

# Adversarial Deepfakes: Evaluating Vulnerability of Deepfake Detectors to Adversarial Examples

Paarth Neekhara<sup>\*1</sup> Shehzeen Hussain<sup>\*2</sup> Malhar Jere<sup>2</sup> Farinaz Koushanfar<sup>2</sup> Julian McAuley<sup>1</sup>

## 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 (Thies et al., 2016), Neural Textures (Thies et al., 2019) and FaceSwap (Kowalski) 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 (Suwajanakorn et al., 2017). 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 (Rossler et al., 2019) detector. Top: Frames 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) (Dolhansky et al., 2019; Rossler et al., 2019; Afchar et al., 2018; Amerini et al., 2019; Li & Lyu, 2019; Rahmouni et al., 2017). 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 (Afchar et al., 2018; Chollet, 2017).

While the above pipeline achieves promising results in accurately detecting manipulated videos, in this paper we demonstrate that it is susceptible to *adversarial examples* which can fool the detector to classify fake videos as real<sup>1</sup>. An adversarial example is an intentionally perturbed input

<sup>\*</sup>Equal contribution <sup>1</sup>Department of Computer Science, UC San Diego <sup>2</sup>Department of Electrical and Computer Engineering, UC San Diego. Correspondence to: Shehzeen Hussain <ssh028@eng.ucsd.edu>, Paarth Neekhara <pneekhar@eng.ucsd.edu>.

<sup>1</sup>Video Examples: <https://adversarialdeepfakes.github.io/>

that can fool a victim classification model (Szegedy et al., 2014). 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 have 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 the following methods:

**FaceSwap (FS):** FaceSwap (Kowalski) is a classical computer graphics-based approach for face replacement in videos. In this method, sparse facial landmarks are detected to extract the face region in an image. These landmarks are then used to fit a 3D template model which is back-projected onto the target image by minimizing the distance between the projected shape and localized landmarks. Finally, the rendered model is blended with the image and color correction is applied.

**Face2Face (F2F):** Face2Face (Thies et al., 2016) is a facial reenactment system that transfers the expressions of a person in a source video to another person in a target video, while maintaining the identity of the target person. In this method, faces are compressed into a low-dimensional expression space, where expressions can be easily transferred from the source to the target.

**DeepFakes (DF):** While the term Deepfakes has commonly been used in mainstream media as a blanket term for deep-learning based face replacement, it is also the name of a specific manipulation (DeepFakes) method that was spread via online forums. In the learning phase, two auto-encoders with a shared encoder are trained to reconstruct the images of source and target face. To create a fake image, the encoded source image is passed as input to the target image decoder.

**NeuralTextures (NT):** NeuralTextures (Thies et al., 2019) is a Generative Adversarial Network (GAN) based facial reenactment technique. In this method, a generative model is trained to learn the neural texture of a target person using original video data. The GAN objective is a combination of adversarial and photometric reconstruction loss.

The authors of (Rossler et al., 2019) curated a dataset of manipulated videos named FaceForensics++ using the above methods. While our proposed attacks are generally applicable to CNN detectors for any kind of forgery in images and videos, we perform our experiments on FaceForensics++ dataset containing videos with only facial forgery.

### 2.2. Detecting Manipulated Videos

Traditionally, multimedia forensics investigated the authenticity of images (Wang & Farid, 2007; Bohme & Kirchner, 2013; Farid, 2016) 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 Barni et al. (2018); Böhme & Kirchner (2013) 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, in order to detect facial tampering resulting from face swapping (Zhou et al., 2017), face morphing (Raghavendra et al., 2017) and splicing attacks (Bappy et al., 2017; 2019). Other works (Li et al., 2018; Afchar et al., 2018; Güera & Delp, 2018; Rossler et al., 2019; Sabir et al., 2019) have focused on detecting face manipulation artifacts resulting from Deepfake generation methods. Li et al. (2018) reported that eye blinking is not well reproduced in fake videos, and therefore proposed a temporal approach using CNN/RNN based model to detect a lack of eye blinking when exposing deepfakes. Similarly, Yang et al. (2019) 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 (Vougioukas et al., 2019; Duarte et al., 2019).

The Deepfake detectors proposed in Rossler et al. (2019); Afchar et al. (2018); Dolhansky et al. (2019) model Deepfake detection as a per-frame binary classification problem. In our work, we focus on exposing 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 (Thies et al., 2016) 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 (Rossler et al., 2019) 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 (Rossler et al., 2019) were both CNN based models: XceptionNet and MesoNet. XceptionNet has also been reported as the best performing classifier for Deepfake detection in the Deepfake Detection Challenge (Dolhansky et al., 2019) paper.

