Exposing Vulnerabilities of Deepfake Detection Systems with Robust Attacks

SHEHZEEN HUSSAIN* and PAARTH NEEKHARA*, University of California, San Diego, USA
BRIAN DOLHANSKY, Facebook AI, USA
JOANNA BITTON, Facebook AI, USA
CRISTIAN CANTON FERRER, Facebook AI, USA
JULIAN MCAULEY, University of California, San Diego, USA
FARINAZ KOUSHANFAR, University of California, San Diego, USA

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. Finally, we study the extent to which adversarial perturbations transfer across different Deepfake detectors and create more accessible attacks using universal adversarial perturbations that pose a very feasible attack scenario since they can be easily shared amongst attackers. [1].

CCS Concepts: • **Computing methodologies** → **Computer vision**; • **Security and privacy** → **Social aspects of security and privacy**; • **Human-centered computing** → *Collaborative and social computing theory, concepts and paradigms*; • **Applied computing** → *Computer forensics.*

Additional Key Words and Phrases: misinformation detection, neural networks, adversarial machine learning, digital threats, fake news, fake news detection, news verification

ACM Reference Format:
Shehzeen Hussain, Paarth Neekhara, Brian Dolhansky, Joanna Bitton, Cristian Canton Ferrer, Julian McAuley, and Farinaz Koushanfar. 2021. Exposing Vulnerabilities of Deepfake Detection Systems with Robust Attacks. *Digit. Threat. Res. Pract.* 1, 1, Article 1 (April 2021), 24 pages. https://doi.org/10.1145/3464307

[*] Both authors contributed equally to this research.
[1] Video Examples: https://adversarialdeepfakes.github.io/

Authors' addresses: Shehzeen Hussain, ssh028@eng.ucsd.edu; Paarth Neekhara, pneekhar@eng.ucsd.edu, University of California, San Diego, San Diego, USA, 92122; Brian Dolhansky, Facebook AI, Seattle, USA, bdol@fb.com; Joanna Bitton, Facebook AI, Seattle, USA, jbitton@fb.com; Cristian Canton Ferrer, Facebook AI, Seattle, USA, ccanton@fb.com; Julian McAuley, University of California, San Diego, San Diego, USA; Farinaz Koushanfar, University of California, San Diego, San Diego, USA.

2576-5337/2021/4-ART1 $15.00
https://doi.org/10.1145/3464307

1 INTRODUCTION

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. With the advent of sophisticated image and video synthesis techniques, it has become increasingly easier to generate high-quality convincing fake videos. Video manipulation methods like Face2Face [67], Neural Textures [66] and FaceSwap [38] 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 has led to advances in research on synthetic video generation for movies, storytelling, and modern-day streaming services. However, they can also be used maliciously to spread misinformation, harass individuals or defame famous personalities [63]. These videos are now an emerging threat, especially within the realms of politics and misinformation. Deepfakes have been used to create fake news aggravating political and religious tensions, with the aim to influence results in election campaigns [31, 36, 49]. Such extensive spread of fake videos through social media platforms has raised significant concerns worldwide, particularly hampering the credibility of digital media. Recent research has found evidence that widespread misinformation not only misleads individuals and reduces public trust on digital media but also leads to increased cynicism within democratic societies [68].

While there are plenty of potential good uses for deepfakes, its misuse has been found in several cases of defamation or impersonation. Moreover, the fact that there are plenty of free and easy to use applications that allow users to swap their faces into movie clips and videos has accelerated the widespread usage of deepfake to create new content thus increasing the prospects for abuse of this technology.

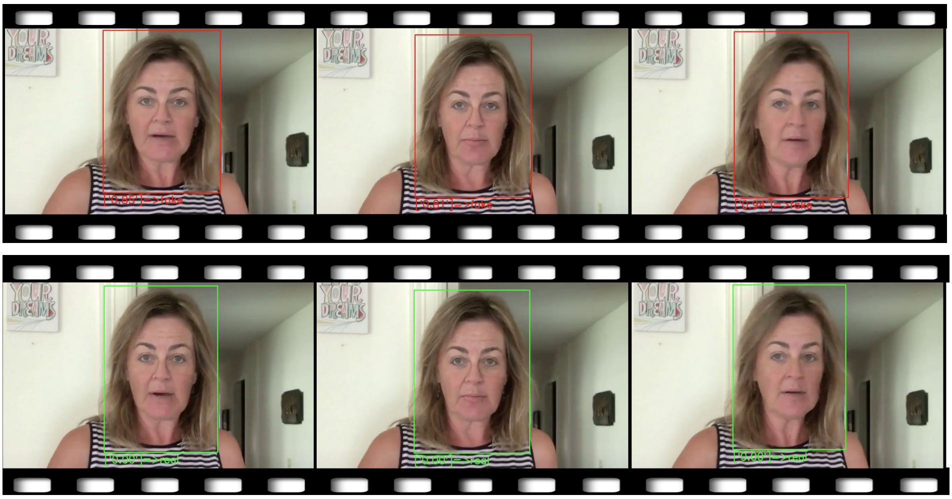


Fig. 1. Adversarial Deepfakes for an EfficientNet [65] based Deepfake detector. Top: Frames of of a *fake* video generated by Face2Face being correctly identified as *fake* by the detector. Bottom: Corresponding frames of the adversarially modified fake video being classified as real by the detector.

