Adversarial Deepfakes: Evaluating Vulnerability of Deepfake Detectors to Adversarial Examples

*Shehzeen Hussain, *Paarth Neekhara, Malhar Jere, Farinaz Koushanfar, Julian McAuley
University of California San Diego
{pneekhar,ssh028}@ucsd.edu
* Equal contribution

Abstract

Recent advances in video manipulation techniques have made the generation of fake videos more accessible than ever before. Manipulated videos can fuel disinformation and reduce trust in media. Therefore detection of fake videos has garnered immense interest in academia and industry. Recently developed Deepfake detection methods rely on Deep Neural Networks (DNNs) to distinguish AI-generated fake videos from real videos. In this work, we demonstrate that it is possible to bypass such detectors by adversarially modifying fake videos synthesized using existing Deepfake generation methods. We further demonstrate that our adversarial perturbations are robust to image and video compression codecs, making them a real-world threat. We present pipelines in both white-box and black-box attack scenarios that can fool DNN based Deepfake detectors into classifying fake videos as real.

1. Introduction

With the advent of sophisticated image and video synthesis techniques, it has become increasingly easier to generate high-quality convincing fake videos. Deepfakes are a new genre of synthetic videos, in which a subject's face is modified into a target face in order to simulate the target subject in a certain context and create convincingly realistic footage of events that never occurred. Video manipulation methods like Face2Face [49], Neural Textures [48] and FaceSwap [26] operate end-to-end on a source video and target face and require minimal human expertise to generate fake videos in real-time.

The intent of generating such videos can be harmless and have advanced the research of synthetic video generation for movies, storytelling and modern-day streaming services. However, they can also be used maliciously to spread disinformation, harass individuals or defame famous personalities [45]. The extensive spread of fake videos through social media platforms has raised significant concerns worldwide, particularly hampering the credibility of digital media.

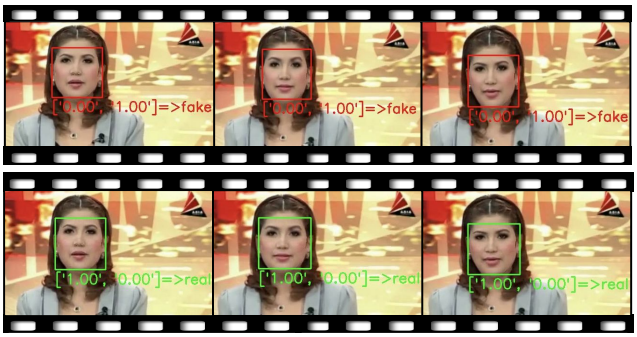

Figure 1. Adversarial Deepfakes for XceptionNet [40] detector. Top: Frames of of a *fake* video generated by Face2Face being correctly identified as *fake* by the detector. Bottom: Corresponding frames of the adversarially modified fake video being classified as real by the detector.

To address the threats imposed by Deepfakes, the machine learning community has proposed several countermeasures to identify forgeries in digital media. Recent state-of-the-art methods for detecting manipulated facial content in videos rely on Convolutional Neural Networks (CNNs) [17, 40, 1, 2, 30, 39]. A typical Deepfake detector consists of a face-tracking method, following which the cropped face is passed on to a CNN-based classifier for classification as real or fake [1, 13]. Some of the recent DeepFake detection methods use models operate on a sequence of frames as opposed to a single frame to exploit temporal dependencies in videos [15].

While the above neural network based detectors achieve promising results in accurately detecting manipulated videos, in this paper we demonstrate that they are susceptible to *adversarial examples* which can fool the detectors to classify fake videos as real [1]. An adversarial example is an intentionally perturbed input that can fool a victim classification

[1] Video Examples: https://adversarialdeepfakes.github.io/

978-0-7381-4266-1/21/$31.00 ©2021 IEEE
DOI 10.1109/WACV48630.2021.00339

model [46]. Even though several works have demonstrated that neural networks are vulnerable to adversarial inputs (Section 2.3), we want to explicitly raise this issue that has been ignored by existing works on Deepfake detection (Section 2.2). Since fake video generation can potentially be used for malicious purposes, it is critical to address the vulnerability of Deepfake detectors to adversarial inputs.

To this end, we quantitatively assess the vulnerability of state-of-the-art Deepfake detectors to adversarial examples. Our proposed methods can augment existing techniques for generating fake videos, such that they can bypass a given fake video detector. We generate adversarial examples for each frame of a given fake video and combine them together to synthesize an adversarially modified video that gets classified as real by the victim Deepfake detector. We demonstrate that it is possible to construct fake videos that are robust to image and video compression codecs, making them a real world threat since videos shared over social media are usually compressed. More alarmingly, we demonstrate that it is possible to craft robust adversarial Deepfakes in black-box settings, where the adversary may not be aware of the classification model used by the detector. Finally, we discuss normative points about how the community should approach the problem of Deepfake detection.

2. Background

2.1. Generating Manipulated Videos

Until recently, the ease of generating manipulated videos has been limited by manual editing tools. However, since the advent of deep learning and inexpensive computing services, there has been significant work in developing new techniques for automatic digital forgery. In our work, we generate adversarial examples for fake videos synthesized using FaceSwap (FS) [26], Face2Face (F2F) [49], DeepFakes (DF) [16] and NeuralTextures (NT) [48]. We perform our experiments on this FaceForesics++ dataset [40], which is a curated dataset of manipulated videos containing facial forgery using the above methods. Another recently proposed dataset containing videos with facial forgery is the DeepFake Detection Challenge (DFDC) Dataset [17], which we utilize when evaluating our attacks against sequence based detection frameworks (Section 3.1).

2.2. Detecting Manipulated Videos

Traditionally, multimedia forensics investigated the authenticity of images [51, 10, 21] using hand-engineered features and/or a-priori knowledge of the statistical and physical properties of natural photographs. However, video synthesis methods can be trained to bypass hand-engineered detectors by modifying their training objective. We direct readers to [7, 9] for an overview of counter-forensic attacks to bypass traditional (non-deep learning based) methods of detecting forgeries in multimedia content.

