2023 IEEE International Conference on Multimedia and Expo (ICME)

2023 IEEE International Conference on Multimedia and Expo (ICME) | 978-1-6654-6891-6/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICME55011.2023.00213

Face Poison: Obstructing DeepFakes by Disrupting Face Detection

Yuezun Li[1], Jiaran Zhou[1,*], Siwei Lyu[2]
[1] Department of Computer Science and Technology, Ocean University of China, Qingdao, China
[2] University at Buffalo, State University of New York, USA

***Abstract*—Recent years have seen fast development in synthesizing realistic human faces using AI-based forgery technique called DeepFake, which can be weaponized to cause negative personal and social impacts. In this work, we develop a defense method, namely *FacePosion*, to prevent individuals from becoming victims of DeepFake videos by sabotaging would-be training data. This is achieved by disrupting face detection, a prerequisite step to prepare victim faces for training DeepFake model. Once the training faces are wrongly extracted, the DeepFake model can not be well trained. Specifically, we propose a multi-scale feature-level adversarial attack to disrupt the intermediate features of face detectors using different scales. Extensive experiments are conducted on seven various DeepFake models using six face detection methods, empirically showing that disrupting face detectors using our method can effectively obstruct DeepFakes.**

***Index Terms*—DeepFake defense, video forensics, adversarial perturbation, face detection**

I. Introduction

Recent advances in deep learning and the availability of a vast volume of online personal images and videos have drastically improved the synthesis of highly realistic human faces [1], [2]. DeepFake is one of the most prevalent face forgery techniques that can swap the face with a synthesized face while retaining the same attributes such as facial expression and orientation, see Fig. 1. While there are interesting and creative applications of DeepFakes, they can also be weaponized to create illusions of a person's presence and activities that do not occur in reality, leading to serious political, social, financial, and legal consequences [3].

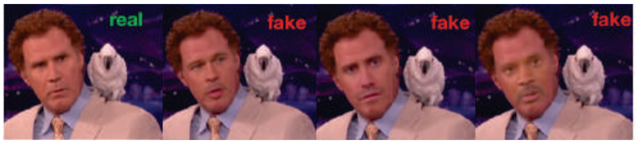

Fig. 1. Examples of DeepFake, which involves replacing the original faces with synthesized faces while keeping the same facial expressions. These examples are from [4].

Foreseeing this threat, many forensic methods aiming to detect DeepFake faces have been proposed recently [5]–[10]. However, given the speed and reach of the propagation of online media, even the currently best forensic method will largely operate in a postmortem fashion, applicable only after the fake face images or videos emerge. In this work, we aim to develop *proactive* approaches to protect individuals from becoming the victims of such attacks. Our solution is to add specially designed patterns known as adversarial perturbations that are imperceptible to human eyes but can result in face detection failures. The rationale of our method is the following. High-quality DeepFake models need a large number of, typically in the range of thousands, sometimes even millions, standard training faces collected using automatic face detection methods, *i.e.*, the face sets. Our method can "pollute" the face set to have low or no utility faces as training data for DeepFake models (see Fig. 2).

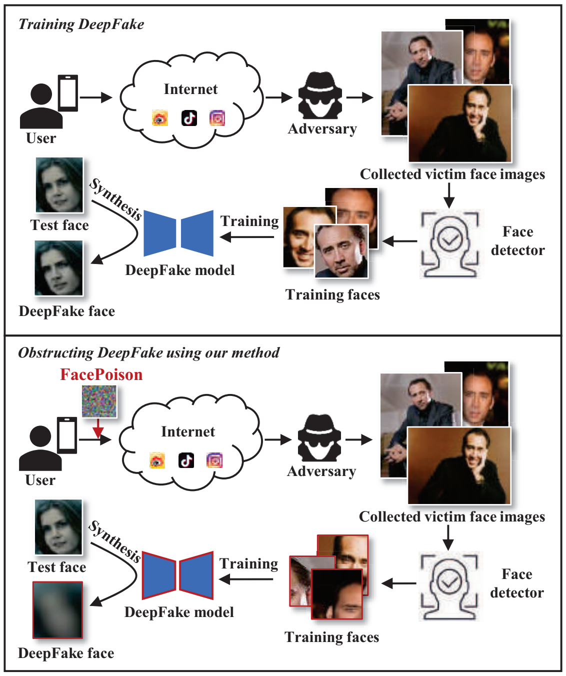

Fig. 2. Overview of FacePoison. Our method can disrupt face detection, resulting in inappropriate training faces, which subsequently obstructs DeepFake training.