**XceptionNet** (Chollet, 2017), first proposed in 2017, is a modification to the Inception architecture to include depth-wise separable convolutions and residual connections, which allow for faster convolutional operations and improve computational efficiency. In order to detect deepfakes, (Rossler et al., 2019) utilized transfer learning on XceptionNet and retrained it on forged images. The authors demonstrated that XceptionNet can outperform several alternative classifiers in detecting forgeries in both uncompressed and compressed videos, and identifying forged regions in them. The accuracy of this model on the FaceForensics++ Dataset (Rossler et al., 2019) is reported in Table 1.

**MesoNet** consists of two fast neural network architectures, Meso-4 and MesoInception-4, and was first proposed in Afchar et al. (2018) to capture mesoscopic properties of images. Meso-4 consists of 4 convolutional layers followed by Dropout and a binary sigmoid classification and consists of 27,977 trainable parameters, while MesoInception-4 replaces the first 2 convolutional layers of the Meso-4 architecture with a variant of the Inception module inspired by the Inception (Szegedy et al., 2015) architecture. The accuracy of this model on the FaceForensics++ Dataset (Rossler et al., 2019) is reported in Table 1.

### 2.3. Adversarial Examples

Adversarial examples are intentionally designed inputs to a machine learning (ML) model that cause the model to make a mistake (Szegedy et al., 2014). Prior work has shown a series of first-order gradient-based attacks to be fairly effective in fooling DNN based models in both image (Papernot et al., 2016a; 2017; Goodfellow et al., 2015;

Moosavi-Dezfooli et al., 2017; Carlini & Wagner, 2017; Shi et al., 2019) and audio (Carlini & Wagner, 2018; Qin et al., 2019; Neekhara et al., 2019) 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 (Xie et al., 2018) 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 (Athalye et al., 2018a;b) 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

In this work, we demonstrate the effectiveness of our attack on Deepfake detectors which rely on 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 (Thies et al., 2016) 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 (Chollet, 2017) and MesoNet (Afchar et al., 2018).

### 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 (Goodfellow et al., 2015) recommend constraining the maximum distortion of any individual pixel by a given threshold  $\epsilon$ , i.e., constraining the perturbation using an  $L_\infty$  metric. Additionally, *Fast Gradient Sign Method* (FGSM) (Goodfellow et al., 2015) based attacks, which are optimized for the  $L_\infty$  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

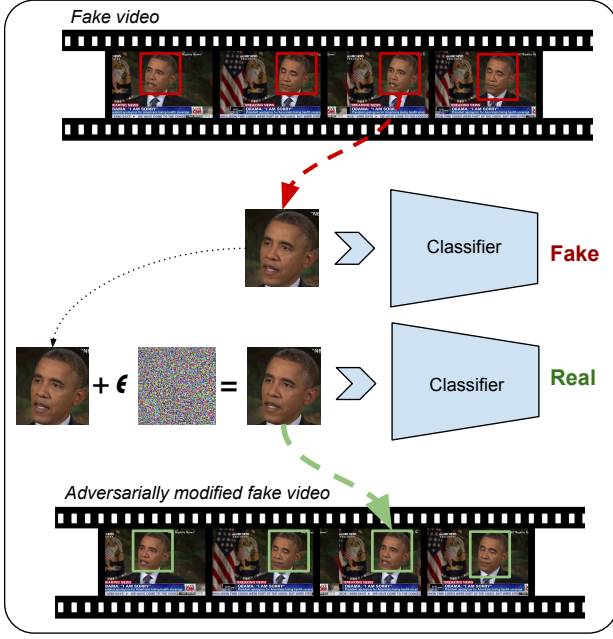


Figure 2. An overview of our attack pipeline to generate Adversarial Deepfakes. We generate an adversarial example for each frame in the given fake video and combine them together to create an adversarially modified fake video.

used in practice. Therefore, in this work, we use the  $L_\infty$  distortion metric for constraining our adversarial perturbation and optimize for it using gradient sign based methods.

