vector is generated according to a uniform distribution, the number of iterations is 2, and the step size $\eta_{odi}$ is set to 0.02. In PGD attack stage, the iteration is continued on the basis of ODI, the number of iterations is 8, $\alpha$ is set to a quarter of $\eta_{odi}$, $\epsilon$ equals 0.2, and the final output data will be clipped to the range of $(-1,1)$. All experiments are done using Pytorch 1.2.0 and GeForce RTX 2080S, and the GPU memory is 8G.

4.3. Results
In this section, we first give the basic results of the experiment. Then, we conduct a quantitative analysis of the attack results by testing image distance and referring to the distortion scores given in the literature [21]. The comparison results demonstrates that the adversarial examples produced turn out to be more effective when the ODI is used for initialization. Finally, the adversarial perturbation generated by the ODI method is tested.

4.3.1. Result Case
Figure 1 illustrates the result of the adversarial attack performed by our method on CycleGAN with the first row displaying the original image, the second row is the output image of style transfer with the first row as the input, and the third one is the output after adding the adversarial example into the original image. According to the role of CycleGAN, we select four pre-trained models as target models for attack. They are: (a) Smile: style transfer between seriousness and smile; (b) Blond: style transfer between blonde hair and other hair colours; (c) Bald: style transfer between baldness and hair; (d) Glass: style transfer between wearing glasses and not wearing glasses. Comparing the third row with the second row of Figure 1, we can conclude that we successfully completed the adversarial attack on the image for the adversarial face with added perturbation can no longer be generated normally by the style transfer network. The image quality of the confrontational face after style transfer is so poor, that the naked eye can easily distinguish them from normal images.

In addition, we found that, though ODI decreases the randomness of the starting point by measuring the output space of target model, the results produced are different among multiple implementations of ODI since the random selection of $w_d$ ensures that the direction of each update is
different. Therefore, it is verified that ODI can not only generate diversified adversarial perturbations but also produce different attack results.

4.3.2. Image Distance Comparison
In order to give a more objective evaluation on the result of the adversarial attack against the generative model, we use the $L_2$ distance between images: $L_2 = \|G(x_{adv}) - G(x)\|_2$ as an indicator of the judgment. When $L_2 \geq 0.05$, the impact of the adversarial attack on the image is obvious for visual perception, i.e., the forged image content can be subjectively identified as a false image, which signifies a successful attack. This paper uses the CelebA to generate 20,000 images of style transfer for the four pre-training models, and conducts an attack test against these 80,000 images. We compare the average attack success rates of our method using ODI with that of PGD method using Gaussian random initialization as follows.

| Pretrained Model | PGD(Gaussian) | ODI-PGD |
|------------------|---------------|---------|
|                  | $L_2$ distance | Success Rate | $L_2$ distance | Success Rate |
| Smile            | 0.054         | 100     | 0.062         | 100          |
| Blond            | 0.077         | 100     | 0.080         | 100          |
| Bald             | 0.922         | 100     | 0.930         | 100          |
| Glass            | 0.056         | 100     | 0.060         | 100          |

As shown in Table 1, whether to initialize the noise using ODI or Gaussian distribution causes no difference in attack success rate. But the adversarial image generated by ODI is farther from the original output, which indicates that ODI has higher attack performance than random initialization and can cope with more demanding judgment conditions. Different from PGD method using Gaussian random initialization, our method performs iterations for the final output with a random direction vector. Even if the direction vector still guarantees the direction randomness of the perturbation generation, the result of ODI is more easily reflected in the output space, so it will get a better attack success rate than the original PGD.

4.3.3. Distortion Scores Comparison
In order to prove that ODI can bring a more effective starting point for the gradient-based adversarial example generation method, we introduce the distortion score [21] to measure the impact of the attack method on the output result. The measurement formula of the distortion score is as follows:

$$s_{dist} = \max\left(0, \log L(y_{adv} - y) \cdot \left(\log(L(x_{adv} - x))^{-1} - 1\right)\right)\quad (11)$$

Where $y_{adv}$ represents the adversarial output image, $y$ is the normal output image of the model, and $L(x) = x^2$ is taken as the distance measurement function. Distortion score represents the ratio between the adversarial perturbation and the output distance of the model, which is an evaluation of the overall attack effect of the method. The higher the distortion score is, the greater the impact of the added anti-disturbance has on the distance between the output images, and the better the effect of attack is.

We calculates the distortion scores of all images in each pre-training model and takes the average value. The comparative results are shown in Table 2:
Table 2. Distortion Scores Comparison

| Initialization Method | Distortion Scores |
|-----------------------|-------------------|
|                       | Smile  | Blond | Bald  | Glass |
| PGD(Gaussian)         | 0.16   | 0.16  | 0.20  | 0.14  |
| ODI-PGD               | 0.43   | 0.47  | 0.45  | 0.53  |

According to Table 2, the distortion score obtained by ODI is generally greater than the distortion score of random initializations. It can be seen that the ODI initialization method has a greater impact on the output of the model than random initialization. This is because the attack starting point obtained by ODI is farther away from the original image than the attack starting point obtained by random initialization. Therefore, it can produce visually more destructive fake images easy for the users to distinguish. Hence, the effectiveness and advantage of ODI over random initialization in attacks against GAN is verified.