Our study is focused on the adversarial attack to the deep neural network (DNN) based face detectors *e.g.*, [11], [12], as they achieve superior performance in comparison to the non-DNN-based methods and are more robust under variations in pose, expression and occlusion. Specifically, we propose a new multi-scale feature-level adversarial attack, which disrupts detection results by disturbing multiple intermediate features.

[*] Corresponding author

1945-788X/23/$31.00 ©2023 IEEE
DOI 10.1109/ICME55011.2023.00213

Attacking features instead of final detection results has the following advantages: 1) it has better transferability as it is not overfitted to a specific task [13], and 2) it can generate adversarial images with an arbitrary size without interpolation, thus it can be better used to attack other detectors [14]. Since the different levels of features represent different scales of information, our method jointly disrupts multiple levels of features, using a newly designed objective function. In particular, this objective contains two terms: error loss and key loss. The error loss encourages the error between attacked features and original features, and the key loss disturbs the key features that can largely impact the detection results. We perform extensive experimental evaluations on seven various DeepFake models using six face detection methods, demonstrating our method's effectiveness in disrupting the DNN-based face detectors to obstruct DeepFakes. *We would like to highlight that our contribution is not just the development of an adversarial attack to face detectors, but proposes a new paradigm to obstruct DeepFakes by utilizing the vulnerability of face detectors.*

The danger of DeepFake faces suggests that the mere fact that our faces can be automatically detected poses a threat to our privacy. Therefore, the proposed adversarial perturbation generation method can be implemented as a service of photo/video sharing platforms before a user's personal images/videos are uploaded or as a standalone tool that the user can use to process the images and videos before they are uploaded online. It is also noteworthy that the proposed method is not expected to substitute current forensics tools but is complementary to the current forensic tools.

II. Background and Related Works

Face Detection using DNNs have become mainstream with their high performance and robustness regarding variations in pose, expression, and occlusion. There has been a plethora of DNN-based face detectors *e.g.*, [11], [12], [15]–[18]. Regardless of the idiosyncrasies of different detectors, they all follow a similar workflow, which predicts the location and confidence score of the potential candidate regions corresponding to faces in an end-to-end fashion. The prohibitive cost of searching optimal network structures and architectures makes the choice of the backbone network limited to two well-tested DNN models, namely, the VGG network [19] or the ResNet [20], as reported on the leader board of WIDER challenge [21].

Adversarial Perturbations are intentionally designed noises that are imperceptible to human observers, yet can seriously reduce the deep neural network performance if added to the input image. Many methods [22]–[24] have been proposed to impair image classifiers by adding adversarial perturbations on the entire image. Recently, there have been several works on adversarial perturbation generation for general object detectors [25]–[28]. But there are not many studies on the vulnerability of face detectors. Bose *et al.* [29] used a GAN model to attack only one type of face detector, thus the feasibility is highly limited.

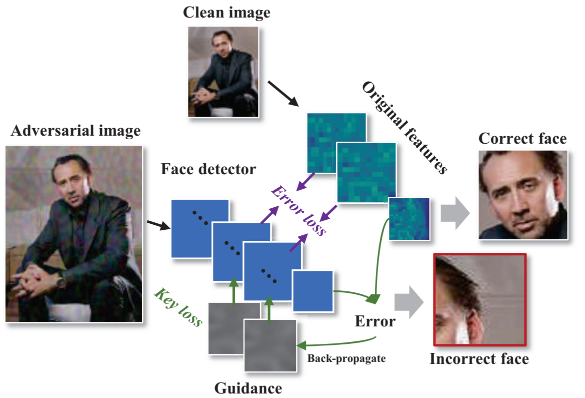

Fig. 3. Overview of our method on disrupting face detection. Our method attacks multiple intermediate features using error loss and key loss. The error loss enlarges the error between attacked features and original features, while key loss amplifies the disturbance on key elements indicated by the guidance map. See text for details.