**Notation:** We follow the notation previously used in Carlini & Wagner (2017); Papernot et al. (2016b): 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

$$F(x) = \text{softmax}(Z(x)) = y$$

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,

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

**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 (Kurakin et al., 2016) to optimize the following loss function:

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

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 by Carlini & Wagner (2017) 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:

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

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 (Dziugaite et al., 2016; Das et al., 2017; Guo et al., 2017). In order 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 (Athalye et al., 2018b). Given a distribution of input transformations  $T$ ,



input image  $x$ , and target class  $y$ , our objective is as follows:

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

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:

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

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

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

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. Empirically, we find that ensuring robustness of adversarial examples to these transforms ensures robustness to JPEG compression. Intuitively, image translation transform ensures robustness to different crops of the face, while gaussian blur, gaussian noise and resizing ensure robustness to JPEG artifacts.

### 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 (Wierstra et al., 2014; Ilyas et al., 2018). 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:

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

This allows efficient gradient estimation in fewer queries as compared to finite-difference methods. From (Wierstra et al., 2014), we know the gradient of expectation can be derived as follows:

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

Similar to (Ilyas et al., 2018; Wierstra et al., 2014), 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  $\delta_i$  similar to (Salimans et al., 2017; Ilyas et al., 2018). That is, instead of generating  $n$  values  $\delta \sim \mathcal{N}(0, I)$ , we sample Gaussian noise for  $i \in \{1, \dots, \frac{n}{2}\}$  and set  $\delta_j = -\delta_{n-j+1}$  for  $j \in \{(\frac{n}{2} + 1), \dots, n\}$ . This optimization has been empirically shown to improve performance of NES. Algorithm 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:

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

### 3.6. Robust Black-box Attack

In order 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:

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

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  $\delta_i$ :

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

---

**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  $\sigma$ , number of samples  $n$ , image dimensionality  $N$   
 $g \leftarrow \mathbf{0}_n$   
**for**  $i = 1$  **to**  $n$  **do**  
     $t_i \sim T$   
     $u_i \leftarrow \mathcal{N}(\mathbf{0}_N, \mathbf{I}_{N \cdot N})$   
     $g \leftarrow g + F(t_i(x + \sigma \cdot u_i))_y \cdot u_i$   
     $g \leftarrow g - F(t_i(x - \sigma \cdot u_i))_y \cdot u_i$   
**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 (Chollet, 2017) and MesoNet (Afchar et al., 2018). In our experiments, we perform our attack on the test set of the FaceForensics++ Dataset (Rossler et al., 2019), 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<sup>2</sup>.

	DF	F2F	FS	NT
<b>XceptionNet</b>	97.49	97.69	96.79	92.19
<b>MesoNet</b>	89.55	88.6	81.24	76.62

Table 1. Accuracy of Deepfake detectors on the FaceForensics++ HQ Dataset as reported in Rossler et al. (2019). 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.

**Mean distortion ( $L_\infty$ ):** The average  $L_\infty$  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  $L_\infty$  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

<sup>2</sup>Code released upon publication

that can be placed back in the frame. We set the maximum number of iterations to 100, learning rate  $\alpha$  to  $1/255$  and max  $L_\infty$  constraint  $\epsilon$  to  $16/255$  for both our attack methods described in Sections 3.3 and 3.4.

Dataset	XceptionNet			MesoNet		
	$L_\infty$	SR - U	SR - C	$L_\infty$	SR - U	SR - C
DF	0.004	99.67	43.11	0.006	97.30	92.27
F2F	0.004	99.85	52.50	0.007	98.94	96.30
FS	0.004	100.00	43.13	0.009	97.12	86.10
NT	0.004	99.89	95.10	0.007	99.22	96.20
All	0.004	99.85	58.46	0.007	98.15	92.72

Table 2. Success Rate of White-box attack on XceptionNet and MesoNet. We report the average  $L_\infty$  distortion between the adversarial and original frames and the attack success rate on uncompressed (SR-U) and compressed (SR-C) 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 (Dziugaite et al., 2016; Das et al., 2017; Guo et al., 2017) that employ JPEG compression and image transformations to defend against adversarial examples.