4.3.4. Adversarial Perturbation Comparison

The above experiments show that the method in this paper can produce more diverse and more effective experimental results. In view that the main purpose of the adversarial attack on CycleGAN is to avoid its malicious tampering with the image, adversarial perturbation generated by the method should not affect the normal use of the image. ODI have a greater impact on the output of the model. In order to test whether the disturbance generated by the method in this paper is within a certain range, we compare the adversarial perturbation generated by our method with normal PGD, and calculate the perturbation with the $\ell^2$ norm.

Table 3. Adversarial Perturbation Comparison

| Pretrained Model | Norm of Perturbation |
|------------------|----------------------|
|                  | PGD(Gaussian) | ODI-PGD |
| Smile            | 0.081       | 0.075   |
| Blond            | 0.147       | 0.125   |
| Bald             | 0.143       | 0.119   |
| Glass            | 0.085       | 0.060   |

According to the experimental results, perturbations generated by our method satisfy the requirements of parameter setting $\varepsilon$, and its $\ell^2$ norm is not much different in the four pre-trained models compared with the randomly initialized PGD. It means that the introduction of ODI does not significantly increase the modification degree of original image, and the addition of perturbation is still invisible to the human eye. For the entire attack algorithm, ODI affects the model output of the style transfer network more effectively. The increase in the gap between the adversarial output and the original output is the main reason for the increase in the distortion score.

5. Conclusion

This paper proposes CycleGAN adversarial attack method based on output diversification initialization. We firstly use the output diversity initialization to generate noise disturbances and set the direction vector to ensure diversity. We iterate with the output of the model as the metric to generate effective and diverse attack starting points. Then we use PGD starting from the attack starting point and modify the adversarial loss function so as to complete the attack on the pre-trained model. Experiments show that extending the ODI-PGD method from the classification model to the style transfer generation model can improve the performance of the attack. Under the premise that the
disturbance size is not much different from the existing methods, the success rate of the attack is improved, the distance between the output result and the original image is enlarged, and a more effective and intuitive attack result is produced. The disadvantage is that this method currently targets fewer GAN models and is limited to gradient-based adversarial sample methods, the scope of which need to be extended in the follow-up.

References

[1] Goodfellow I, Pougetabadie J, Mirza M, et al. 2014 Generative Adversarial Nets. Neural Information Processing Systems 2672-2680

[2] R Yu, X Wang and X Xie 2019 VTNFP: An Image-Based Virtual Try-On Network With Body and Clothing Feature Preservation. IEEE/CVF International Conference on Computer Vision (ICCV) pp 10510-10519

[3] J Y Zhu, T Park, P Isola, and A A Efros 2017 Unpaired Image-to-image Translation Using Cycle-consistent Adversarial Networks. IEEE International Conference on Computer Vision

[4] P Isola, J Y Zhu, T Zhou, and A A Efros 2017 Image-to-image Translation with Conditional Adversarial Networks. IEEE Conference on Computer Vision and Pattern Recognition pp 1125–1134

[5] Natsume R, Yatagawa T, and Morishima S 2018 Rsgan: Faceswapping and editing using face and hair representation in latent spaces. ACM SIGGRAPH.

[6] Nirkin Y, Keller Y, Hassner T 2014 FSGAN: Subject agnostic face swapping and reenactment International Conference on Learning Representations

[7] Faceswap: Deepfakes software for all https://github.com/deepfakes/faceswap

[8] Mccloskey S, Albright M 2018 Detecting GAN-generated Imagery using Color Cues Computer Vision and Pattern Recognition

[9] Goodfellow I, Shlens J, Szegedy C 2015 Explaining and Harnessing Adversarial Examples. International Conference on Learning Representations

[10] Madry A, Makelov A, Schmidt L, et al. 2018 Towards Deep Learning Models Resistant to Adversarial Attacks International Conference on Learning Representations

[11] Tashiro Y, Song Y, Ermon S 2020 Diversity can be transferred: Output Diversification for White- and Black-box Attacks Neural Information Processing Systems

[12] Choi Y, Choi M, Kim M, et al. 2019 Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. the IEEE Conference on Computer Vision and Pattern Recognition 8789-8797

[13] Ming Y L, Oncel T 2016 Coupled generative adversarial networks Neural Information Processing Systems 469–477.

[14] Amélie R, Bousmalis K, Gouws S, et al. 2020 XGAN: Unsupervised Image-to-Image Translation for many-to-many Mappings Domain Adaptation for Visual Understanding

[15] Szegedy C, Zaremba W, Sutskever I, et al. 2014 Intriguing properties of neural networks International Conference on Learning Representations

[16] Kurakin A, Goodfellow I, Bengio S 2017 Adversarial examples in the physical world International Conference on Learning Representations

[17] Baluja S, Fischer I 2017 Adversarial transformation networks: Learning to generate adversarial examples arXiv preprint arXiv:1703.09387

[18] Xiao C W, Li B, Zhu J Y, et al 2018 Generating Adversarial Examples with Adversarial Networks[C]. Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI) 3905-3911

[19] Dario P, Marco M and Massimo B 2019 Adversarial out-domain examples for generative models IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)

[20] Shan S, Wenger E, Zhang J, et al. 2020 Fawkes: protecting privacy against unauthorized deep learning models USENIX Security Symposium
[21] Yeh C Y, Chen H W, Tsai S L, et al. 2020 Disrupting image-translation-based deepfake algorithms with adversarial attacks. *IEEE Winter Applications of Computer Vision Workshops (WACVW)*

[22] Ruiz N, Bargal S A, Sclaroff S. 2020 Disrupting deepfakes: adversarial attacks against conditional image translation networks and facial manipulation systems. *arXiv preprint arXiv: 2004.01279*