III. Method

A. Overview

Our goal is to obstruct the training of DeepFakes by disrupting the function of face detectors. Denote $\mathcal{D}_{\mathcal{I}} = \{\mathcal{I}_i\}_{i=1}^{N}$ as the set of collected images containing faces for training. Denote $\mathcal{F}$ as the face detector and $x_i = \mathcal{F}(\mathcal{I}_i)$ denotes the detected face given image $\mathcal{I}_i$. We assume only one face exists in an image for simplicity. Denote $\mathcal{D}_x = \{x_i\}_{i=1}^{N}$ as the set of extracted faces using face detector $\mathcal{F}$. Note that $\mathcal{D}_x$ is used to train DeepFake model. By poisoning the face detection, $\mathcal{D}_x$ is contaminated with incorrect faces, thus impairing the generation of DeepFake model.

B. FacePoison: Disrupting Face Detection

Problem Setting. Denote θ as the parameters of the face detector $\mathcal{F}$. Let $\mathcal{I}$ be the clean image containing the victim's faces. Disrupting face detection is to find the adversarial image $\mathcal{I}^{adv}$ that can fool $\mathcal{F}$, while is visually similar to clean image $\mathcal{I}$. Let $\mathcal{L}$ be the adversary objective function to disrupt the face detector. This problem can be formulated as

$$\arg\min_{x^{adv}} \quad \mathcal{L}(\mathcal{I}^{adv}, \mathcal{I}; \theta), \quad \text{s.t. } ||\mathcal{I}^{adv} - \mathcal{I}||_{\infty} \leq \epsilon, \qquad (1)$$

where ϵ is the bound of distortion. Minimizing this equation can find an adversarial image to fool the face detector.

Multi-scale Feature-level Adversarial Attack. Our method targets attacking the intermediate features instead of the detection results. Intuition is that the features contain the discriminative information that determines the results. Thus the disruption of features can lead to a wrong face detection subsequently. In our method, we attack multiple feature layers of the backbone network. Denote $\mathcal{H} = \{h_i\}_{i=1}^{K}$ as the feature set obtained from K feature layers when given a clean image $\mathcal{I}$. Our goal is to mislead face detector $\mathcal{F}$ by disturbing $\mathcal{H}$. Specifically, we design a new objective function $\mathcal{L}$ containing two terms: error loss $\mathcal{L}_e$ and key loss $\mathcal{L}_k$ respectively.

Denote $\mathcal{H}' = \{h'_i\}_{i=1}^K$ as the corresponding set given the adversarial image $\mathcal{I}^{adv}$. The error loss $\mathcal{L}_e$ aims to enlarge the discrepancy between original and disturbed features, as

$$\mathcal{L}_e(\mathcal{I}^{adv}, \mathcal{I}; \theta) = \sum_{i=1}^{K} \alpha_i \cdot \frac{h_i \cdot {h'_i}^\top}{||h_i|| \cdot ||h'_i||}, \qquad (2)$$

where α_i is the weight controlling the loss on the i-th layer. We use cosine similarity to measure the error of two features.

Note that the elements in features have different impacts on the results. Disturbing the elements with high impact can more effectively disrupt the results. Thus the key loss $\mathcal{L}_k$ emphasizes attacking the key elements of features. To decide whether an element is a key, one straightforward way is to create a guidance map by back-propagating the gradients from the objective of training the face detectors with respect to the feature that is going to be attacked. However, the objective of training face detectors may vary due to different architectures, which hinders their general use, as a prior of the type of face detectors needs to know. To make a general form, we calculate the error of the last feature layer of the network to substitute the training objective, as $\mathcal{M}_i = \partial\mathcal{L}'_e / \partial h'_i$, where $\mathcal{L}'_e$ denotes the corresponding error loss, $\mathcal{M}_i$ is the guidance map for the i-th feature, and a larger value means a higher impact of an element. The key loss $\mathcal{L}_k$ can be defined as

$$\mathcal{L}_k(\mathcal{I}^{adv}, \mathcal{I}; \theta) = \sum_{i=1}^{K} \alpha_i \cdot (\mathcal{M}_i \cdot h'_i). \qquad (3)$$

The overall objective is $\mathcal{L} = \lambda_1 \mathcal{L}_e + \lambda_2 \mathcal{L}_k$, where λ_1, λ_2 are the weights for two losses. We optimize this objective using iterative steps as in MIM [24].

IV. Experiments

A. Experimental Settings

Datasets. To demonstrate the efficacy of disrupting face detection, we use WIDER [21] and UMDFaces datasets [30]. To validate the DeepFake defense ability, we use Celeb-DF dataset [4], which is a large-scale recent dataset containing more than 5500 DeepFake videos, covering 59 celebrities.