To address the threats imposed by Deepfakes, the machine learning community has proposed several countermeasures to identify forgeries in digital media [69]. The state-of-the-art methods for detecting manipulated facial content in videos rely on Convolutional Neural Networks (CNNs) [1, 2, 21, 43, 56, 57]. Deepfake detection is typically modeled as a per-frame classification problem.

Additionally, the best performing models employ a face-tracking method following which the cropped face from a frame is passed on to a CNN-based classifier for classification as real or fake [1, 15, 33, 60]. Some of the recent Deepfake detection methods also use models that operate on a sequence of frames as opposed to a single frame to exploit temporal dependencies in videos [17].

While the above neural network based detection methods achieve promising results in accurately detecting manipulated videos, in this paper we examine their vulnerabilities to *adversarial examples*. An adversarial example is an intentionally perturbed input that can fool a victim classification model [64]. Even though several works have demonstrated that neural networks are vulnerable to adversarial inputs (Section 2.4), we want to explicitly raise this issue that has been ignored by existing works on Deepfake detection (Section 2.2). Deepfakes have the potential to be very damaging, therefore attacks designed to evade Deepfake detectors can cause immense harm when compared to other attack scenarios. In addition, as Deepfakes are rare compared to the set of all videos, detection of Deepfakes is already an extremely difficult problem.

To this end, we quantitatively assess the vulnerability of Deepfake detectors to adversarial examples in different threat scenarios. Assuming a complete access (white-box) threat scenario, we find that it is trivial to bypass a Deepfake detector with an imperceptible adversarial modification to a given video. However, in a practical threat scenario the attacker may not have knowledge of the victim detection model and parameters. Next we assume a more challenging threat scenario in which the attacker can only query a victim model to get the detection scores for a video frame. Even in this attack scenario, we find that it is possible to bypass the detector with a slightly higher amount of adversarial perturbation. Additionally, to ensure the adversarial videos remain effective even after video compression, we incorporate expectation over input transforms [4] while training the adversarial perturbation to craft robust adversarial videos.

While the above attacks can effectively bypass Deepfake detectors, they can be easily thwarted by the service provider. Detection models and parameters can be kept private to prevent the white-box attack and query access can be limited to prevent the black-box attack. Adversarial examples pose a practical threat to Deepfake detection if they are transferable across different detection methods. That is, if adversarial videos designed to fool some open source Deepfake detection method can also reliably fool other unseen CNN-based detection methods, this would pose a real security threat to deploying CNN-based detectors in production. We experimentally demonstrate that it is possible to design highly transferable adversarial examples by ensuring robustness to input-transformation functions while training the perturbation. Finally, we design more accessible adversarial attacks by creating transferable universal adversarial perturbations that can be universally added across all frames of all videos to reliably fool a number of Deepfake detection methods.

A preliminary version of this work will appear in the conference: IEEE WACV, 2021 [34]. In this work we extend our previous version of the paper in the following ways:

- We study adversarial threats to Deepfake detection from a practical perspective. First, we address explainability of Deepfake detectors and provide insightful visualizations to explain how the decisions are made by a detector (Section 2.3). We perform a transferability study (Section 5.2) of adversarial perturbations, and report that our robust attack (Section 3.4) results in highly transferable adversarial examples that can bypass unseen detection models thereby posing a feasible threat in a black-box scenario.
- We propose a more accessible attack on Deepfake detectors using universal adversarial perturbations (Section 3.7). Our algorithm finds a single universal perturbation that can be added to all frames of a given video and fool multiple seen and unseen detectors to a significant extent (Section 5.3).

- We extend our attacks to new models and the DeepFake Detection Challange (DFDC) dataset (Section 4, 5.2, 5.3). We attack the state-of-the-art detectors from the DFDC hosted by Facebook, Inc in 2020. The three detectors we choose to evaluate are the top three winners of the challenge.

2 BACKGROUND

2.1 Generating Deepfakes

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. To aid research on Deepfake detection, there have been efforts in curating datasets of real and deepfake videos. In our work, we generate adversarial examples for videos in the FaceForensics++[2] [57] and DeepFake Detection Challenge (DFDC) Dataset [21]. The FaceForensics++ dataset employs the following four Deepfake generation techniques:
1. FaceSwap (FS): FaceSwap [38] 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.
2. Face2Face (F2F): Face2Face [67] 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.
3. DeepFakes (DF): While the term 'Deepfake' 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 [18] 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.
4. NeuralTextures (NT): NeuralTextures [66] 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 an adversarial and photometric reconstruction loss.

