DeepFake Disrupter: The Detector of DeepFake Is My Friend

Xueyu Wang*,[1] Jiajun Huang*,[1] Siqi Ma[2] Surya Nepal[3] Chang Xu[1]
[1]School of Computer Science, Faculty of Engineering, The University of Sydney
[2]The University of New South Wales, Canberra [3]CSIRO, Data61
{xwan6266, jhua7177}@uni.sydney.edu.au,siqi.ma@adfa.edu.au,
surya.nepal@data61.csiro.au,c.xu@sydney.edu.au

Abstract

In recent years, with the advances of generative models, many powerful face manipulation systems have been developed based on Deep Neural Networks (DNNs), called DeepFakes. If DeepFakes are not controlled timely and properly, they would become a real threat to both celebrities and ordinary people. Precautions such as adding perturbations to the source inputs will make DeepFake results look distorted from the perspective of human eyes. However, previous method doesn't explore whether the disrupted images can still spoof DeepFake detectors. This is critical for many applications where DeepFake detectors are used to discriminate between DeepFake data and real data due to the huge cost of examining a large amount of data manually. We argue that the detectors do not share a similar perspective as human eyes, which might still be spoofed by the disrupted data. Besides, the existing disruption methods rely on iteration-based perturbation generation algorithms, which is time-consuming. In this paper, we propose a novel DeepFake disruption algorithm called "DeepFake Disrupter". By training a perturbation generator, we can add the human-imperceptible perturbations to source images that need to be protected without any backpropagation update. The DeepFake results of these protected source inputs would not only look unrealistic by the human eye but also can be distinguished by DeepFake detectors easily. For example, experimental results show that by adding our trained perturbations, fake images generated by StarGAN [5] can result in a 10 $\sim$ 20% increase in F1-score evaluated by various DeepFake detectors.

1. Introduction

Face Manipulation has raised significant concerns within our digital society. It is a kind of technique that allows people to modify the face's identity, expression, and attributes in a given image or video. With the development and implementation of Deep Neural Networks (DNN), the recent manipulation methods could produce verisimilar results that might fool human eyes. These DNN based methods, called DeepFakes [3, 4, 11, 24, 25, 28], have attracted much attention from the public and researchers because the high-quality DeepFake results could lead to social and security problems. For example, the public might think the victims did somethings that they never did due to the presence of their identities in the DeepFake videos, thus ruining their reputations. The forgery data could also fool the security protocol by verifying the payment authorization system through fake personal information so that putting the victims' wealth at great risk.

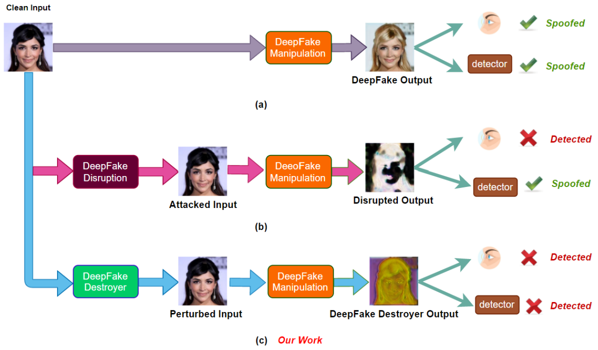

Figure 1. (a) shows that advanced DeepFake manipulation models can easily spoof human naked eye and DeepFake detector. (b) shows that after DeepFake Disruption, fake outputs become apparently distorted from the perspective of human eye, but can still spoof the DeepFake detector. (c) shows that our proposed method DeepFake disrupter can invalidate the DeepFake manipulation process from both human end and machine end.

As a response to the increasing concern of DeepFake, many defense methods are proposed. The first way is using detection models to distinguish Real and DeepFake data. Multiple detection algorithms are introduced, including using traditional DNN models for detection [7, 18, 21], analyzing the inconsistent within the DeepFake data [12, 30], and

*Equal Contribution

extracting the synthesis signal as the evidence for discrimination [27]. On the other hand, recent research provides a new direction for defense, preventing attackers from synthesizing DeepFake images. These methods, called DeepFake Disruption, attempt to add small perturbations to the original images such that the corresponding DeepFake results might be heavily distorted in visualization. Disrupting DeepFakes [22] is a related work on image translation disruption framework to make image manipulation models generates fake images with human-perceptible distortions.

Although the recent DeepFake Disruption methods could prevent the DeepFake models from generating realistic results, these kinds of methods still have some problems. In the real-world multi-media systems, it is extremely expensive to employ human observers to defend the DeepFake by manually examining every input image in the large volume of vision data, even though the defects in the image produced by DeepFake are obvious. Instead, to automate the defense of DeepFake, it is prevalent and preferable to develop DeepFake detectors. However, although the existing disruption methods could make the DeepFake's output become distorted from the human eye, our experiment demonstrates that these visually unnatural samples can still spoof the DeepFake detectors since the human eye and neural network share a different decision logic. What's more, these recent disruption methods rely on iteration-based adversarial attack algorithms, e.g. Iterative Fast Gradient Sign Method(I-FGSM) [10] and Projected Gradient Descent (PGD) [14], to find out the perturbation for each data, which is normally time-consuming.

We argue that we also need to consider the loss of the DeepFake detector, such that the generated DeepFake results of protected data are not only being recognized by the human eye but also can be detected by the detectors, and at the same time, the original data injected with perturbations can still be recognized as the real one. We should also use a perturbation generator to generate perturbation, which provides an end-to-end protection algorithm that can save time. Figure 1 shows the development of DeepFake disruption methods.