More recent works have employed CNN-based approaches that decompose videos into frames to automatically extract salient and discriminative visual features pertinent to Deepfakes. Some efforts have focused on segmenting the entire input image to detect facial tampering resulting from face swapping [56], face morphing [38] and splicing attacks [5, 6]. Other works [28, 29, 1, 23, 40, 41] have focused on detecting face manipulation artifacts resulting from Deepfake generation methods. The authors of [29] reported that eye blinking is not well reproduced in fake videos, and therefore proposed a temporal approach using a CNN + Recurrent Neural Network(RNN) based model to detect a lack of eye blinking when exposing deepfakes. Similarly, [54] used the inconsistency in head pose to detect fake videos. However, this form of detection can be circumvented by purposely incorporating images with closed eyes and a variety of head poses in training [50, 18].

The Deepfake detectors proposed in [40, 1, 17] model Deepfake detection as a per-frame binary classification problem. The authors of [40] demonstrated that XceptionNet can outperform several alternative classifiers in detecting forgeries in both uncompressed and compressed videos, and identifying forged regions in them. In our work, we expose the vulnerability of such state-of-the-art Deepfake detectors. Since the task is to specifically detect facial manipulation, these models incorporate domain knowledge by using a face tracking method [49] to track the face in the video. The face is then cropped from the original frame and fed as input to classification model to be labelled as *Real* or *Fake*. Experimentally, the authors of [40] demonstrate that incorporation of domain knowledge helps improve classification accuracy as opposed to using the entire image as input to the classifier. The best performing classifiers amongst others studied by [40] were both CNN based models: XceptionNet [13] and MesoNet [1]. More recently, some detectors have also focused on exploiting temporal dependencies while detecting DeepFake videos. Such detectors work on sequence of frames as opposed to a single frame using a CNN + RNN model or a 3-D CNN model. One such model based on a 3-D EfficientNet [47] architecture, was used by the third place winner [15] of the recently conducted DeepFake Detection Challenge (DFDC) [17]. The first two winning submissions were CNN based per-frame classification models similar to ones described above. We evaluate our attacks against the 3D CNN model to expose the vulnerability of temporal Deepfake detectors.

2.3. Adversarial Examples

Adversarial examples are intentionally designed inputs to a machine learning (ML) model that cause the model to make a mistake [46]. Prior work has shown a series of first-order gradient-based attacks to be fairly effective in fooling DNN

based models in both image [35, 34, 22, 31, 11, 44, 43], audio [12, 37, 33] and text [20, 8, 32] domains. The objective of such adversarial attacks is to find a good trajectory that (i) maximally changes the value of the model's output and (ii) pushes the sample towards a low-density region. This is equivalent to the ML model's gradient with respect to input features. Prior work on defenses [53] against adversarial attacks, propose to perform random operations over the input images, e.g., random cropping and JPEG compression. However, such defenses are shown to be vulnerable to attack algorithms that are aware of the randomization approach. Particularly, one line of adversarial attack [3, 4] computes the expected value of gradients for each of the sub-sampled networks/inputs and performs attacks that are robust against compression.

3. Methodology

3.1. Victim Models: Deepfake Detectors

Frame-by-Frame detectors: To demonstrate the effectiveness of our attack on Deepfake detectors, we first choose detectors which rely on frame level CNN based classification models. These victim detectors work on the frame level and classify each frame independently as either *Real* or *Fake* using the following two-step pipeline:

1. A face tracking model [49] extracts the bounding box of the face in a given frame.
2. The cropped face is then resized appropriately and passed as input to a CNN based classifier to be labelled as either real or fake.

In our work, we consider two victim CNN classifiers: XceptionNet [13] and MesoNet [1]. Detectors based on the above pipeline have been shown to achieve state-of-the-art performance in Deepfake detection as reported in [17, 40, 55]. The accuracy of such models on the FaceForensics++ Dataset [40] is reported in Table 1.

Sequence based models: We also demonstrate the effectiveness of our attacks on detectors that utilize temporal dependencies. Such detection methods typically use a CNN + RNN or a 3D-CNN architecture to classify a *sequence* of frames as opposed to a single frame. A 3D-CNN architecture performs convolutions across height, width and time axis thereby exploiting temporal dependencies. In Section 5, we evaluate our attacks against one such detection method [15] that uses a 3-D EfficientNet [47] CNN model for classifying a sequence of face-crops obtained from a face tracking model. In this model, a 3-D convolution is added to each block of the EfficientNet model to perform convolutions across time. The length of the input sequence to the model is 7 frames and the step between frames is 1/15 of a second. This 3-D CNN model was used by the third place winner of the recently conducted DFDC challenge.

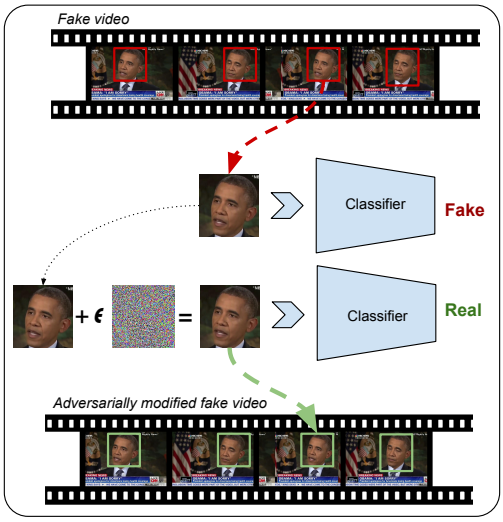

Figure 2. An overview of our attack pipeline to generate Adversarial Deepfakes. We generate an adversarial example for each frame in the given fake video and combine them together to create an adversarially modified fake video.

3.2. Threat Model

Given a facially manipulated (fake) video input and a victim Deepfake detector, our task is to adversarially modify the fake video such that most of the frames get classified as *Real* by the Deepfake detector, while ensuring that the adversarial modification is quasi-imperceptible.