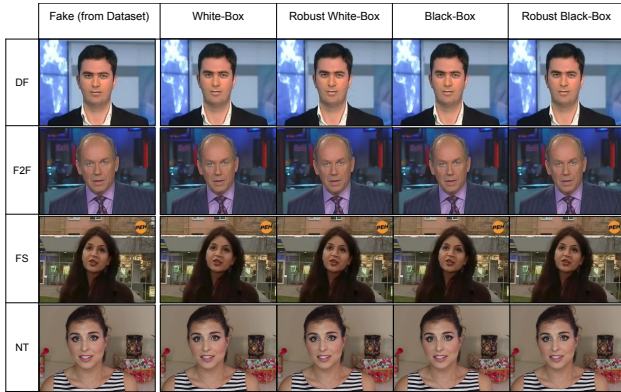


Figure 3. Randomly selected frames of Adversarial Deepfakes from successful attacks. The frame from the dataset in the first column is correctly identified as *Fake* by the detectors, while the corresponding frames generated by each of our attacks are labelled as *Real* with a probability of 1. Video 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 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 $k(d, d, \sigma)$ , $d \sim \mathcal{U}[3, 7]$ , $\sigma \sim \mathcal{U}[5, 10]$ $\sigma \sim \mathcal{U}[0.01, 0.02]$ $d_x \sim \mathcal{U}[-20, 20]$ , $d_y \sim \mathcal{U}[-20, 20]$ Scaling factor $r \sim \mathcal{U}[2, 5]$
Gaussian Noise	
Translation	
Down-sizing & Up-sizing	

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.

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.

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 Rossler et al. (2019), 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.

Dataset	XceptionNet			MesoNet		
	$L_\infty$	SR - U	SR - C	$L_\infty$	SR - U	SR - C
DF	0.016	99.56	98.71	0.030	99.94	99.85
F2F	0.013	100.00	99.00	0.020	99.71	99.67
FS	0.013	100.00	95.33	0.026	99.02	98.50
NT	0.011	100.00	99.89	0.025	99.99	99.98
All	0.013	99.89	98.23	0.025	99.67	99.50

Table 4. Success Rate of Robust White-box attack on XceptionNet and MesoNet.

## 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  $\alpha$  to  $1/255$  and max  $L_\infty$  constraint  $\epsilon$  to  $16/255$ .

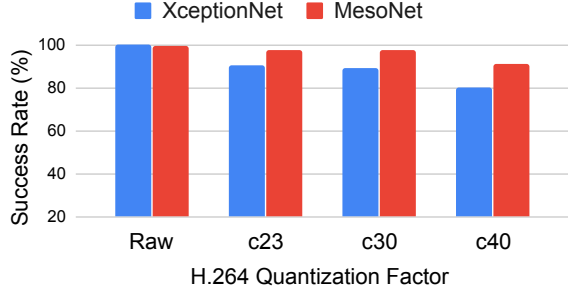


Figure 4. Attack success rate vs Quantization factor used for compression in H264 codec for robust white box attack.

Table 5 shows the results of our Black-box attack (Section 3.5) without robust transforms. Note that the average  $L_\infty$  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 6 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 compressed 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.

Dataset	XceptionNet			MesoNet		
	$L_\infty$	SR - U	SR - C	$L_\infty$	SR - U	SR - C
DF	0.055	89.72	55.64	0.062	96.05	93.33
F2F	0.055	92.56	81.40	0.0627	84.08	77.68
FS	0.045	96.77	23.50	0.0627	77.55	62.44
NT	0.024	99.86	94.23	0.0627	85.98	79.25
All	0.045	94.73	63.69	0.0626	85.92	78.18

Table 5. Success Rate of Black-box attack on XceptionNet and MesoNet.

Dataset	XceptionNet			MesoNet		
	$L_\infty$	SR - U	SR - C	$L_\infty$	SR - U	SR - C
DF	0.060	88.47	79.18	0.047	96.19	93.80
F2F	0.058	97.68	94.42	0.054	84.14	77.50
FS	0.052	98.97	63.26	0.061	77.34	61.77
NT	0.018	99.65	98.91	0.053	88.05	80.27
All	0.047	96.19	83.94	0.053	86.43	78.33

Table 6. Success Rate of Robust Black-box attack on XceptionNet and MesoNet.

## 5. Discussion and Conclusion

The intent of Deepfake generation can be malicious and their detection is a security concern. Current works on CNN-based Deepfake detection assume a non-adaptive adversary whose aim is to fool the human-eye by generating a realistic fake video. In order 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, 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 (Goodfellow et al., 2015) 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 CNN based Deepfake detectors and therefore recommend future work on provably robust Deepfake detectors.

## References

- Afchar, D., Nozick, V., Yamagishi, J., and Echizen, I. Mesonet: a compact facial video forgery detection network. In *2018 IEEE International Workshop on Information Forensics and Security (WIFS)*. IEEE, 2018.