In this work, we propose a novel framework, called DeepFake Disrupter, to defend against DeepFake with the help of the DeepFake detector. The DeepFake Disrupter is a perturbation generator that takes as input real images and outputs a human-imperceptible perturbation so as to make the data generated by the DeepFake models be identified as fake by DeepFake detector and human eyes; meanwhile, the original real inputs injected with perturbations can still be identified as real by DeepFake detector. We show that just making DeepFake outputs distorted from the human eye's view is insufficient because the DeepFake detector may still be fooled by classifying the fake videos as real. Experimental results on CelebA [13] and VoxCeleb1 [16] datasets demonstrate that the proposed DeepFake disrupter can effectively protect original real images/videos from being used as a source for making DeepFake data.

2. Related Work

Adversarial attack and Adversarial Training After realizing the vulnerability of normal neural networks, a plethora of works have been done in the area of adversarial attacks. The first work is Fast Gradient Sign Method(FGSM) proposed by [6], in which they suggest a one-step gradient ascent method to generate the adversarial examples. Based on FGSM, [14] propose Projected Gradient Descent(PGD) attacks. Instead only updating once, PGD just iteratively update the adversarial examples to make it stronger than FGSM. While the above methods collectively use the iterative gradient update method for perturbation generation, there are also some works producing perturbations via generative models [19]. However, all the above works focus on adversarial attack and training on normal classification tasks and researches about adversarial attacks on generative models attract less attention. Works like [22] and [23] use the idea of adversarial attack to disrupt the ability of DeepFakes generators, which are more close to our proposed method.

DeepFake Data Generation The DeepFake techniques can be separated into two major categories, identity manipulation and attribute manipulation. For identity manipulation, RSGAN [17] extracts the embedding information of the face and hair for generating results. FSGAN [9] implement multi-scale architecture to handle different pixel situations while using an occlusion-wise algorithm to preserve the occlusion region of the target face. Facial expression and attribute manipulation form another category of DeepFake. Rather than changing the identity information, these techniques try to change some facial attributes (such as hair and skin color), or expressions (such as smiling or blinking). StarGAN [5] is a famous work for attribute manipulation, which encodes the facial attributes into latent space. Furthermore, the researcher extent the image expression manipulation into video level called facial animation. Given a driving video and a source face image, the animation methods would generate a new video that the source image is performing the same expression and action as the face in the driving video [8, 24, 26].

DeepFake Detection and Defense As a response to the increasing quality of DeepFake, many methods are proposed to detect DeepFake attacks. A common way is to develop DNN detection modules. Rossler [21] and Selim* implement DNN to achieve promising detection accuracy. Others try to detect the inconsistency within the generated data, such as detecting the identity swapping boundary, the in-

*https://github.com/selimsef/dfdc_deepfake_challenge consistent angles between the face and head, and the difference between face and background [15]. In addition, some researchers argue it contains synthesis signals [7, 27, 29] for the GAN-based DeepFake generation method. Utilizing these signals could detect the forgery data easily.

3. Methodology

In this section, we will first provide the preliminaries on Adversarial Attacks to DeepFake and then describe our proposed pipeline and optimization algorithm.

3.1. Adversarial Attacks to DeepFake

As a new way for DeepFake defence, disrupting DeepFake models is to add human-imperceptible perturbations on the source images [23]. The disruption on the output fake images can be taken as an adversarial attack to the DeepFake model and will make the DeepFake models less effective in generating realistic images. That is, the output by DeepFake will be highly unrealistic from the perspective of human naked eyes. Formally, we denote x as the source image, and $\widehat{x}$ represents the adversarial input, i.e., $\widehat{x} = x + \eta$, where η is the human-imperceptible perturbation with a common norm constraint $\|\eta\|_2 \leq \epsilon$. Suppose there is a DeepFake generator $G(\cdot)$. By taking the source image x, and the perturbed image $\widehat{x}$ as the input, the DeepFake generator will produce $G(x)$ and $G(\widehat{x})$, respectively. A successful DeepFake disruption η on x will make the human observers easily notice that the generated $G(\widehat{x})$ is an image after manipulations. By considering r as an attacking target, the objective function can be written as

$$\max_{\eta} L_D(G(x+\eta), r), \quad \text{s.t.} \quad ||\eta||_2 \leq \epsilon, \tag{1}$$

where L_D is a distance function normally using the L^0, L^2 or L^∞ norms. If r is set to be the original DeepFake output, which is $r = G(x)$, we will get the ideal disruption that can maximize the distortion of the output. Eq. (1) can be further generalized to consider the conditional image generation, i.e., $G(x, c)$, where c denotes the target class. Moreover, we can also choose r_{target} to be a specific predefined image as the attacking target.

The optimal perturbation η for the source image x in Eq. (1) can be effectively optimized with those methods developed for generating adversarial examples, e.g., Iterative Fast Gradient Sign Method (IFGSM) [6] or Projected Gradient Descent (PGD) [14]. Though the optimal η can be effectively solved through the iterations of IFGSM or PGD, it could be time-consuming to deal with the large-scale image dataset. For each source image x, we have to run a separate optimization procedure to discover its corresponding optimal perturbation η. Most importantly, source images in the same dataset often have shared low-level or high-level patterns. The separate optimization of the perturbation for these images thus cannot well exploit the useful structure information in the dataset.

In the real-world multi-media systems, it is extremely expensive to employ human observers to defend the DeepFake by manually examining every input image in the large volume of vision data, even though the defects in the image produced by DeepFake are obvious. Instead, to automate the defense of DeepFake, it is prevalent to develop DeepFake detectors. Both DeepFake detectors and disrupting DeepFake are to defend DeepFake but from two different perspectives. The remaining question is whether the human perceptible images produced by disrupting DeepFake will indeed benefit the future DeepFake detection, rather than doing a disservice.