Face Detectors. We consider six state-of-the-art DNN-based face detectors in experiments, which are MTCNN [15], FaceBoxes [16], TinyFace [17], PyramidBox [12], S3FD [11] and DSFD [18] respectively. MTCNN is the combination of three self-designed networks called PNet, RNet and ONet. FaceBoxes is built upon InceptionNet [31]. TinayFace is constructed on ResNet101. PyramidBox, S3FD and DSFD utilize VGG16 as their basenetwork. All of these face detectors are trained under their default settings.

DeepFake Models. To fully demonstrate the effectiveness of our method, we study seven DeepFake models, namely Origin, DFaker, IAE, LightWeight, DFLH, CDFv1 and CDFv2 respectively. The first five models are from the repository of open-source tool `FaceSwap` [32] and the last two models are from [4]. These DeepFake models are various in input resolution and architecture [33], see Table I.

TABLE I
DeepFake models used in our experiments.

DF model	Input	Output	Encoder	Decoder	Variation
Origin	64	64	4Conv+1Ups	3Ups+1Conv	-
IAE	64	64	4Conv	4Ups+1Conv	Shared En&Decoder
LightWeight	64	64	3Conv+1Ups	3Ups+1Conv	Encoder
DFaker	64	128	4Conv+1Ups	4Ups+3Residual+1Conv	Decoder
DFLH	128	128	4Conv+1Ups	3Ups+1Conv	Resolution
CDFv1	128	128	5Conv+1Ups	4Ups+1Conv	Resolution; En&Decoder
CDFv2	256	256	5Conv+1Ups	5Ups+1Conv	Resolution; En&Decoder

TABLE II
F1-score (%) of different methods on WIDER dataset.

Methods	MTCNN	FaceBoxes	TinyFace	PyramidBox	S3FD	DSFD
None	86.8	79.5	81.6	98.7	94.6	96.3
Random	85.9	78.8	81.7	98.5	94.4	96.5
NNCO	80.2	77.2	80.4	95.6	90.1	92.1
SSOD	9.1	8.0	10.7	6.2	12.5	10.6
BIM	7.7	8.3	6.3	7.1	20.4	8.0
DIM	23.3	26.8	22.2	14.2	40.2	16.7
MIM	6.8	7.8	6.4	6.6	19.3	9.7
Ours	2.1	1.4	3.1	0.0	0.4	0.4

Implementation Details. The experiments are conducted on Ubuntu 18.04 with one Nvidia 2080ti GPU. The number of feature layers is set to $K = 3$. For VGG16 based face detectors, we attack 15-th, 22-nd and 28-th feature layers. For ResNet based face detectors, we attack the features out from first three residual blocks. It is similar to InceptionNet based face detector. For MTCNN, we select one layer from each sub-network. The parameter settings are as follows: $\alpha_1 = 0.2, \alpha_2 = 0.3, \alpha_3 = 0.5, \lambda_1 = 1, \lambda_2 = 1$. The maximum iteration number is 10 and the bound of distortion is set to $\epsilon = 8$. Note that the bound of distortion in attacking general image classification is usually set to 16, see [24], [34], [35]. But in our case, we restrict the bound to 8 for the imperceptible attack, as we are more sensitive to face images.

B. Disrupting Face Detection

We evaluate the performance of our method in comparison to the following methods. *Random* is a simple baseline that adds random Gaussian noise to images. *NNCO* [29] uses a GAN model with a generator of adversarial perturbations targeting face detectors. We use its default setting for comparison. *SSOD* is adapted from the method for attacking general object detectors [26]. Since our method attacks the intermediate features of base networks, we also adapt the methods of attacking classifiers for comparison, denoted as *BIM* [34], *DIM* [35] and *MIM* [24].

We use F1-score to evaluate the attacking performance of our method, which is calculated as $2 \cdot \frac{\text{Recall} \cdot \text{Precision}}{\text{Recall} + \text{Precision}}$, which considers both the precision and recall, which is more comprehensive than Average Precision (AP) metric (See supplementary for more details). The less score of these metrics means the faces are not well detected.

Table II and Table III show the F1-score $(\%)$ of different methods attacking face detectors on WIDER and UMDFaces

TABLE III
F1-SCORE (%) OF DIFFERENT METHODS ON UMDFACES DATASET.