Aside from the FaceForensics++ dataset, another prominent collection of Deepfake videos was released by Facebook, Inc in 2019. To the best of our knowledge, this recently developed DeepFake Detection Challenge (DFDC) dataset [20, 21] is the largest collection of real and Deepfake videos, consisting of over one million training clips of face swaps produced with a variety of methods. For synthesizing the fake videos in the DFDC dataset, 8 different video manipulation techniques were used, many of which are CNN-based techniques. These methods include the traditional Deepfake auto-encoder architecture, a non-learned morphable mask face swap algorithm, and several Generative Adversarial Networks (GAN) techniques like Neural Talking Heads [76], FSGAN [50] and StyleGAN [37]. In conjunction with the dataset, a corresponding competition[3] was launched in

[2] Handling, analysis and processing of data related to the FaceForensics++ dataset was conducted by the two first authors (S.Hussain and P.Neekhara) while at University of California in San Diego. Facebook researchers were not involved with the FaceForensics++ dataset in any extent.

[3] https://www.kaggle.com/c/deepfake-detection-challenge

which competitors were encouraged to submit models trained for Deepfake detection on the training set. These models were then ranked on a hidden, held-out test set, and the winning competitors released their architectures and training strategies publicly.

2.2 Detecting Manipulated Videos

Traditionally, multimedia forensics investigated the authenticity of images [11, 27, 71] 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, 10] 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 [79], face morphing [55] and splicing attacks [5, 6]. Other works [1, 30, 41, 42, 57, 58] have focused on detecting face manipulation artifacts resulting from Deepfake generation methods. The authors of [42] 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, [75] 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 [24, 70].

The Deepfake detectors proposed in [1, 21, 57] model Deepfake detection as a per-frame binary classification problem. The authors of [57] demonstrated that XceptionNet can outperform several alternative classifiers in detecting forgeries in both uncompressed and compressed videos, and identifying forged regions in them. Since the task is to specifically detect facial manipulation, these models incorporate domain knowledge by using a face tracking method [67] to track the face in the video. The face is then cropped from the original frame and fed as input to a classification model to be labelled as real or fake. Experimentally, the authors of [57] 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 [57] were both CNN based models: XceptionNet [15] and MesoNet [1].

Some detectors have also focused on exploiting temporal dependencies for detecting Deepfake videos. Such detectors work on sequence of frames as opposed to a single frame using a CNN + RNN model or a 3D CNN model. One such model based on a 3D EfficientNet [65] architecture, was used by the third place winner [17] of DFDC in addition to a per-frame classification model. For completeness, we also evaluate our attacks against the 3D CNN model to expose the vulnerability of temporal Deepfake detectors. While intuitively, exploiting temporal dependencies using sequence models should improve a detector's ability to detect manipulated videos, the insights from the results of the DFDC challenge [21] show that the best performing models operate on a frame level. In fact, the winning team [60] of the DFDC challenge explicitly noted that other ideas besides frame-by-frame detection did not improve their performance on the public leaderboard. The first two winning submissions were both CNN based per-frame classification models similar to the ones described above.

2.3 Explainability of Deepfake Detectors

It is important to gain insight about what the detector is looking at when it makes a decision about a video being Real or Fake. This is typically done by obtaining the gradient of the score of the predicted class with respect to the input image and plotting the magnitude of these gradients as a heat-map. Back-propagating gradients naively does not result in very interpretable visualizations. This is because it is more important to consider pixels which activate a neuron and do not suppress it (suppression is indicated by negative gradients). Therefore, we use guided back-propagation which defines custom gradient estimates for activation functions like ReLU and suppresses negative gradients during the backward pass. We then standardize the gradient obtained with respect to the input and overlay the heat-map on the frame to visualize the areas of an image that trigger the network's output. Figure 2 shows some examples of the saliency maps obtained while analyzing two different detectors on Deepfake videos.

Our initial observations on these saliency maps suggest that different CNN based detection methods attend to similar aspects of the input frame for predicting the label. These aspects include the edges of the face, the eyes, lips, teeth etc. These similarities across different detection methods indicate that adversarially modifying such aspects of the image could potentially fool multiple detection methods. We validate this hypothesis in our work by studying the transferability of adversarial examples (Section 5.2) across different detection methods and proposing techniques (Section 3.4) that improve the transferability.

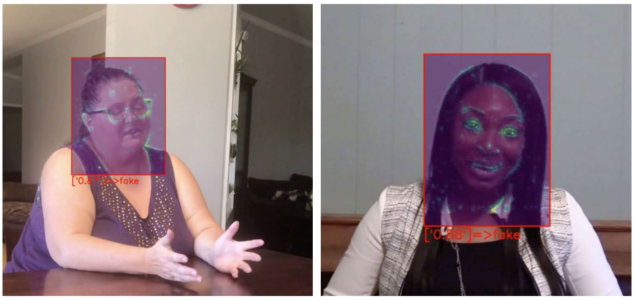


Fig. 2. Gradient saliency maps obtained on Deepfake videos using guideded-backpropogation on a CNN-based detector [60]. The highlighted areas indicate the image regions that strongly influence the detector's predictions.