Distortion Metric: To ensure imperceptibility of the adversarial modification, the L_p norm is a widely used distance metric for measuring the distortion between the adversarial and original inputs. The authors of [22] recommend constraining the maximum distortion of any individual pixel by a given threshold ϵ, i.e., constraining the perturbation using an L_∞ metric. Additionally, *Fast Gradient Sign Method* (FGSM) [22] based attacks, which are optimized for the L_∞ metric, are more time-efficient than attacks which optimize for L_2 or L_0 metrics. Since each video can be composed of thousands of individual frames, time-efficiency becomes an important consideration to ensure the proposed attack can be reliably used in practice. Therefore, in this work, we use the L_∞ distortion metric for constraining our adversarial perturbation and optimize for it using gradient sign based methods.

Notation: We follow the notation previously used in [11, 36]: Define F to be the full neural network (classifier) including the softmax function, $Z(x) = z$ to be the output of all layers except the softmax (so z are the logits), and

$$F(x) = softmax(Z(x)) = y$$

The classifier assigns the label $C(x) = \arg\max_i(F(x)_i)$ to input x.

Problem Formulation: Mathematically, for any given

frame x_0 of a fake video, and a victim frame-forgery detector model C, we aim to find an adversarial frame x_{adv} such that,

$$C(x_{adv}) = \textit{Real} \text{ and } ||x_{adv} - x_0||_\infty < \epsilon$$

Attack Pipeline: An overview of the process of generating adversarial fake videos is depicted in Figure 2. For any given frame, we craft an adversarial example for the cropped face, such that after going through some image transformations (normalization and resizing), it gets classified as *Real* by the classifier. The adversarial face is then placed in the bounding box of face-crop in the original frame, and the process is repeated for all frames of the video to create an adversarially modified fake video. In the following sections, we consider our attack pipeline under various settings and goals.

Note that, the proposed attacks can also be applied on detectors that operate on entire frames as opposed to face-crops. We choose face-crop based victim models because they have been shown to outperform detectors that operate on entire frames for detecting facial-forgeries.

3.3. White-box Attack

In this setting, we assume that the attacker has complete access to the detector model, including the face extraction pipeline and the architecture and parameters of the classification model. To construct adversarial examples using the attack pipeline described above, we use the iterative gradient sign method [27] to optimize the following loss function:

$$\begin{aligned} &\text{Minimize } loss(x') \text{ where} \\ &loss(x') = max(Z(x')_{Fake} - Z(x')_{Real}, 0) \end{aligned} \tag{1}$$

Here, $Z(x)_y$ is the final score for label y before the softmax operation in the classifier C. Minimizing the above loss function maximizes the score for our target label $Real$. The loss function we use is recommended in [11] because it is empirically found to generate less distorted adversarial samples and is robust against defensive distillation. We use the iterative gradient sign method to optimize the above objective while constraining the magnitude of the perturbation as follows:

$$x_i = x_{i-1} - \text{clip}_\epsilon(\alpha \cdot \text{sign}(\nabla loss(x_{i-1}))) \tag{2}$$

We continue gradient descent iterations until success or until a given number number of maximum iterations, whichever occurs earlier. In our experiments, we demonstrate that while we are able to achieve an average attack success rate of 99.05% when we save videos with uncompressed frames, the perturbation is not robust against video compression codecs like *MJPEG*. In the following section, we discuss our approach to overcome this limitation of our attack.

3.4. Robust White-box Attack

Generally, videos uploaded to social networks and other media sharing websites are compressed. Standard operations like compression and resizing are known for removing adversarial perturbations from an image [19, 14, 24]. To ensure that the adversarial videos remain effective even after compression, we craft adversarial examples that are robust over a given distribution of input transformations [4]. Given a distribution of input transformations T, input image x, and target class y, our objective is as follows:

$$x_{adv} = argmax_x \mathbb{E}_{t\sim T}[F(t(x))_y] \text{ s.t. } ||x - x_0||_\infty < \epsilon$$

That is, we want to maximize the expected probability of target class y over the distribution of input transforms T. To solve the above problem, we update the loss function given in Equation 1 to be an expectation over input transforms T as follows:

$$loss(x) = \mathbb{E}_{t\sim T}[max(Z(t(x))_{Fake} - Z(t(x))_{Real}, 0)]$$

Following the law of large numbers, we estimate the above loss functions for n samples as:

$$loss(x) = \frac{1}{n}\sum_{t_i\sim T}[max(Z(t_i(x))_{Fake} - Z(t_i(x))_{Real}, 0)]$$

Since the above loss function is a sum of differentiable functions, it is tractable to compute the gradient of the loss w.r.t. to the input x. We minimize this loss using the iterative gradient sign method given by Equation 2. We iterate until a given a number number of maximum iterations or until the attack is successful under the sampled set of transformation functions, whichever happens first.

Next we describe the class of input transformation functions we consider for the distribution T:

Gaussian Blur: Convolution of the original image with a Gaussian kernel k. This transform is given by $t(x) = k * x$ where $*$ is the convolution operator.

Gaussian Noise Addition: Addition of Gaussian noise sampled from $\Theta \sim \mathcal{N}(0, \sigma)$ to the input image. This transform is given by $t(x) = x + \Theta$

Translation: We pad the image on all four sides by zeros and shift the pixels horizontally and vertically by a given amount. Let t_x be the transform in the x axis and t_y be the transform in the y axis, then $t(x) = x'_{H,W,C}$ s.t $x'[i, j, c] = x[i + t_x, j + t_y, c]$

Downsizing and Upsizing: The image is first downsized by a factor r and then up-sampled by the same factor using bilinear re-sampling.

The details of the hyper-parameter search distribution used for these transforms can be found in the Section 4.1.