- Amerini, I., Galteri, L., Caldelli, R., and Del Bimbo, A. Deepfake video detection through optical flow based cnn. In *The IEEE International Conference on Computer Vision (ICCV) Workshops*, Oct 2019.
- Athalye, A., Carlini, N., and Wagner, D. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. *arXiv preprint arXiv:1802.00420*, 2018a.
- Athalye, A., Engstrom, L., Ilyas, A., and Kwok, K. Synthesizing robust adversarial examples. In *Proceedings of the 35th International Conference on Machine Learning*, 2018b.
- Bappy, J. H., Roy-Chowdhury, A. K., Bunk, J., Nataraj, L., and Manjunath, B. Exploiting spatial structure for localizing manipulated image regions. In *Proceedings of the IEEE international conference on computer vision*, pp. 4970–4979, 2017.
- Bappy, J. H., Simons, C., Nataraj, L., Manjunath, B., and Roy-Chowdhury, A. K. Hybrid lstm and encoder–decoder architecture for detection of image forgeries. *IEEE Transactions on Image Processing*, 28(7):3286–3300, 2019.
- Barni, M., Stamm, M. C., and Tondi, B. Adversarial multimedia forensics: Overview and challenges ahead. In *2018 26th European Signal Processing Conference (EUSIPCO)*, pp. 962–966. IEEE, 2018.
- Böhme, R. and Kirchner, M. Counter-forensics: Attacking image forensics. In *Digital image forensics*, pp. 327–366. Springer, 2013.
- Bohme, R. and Kirchner, M. Digital image forensics: There is more to a picture than meets the eye, chapter counter-forensics: Attacking image forensics, 2013.
- Carlini, N. and Wagner, D. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (sp)*, pp. 39–57. IEEE, 2017.
- Carlini, N. and Wagner, D. Audio adversarial examples: Targeted attacks on speech-to-text. In *2018 IEEE Security and Privacy Workshops (SPW)*. IEEE, 2018.
- Chollet, F. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1251–1258, 2017.
- Das, N., Shanbhogue, M., Chen, S.-T., Hohman, F., Chen, L., Kounavis, M. E., and Chau, D. H. Keeping the bad guys out: Protecting and vaccinating deep learning with jpeg compression. *arXiv preprint arXiv:1705.02900*, 2017.
- DeepFakes. Deepfakes. URL <https://github.com/deepfakes/faceswap>.
- Dolhansky, B., Howes, R., Pflaum, B., Baram, N., and Ferrer, C. C. The deepfake detection challenge (dfdc) preview dataset. *arXiv preprint arXiv:1910.08854*, 2019.
- Duarte, A., Roldan, F., Tubau, M., Escur, J., Pascual, S., Salvador, A., Mohedano, E., McGuinness, K., Torres, J., and Giro-i Nieto, X. Wav2pix: speech-conditioned face generation using generative adversarial networks. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 3, 2019.
- Dziugaite, G. K., Ghahramani, Z., and Roy, D. M. A study of the effect of jpg compression on adversarial images. *arXiv preprint arXiv:1608.00853*, 2016.
- Farid, H. *Photo Forensics*. The MIT Press, 2016. ISBN 0262035340.
- Goodfellow, I. J., Shlens, J., and Szegedy, C. Explaining and harnessing adversarial examples. *stat*, 2015.
- Güera, D. and Delp, E. J. Deepfake video detection using recurrent neural networks. In *2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pp. 1–6. IEEE, 2018.
- Guo, C., Rana, M., Cisse, M., and Van Der Maaten, L. Countering adversarial images using input transformations. *arXiv preprint arXiv:1711.00117*, 2017.
- Ilyas, A., Engstrom, L., Athalye, A., and Lin, J. Black-box adversarial attacks with limited queries and information. In *International Conference on Machine Learning*, pp. 2137–2146, 2018.
- Kowalski, M. Faceswap. URL <https://github.com/MarekKowalski/FaceSwap/>.
- Kurakin, A., Goodfellow, I., and Bengio, S. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*, 2016.
- Li, Y. and Lyu, S. Exposing deepfake videos by detecting face warping artifacts. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 46–52, 2019.
- Li, Y., Chang, M.-C., and Lyu, S. In icu oculi: Exposing ai created fake videos by detecting eye blinking. In *2018 IEEE International Workshop on Information Forensics and Security (WIFS)*, pp. 1–7. IEEE, 2018.
- Moosavi-Dezfooli, S.-M., Fawzi, A., Fawzi, O., and Frossard, P. Universal adversarial perturbations. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