3.2. DeepFake Disrupter

The model pipeline of our proposed method consists of Perturbation Generator, DeepFake Generator, and DeepFake Detector. Next, we will introduce them one by one followed by an overall optimization framework.

Instead of independently treating the perturbations on the source images, we tend to learn a disrupter to generate the perturbation $P(x)$ for the image x. Hence, given a DeepFake generator G, the generated images for the source image x and perturbed image $x + P(x)$ can be written as $G(x)$ and $G(x + P(x))$, respectively. The objective function of disrupting DeepFake can thus be rewritten as

$$\max_{P} \quad \mathbb{E}_x[L_D(G(x+P(x)), r)], \quad \text{s.t.} \quad ||P(x)||_2 \leq \epsilon \quad \forall x, \tag{2}$$

where we have calculated distance loss function over all images in the training set. As the inequality constraint in Eq. (2) cannot be conveniently handled in the end-to-end training of the networks, we further have a soft constrained version of the objective function,

$$\max_{P} \quad \mathbb{E}_x\big[L_D(G(x+P(x)), r)\big] - C_1\mathbb{E}_x\big[(\|P(x)\|_2 - \epsilon)_+\big], \tag{3}$$

where $(\cdot)_+$ denotes the hinge loss, ϵ is a small constant to constrain $\|P(x)\|_2$, and C_1 is a hyper parameter to balance the two items. We will have a non-zero loss for the second item when $\|P(x)\|_2 > \epsilon$, which encourages that the generated perturbation will not be too severe on the source image.

Along with the development of advanced DeepFake manipulation methods, there also emerge a few effective DeepFake detection techniques. These DeepFake detection techniques are essential to screen the DeepFake data for the multimedia systems, while the aforementioned disrupting DeepFake is an preventive measure to protect the data from the DeepFake. But a safe DeepFake disrupter should take the downstream DeepFake detector into consideration. As the proposed work is not focusing on developing a new

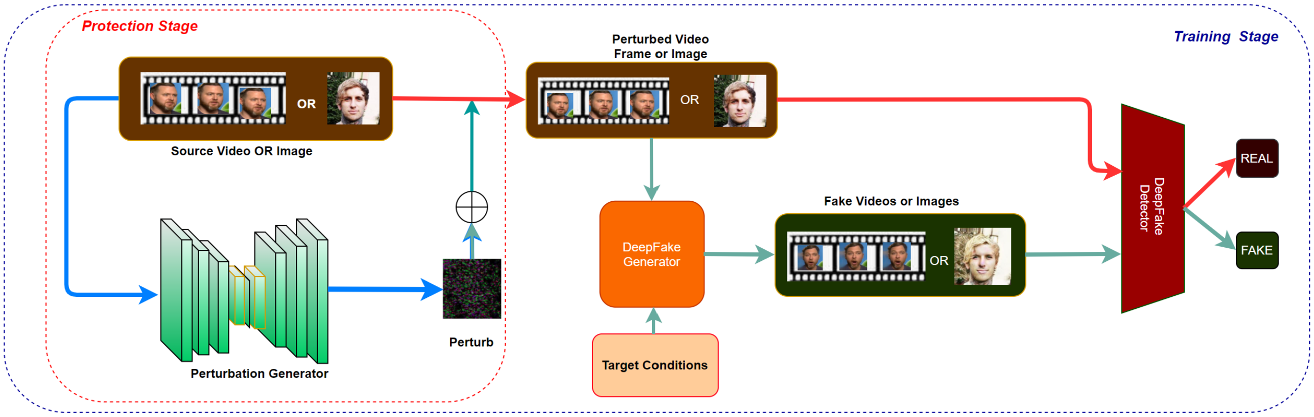

Figure 2. Overview of DeepFake disrupter. A source video or image will firstly feed into the perturbation generator to produce a human-imperceptible perturbation. The perturbation will be added back to the source inputs. After that, we pass the adversarial inputs into the DeepFake Generator together with a target condition to get a fake image or fake video. Lastly, the fake outputs and the adversarial inputs will be fed into the downstream DeepFake discriminator model.

DeepFake detector, we directly adopt a well-trained DeepFake detector D as an auxiliary for the DeepFake disrupter. The detector actually considers a binary classification task,

$$D(x) = \begin{cases} 1, & if\ x\ is\ real \\ 0, & if\ x\ is\ fake \end{cases}, \tag{4}$$

where D denotes the DeepFake detector model, x is the input data to be examined by the detector. If the values of model output logits is over 0.5, we classify the input to be real; otherwise, we classify the input as fake.

The DeepFake detector D can well recognize that the clean source image x is real. A DeepFake generator G might generate the image $G(x)$ that deceives the DeepFake detector D. But with the DeepFake disrupter P, we expect the generated fake image $G(x+P(x))$ could be well identified by D. That is, we aim to minimize the predicted logits for the fake data $G(x+P(x))$,

$$\mathcal{L}_{fake} = \mathbb{E}_x[D(G(x+P(x)))], \tag{5}$$

where $\mathcal{L}_{fake}$ is thus denoted as the DeepFake detection loss for the fake data.