3.5. Black-box Attack

In the black-box setting, we consider the more challenging threat model in which the adversary does not have access to the classification network architecture and parameters. We assume that the attacker has knowledge of the detection pipeline structure and the face tracking model. However, the attacker can solely query the classification model as a black-box function to obtain the probabilities of the frame being *Real* or *Fake*. Hence there is a need to estimate the gradient of the loss function by querying the model and observing the change in output for different inputs, since we cannot backpropagate through the network.

We base our algorithm for efficiently estimating the gradient from queries on the Natural Evolutionary Strategies (NES) approach of [52, 25]. Since we do not have access to the pre-softmax outputs Z, we aim to maximize the class probability $F(x)_y$ of the target class y. Rather than maximizing the objective function directly, NES maximizes the expected value of the function under a search distribution $\pi(\theta|x)$. That is, our objective is:

$$\textit{Maximize: } \mathbb{E}_{\pi(\theta|x)}[F(\theta)_y]$$

This allows efficient gradient estimation in fewer queries as compared to finite-difference methods. From [52], we know the gradient of expectation can be derived as follows:

$$\nabla_x \mathbb{E}_{\pi(\theta|x)}\left[F(\theta)_y\right] = \mathbb{E}_{\pi(\theta|x)}\left[F(\theta)_y \nabla_x \log\left(\pi(\theta|x)\right)\right]$$

Similar to [25, 52], we choose a search distribution $\pi(\theta|x)$ of random Gaussian noise around the current image x. That is, $\theta = x + \sigma\delta$ where $\delta \sim \mathcal{N}(0, I)$. Estimating the gradient with a population of n samples yields the following variance reduced gradient estimate:

$$\nabla\mathbb{E}[F(\theta)] \approx \frac{1}{\sigma n}\sum_{i=1}^{n} \delta_i F(\theta + \sigma\delta_i)_y$$

We use antithetic sampling to generate δ_i similar to [42, 25]. That is, instead of generating n values $\delta \sim \mathcal{N}(0, I)$, we sample Gaussian noise for $i \in \{1, \ldots, \frac{n}{2}\}$ and set $\delta_j = -\delta_{n-j+1}$ for $j \in \{(\frac{n}{2}+1), \ldots, n\}$. This optimization has been empirically shown to improve performance of NES. Algorthim 1 details our implementation of estimating gradients using NES. The transformation distribution T in the algorithm just contains an identity function i.e., $T = \{I(x)\}$ for the black-box attack described in this section.

After estimating the gradient, we move the input in the direction of this gradient using iterative gradient sign updates to increase the probability of target class:

$$x_i = x_{i-1} + \text{clip}_\epsilon(\alpha \cdot \text{sign}(\nabla F(x_{i-1})_y)) \qquad (3)$$

3.6. Robust Black-box Attack

To ensure robustness of adversarial videos to compression, we incorporate Expectation over Transforms (Section 3.4) method in the black-box setting for constructing adversarial videos.

To craft adversarial examples that are robust under a given set of input transformations T, we maximize the expected value of the function under a search distribution $\pi(\theta|x)$ and our distribution of input transforms T. That is, our objective is to maximize:

$$\mathbb{E}_{t\sim T}[\mathbb{E}_{\pi(\theta|x)}\left[F(t(\theta))_y\right]]$$

Following the derivation in the previous section, the gradient of the above expectation can be estimated using a population of size n by iterative sampling of t_i and δ_i:

$$\nabla\mathbb{E}[F(\theta)] \approx \frac{1}{\sigma n}\sum_{i=1, t_i\sim T}^{n} \delta_i F(t_i(\theta + \sigma\delta_i))_y$$

Algorithm 1 NES Gradient Estimate

Input: Classifier $F(x)$,target class y, image x
Output: Estimate of $\nabla_x F(x)_y$
Parameters: Search variance σ, number of samples n, image dimensionality N
$g \leftarrow \mathbf{0}_n$
for $i = 1$ **to** n **do**
  $t_i \sim T$
  $u_i \leftarrow \mathcal{N}(\mathbf{0}_N, \boldsymbol{I}_{N\cdot N})$
  $g \leftarrow g + F(t_i(x + \sigma \cdot u_i))_y \cdot u_i$
  $g \leftarrow g - F(t_i(x - \sigma \cdot u_i))_y \cdot u_i$
end for
return $\frac{1}{2n\sigma} g$

We use the same class of transformation functions listed in Section 3.4 for the distribution T. Algorithm 1 details our implementation for estimating gradients for crafting robust adversarial examples. We follow the same update rule given by Equation 3 to generate adversarial frames. We iterate until a given a number of maximum iterations or until the attack is successful under the sampled set of transformation functions.

4. Experiments

Dataset and Models: We evaluate our proposed attack algorithm on two pre-trained victim models: XceptionNet [13] and MesoNet [1]. In our experiments, we perform our attack on the test set of the FaceForensics++ Dataset [40], consisting of manipulated videos from the four methods described in Section 2.1. We construct adversarially modified fake videos on the FaceForensics++ test set, which contains

70 videos (total 29,764 frames) from each of the four manipulation techniques. For simplicity, our experiments are performed on high quality (HQ) videos, which apply a light compression on raw videos. The accuracy of the detector models for detecting facially manipulated videos on this test set is reported in Table 1. We will be releasing code for all our attack algorithms in PyTorch[2].

	DF	F2F	FS	NT
XceptionNet Acc %	97.49	97.69	96.79	92.19
MesoNet Acc %	89.55	88.6	81.24	76.62

Table 1. Accuracy of Deepfake detectors on the FaceForensics++ HQ Dataset as reported in [40]. The results are for the entire high-quality compressed test set generated using four manipulation techniques (DF: DeepFakes, F2F: Face2Face, FS: FaceSwap and NT: NeuralTextures).

Evaluation Metrics: Once the adversarial frames are generated, we combine them and save the adversarial videos in the following formats:
1) *Uncompressed (Raw):* Video is stored as a sequence of uncompressed images.
2) *Compressed (MJPEG):* Video is saved as a sequence of JPEG compressed frames.
3) *Compressed (H.264):* Video is saved in the commonly used mp4 format that applies temporal compression across frames.