Methods	MTCNN	FaceBoxes	TinyFace	PyramidBox	S3FD	DSFD
None	84.4	78.2	62.9	95.0	94.6	90.3
Random	83.2	78.0	60.4	94.5	93.2	88.3
NNCO	78.4	77.6	55.9	88.4	87.2	82.2
SSOD	8.8	7.4	5.3	7.1	15.6	11.3
BIM	5.2	7.3	4.5	6.2	18.5	7.6
DIM	19.9	24.4	20.2	17.3	33.6	14.4
MIM	6.9	9.2	5.9	5.3	16.2	7.9
Ours	2.6	1.8	1.1	0.1	0.3	0.6

TABLE IV
ABLATION STUDY OF OUR METHOD.

	MTCNN	FaceBoxes	TinyFace	PyramidBox	S3FD	DSFD
$\mathcal{L}_e$	3.1	2.5	4.1	2.2	1.2	1.0
$\mathcal{L}_k$	2.9	3.2	6.2	1.8	2.2	1.9
L1	4.1	5.5	7.1	10.2	7.2	6.0
L2	3.3	4.1	4.5	5.8	3.2	2.4
L3	3.0	2.4	3.9	0.8	1.0	1.2
Ours	2.1	1.4	3.1	0.0	0.4	0.4

datasets. *None* denotes no noises are added to images. By adding random noises, the performance of these methods merely drops, which indicates their robustness against random noises. It can be seen that NNCO is highly degraded on these face detectors. SSOD directly attacks the task-specific loss function, it can notably reduce the F1-score. The adapted methods of BIM, DIM and MIM can also disrupt face detection by only disturbing the intermediate features. Compared to these methods, our method performs best, which reduces the F1-score to $\sim 1\%$ on average, significantly disrupting face detection.

Ablation Study. We also study the effect of each loss term and multiple layers attack on WIDER dataset in Table IV. The top two rows are the performance of using error loss $\mathcal{L}_e$ and key loss $\mathcal{L}_k$ respectively. It can be seen that these two terms can disrupt face detectors effectively, but still can not outperform the combination of these terms. The third to fifth row shows the performance of only attacking one of the layers (L1, L2, L3), which reveals that our method can outperform them by $\sim 1\%$.

Transferability. Moreover, we investigate the transferability of our method, *i.e.*, attacking one face detector (target) using adversarial perturbations from another face detector (source), see Table V. "S" and "T" denote source and target face detectors respectively. The results in Table V indicate our method has transferability in most cases, *e.g.*, using PyramidBox to attack MTCNN can reduce its performance from 86.8% to 37.0%, to attack TinyFace can reduce the performance from 81.6% to 37.0%. It can be seen more clearly among PyramidBox, S3FD and DSFD, which can greatly disrupt the detection to less than 20%. However, FaceBoxes is difficult to attack using other face detectors, which is likely due to the large architectural discrepancy between FaceBoxes and other detectors.

Robustness. Since the social platform usually processes the uploaded images, we study the robustness of our method in this part, including the resistance to image compression and adversarial defense. 1) Image compression: we control the quality of images using OpenCV ranging in $[20, 100]$. The larger factor means higher image quality. Fig. 4 (left) shows the results of our method confronting different image compression. We can see that the F1-score of these face detectors are still around 20% at quality 50, which means our method can resist a certain image compression. 2) Adversarial Defense: Image transformation-based defense is a typical solution to remove adversarial perturbations. Thus we adapt method [36] to our task, which utilizes random resizing and padding on input images to resist adversarial attack. In this part, we set resizing ratio in $[0.75, 1]$. Fig. 4 (right) shows the results of our method against the adversarial defense. It can be seen that the F1-score only slightly drops when the reducing of resizing ratio, which demonstrates our method is robust to such defense to some extent.

TABLE V
TRANSFERABILITY OF OUR METHOD ON DISRUPTING FACE DETECTION. "S" AND "T" DENOTE SOURCE AND TARGET FACE DETECTORS RESPECTIVELY.

S↓, T→	MTCNN	FaceBoxes	TinyFace	PyramidBox	S3FD	DSFD
MTCNN	2.1	78.3	77.8	98.3	98.7	96.2
FaceBoxes	85.9	1.4	76.3	98.5	98.3	96.0
TinyFace	84.3	79.9	3.1	98.2	98.8	95.3
PyramidBox	37.0	74.3	31.8	0.0	4.6	12.9
S3FD	76.0	74.4	79.6	28.0	0.4	12.7
DSFD	48.0	75.1	50.1	8.6	11.1	0.4

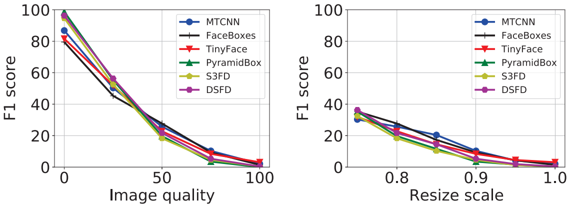

Fig. 4. Performance of different face detectors under different image quality (left) and resize scale (right).