It is instructive to note that the aim of disrupting DeepFake is to protect the data from DeepFake, rather than hurting the quality of the data. We have already constrained that the perturbation $P(x)$ will not be too large by penalizing its norm. Here from the lens of the DeepFake detection, we introduce another quality measure of the perturbed data. If the perturbed data $x+P(x)$ can still be recognized as real by the DeepFake detector D, we think its quality can be guaranteed to some extend. Formally, we obtain the detection loss for the perturbed data $x+P(x)$,

$$\mathcal{L}_{real} = \mathbb{E}_x[1 - D(x+P(x))]. \tag{6}$$

By incorporating Eq. (3) and the above loss functions, we thus achieve the resulting objective function,

$$\begin{aligned}\mathcal{L} = & -\mathbb{E}_x\big[L_D(G(x+P(x)), r)\big] + C_1\mathbb{E}_x\big[(\|P(x)\|_2 - \epsilon)_+\big] \\ & + C_2\mathbb{E}_x[D(G(x+P(x)))] + C_3\mathbb{E}_x[1 - D(x+P(x))],\end{aligned} \tag{7}$$

where C_1, C_2 and C_3 are hyper parameters to balance different loss items. The DeepFake Generator and DeepFake Detector will be pretrianed or selected from pretrained SOTA models and during training their parameters will be freezed, which means only the parameter of our proposed Perturbation Generator will be updated. The setting of of the hyper parameters and other training details like the structure of perturbation generators will be discussed in Appendix. By optimizing Eq. (7), an effective disrupter can be learned from the training dataset. At the inference stage, we can feed new images into the disrupter P and conduct efficient feed-forward processes to generate their corresponding optimal perturbation. The generated perturbation can increase the difficulty of the DeepFake methods in generating effective fake images to deceive humans' eyes and the downstream DeepFake detectors.

Generalization over D. In the real world, the DeepFake Disrupter P and D are in the same camp to defend the attack by DeepFake Generator G. Both P and D are thus probably trained and maintained by the same developer. That is to say, when we are training disrupter P, we may already know and get access to the detector D that is to be used online. This thus naturally addresses the concern on the generalization over D. Nevertheless, we also evaluate the generalization over different Ds' in our experiments, e.g., training with an Xception-based D while testing on a Resnet-based D. Although the overall detection accuracy drop, an image with the optimized perturbation tend to be more easily identified than the clean one without protection.

Generalization over G. The generalization over Deep-

Fake Generator G manifests in two aspects. First, well-trained perturbation P generalizes well across different DeepFake Generators G evaluated by the percentage of successfully disrupted images. e.g. trained using GANimation and tested using StarGAN. This is because most DeepFake Generators are adversarially vulnerable to perturbations. A tiny change to the input could easily distort the output. Second, many DeepFake Generators are controlled by different conditions to generate specific fake images. The well-trained perturbation shows a good generalization property across different conditions. e.g. trained under blackhair condition and tested under brownhair condition in StarGAN. The detection accuracy for these cross-condition evaluations is higher than the clean image without protection.

4. Experiment

This section illustrates the experimental results to demonstrate the effectiveness of DeepFake Disrupter using our proposed framework. We will firstly describe the datasets we used in our experiments. Then we will briefly discuss the baseline and evaluation metrics we used. Lastly, we will discuss our experimental results in detail.

4.1. Datasets

We mainly use two datasets in our experiments: CelebA [13] and VoxCeleb1 [16]. The Large-scale CelebFaces Attributes Dataset (CelebA) has more than 200k celebrity images. In addition, this dataset covers large pose variations and background clutter with 10,177 identities, 202,599 face images, and 5 landmark locations, 40 binary attributes annotations per image. The VoxCeleb1 dataset contains more than 100,000 videos extracted from Youtube, which are utterances of more than 1,000 celebrities. For the preprocessing of these videos, we follow the guideline and implementation details of [24] to crop the video frames according to annotated bounding boxes because the cropping process can provide better alignment for DeepFake generator to produce good quality fake videos. After that, we recorded five different motion videos as the driving videos that serve as an input of the DeepFake Generator, namely mouth, blink, yaw, nod, smile. Then, we follow the official implementation of [24], for each video, we extract the best frame and together with a random driving video as the input pairs for the selected DeepFake Generator.

4.2. Baseline and Evaluation Metrics

We use Disrupting DeepFakes proposed by [22] as our baseline. This work disrupts the DeepFake generation by making the DeepFake Generator produce distorted images using the loss function same with Eq. 1, which is trying to make the DeepFake results visually unnatural. It is different from our method which is disrupting DeepFake generation by significantly reducing its passing rate on DeepFake detectors. We will empirically demonstrate that just making the DeepFake generator produce fake images with human-perceptible distortions doesn't necessarily guarantee that the fake videos will be recognized by DeepFake Detectors. To be specific, we will use ***Precision***, ***Recall*** and ***F1*** score to quantify the disrupting performance for both the baseline method and our proposed method. For evaluation on the successful disruptions by human eye, we follow [22] to set per-pixel errors $L_2 \geq 0.05$ as our criteria. If the fake outputs of our method and the original fake outputs have $L_2 \geq 0.05$, we consider it a successful disruption visually.

4.3. Results

Disruption Performance for specific target domains In this section, we use StarGAN to demonstrate that our proposed method can work on any specific target domains i.e. Black Hair, Blond Hair, Brown Hair, Male, Young. Here, target domain refers to c in conditional image translation model $G(x, c)$. The comparison comes from three parts. The first part is original fake images produced under [5], the second part is disrupted fake images generated under [22], the third part is fake images generated under our proposed framework. Specifically, we choose 100 real images to generate 100 fake images according to the aforementioned three-generation processes to conduct the comparison. That is to say, we calculate the precision, recall, and F1 score using the 100 real images plus 100 fake images for comparison, in which the higher the precision, recall, and F1 score, the better the algorithm. Because all detectors are well pretrained and thus have high accuracy on predicting real images, the case of false negative is rare, resulting in high recall in all cases. Therefore, in the following experiments, we mainly focus on analyzing the performance change on precision and F1-score. Table 1 shows the detailed comparisons under different target settings. Notice that we choose the same set of real images for a fair comparison, therefore, the recall score will only change based on different detection models. In most cases, baseline method [22] can only archive $2-10\%$ performance gain with regarding to precision and F1 score, while our method can boost the performance to more than 90%.