2.4 Adversarial Examples

Adversarial examples are intentionally designed inputs to a machine learning (ML) model that cause the model to make a mistake [64]. Prior work has shown a series of first-order gradient-based attacks to be fairly effective in fooling DNN based models in images [13, 28, 45, 51, 52, 61, 62], audio [14, 48, 54] and text [9, 26, 47] 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. One type of adversarial perturbation known as universal adversarial perturbations (UAPs), requires relatively low computing power during the attacks because the same perturbation can be added to any input image to cause mis-classification by a victim model [45]. UAPs have also been reported to show excellent transferability on attacking different models rather than the target in various data domains [8, 45, 46, 48]. An adversarial attack typically involves several attack parameters, such as the number of steps for iterative attack [22], the distance metric to quantify the perturbation magnitude, the bound used to clip the perturbation, and the weight ratio of two losses (adversarial goal and perturbation bound). The effectiveness of an adversarial attack method can be quantified by the *'attack success rate'* metric. Attack success rate is typically defined as the probability that an image with the crafted adversarial perturbation is predicted as the attack target class (for targeted attacks) or as the non-original class (for untargeted attacks). Prior work on defenses [73] 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

In this section, we propose threat models for Deepfake detectors in various attack settings assuming different attacker capabilities. We first describe the victim Deepfake detectors we wish to attack (Section 3.1). Then we mathematically define the threat model and attack goal (Section 3.2). Next, we propose a white-box attack to achieve the attack goal in a scenario when the attacker has complete access to the victim model architecture and parameters (Section 3.3). In our experiments, we find that while the simple white-box attack works well on uncompressed videos, the attack success rate drops significantly on compressed videos. Another challenge in the simple white-box attack is the limited transferability of the attack to unseen models. We tackle these two challenges using our robust and transferable attack which poses a real world threat — the adversarial videos are more robust to compression and can also fool unseen detectors to a significant extent thereby posing a real-world threat (Section 3.4). Next we propose query based black-box attacks which do not require access to any surrogate model but only require query access to the model scores (Section 3.5, 3.6). Finally, we propose a highly accessible attack using universal adversarial perturbations — we find that it is possible to craft a single input-agnostic perturbation that can be added across all frames of any given video to cause classification to the target label by many seen and unseen detectors. Once crafted, this perturbation can be easily shared amongst adversaries thereby posing a very practical challenge to Deepfake detection (Section 3.7).

3.1 Victim Models: Deepfake Detectors

3.1.1 ***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 [67] 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 [15] 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 [21, 57, 77]. The accuracy of such models on the FaceForensics++ Dataset [57] is reported in Table 1.

3.1.2 ***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 5.4, we evaluate our attacks against one such detection method [17] that uses a 3D EfficientNet [65] CNN model for classifying a sequence of face-crops obtained from a face tracking model. In this model, a 3D 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

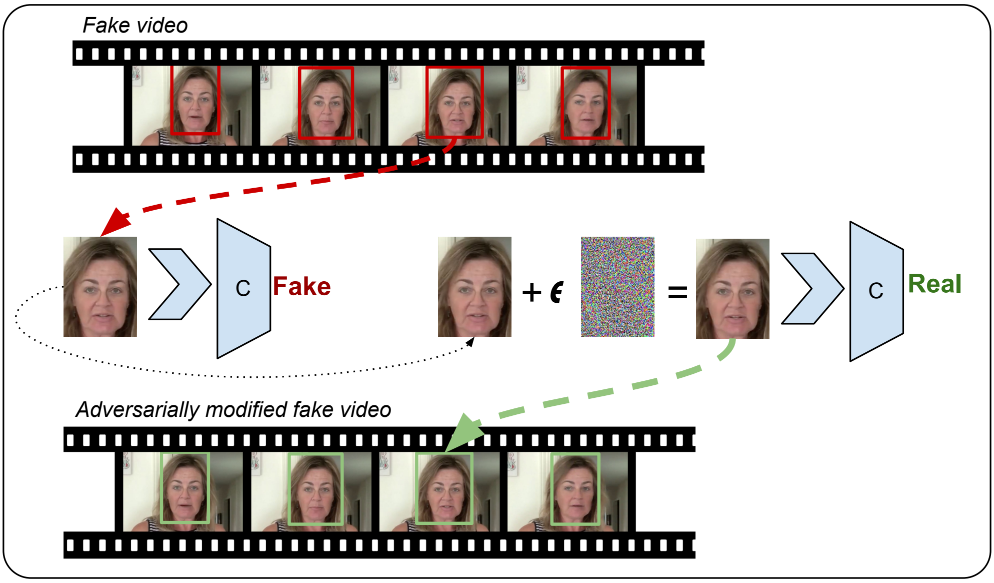


Fig. 3. An overview of our attack pipeline to generate Adversarial Deepfakes. We generate an adversarial example for each frame in the given fake video and combine them together to create an adversarially modified fake video.

between frames is 1/15th of a second. This 3D CNN model was used by the third place winner of the recently conducted DFDC challenge.