C. Obstructing DeepFake

Since PyramidBox, S3FD and DSFD face detectors perform best, we use them as examples to demonstrate the efficacy of our method in obstructing DeepFake. Specifically, we select three identities (id0, id1, id2) and train DeepFake models using these pairs of identities (three pairs in total). Since we employ seven types of DeepFake models and each model is trained using three pairs of identities, we obtain 21 DeepFake models. For obstructing DeepFake, we apply our method to pollute the training faces of one identity and keep the other one untouched. Then we train another 21 DeepFake models. In DeepFake synthesis (inference), we feed correct face images into obstructed DeepFake models to see whether the synthesized faces are significantly disrupted. To evaluate the synthesis quality, we utilize SSIM, where less score denotes the synthesized faces are worse, indicating our method is more effective to obstruct DeepFake. Table VI shows the SSIM score of synthesized faces after using our method. *Random*

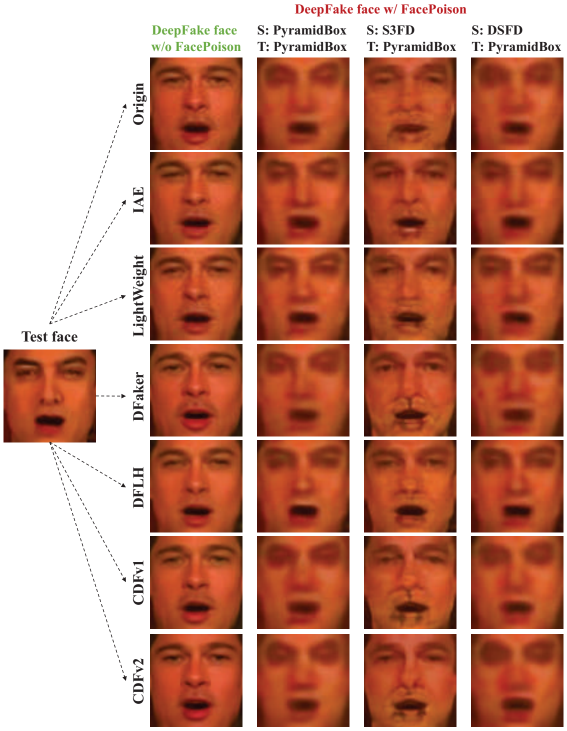

Fig. 5. Visual examples of DeepFake faces. The second column shows DeepFake faces without our method. The other right columns show DeepFake faces with our method. "S:A" and "T:B" denote the faces are extracted using B, which is attacked by adversarial perturbations from A.

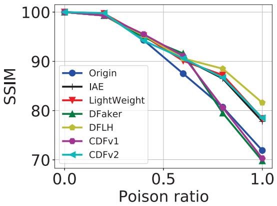

Fig. 6. Effect of different poison ratios of training faces.

denotes sending faces with random noise in the same strength of the distortion. We can observe that the random noise can hardly affect the synthesis quality, but our method can greatly reduce the SSIM score (at least 10%), despite the source and target face detectors are not the same. The results demonstrate our method can effectively obstruct the training of DeepFake models. Fig.5 shows visual examples of DeepFake faces with our method. It can be seen our method can effectively damage the visual quality of DeepFake faces.

Effect of Poison Ratio. This part investigates the effect of the poison ratio of training faces, *i.e.*, the relationship between the synthesis quality and the ratio of incorrect faces in training. Specifically, we change the poison ratio in a range of $[0, 1]$, where 0 denotes our method is not applied and 1 denotes our method is applied to all training faces. Fig. 6 illustrates the curve of the SSIM score with different poison ratios. It can be seen that the training faces do not need to be fully disrupted by our method, in order to reduce the synthesis quality, *e.g.*, the SSIM score is still under 90% at poison ratio 0.8.

TABLE VI
SSIM SCORE ($\%$) OF SYNTHESIZED FACES USING OUR METHOD.