Disruption Performance for class transferable attacks The DeepFake architectures used in our framework all have conditional targets as their inputs. StarGAN has facial attribute encodings; GANimation has action units for different expressions and the First-Order Motion model has different kinds of driving videos. It is beneficial to train a perturbation generator that can work under arbitrary target conditions. The training strategy is simple, we choose conditional targets randomly during each training iteration, i.e., at each iteration, we choose random action unit index from the range between 0 to 80 when training our Perturbation

Attributes	Type	Xception			Resnet18			Resnet50		
		precision	recall	F1-score	precision	recall	F1-score	precision	recall	F1-score
Blackhair	StarGAN-fake [5]	0.51	0.99	0.68	0.39	0.60	0.47	0.43	0.75	0.55
	disrupted-fake [22]	0.53	0.99	0.69	0.50	0.60	0.55	0.46	0.75	0.57
	DeepFake disrupter(ours)	**0.90**	0.99	**0.94**	**1.00**	0.60	**0.75**	**0.94**	0.75	**0.83**
Blondhair	StarGAN-fake [5]	0.51	0.99	0.67	0.38	0.60	0.46	0.43	0.75	0.55
	disrupted-fake [22]	0.55	0.99	0.71	0.52	0.60	0.56	0.56	0.75	0.64
	DeepFake disrupter(ours)	**0.98**	0.99	**0.99**	**1.00**	0.60	**0.75**	**0.99**	0.75	**0.85**
Brownhair	StarGAN-fake [5]	0.51	0.99	0.68	0.38	0.60	0.47	0.43	0.75	0.55
	disrupted-fake [22]	0.56	0.99	0.71	0.49	0.60	0.54	0.46	0.75	0.57
	DeepFake disrupter(ours)	**0.96**	0.99	**0.98**	0.75	0.60	0.67	0.54	0.75	0.63
male	StarGAN-fake [5]	0.51	0.99	0.68	0.39	0.60	0.47	0.43	0.75	0.55
	disrupted-fake [22]	0.54	0.99	0.70	0.42	0.60	0.49	0.51	0.75	0.61
	DeepFake disrupter(ours)	**0.97**	0.99	**0.98**	**1.00**	0.60	**0.75**	**0.98**	0.75	**0.85**
young	StarGAN-fake [5]	0.51	0.99	0.67	0.38	0.60	0.47	0.43	0.75	0.54
	disrupted-fake [22]	0.53	0.99	0.69	0.48	0.60	0.53	0.51	0.75	0.61
	DeepFake disrupter(ours)	**0.90**	0.99	**0.95**	**1.00**	0.60	**0.75**	**0.83**	0.75	**0.79**

Table 1. Disruption Performance for StarGAN with 5 different target conditions. Higher figure implies better performance

DeepFake Detector	Xception			Resnet18			Resnet50		
	precision	recall	F1-score	precision	recall	F1-score	precision	recall	F1-score
StarGAN [5]	0.58	0.99	0.72	0.53	0.60	0.56	0.56	0.75	0.64
Disrupring StarGAN [22]	0.64	0.99	0.78	0.59	0.60	0.60	0.59	0.75	0.66
DeepFake disrupter (ours)	**0.86**	0.99	**0.92**	**0.87**	0.60	**0.71**	**0.72**	0.75	**0.74**
GANimation [20]	0.60	0.97	0.74	0.47	0.65	0.55	0.53	0.78	0.63
Disrupting GANimation [22]	0.70	0.97	0.82	0.56	0.65	0.60	0.67	0.78	0.72
DeepFake disrupter (ours)	**0.82**	0.97	**0.89**	**0.89**	0.65	**0.75**	**0.98**	0.78	**0.87**
First-Order-Motion [24]	0.56	0.72	0.63	0.50	0.68	0.58	–	–	–
DeepFake disrupter (ours)	**0.91**	0.72	**0.80**	**0.89**	0.68	**0.78**	–	–	–

Table 2. Disruption Performance under different DeepFake Manipulation Models and DeepFake Detection Models

generator under the GANimation setting. The trained perturbation generator would be enabled to produce perturbations that can satisfy our problem constraints. We tested the performance of the aforementioned three DeepFake manipulation algorithms using the pretrained DeepFake detection models via precision, recall and, F1 score. For First Order Motion Model [24], we didn't use Resnet50 for detection testing due to VRAM budget limitation. As there is no previous work for disrupting the outcomes of the First-Order-Motion model, we directly use its original fake videos as our baseline. Table 2 shows the detailed comparisons, and from the table, we can see that our proposed framework not only works on image transformation algorithms like StarGAN [5] and GANimation [20] but also works on face image animation algorithms First-Order-Motion Model [24], and they all show superior performance compared with simply maximizing output norm distances in baseline method.

Disruption Performance with SOTA DeepFake Detection Algorithms We use basic backbones like Xception and Resnet in the above experiments simply because we aim to prove the effectiveness of our trained perturbation generator. However, we also tested our proposed pipeline against state-of-the-art DeepFake detection algorithms [1, 2, 32].