We conduct our primary evaluation on the *Raw* and *MJPEG* video formats across all attacks. We also study the effectiveness of our white box robust attack using different compression levels in the *H264* codec. We report the following metrics for evaluating our attacks:
Success Rate (SR): The percentage of frames in the adversarial videos that get classified to our target label *Real*. We report: **SR-U**- Attack success rate on uncompressed adversarial videos saved in Raw format; and **SR-C**- Attack success rate on compressed adversarial videos saved in MJPEG format.
Accuracy: The percentage of frames in videos that get classified to their original label *Fake* by the detector. We report **Acc-C**- accuracy of the detector on compressed adversarial videos.
Mean distortion ($\mathbf{L_\infty}$): The average L_∞ distortion between the adversarial and original frames. The pixel values are scaled in the range [0,1], so changing a pixel from full-on to full-off in a grayscale image would result in L_∞ distortion of 1 (not 255).

4.1. White-box Setting

To craft adversarial examples in the white-box setting, in our attack pipeline, we implement differentiable image pre-processing (resizing and normalization) layers for the CNN. This allows us to backpropagate gradients all the way to the cropped face in-order to generate the adversarial image that can be placed back in the frame. We set the maximum number of iterations to 100, learning rate α to $1/255$ and max L_∞ constraint ϵ to $16/255$ for both our attack methods described in Sections 3.3 and 3.4.

	XceptionNet				*MesoNet*			
Dataset	$\mathbf{L_\infty}$	**SR - U**	**SR - C**	**Acc-C%**	$\mathbf{L_\infty}$	**SR - U**	**SR - C**	**Acc-C%**
DF	0.004	99.67	43.11	56.89	0.006	97.30	92.27	7.73
F2F	0.004	99.85	52.50	47.50	0.007	98.94	96.30	4.70
FS	0.004	100.00	43.13	56.87	0.009	97.12	86.10	13.90
NT	0.004	99.89	95.10	4.90	0.007	99.22	96.20	3.80
All	0.004	99.85	58.46	41.54	0.007	98.15	92.72	7.28

Table 2. Success Rate of White-box attack on XceptionNet and MesoNet. We report the average L_∞ distortion between the adversarial and original frames and the attack success rate on uncompressed (SR-U) and compressed (SR-C) videos. Acc-C denotes the accuracy of the detector on compressed adversarial videos.

Table 2 shows the results of the white-box attack (Section 3.3). We are able to generate adversarial videos with an average success rate of 99.85% for fooling XceptionNet and 98.15% for MesoNet when adversarial videos are saved in the Raw format. However, the attack average success rate drops to 58.46% for XceptionNet and 92.72% for MesoNet when MJPEG compression is used. This result is coherent with past works [19, 14, 24] that employ JPEG compression and image transformations to defend against adversarial examples.

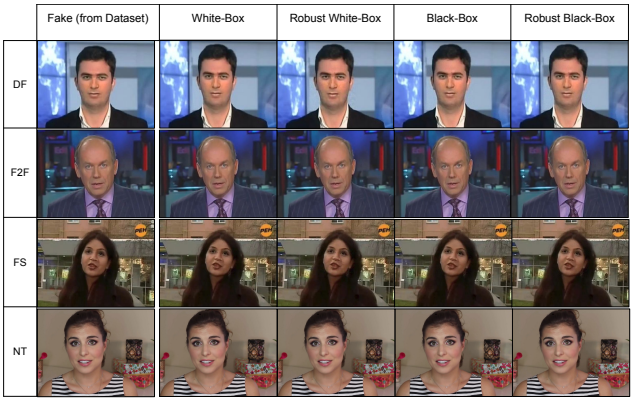

Figure 3. Randomly selected frames of Adversarial Deepfakes from successful attacks. The frame from the dataset in the first column is correctly identified as *Fake* by the detectors, while the corresponding frames generated by each of our attacks are labelled as *Real* with a probability of 1. Video examples are linked in the footnote on the first page.

Robust White-Box: For our robust white box attack, we sample 12 transformation functions from the distribution T for estimating the gradient in each iteration. This includes three functions from each of the four transformations listed

[2]Code released upon publication

in Section 3.4. Table 3 shows the search distribution for different hyper-parameters of the transformation functions.

Transform	Hyper-parameter search distribution
Gaussian Blur	Kernel $k(d, d, \sigma)$, $d \sim \mathcal{U}[3, 7]$, $\sigma \sim \mathcal{U}[5, 10]$
Gaussian Noise	$\sigma \sim \mathcal{U}[0.01, 0.02]$
Translation	$d_x \sim \mathcal{U}[-20, 20]$, $d_y \sim \mathcal{U}[-20, 20]$
Down-sizing & Up-sizing	Scaling factor $r \sim \mathcal{U}[2, 5]$

Table 3. Search distribution of hyper-parameters of different transformations used for our Robust White box attack. During training, we sample three functions from each of the transforms to estimate the gradient of our expectation over transforms.

	XceptionNet				*MesoNet*			
Dataset	L_∞	**SR - U**	**SR - C**	**Acc-C%**	L_∞	**SR - U**	**SR - C**	**Acc-C%**
DF	0.016	99.67	98.71	1.29	0.030	99.94	99.85	0.15
F2F	0.013	100.00	99.00	1.00	0.020	99.71	99.67	0.33
FS	0.013	100.00	95.33	4.67	0.026	99.02	98.50	1.50
NT	0.011	100.00	99.89	0.11	0.025	99.99	99.98	0.02
All	0.013	99.91	98.23	1.77	0.025	99.67	99.50	0.50

Table 4. Success Rate of Robust White-box attack on XceptionNet and MesoNet. Acc-C denotes the accuracy of the detector on compressed adversarial videos.