3.2 Threat Model

Given a video (*Real* or *Fake*), our task is to adversarially modify the video such that the label predicted by a victim Deepfake detection method is incorrect. That is, we want to modify the videos such that the *Fake* videos are classified as *Real* and vice-versa. Misclassifying a *Fake* video as *Real* can be used by the adversary to propagate false information. Misclassifying a *Real* video as *Fake* can be used by the adversary to cover up an event that did actually happen.

3.2.1 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 [28] 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) [28] 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.

3.2.2 Notation. We follow the notation previously used in [13, 53]; we 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 (that is z are the logits). That is:

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

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

3.2.3 Problem Formulation. Mathematically, for each video frame x, we aim to find an adversarial frame x_{adv} such that:

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

where y is the target label. In our case the target label is *Real* for *Fake* videos and *Fake* for *Real* videos. In the upcoming sections, we study this attack goal in various attacker knowledge settings and constraints.

3.2.4 Attack Pipeline. An overview of the process of generating adversarial fake videos is depicted in Figure 3. 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 Simple White-box attack

In this setting, we assume that the attacker has complete access to the detection 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 [39] to optimize the following objective:

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

Here, $Z(x)_y$ is the final score for target label y and $Z(x)_o$ is the score of the original label o before the softmax operation in the classifier C. The loss function we use is recommended by [13] 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 loss function 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 of maximum iterations, whichever occurs earlier. We solve the optimization problem for each frame of the given video and combine all the adversarial frames together to generate the adversarial video. In our experiments, we demonstrate that we are able to successfully fool all the detection methods studied in our work in the white-box attack setting using the above attack. However, the transferability of adversarial examples generated using this attack across different methods is limited. In the next section we propose techniques to overcome this challenge.

3.4 Robust and Transferable attack

Generally, videos uploaded to social networks and other media sharing websites are compressed. Standard operations like compression and resizing are known to remove adversarial perturbations from an image [16, 25, 32]. To ensure that the adversarial videos remain effective even after compression, it is important to ensure robustness to input-transformation functions while training the perturbation.

Also, past works [22, 23, 44, 51, 74, 80] have studied that adversarial inputs can transfer across different models. That is, an adversarial input that was designed to fool a particular victim model can possibly fool other models that were trained for the same task. This is because different models learn similar decision boundaries and therefore have similar vulnerabilities. However, for Deepfake detectors, the goal of making transferable adversarial videos is more challenging due to multiple steps involved in the Deepfake detection pipeline and the differences in these steps across various methods.

- Different face detection methods result in different face-crops.
- Different data-augmentation procedures during training result in different levels of robustness to adversarial examples.
- Different input pre-processing pipelines, such as image resizing, cropping and channel normalization parameters vary across different detection methods.

Therefore ensuring robustness to input transformation functions not only helps create adversarial videos that are robust to compression, but can also potentially result in adversarial videos that are transferable across different detection methods. We use the expectation over transforms [4] attack to craft robust and transferable adversarial examples. 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:

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

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

$$loss(x) = \frac{1}{n} \sum_{t_i \sim T} [max(Z(t_i(x))_o - Z(t_i(x))_y, 0)] \tag{3}$$

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 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 5.1.2.

3.5 Query based 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 [35, 72]. 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 [72], 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 [35, 72], 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 [35, 59]. 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 the 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 the target class:

$$x_i = x_{i-1} + \text{clip}_\epsilon(\alpha \cdot \text{sign}(\nabla F(x_{i-1})_y)) \tag{4}$$

3.6 Query based Robust Black-box Attack

To ensure robustness of adversarial videos to compression, we incorporate the 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 $\sigma$, number of samples $n$, image dimensionality $N$
$g \leftarrow 0_n$
**for** $i = 1$ **to** $n$ **do**
  $t_i \sim T$
  $u_i \leftarrow \mathcal{N}(0_N, I_{N \cdot N})$
  $g \leftarrow g + F(t_i(x + \sigma \cdot u_i))_y \cdot u_i$
  $g \leftarrow g - F(t_i(x - \sigma \cdot u_i))_y \cdot u_i$
**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 4 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.

3.7 Universal attack

While the transferability of adversarial perturbations poses a practical threat to Deepfake detectors in production, creating an adversarial video requires significant technical expertise in adversarial machine learning — the attacker needs to solve an optimization problem for each frame of the video to fool the detector.

To ease the process of fooling Deepfake detectors, we aim to design more accessible adversarial attacks that can be easily shared amongst attackers. Past works [8, 45, 48] have shown the existence of universal adversarial perturbations that can fool classification models in various input domains. We aim to find a single universal adversarial perturbation which when added across all frames of any video, will cause the victim Deepfake Detector to classify the video to a target label.