DeepFake Detector	Multi-Attentional [32]		
	precision	recall	F1-score
StarGAN [5]	0.53	0.98	0.69
Disrupring StarGAN [22]	0.67	0.98	0.77
DeepFake disrupter (ours)	**0.88**	0.98	**0.93**
GANimation [20]	0.62	0.96	0.75
Disrupting GANimation [22]	0.73	0.96	0.84
DeepFake disrupter (ours)	**0.85**	0.96	**0.91**

Table 3. Disruption Performance Against Multi-Attentional Deepfake Detection models

DeepFake Detector	F3-Net [1]	RFM [2]
StarGAN[3]	0.74	0.69
Disrupring StarGAN[20]	0.76	0.73
DeepFake disrupter (ours)	**0.92**	**0.90**
GANimation[18]	0.73	0.70
Disrupting GANimation[20]	0.76	0.78
DeepFake disrupter (ours)	**0.94**	**0.96**

Table 4. Disruption Performance Against F3-Net RFM DeepFake Detection models; to save space, we only report the critical metric F1-score in this table

From Table 3 and 4 we can see our proposed pipeline can achieve similar performance gain in terms of precision and

F1-score compared with those using Xception and Resnet, which further proves the efficacy of our proposed pipeline.

Detection Outcomes for Real Inputs, Perturbed Inputs and Fake Inputs We compare the detection outcomes of real inputs, perturbed real inputs, and fake inputs using our proposed method because one goal of our proposed framework is ensuring the perturbed real inputs to be detected as real by the DeepFake detectors. To be specific, we add perturbations generated by the proposed method to 100 testing real images to get 100 perturbed real images for testing under pretrained Xception and Resnet18 detector. Table 5 shows the success rate, i.e., the proportion of images that can be detected as real by the detectors. For real inputs x and perturbed real inputs $x + P(x)$, the higher the success rate the better, for fake inputs $G[x + P(x)]$ the lower the success rate the better. From the table, we can see that the real inputs and perturbed input can all maintain a high success rate while the fake inputs successfully achieve a lower success rate.

Face Manipulation	Inputs Type	Xception	Resnet18
StarGAN	x	0.99	0.98
	$x + P(x)$	0.98	0.96
	$G[x + P(x)]$	0.10	0.07
GANimation	x	0.99	0.97
	$x + P(x)$	0.98	0.96
	$G[x + P(x)]$	0.03	0.09
First-Order-Motion	x	0.99	0.97
	$x + P(x)$	0.98	0.96
	$G[x + P(x)]$	0.13	0.16

Table 5. Detection outcomes for comparison among real inputs, perturbed inputs and fake inputs.

Generalization This section explores the generalization ability of the trained perturbation generator from two aspects: detector and manipulation generator. We test the disruption performance by using a detector that is different from that in the training. F1 scores are reported in Table 6, where the generator is GANimationm and detectors in columns are used in training while those in rows are in testing. The manipulations $G(x)$ over clean images can be only detected by Resnet18 with a 0.55 F1. By incorporating the downstream Resnet18 detector into the proposed algorithm, the F1 by Resnet18 can achieve 0.75, and other unknown detectors like Xception and Resnet50 can also enjoy higher F1 scores than 0.55. In addition, the model trained with Xception and evaluated with Resnet18 leads to 0.77, which is still higher than the detection performance (0.75) over deepfake data from [22], which demonstrates the generalization with respect to the detector.

We proceed to evaluate the generalization w.r.t. different manipulations. StarGAN is used in training, while GANimation is used in tests, and vice versa. The distortion of DeepFake outcomes is reported in Table 7. L_2 norm is calculated between manipulation of the clean image and that of the clean image with our perturbations, i.e., G(x) and G(x+P(x)). Following [22], $\%dis$ shows percentage of successful disruptions of 500 fake images produced by G(x+P(x)), ie. when L2 is over 0.05, the image is successfully disrupted. Conducting both training and test with StarGAN leads to a 100% success. Though the test on a different GANimation has a performance drop, the success rate of 74% is still high enough. When training with GANimation, we find that StarGAN is more vulnerable, as its success rate achieves 100%. We also report the F1-score evaluation in this table on Xception. Though the test on a different GANimation has a performance drop, F1-score is still high enough. E.g., trained with GANimation, the model tested with StarGAN has an F1-score 0.86, which is still higher than the F1-score 0.82 achieved by [22] with GANimation.

Detectors	Xception	Resnet18	Resnet50	G(x)
Xception	0.89	0.77	0.68	0.74
Resnet18	0.61	0.75	0.64	0.55
Resnet50	0.74	0.79	0.87	0.63

Table 6. Generalization w.r.t. different detectors

Face Manipulation	StarGAN			GANimation		
	L2	%dis	F1	L2	%dis	F1
StarGAN	0.326	100%	0.92	0.183	74%	0.79
GANimation	0.987	100%	0.86	0.073	82%	0.89

Table 7. L2 Norm and Percentage of Disruption trained and tested with different generators.

StarGAN depends on the manipulation conditions. The condition set in the training could be different from that in the testing. We evaluate the generalization results of different conditions in Table 8. The results are F1-scores tested by Xception Detector. Baseline is detection results of G(x) over clean images. By considering the condition Blackhair, the detector can achieve a 0.68 F1 over the fake results of clean images, while the proposed algorithm can protect the data and the detector's performance is boosted to the 0.94 F1. Though changing a different condition (Blondhair or Brownhair) in the test leads to a performance drop, the resulting detection performance is still higher than that without protection (i.e., 0.68). This thus suggests the generalization of the manipulation generator.