Table 4 shows the results of our robust white-box attack. It can be seen that robust white-box is effective in both Raw and MJPEG formats. The average distortion between original and adversarial frames in the robust attack is higher as compared to the non-robust white-box attack. We achieve an average success rate (SR-C) of 98.07% and 99.83% for XceptionNet and MesoNet respectively in the compressed video format. Additionally, to assess the gain obtained by incorporating the transformation functions, we compare the robust white-box attack against the non-robust white-box attack at the same level of distortion in Table 5. We observe a significant improvement in attack success rate on compressed videos (SR-C) when using the robust attack as opposed to the simple white-box attack (84.96% vs 74.69% across all datasets at L_∞ norm of 0.008).

		White Box			*Robust White Box*		
Dataset	L_∞	**SR - U**	**SR - C**	**Acc-C%**	**SR - U**	**SR - C**	**Acc-C%**
DF	0.008	99.67	60.36	39.64	99.67	75.06	24.94
F2F	0.008	99.85	80.69	19.31	100.0	90.20	9.80
FS	0.008	100.00	59.63	40.37	100.0	76.12	23.88
NT	0.008	99.89	98.08	1.92	100.0	98.48	1.52
All	0.008	99.85	74.69	25.31	99.91	84.96	15.04

Table 5. Comparison of white-box and robust white-box attacks at the same magnitude of L_∞ norm of the adversarial perturbation. Acc-C denotes the accuracy of the detector on compressed adversarial videos.

We also study the effectiveness of our robust white box attack under different levels of compression in the H.264 format which is widely used for sharing videos over the internet. Figure 4 shows the average success rate of our attack across all datasets for different quantization parameter c used for saving the video in H.264 format. The higher the quantization factor, the higher is the compression level. In [40], fake videos are saved in HQ and LQ formats which use $c = 23$ and $c = 40$ respectively. It can be seen that even at very high compression levels ($c = 40$), our attack is able to achieve 80.39% and 90.50% attack success rate for XceptionNet and MesoNet respectively, without any additional hyper-parameter tuning for this experiment.

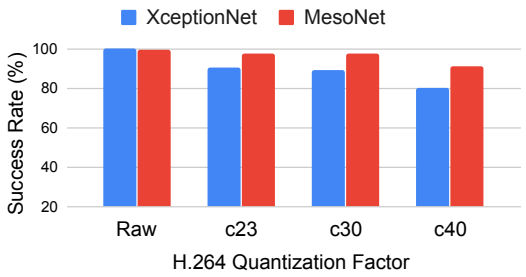

Figure 4. Attack success rate vs Quantization factor used for compression in H264 codec for robust white box attack.

4.2. Black-box Setting

We construct adversarial examples in the black-box setting using the methods described in Sections 3.5 and 3.6. The number of samples n in the search distribution for estimating gradients using NES is set to 20 for black-box attacks and 80 for robust black-box to account for sampling different transformation functions t_i. We set the maximum number of iterations to 100, learning rate α to $1/255$ and max L_∞ constraint ϵ to $16/255$.

Table 6 shows the results of our Black-box attack (Section 3.5) without robust transforms. Note that the average L_∞ norm of the perturbation across all datasets and models is higher than our white-box attacks. We are able to generate adversarial videos with an average success rate of 97.04% for XceptionNet and 86.70% for MesoNet when adversarial videos are saved in the Raw format. Similar to our observation in the white-box setting, the success rate drops significantly in the compressed format for this attack. The average number of queries to the victim model for each frame is 985 for this attack.

Robust Black-box: We perform robust black-box attack using the algorithm described in (Section 3.6). For simplicity, during the robust black-box attack we use the same hyper-parameters for creating a distribution of transformation functions T (Table 3) as those in our robust white-box attack. The average number of network queries for fooling each frame is 2153 for our robust black-box attack. Table 7 shows the results for our robust black-box attack. We observe a significant improvement in the attack success rate for XceptionNet when we save adversarial videos in the com-

	XceptionNet				*MesoNet*			
Dataset	L_∞	**SR - U**	**SR - C**	**Acc-C%**	L_∞	**SR - U**	**SR - C**	**Acc-C %**
DF	0.055	89.72	55.64	44.36	0.062	96.05	93.33	6.67
F2F	0.055	92.56	81.40	18.6	0.063	84.08	77.68	22.32
FS	0.045	96.77	23.50	76.5	0.063	77.55	62.44	37.56
NT	0.024	99.86	94.23	5.77	0.063	85.98	79.25	20.75
All	0.045	94.73	63.69	36.31	0.063	85.92	78.18	21.82

Table 6. Success Rate of of Black-box attack on XceptionNet and MesoNet. Acc-C denotes the accuracy of the detector on compressed adversarial videos.

	XceptionNet				*MesoNet*			
Dataset	L_∞	**SR - U**	**SR - C**	**Acc-C%**	L_∞	**SR - U**	**SR - C**	**Acc-C%**
DF	0.060	88.47	79.18	20.82	0.047	96.19	93.80	6.20
F2F	0.058	97.68	94.42	5.58	0.054	84.14	77.50	22.50
FS	0.052	98.97	63.26	36.74	0.061	77.34	61.77	38.23
NT	0.018	99.65	98.91	1.09	0.053	88.05	80.27	19.73
All	0.047	96.19	83.94	16.06	0.053	86.43	78.33	21.67

Table 7. Success Rate of Robust Black-box attack on XceptionNet and MesoNet. Acc-C denotes the accuracy of the detector on compressed adversarial videos.

pressed format as compared to that in the naive black-box attack setting. When attacking MesoNet in robust black-box setting, we do not observe a significant improvement even though overall success rate is higher when using robust transforms.

5. Evaluation on Sequence Based Detector

We consider the 3D CNN based detector described in Section 3.1. The detector performs 3D convolution on a sequence of face-crops from 7 consecutive frames. We perform our attacks on the pre-trained model checkpoint (trained on DFDC [17] train set) released by the NTech-Lab team [15]. We evaluate our attacks on the DeepFake videos from the DFDC public validation set which contains 200 Fake videos. We report the accuracy of the detector on the 7-frame sequences from this test set in the first row of Table 8.