That is, we aim to find a targeted universal perturbation δ such that:

$$\begin{aligned} &C(x + \delta) = y \quad s.t \quad ||\delta||_\infty < \epsilon \\ &\text{for “most” } x \text{ in our dataset} \end{aligned} \tag{5}$$

where y is the target class. We train separate perturbations for Real and Fake target labels. In order to ensure robustness to differences across detection methods, we incorporate the transformation functions described in Section 3.4. We train the universal adversarial perturbation on a dataset of videos that are labelled opposite from our target label. On this dataset of videos, we aim to maximize the log-likelihood of predicting our target label y. Additionally to ensure the imperceptibility of the adversarial perturbation we penalize the L_2 distortion of the perturbation by adding a regularization term in our objective. Thus, our final objective to train a universal perturbation for a target label y

is as follows:

$$\begin{aligned} &Minimize \sum_{x \text{ in } D} \mathbb{E}_{t \sim T}[L(F(t(x+\delta)), y)] + c||\delta||_2 \\ &\qquad\qquad such\ that \quad ||\delta||_\infty < \epsilon \end{aligned} \tag{6}$$

Here, L is the cross-entropy loss between the predictions and our target label, c is a hyper-parameter to control the regularization loss and x is an input frame of a video from our dataset D. Similar to Equation 3, we estimate the above expectation using n samples as follows:

$$\mathbb{E}_{t \sim T}[L(F(t(x+\delta)), y)] = \frac{1}{n} \sum_{t_i \sim T} [L(F(t_i(x+\delta)), y)] \tag{7}$$

To ensure the constraint $||\delta||_\infty < \epsilon$, we express δ as follows:

$$\delta = \epsilon \cdot tanh(p)$$

where p is a trainable unconstrained parameter having the same dimensions as δ. We fix the size of the perturbation vector p to be $3 \times 256 \times 256$ in our experiments, but resize the perturbation using bilinear interpolation to match the size of our input x. We iteratively optimize the objective given by Equation 6 using gradient descent. In our experiments, we find that targeting certain Deepfake detectors not only results in input-agnostic universal perturbations but also model-agnostic universal perturbations.

4 EXPERIMENTAL SETUP

We conduct our experiments on the FaceForensics++ [57] and the DFDC datasets [21] and choose the best performing models on these datasets as the victim models for our attacks. We first craft adversarial videos for the FaceForensics++ dataset and target the XceptionNet and MesoNet models which are the best reported architectures reported in the paper [57] introducing this dataset (Section 5.1). We use these two models as a test-bed to study the robustness of our attacks to video compression and demonstrate the using our robust attack helps significantly improve attack performance on compressed videos. Next we conduct the transferable and universal attack experiments on the DFDC dataset. We choose the models from top three winning entries in the DFDC Kaggle competition as the victim models for these experiments (Section 5.2, 5.3). Finally, we evaluate our attacks on a sequence based 3D CNN model to demonstrate that adversarial examples are a threat to not only frame by frame detectors but also sequence based models (Section 5.4).

4.1 Dataset and Models

On the FaceForensics++ dataset, XceptionNet [15] and MesoNet [1] CNN classifiers have been reported to achieve the best performance in the paper introducing the dataset [57]. For these two models, we perform our attack on the test set of the FaceForensics++ Dataset [57], 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.

For the DFDC dataset, we choose the top three winners of the challenge, which was hosted by Facebook on the Kaggle website. The top two winning entries of the challenge rely solely on face detection models and per-frame CNN classifiers similar to the best performing models on the FaceForensics++ dataset. The third place winner of the challenge uses a combination of per-frame classifiers and a 3D CNN based sequence model (Section 3.1). Table 2 lists the Deepfake detection

	DF	**F2F**	**FS**	**NT**
XceptionNet [57] Acc %	97.49	97.69	96.79	92.19
MesoNet [57] Acc %	89.55	88.6	81.24	76.62

Table 1. Accuracy of Deepfake detectors on the FaceForensics++ HQ Dataset as reported in [57]. 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).

Model	*Team Name*	*Classifier*	*Face detection*	*AUC*
EN-B7 Selim [60]	Selim	EfficientNet B7 [65]	MTCNN [78]	0.717
XN WM [33]	Team WM	XceptionNet [15]	RetinaFace [19]	0.724
EN-B3 WM [33]	Team WM	EfficientNet B3 [65]	RetinaFace [19]	0.724
EN-B7 NLab [17]	NTech Lab	EfficientNet B7 [65]	DSFD [40]	0.717

Table 2. Different Deepfake detection systems studied in our work with their respective classification models, face detection models and detection AUC scores on the DFDC test set.

methods studied in this work along with their respective CNN architectures used for classification and face detection. We use the DFDC dataset and these top three winning models as the test bed for evaluating the transferability of our attacks across different models. In our transferability experiments we use the terms *victim model* and *test model* and define them as:

- *Victim model:* The detection model that the attack/adversarial perturbation is trained on, in the complete-knowledge (white-box) attack scenario.
- *Test model:* The model on which we evaluate the attack. This can be the same as the victim model (white-box) or an unseen detection model (black-box).