Conditions	Blackhair	Blondhair	Brownhair	G(x)
Blackhair	0.94	0.79	0.81	0.68
Blondhair	0.73	0.99	0.70	0.67
Brownhair	0.69	0.74	0.98	0.68

Table 8. Generalization to different conditions in StarGAN reported by F1-score

DeepFake Visualization In this section, we will show that the DeepFake outputs have a high probability of being dis-

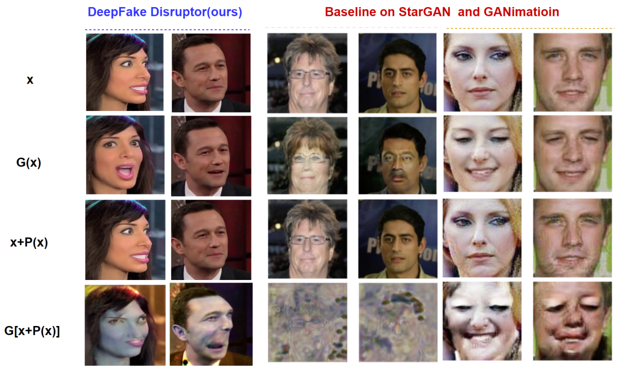

Figure 3. Comparison between the result of our framework and baseline methods.

torted from the perspective of the human eye. Figure 3 shows the visualization comparison of our method with the baseline method on StarGAN and GANimation. We can see that the patterns of disrupted outcome vary across different attacking methods and DeepFake manipulation models, but they all show noticeable distortions compared with the fake images without disruption. Figure 4 shows visualizations on video frames under [24]. We can see that the disrupted fake video frames in the last row all have significant noises or distortions. Apart from qualitatively visualizing the DeepFake outcomes, we also evaluate the proportions of disrupted outcomes quantitatively. We follow [31] to set $L_2 \geq 0.05$ as the criteria for successful disruption as there are noticeable distortions when $L_2 \geq 0.05$. Table 9 shows the per-pixel L_2 for generated perturbations η and the difference between disrupted fake images and original fake images as well as the percentage of successful disruptions under different attack methods. Although our reported L_2 results are lower than those of I-FGSM and PGD, they are still higher than 0.05. From visualizations in Figures 3 and 4, our outputs do have noticeable distortions.

Algorithm	Type	StarGAN	GANimation	First-Order-Motion
I-FGSM	L_2	1.324	0.120	0.240
	$\%dis$	100%	89%	92%
PGD	L_2	1.532	0.064	0.280
	$\%dis$	100%	79%	100%
disrupter (Ours)	L_2	0.326	0.073	0.320
	$\%dis$	100%	82%	91%

Table 9. Comparison of L_2 pixel-wise errors and the percentage of disrupted images(%dis.) for baseline disruption methods and our method.

Inference Efficiency We also compare the inference efficiency between the baseline method [22] and our proposed Disrupter. Specifically, we choose 100 testing images to run the inference using the PGD method and Our Disrupter and calculate the average time of generating perturbation for a single input image measured by seconds. Table 10 shows that our method runs 8-10 times faster compared with the traditional iteration-based attack method PGD.

Attack Method	PGD	**Disrupter (ours)**
StarGAN	0.551	**0.062**
GANimation	0.628	**0.057**

Table 10. Inference Efficiency measured by seconds per image

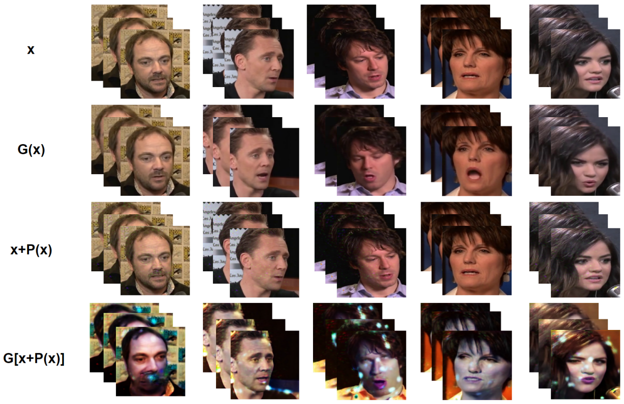

Figure 4. Examples of visualizations on First Order Motion Model with and without perturbations.

5. Conclusion

We propose an effective pipeline called DeepFake Disrupter based on generative networks to train perturbation generators that can help protect the source images or videos from being manipulated by various DeepFake manipulation algorithms. By adding adversarial perturbations, the DeepFake models would output fake images or videos to be successfully detected as fake by the DeepFake detector. Meanwhile, the adversarial images would still be detected as real by the DeepFake detector. The objective is achieved by adversarial loss and hinge loss, the former one controls the detection accuracy, while the latter controls the magnitude of the perturbations. Experiments show that (a) the baseline method [22] can only ensure the effectiveness of disruption by the naked eye, but failed to guarantee that the disrupted outputs can be effectively detected as fake by DeepFake detectors. (b). The proposed method can significantly improve the detection outcomes measured by precision, recall, and F1 score compared with the baseline method, and this performance is class transferable when training with random class attributes. (c) The proposed method also provides an extra benefit, which is maintaining the unnatural-looking property of the fake outcomes. That is to say, our proposed pipeline can destroy the ability of DeepFake manipulation models both visually by the human eye and logically by DeepFake detectors.

Acknowledgements

This work was supported in part by the Australian Research Council under Project DP210101859 and the University of Sydney SOAR Prize.