Similar to our attacks on frame-by-frame detectors, in the white-box setting we back-propagate the loss through the entire model to obtain gradients with respect to the input frames for crafting the adversarial frames. While both white-box and robust white-box attacks achieve 100% success rate on uncompressed videos, the robust white-box attack performs significantly better on the compressed videos and is able to completely fool the detector. As compared to frame-by-frame detectors, a higher magnitude of perturbation is required to fool this sequence model in both the white-box attacks. In the black-box attack setting, while we achieve similar attack success rates on uncompressed videos as the frame-by-frame detectors, the attack success rate drops after compression. The robust black-box attack helps improve robustness of adversarial perturbations to compression as observed by higher success rates on compressed videos (51.02% vs 24.43% SR-C).

	3D CNN Sequence Model			
Attack Type	L_∞	**SR - U**	**SR - C**	**Acc. - C%**
None	-	-	-	91.74
White-Box	0.037	100.00	77.67	22.33
Robust White-Box	0.059	100.00	100.00	0.00
Black-Box	0.061	87.99	24.43	75.57
Robust Black-Box	0.062	88.21	51.02	48.98

Table 8. Evaluation of different attacks on a sequence based detector on the DFDC validation dataset. The first row indicates the performance of the classifier on benign (non adversarial) videos.

6. Discussion and Conclusion

The intent of Deepfake generation can be malicious and their detection is a security concern. Current works on DNN-based Deepfake detection assume a non-adaptive adversary whose aim is to fool the human-eye by generating a realistic fake video. To use these detectors in practice, we argue that it is essential to evaluate them against an adaptive adversary who is aware of the defense being present and is intentionally trying to fool the defense. In this paper, we show that the current state-of-the-art methods for Deepfake detection can be easily bypassed if the adversary has complete or even partial knowledge of the detector. Therefore, there is a need for developing provably robust detectors that are evaluated under different attack scenarios and attacker capabilities.

In order to use DNN based classifiers as detectors, ensuring robustness to adversarial examples is necessary but not sufficient. A well-equipped attacker may devise other methods to by-pass the detector: For example, an attacker can modify the training objective of the Deepfake generator to include a loss term corresponding to the detector score. Classifiers trained in a supervised manner on existing Deepfake generation methods, cannot be reliably secure against novel Deepfake generation methods not seen during training. We recommend approaches similar to Adversarial Training [22] to train robust Deepfake detectors. That is, during training, an adaptive adversary continues to generate novel Deepfakes that can bypass the current state of the detector and the detector continues improving in order to detect the new Deepfakes. In conclusion, we highlight that adversarial examples are a practical concern for current neural network based Deepfake detectors and therefore recommend future work on designing provably robust Deepfake detectors.

7. Acknowledgements

This work was supported by ARO under award number W911NF-19-1-0317 and SRC under Task ID: 2899.001.

References

[1] Darius Afchar, Vincent Nozick, Junichi Yamagishi, and Isao Echizen. Mesonet: a compact facial video forgery detection network. In *2018 IEEE International Workshop on Information Forensics and Security (WIFS)*. IEEE, 2018.

[2] Irene Amerini, Leonardo Galteri, Roberto Caldelli, and Alberto Del Bimbo. Deepfake video detection through optical flow based cnn. In *The IEEE International Conference on Computer Vision (ICCV) Workshops*, Oct 2019.

[3] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. *arXiv preprint arXiv:1802.00420*, 2018.

[4] Anish Athalye, Logan Engstrom, Andrew Ilyas, and Kevin Kwok. Synthesizing robust adversarial examples. In *Proceedings of the 35th International Conference on Machine Learning*, 2018.

[5] Jawadul H Bappy, Amit K Roy-Chowdhury, Jason Bunk, Lakshmanan Nataraj, and BS Manjunath. Exploiting spatial structure for localizing manipulated image regions. In *Proceedings of the IEEE international conference on computer vision*, pages 4970–4979, 2017.

[6] Jawadul H Bappy, Cody Simons, Lakshmanan Nataraj, BS Manjunath, and Amit K Roy-Chowdhury. Hybrid lstm and encoder–decoder architecture for detection of image forgeries. *IEEE Transactions on Image Processing*, 28(7):3286–3300, 2019.

[7] Mauro Barni, Matthew C Stamm, and Benedetta Tondi. Adversarial multimedia forensics: Overview and challenges ahead. In *2018 26th European Signal Processing Conference (EUSIPCO)*, pages 962–966. IEEE, 2018.

[8] Yonatan Belinkov and Yonatan Bisk. Synthetic and natural noise both break neural machine translation. In *International Conference on Learning Representations*, 2018.

[9] Rainer Böhme and Matthias Kirchner. Counter-forensics: Attacking image forensics. In *Digital image forensics*, pages 327–366. Springer, 2013.

[10] R Bohme and M Kirchner. Digital image forensics: There is more to a picture than meets the eye, chapter counter-forensics: Attacking image forensics, 2013.

[11] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (sp)*, pages 39–57. IEEE, 2017.

[12] Nicholas Carlini and David Wagner. Audio adversarial examples: Targeted attacks on speech-to-text. In *2018 IEEE Security and Privacy Workshops (SPW)*. IEEE, 2018.

[13] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1251–1258, 2017.

[14] Nilaksh Das, Madhuri Shanbhogue, Shang-Tse Chen, Fred Hohman, Li Chen, Michael E Kounavis, and Duen Horng Chau. Keeping the bad guys out: Protecting and vaccinating deep learning with jpeg compression. *arXiv preprint arXiv:1705.02900*, 2017.

[15] Azat Davletshin. https://github.com/ntech-lab/deepfake-detection-challenge.

[16] DeepFakes. https://github.com/deepfakes/faceswap.

[17] Brian Dolhansky, Russ Howes, Ben Pflaum, Nicole Baram, and Cristian Canton Ferrer. The deepfake detection challenge (dfdc) dataset. *arXiv preprint arXiv:2006.07397*, 2020.