We craft adversarial videos for the first 100 Fake and 100 Real videos in the public DFDC validation set [21]. These videos contain a total of 30,300 frames. The videos are recorded in various lighting and background conditions and include people with different skin-tones.

4.2 Evaluation Metrics

Once the adversarial frames are generated, we combine them and save the adversarial videos in the following formats:

- *Uncompressed (Raw):* Video is stored as a sequence of uncompressed images.
- *Compressed (MJPEG):* Video is saved as a sequence of JPEG compressed frames.
- *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*. 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. 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 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).

5 RESULTS

5.1 Evaluation on FaceForensics++ dataset

5.1.1 Simple white-box attack. 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			
Attack	Dataset	L_∞	SR - U	SR - C	Acc-C %	L_∞	SR - U	SR - C	Acc-C %
Simple White-box (Section 3.3)	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	7.53
Robust and Transferable (Section 3.4)	DF	0.016	99.56	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	99.89	98.23	1.77	0.025	99.67	99.50	0.50
Query based Black-box (Section 3.5)	DF	0.055	89.72	55.64	44.36	0.062	96.05	93.33	6.67
	F2F	0.055	92.56	81.40	18.60	0.0627	84.08	77.68	22.32
	FS	0.045	96.77	23.50	76.50	0.0627	77.55	62.44	37.56
	NT	0.024	99.86	94.23	5.77	0.0627	85.98	79.25	20.75
	All	0.045	94.73	63.69	36.31	0.0626	85.92	78.18	21.83
Query based Robust Black-box (Section 3.6)	DF	0.060	88.47	79.18	20.82	0.047	96.19	93.80	93.80
	F2F	0.058	97.68	94.42	5.58	0.054	84.14	77.50	77.50
	FS	0.052	98.97	63.26	36.74	0.061	77.34	61.77	61.77
	NT	0.018	99.65	98.91	1.09	0.053	88.05	80.27	80.27
	All	0.047	96.19	83.94	16.06	0.053	86.43	78.33	78.33

Table 3. Evaluation of various attacks on the two models XceptionNet and MesoNet on the FaceForensics++ dataset. 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 3 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 [16, 25, 32] that employ JPEG compression and image transformations to defend against adversarial examples.

5.1.2 *Robust attack.* 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 4 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 4. 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 3 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.

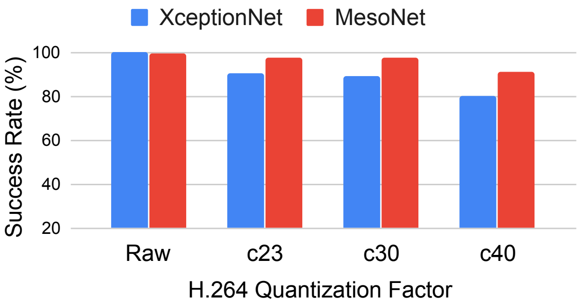


Fig. 4. Attack success rate vs Quantization factor used for compression in H264 codec for robust white box attack.

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 the compression level. In [57], 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 rates for XceptionNet and MesoNet respectively, without any additional hyper-parameter tuning for this experiment.

5.1.3 Query based black-box attack. We construct adversarial examples in the black-box setting using the methods described in Sections 3.5 and **??**. 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 3 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.

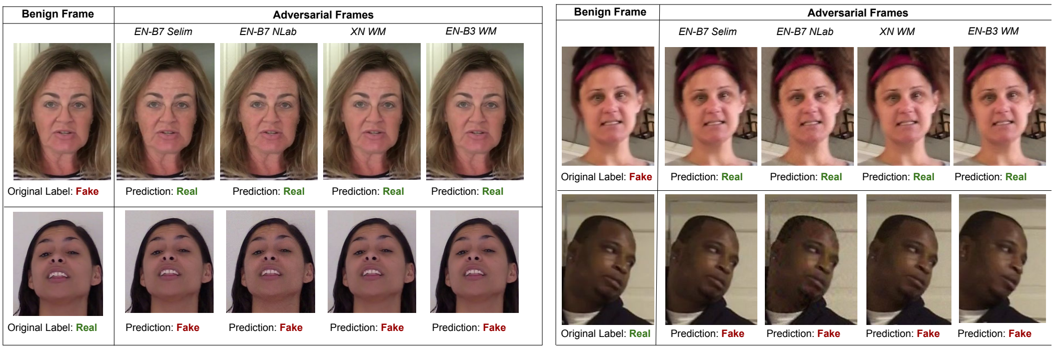
5.1.4 Query based Robust Black-box Attack: We perform robust black-box attack using the algorithm described in (Section 3.6). For simplicity, during the robust black-box attack we use the same hyper-parameters for creating a distribution of transformation functions T (Table 4) as those in our robust white-box attack. The average number of network queries for fooling each frame is 2153 for our robust black-box attack. Table 3 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 the overall success rate is higher when using robust transforms.