References

[1] Thinking in frequency: Face forgery detection by mining frequency-aware clues. In *ECCV*, 2020. 6

[2] Representative forgery mining for fake face detection. In *CVPR*, 2021. 6

[3] Aayush Bansal, Shugao Ma, Deva Ramanan, and Yaser Sheikh. Recycle-gan: Unsupervised video retargeting. In *Proceedings of the European conference on computer vision (ECCV)*, pages 119–135, 2018. 1

[4] Egor Burkov, Igor Pasechnik, Artur Grigorev, and Victor Lempitsky. Neural head reenactment with latent pose descriptors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13786–13795, 2020. 1

[5] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8789–8797, 2018. 1, 2, 5, 6

[6] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014. 2, 3

[7] David Güera and Edward J Delp. Deepfake video detection using recurrent neural networks. In *2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 1–6. IEEE, 2018. 1, 3

[8] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017. 2

[9] Iryna Korshunova, Wenzhe Shi, Joni Dambre, and Lucas Theis. Fast face-swap using convolutional neural networks. In *Proceedings of the IEEE international conference on computer vision*, pages 3677–3685, 2017. 2

[10] Alexey Kurakin, Ian Goodfellow, Samy Bengio, et al. Adversarial examples in the physical world, 2016. 2

[11] Lingzhi Li, Jianmin Bao, Hao Yang, Dong Chen, and Fang Wen. Advancing high fidelity identity swapping for forgery detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5074–5083, 2020. 1

[12] Lingzhi Li, Jianmin Bao, Ting Zhang, Hao Yang, Dong Chen, Fang Wen, and Baining Guo. Face x-ray for more general face forgery detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5001–5010, 2020. 1

[13] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015. 2, 5

[14] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017. 2, 3

[15] Yisroel Mirsky and Wenke Lee. The creation and detection of deepfakes: A survey. *ACM Computing Surveys (CSUR)*, 54(1):1–41, 2021. 3

[16] Arsha Nagrani, Joon Son Chung, and Andrew Zisserman. Voxceleb: a large-scale speaker identification dataset. *arXiv preprint arXiv:1706.08612*, 2017. 2, 5

[17] Ryota Natsume, Tatsuya Yatagawa, and Shigeo Morishima. Rsgan: face swapping and editing using face and hair representation in latent spaces. *arXiv preprint arXiv:1804.03447*, 2018. 2

[18] Huy H Nguyen, Junichi Yamagishi, and Isao Echizen. Capsule-forensics: Using capsule networks to detect forged images and videos. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2307–2311. IEEE, 2019. 1

[19] Omid Poursaeed, Isay Katsman, Bicheng Gao, and Serge Belongie. Generative adversarial perturbations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4422–4431, 2018. 2

[20] Albert Pumarola, Antonio Agudo, Aleix M Martinez, Alberto Sanfeliu, and Francesc Moreno-Noguer. Ganimation: Anatomically-aware facial animation from a single image. In *Proceedings of the European conference on computer vision (ECCV)*, pages 818–833, 2018. 6

[21] Andreas Rossler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Nießner. Faceforensics++: Learning to detect manipulated facial images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1–11, 2019. 1, 2

[22] Nataniel Ruiz, Sarah Adel Bargal, and Stan Sclaroff. Disrupting deepfakes: Adversarial attacks against conditional image translation networks and facial manipulation systems. In *European Conference on Computer Vision*, pages 236–251. Springer, 2020. 2, 5, 6, 7, 8

[23] Eran Segalis. Disrupting deepfakes with an adversarial attack that survives training. *arXiv preprint arXiv:2006.12247*, 2020. 2, 3

[24] Aliaksandr Siarohin, Stéphane Lathuilière, Sergey Tulyakov, Elisa Ricci, and Nicu Sebe. First order motion model for image animation. *arXiv preprint arXiv:2003.00196*, 2020. 1, 2, 5, 6, 8

[25] Supasorn Suwajanakorn, Steven M Seitz, and Ira Kemelmacher-Shlizerman. Synthesizing obama: learning lip sync from audio. *ACM Transactions on Graphics (ToG)*, 36(4):1–13, 2017. 1

[26] Justus Thies, Michael Zollhöfer, and Matthias Nießner. Deferred neural rendering: Image synthesis using neural textures. *ACM Transactions on Graphics (TOG)*, 38(4):1–12, 2019. 2

[27] Sheng-Yu Wang, Oliver Wang, Richard Zhang, Andrew Owens, and Alexei A Efros. Cnn-generated images are surprisingly easy to spot... for now. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8695–8704, 2020. 2, 3

[28] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Guilin Liu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. Video-to-video synthesis. *arXiv preprint arXiv:1808.06601*, 2018. 1

[29] Yaohui Wang and Antitza Dantcheva. A video is worth more than 1000 lies. comparing 3dcnn approaches for detecting deepfakes. In *FG'20, 15th IEEE International Conference on Automatic Face and Gesture Recognition, May 18-22, 2020, Buenos Aires, Argentina.*, 2020. 3

[30] Xin Yang, Yuezun Li, and Siwei Lyu. Exposing deep fakes using inconsistent head poses. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8261–8265. IEEE, 2019. 1

[31] Chin-Yuan Yeh, Hsi-Wen Chen, Shang-Lun Tsai, and Sheng-De Wang. Disrupting image-translation-based deepfake algorithms with adversarial attacks. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision Workshops*, pages 53–62, 2020. 8

[32] Hanqing Zhao, Wenbo Zhou, Dongdong Chen, Tianyi Wei, Weiming Zhang, and Nenghai Yu. Multi-attentional deepfake detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2185–2194, 2021. 6