[18] Amanda Duarte, Francisco Roldan, Miquel Tubau, Janna Escur, Santiago Pascual, Amaia Salvador, Eva Mohedano, Kevin McGuinness, Jordi Torres, and Xavier Giro-i Nieto. Wav2pix: speech-conditioned face generation using generative adversarial networks. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 3, 2019.

[19] Gintare Karolina Dziugaite, Zoubin Ghahramani, and Daniel M Roy. A study of the effect of jpg compression on adversarial images. *arXiv preprint arXiv:1608.00853*, 2016.

[20] Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. Hotflip: White-box adversarial examples for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2018.

[21] Hany Farid. *Photo Forensics*. The MIT Press, 2016.

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

[23] 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.

[24] Chuan Guo, Mayank Rana, Moustapha Cisse, and Laurens Van Der Maaten. Countering adversarial images using input transformations. *arXiv preprint arXiv:1711.00117*, 2017.

[25] Andrew Ilyas, Logan Engstrom, Anish Athalye, and Jessy Lin. Black-box adversarial attacks with limited queries and information. In *International Conference on Machine Learning*, pages 2137–2146, 2018.

[26] Marek Kowalski. Faceswap https://github.com/marekkowalski/faceswap/.

[27] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*, 2016.

[28] 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.

[29] Yuezun Li, Ming-Ching Chang, and Siwei Lyu. In ictu oculi: Exposing ai created fake videos by detecting eye blinking. In *2018 IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 1–7. IEEE, 2018.

[30] Yuezun Li and Siwei Lyu. Exposing deepfake videos by detecting face warping artifacts. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 46–52, 2019.

[31] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Universal adversarial perturbations. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[32] Paarth Neekhara, Shehzeen Hussain, Shlomo Dubnov, and Farinaz Koushanfar. Adversarial reprogramming of text classification neural networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*

and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). Association for Computational Linguistics, 2019.

[33] Paarth Neekhara, Shehzeen Hussain, Prakhar Pandey, Shlomo Dubnov, Julian McAuley, and Farinaz Koushanfar. Universal adversarial perturbations for speech recognition systems. In *Proc. Interspeech 2019*, 2019.

[34] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*. ACM, 2017.

[35] Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z Berkay Celik, and Ananthram Swami. The limitations of deep learning in adversarial settings. In *2016 IEEE European Symposium on Security and Privacy (EuroS&P)*. IEEE, 2016.

[36] Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In *2016 IEEE Symposium on Security and Privacy (SP)*, pages 582–597. IEEE, 2016.

[37] Yao Qin, Nicholas Carlini, Garrison Cottrell, Ian Goodfellow, and Colin Raffel. Imperceptible, robust, and targeted adversarial examples for automatic speech recognition. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 5231–5240, Long Beach, California, USA, 09–15 Jun 2019. PMLR.

[38] Ramachandra Raghavendra, Kiran B Raja, Sushma Venkatesh, and Christoph Busch. Transferable deep-cnn features for detecting digital and print-scanned morphed face images. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1822–1830. IEEE, 2017.

[39] Nicolas Rahmouni, Vincent Nozick, Junichi Yamagishi, and Isao Echizen. Distinguishing computer graphics from natural images using convolution neural networks. In *2017 IEEE Workshop on Information Forensics and Security (WIFS)*, pages 1–6. IEEE, 2017.

[40] Andreas Rossler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Niessner. Faceforensics++: Learning to detect manipulated facial images. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.

[41] Ekraam Sabir, Jiaxin Cheng, Ayush Jaiswal, Wael AbdAlmageed, Iacopo Masi, and Prem Natarajan. Recurrent convolutional strategies for face manipulation detection in videos. *Interfaces (GUI)*, 3:1, 2019.

[42] Tim Salimans, Jonathan Ho, Xi Chen, Szymon Sidor, and Ilya Sutskever. Evolution Strategies as a Scalable Alternative to Reinforcement Learning, Sept. 2017.

[43] Yucheng Shi, Siyu Wang, and Yahong Han. Curls & whey: Boosting black-box adversarial attacks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[44] Dawn Song, Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Florian Tramèr, Atul Prakash, and Tadayoshi Kohno. Physical adversarial examples for object detectors. In *12th USENIX Workshop on Offensive Technologies (WOOT 18)*. USENIX Association, 2018.

[45] Supasorn Suwajanakorn, Steven M Seitz, and Ira Kemelmacher-Shlizerman. Synthesizing obama: learning lip sync from audio. *ACM Transactions on Graphics (TOG)*, 2017.

[46] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations*, 2014.

[47] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, pages 6105–6114, 2019.

[48] 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.

[49] Justus Thies, Michael Zollhofer, Marc Stamminger, Christian Theobalt, and Matthias Nießner. Face2face: Real-time face capture and reenactment of rgb videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[50] Konstantinos Vougioukas, Stavros Petridis, and Maja Pantic. Realistic speech-driven facial animation with gans. *International Journal of Computer Vision*, pages 1–16, 2019.

[51] Weihong Wang and Hany Farid. Exposing digital forgeries in interlaced and deinterlaced video. *IEEE Transactions on Information Forensics and Security*, 2(3):438–449, 2007.

[52] Daan Wierstra, Tom Schaul, Tobias Glasmachers, Yi Sun, Jan Peters, and Jürgen Schmidhuber. Natural evolution strategies. *J. Mach. Learn. Res.*, 15(1):949–980, Jan. 2014.

[53] Cihang Xie, Jianyu Wang, Zhishuai Zhang, Zhou Ren, and Alan Yuille. Mitigating adversarial effects through randomization. In *International Conference on Learning Representations*, 2018.

[54] 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.

[55] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, 2016.

[56] Peng Zhou, Xintong Han, Vlad I Morariu, and Larry S Davis. Two-stream neural networks for tampered face detection. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1831–1839. IEEE, 2017.