- Neekhara, P., Hussain, S., Pandey, P., Dubnov, S., McAuley, J., and Koushanfar, F. Universal adversarial perturbations for speech recognition systems. In *Proc. Interspeech 2019*, 2019.
- Papernot, N., McDaniel, P., Jha, S., Fredrikson, M., Celik, Z. B., and Swami, A. The limitations of deep learning in adversarial settings. In *2016 IEEE European Symposium on Security and Privacy (EuroS&P)*. IEEE, 2016a.
- Papernot, N., McDaniel, P., Wu, X., Jha, S., and Swami, A. Distillation as a defense to adversarial perturbations against deep neural networks. In *2016 IEEE Symposium on Security and Privacy (SP)*, pp. 582–597. IEEE, 2016b.
- Papernot, N., McDaniel, P., Goodfellow, I., Jha, S., Celik, Z. B., and Swami, A. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*. ACM, 2017.
- Qin, Y., Carlini, N., Cottrell, G., Goodfellow, I., and Raffel, C. Imperceptible, robust, and targeted adversarial examples for automatic speech recognition. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 5231–5240, Long Beach, California, USA, 09–15 Jun 2019. PMLR. URL <http://proceedings.mlr.press/v97/qin19a.html>.
- Raghavendra, R., Raja, K. B., Venkatesh, S., and Busch, C. 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)*, pp. 1822–1830. IEEE, 2017.
- Rahmouni, N., Nozick, V., Yamagishi, J., and Echizen, I. Distinguishing computer graphics from natural images using convolution neural networks. In *2017 IEEE Workshop on Information Forensics and Security (WIFS)*, pp. 1–6. IEEE, 2017.
- Rossler, A., Cozzolino, D., Verdoliva, L., Riess, C., Thies, J., and Niessner, M. Faceforensics++: Learning to detect manipulated facial images. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.
- Sabir, E., Cheng, J., Jaiswal, A., AbdAlmageed, W., Masi, I., and Natarajan, P. Recurrent convolutional strategies for face manipulation detection in videos. *Interfaces (GUI)*, 3:1, 2019.
- Salimans, T., Ho, J., Chen, X., Sidor, S., and Sutskever, I. Evolution Strategies as a Scalable Alternative to Reinforcement Learning, September 2017. URL <http://arxiv.org/abs/1703.03864>.
- Shi, Y., Wang, S., and Han, Y. Curls & whey: Boosting black-box adversarial attacks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.
- Suwajanakorn, S., Seitz, S. M., and Kemelmacher-Shlizerman, I. Synthesizing obama: learning lip sync from audio. *ACM Transactions on Graphics (TOG)*, 2017.
- Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. Intriguing properties of neural networks. In *International Conference on Learning Representations*, 2014.
- Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9, 2015.
- Thies, J., Zollhofer, M., Stamminger, M., Theobalt, C., and Nießner, M. Face2face: Real-time face capture and reenactment of rgb videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.
- Thies, J., Zollhöfer, M., and Nießner, M. Deferred neural rendering: Image synthesis using neural textures. *ACM Transactions on Graphics (TOG)*, 38(4):1–12, 2019.
- Vougioukas, K., Petridis, S., and Pantic, M. Realistic speech-driven facial animation with gans. *International Journal of Computer Vision*, pp. 1–16, 2019.
- Wang, W. and Farid, H. Exposing digital forgeries in interlaced and deinterlaced video. *IEEE Transactions on Information Forensics and Security*, 2(3):438–449, 2007.
- Wierstra, D., Schaul, T., Glasmachers, T., Sun, Y., Peters, J., and Schmidhuber, J. Natural evolution strategies. *J. Mach. Learn. Res.*, 15(1):949–980, January 2014. ISSN 1532-4435.
- Xie, C., Wang, J., Zhang, Z., Ren, Z., and Yuille, A. Mitigating adversarial effects through randomization. In *International Conference on Learning Representations*, 2018.
- Yang, X., Li, Y., and Lyu, S. Exposing deep fakes using inconsistent head poses. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 8261–8265. IEEE, 2019.
- Zhou, P., Han, X., Morariu, V. I., and Davis, L. S. Two-stream neural networks for tampered face detection. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 1831–1839. IEEE, 2017.