DF model	S↓, T→	PyramidBox	S3FD	DSFD
Origin	Random	90.2		
	PyramidBox	71.9	70.5	71.3
	S3FD	71.6	62.5	64.5
	DSFD	72.0	73.9	73.2
IAE	Random	92.1		
	PyramidBox	77.8	77.3	78.9
	S3FD	82.0	72.0	71.2
	DSFD	77.0	81.0	79.4
LightWeight	Random	90.2		
	PyramidBox	78.4	77.8	81.8
	S3FD	78.9	71.2	73.3
	DSFD	77.0	79.2	82.5
DFaker	Random	91.7		
	PyramidBox	68.9	68.4	70.3
	S3FD	69.1	60.7	62.4
	DSFD	68.9	70.1	72.1
DFLH	Random	92.3		
	PyramidBox	81.6	80.9	83.3
	S3FD	80.7	76.4	76.3
	DSFD	81.6	82.4	83.5
CDFv1	Random	92.4		
	PyramidBox	70.3	69.6	73.2
	S3FD	69.9	63.0	61.4
	DSFD	71.1	72.4	74.5
CDFv2	Random	93.0		
	PyramidBox	78.5	77.4	80.6
	S3FD	77.3	73.3	71.6
	DSFD	78.5	79.3	80.5

D. Discussion

On one hand, we expect this technology to spawn countermeasures from the forgery makers. In particular, operations that can destroy or reduce the adversarial perturbation are expected to be developed. It is thus our continuing effort to improve the robustness of the adversarial perturbation generation method. Also, we would like to explore a more generic black-box attack that can be significantly transferred among different types of face detectors. On the other hand, the forensics strategies are impossibly perfect, but they can still raise the bar of creating forgeries.

V. Conclusion

DeepFake is becoming a problem encroaching on our trust in online media. As DeepFake requires automatic face detection as an indispensable pre-processing step in preparing training data, an effective protection scheme can be obtained by disrupting the face detection methods. In this work, we develop a proactive protection method (FacePoison) to deter bulk reuse of automatically detected faces for the production of DeepFake faces. Our method exploits the sensitivity of DNN-based face detectors and uses adversarial perturbation to contaminate the face sets. This is achieved by a new multi-scale feature-level adversarial attack to disrupt face detectors. The experiments are conducted on seven different DeepFake models with six different face detectors, and empirically show the effectiveness of our method in obstructing DeepFakes.

Acknowledgement. This work is supported by the Fundamental Research Funds for the Central Universities and China Postdoctoral Science Foundation under Grant No. 2021TQ0314 and 2021M703036. Jiaran Zhou is supported by NSFC under grant No.62102380, NSF of Shandong province under grant No.ZR2021QF095, and China Postdoc Science Foundation under grant No.2021M693022.

REFERENCES

[1] Tero Karras, Samuli Laine, and Timo Aila, "A style-based generator architecture for generative adversarial networks," in *CVPR*, 2019.

[2] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen, "Progressive growing of GANs for improved quality, stability, and variation," in *ICLR*, 2018.

[3] Robert Chesney and Danielle Keats Citron, "Deep Fakes: A Looming Challenge for Privacy, Democracy, and National Security," *107 California Law Review (2019, Forthcoming); U of Texas Law, Public Law Research Paper No. 692; U of Maryland Legal Studies Research Paper No. 2018-21*.

[4] Yuezun Li, Xin Yang, Pu Sun, Honggang Qi, and Siwei Lyu, "Celeb-df: A large-scale challenging dataset for deepfake forensics," in *CVPR*, 2020.

[5] Darius Afchar, Vincent Nozick, Junichi Yamagishi, and Isao Echizen, "MesoNet: a compact facial video forgery detection network," in *WIFS*, 2018.

[6] Yuezun Li, Ming-Ching Chang, and Siwei Lyu, "In Ictu Oculi: Exposing AI generated fake face videos by detecting eye blinking," in *WIFS*, 2018.

[7] Yuezun Li and Siwei Lyu, "Exposing deepfake videos by detecting face warping artifacts," in *CVPR Workshops*, 2019.

[8] 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 *CVPR*, 2020.

[9] Jianwei Fei, Yunshu Dai, Peipeng Yu, Tianrun Shen, Zhihua Xia, and Jian Weng, "Learning Second Order Local Anomaly for General Face Forgery Detection," in *CVPR*, 2022.

[10] Liang Chen, Yong Zhang, Yibing Song, Lingqiao Liu, and Jue Wang, "Self-supervised Learning of Adversarial Example: Towards Good Generalizations for Deepfake Detection," in *CVPR*, 2022.

[11] Shifeng Zhang, Xiangyu Zhu, Zhen Lei, Hailin Shi, Xiaobo Wang, and Stan Z Li, "S3FD: Single shot scale-invariant face detector," in *ICCV*, 2017.

[12] Xu Tang, Daniel K Du, Zeqiang He, and Jingtuo Liu, "Pyramidbox: A context-assisted single shot face detector," in *ECCV*, 2018.

[13] Wen Zhou, Xin Hou, Yongjun Chen, Mengyun Tang, Xiangqi Huang, Xiang Gan, and Yong Yang, "Transferable adversarial perturbations," in *ECCV*, 2018.

[14] Yuezun Li, Ming-Ching Chang, Pu Sun, Honggang Qi, Junyu Dong, and Siwei Lyu, "Transrpn: Towards the transferable adversarial perturbations using region proposal networks and beyond," *CVIU*, 2021.

[15] Jia Xiang and Gengming Zhu, "Joint face detection and facial expression recognition with mtcnn," in *ICISCE*, 2017.

[16] Shifeng Zhang, Xiangyu Zhu, Zhen Lei, Hailin Shi, Xiaobo Wang, and Stan Z Li, "Faceboxes: A cpu real-time face detector with high accuracy," in *IJCB*, 2017.

[17] Zhiyi Cheng, Xiatian Zhu, and Shaogang Gong, "Low-resolution face recognition," in *ACCV*, 2018.

[18] Jian Li, Yabiao Wang, Changan Wang, Ying Tai, Jianjun Qian, Jian Yang, Chengjie Wang, Jilin Li, and Feiyue Huang, "Dsfd: dual shot face detector," in *CVPR*, 2019.

[19] Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *CVPR*, 2016.

[21] Shuo Yang, Ping Luo, Chen Change Loy, and Xiaoou Tang, "Wider face: A face detection benchmark," in *CVPR*, 2016.

[22] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy, "Explaining and harnessing adversarial examples," in *ICLR*, 2015.

[23] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard, "Deepfool: a simple and accurate method to fool deep neural networks," in *CVPR*, 2016.

[24] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li, "Boosting adversarial attacks with momentum," in *CVPR*, 2018.

[25] Jiajun Lu, Hussein Sibai, and Evan Fabry, "Adversarial examples that fool detectors," *arXiv 1712.02494*, 2017.

[26] Cihang Xie, Jianyu Wang, Zhishuai Zhang, Yuyin Zhou, Lingxi Xie, and Alan Yuille, "Adversarial examples for semantic segmentation and object detection," in *ICCV*, 2017.

[27] Yuezun Li, Daniel Tian, Mingching Chang, Xiao Bian, and Siwei Lyu, "Robust adversarial perturbation on deep proposal-based models," in *BMVC*, 2018.

[28] Yuezun Li, Xiao Bian, Ming-Ching Chang, and Siwei Lyu, "Exploring the vulnerability of single shot module in object detectors via imperceptible background patches," in *BMVC*, 2019.

[29] Avishek Joey Bose and Parham Aarabi, "Adversarial attacks on face detectors using neural net based constrained optimization," in *MMSP*, 2018.

[30] Ankan Bansal, Anirudh Nanduri, Carlos D Castillo, Rajeev Ranjan, and Rama Chellappa, "Umdfaces: An annotated face dataset for training deep networks," *arXiv preprint arXiv:1611.01484v2*, 2016.

[31] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna, "Rethinking the inception architecture for computer vision," in *CVPR*, 2016.

[32] "Faceswap," https://github.com/deepfakes/faceswap.

[33] Shan Jia, Xin Li, and Siwei Lyu, "Model attribution of face-swap deepfake videos," in *ICIP*, 2022.

[34] Alexey Kurakin, Ian J Goodfellow, and Samy Bengio, "Adversarial examples in the physical world," in *Artificial intelligence safety and security*. 2018.

[35] Cihang Xie, Zhishuai Zhang, Yuyin Zhou, Song Bai, Jianyu Wang, Zhou Ren, and Alan L Yuille, "Improving transferability of adversarial examples with input diversity," in *CVPR*, 2019.

[36] Cihang Xie, Jianyu Wang, Zhishuai Zhang, Zhou Ren, and Alan Yuille, "Mitigating adversarial effects through randomization," *arXiv preprint arXiv:1711.01991*, 2017.