5.2 Transferability of adversarial attacks

We evaluate the transferability of adversarial perturbations across different detectors trained on the DFDC dataset. We train adversarial videos targeting a given victim model and test the videos against different test models. For our simple whitebox attack, while we achieve 100% attack success rate for the same test model as the victim model, the attack success rate drops significantly on alternate models. EfficientNet-B7 by NTech Lab requires the highest amount of adversarial perturbation under the L_∞ metric as compared to other methods in this study. We find that perturbations trained to fool EfficientNet-B7 by Team NTech Lab result in the most transferable attacks as indicated by the higher success rates on other test models. This suggests that *EN-B7 NLab* is relatively more robust to adversarial perturbations in comparison to the other models used in this study (also indicated by higher L_∞ perturbation required to fool *EN-B7 NLab*).

To improve the transferability of adversarial examples across different methods, we perform our robust transfer attack described in Section 3.4 and evaluate the adversarial videos against unseen detection methods in a black-box setting. The hyper-parameters of the transformation functions used for the attack have been provided in Table 4. All other attack hyper-parameters are kept the same as our simple white-box attack.

As indicated by the results in Table 5, we are able to significantly improve the transferability of adversarial perturbations across different detection methods as compared to our simple white-box attack. The adversarial perturbations are most transferable across models with the same architecture. For example, we are able to achieve high cross-transferability between *EN-B7 Selim* vs *EN-B7 NLab*. Similar to our observation in the previous section, attacking *EN-B7 NLab* results in the most transferable adversarial attacks - we are able to achieve at least 72% success rate across all other detection methods when attacking *EN-B7 NLab*.



Simple White-box attack examples on DFDC detectors

Robust and Transferable attack examples on DFDC detectors

Fig. 5. Randomly selected frames of adversarial videos from attacks on the DFDC detectors.

			Test Model			
	Victim Model	L_∞	EN-B7 Selim	EN-B7 NLab	XN WM	EN-B3 WM
Simple White-box (Section 3.3)	EN-B7 Selim	0.007	100.0 %	59.5 %	57.0 %	38.5 %
	EN-B7 NLab	0.013	94.0 %	100.0 %	66.5 %	49.5 %
	XN WM	0.006	13.0 %	12.5 %	100.0 %	12.0 %
	EN-B3 WM	0.005	21.0 %	15.5 %	22.0 %	100.0 %
Robust and Transferable (Section 3.4)	EN-B7 Selim	0.010	100.0 %	89.0 %	72.5 %	62.0 %
	EN-B7 NLab	0.018	99.0 %	100.0 %	72.0 %	76.5 %
	XN WM	0.018	49.0 %	33.5 %	100.0 %	46.0 %
	EN-B3 WM	0.008	46.5 %	35.0 %	47.5 %	100.0 %

Table 5. Attack success rates (SR-U) of the *white-box* (Section 3.3) and *robust and transferable attacks* (Section 3.4) on different victim models and their transferability to seen and unseen detectors (test models).

5.3 Universal attacks

To create more accessible attacks, we train a universal adversarial perturbation using the procedure described in Section 3.7. We set the L_2 regularization term $c = 0.01$ and use the Adam optimizer with a learning rate of 0.001. For our initial experiments, we set the L_∞ threshold $\epsilon = 40/255$ for all victim models. Since the goal of finding a single input-agnostic perturbation is more challenging than finding one perturbation per video frame, a higher amount of distortion is required for a successful attack as compared to the per-frame attacks described earlier. We train the universal perturbation on a dataset of 100 videos from the DFDC train set which are separate from our evaluation dataset. We train the perturbation using a batch size of 8 for $10,000$ iterations.

We target one victim model at a time and test the transferability of the universal perturbation on seen and unseen detectors. Table 6 presents the results of performing the universal attack on different victim models at $\epsilon = 40/255 = 0.156$. We are able to achieve 100% attack success rate on the same test model as the victim model using a single perturbation across all frames and videos of the same label. Also, the universal perturbation is transferable to a significant extent across different models which poses an extremely practical threat to Deepfake detectors in production.

		Test Model			
Victim Model	L_∞	EN-B7 Selim	EN-B7 NLab	XN WM	EN-B3 WM
EN-B7 Selim	0.156	100.0%	94.5%	65.0%	69.0%
EN-B7 NLab	0.156	94.5%	100.0%	75.0%	81.5%
XN WM	0.156	77.5%	61.0%	100.0%	20.0%
EN-B3 WM	0.156	66.5%	50.5%	60.0%	100.0%

Table 6. Attack success rates (SR-U) of the universal attacks (Section 3.7) on different victim models and their transferability to unseen detectors (test models).

Attacking *EN-B7 NLab* results in the most transferable perturbations where we are able to achieve at least a 75% success rate across all unseen detectors.

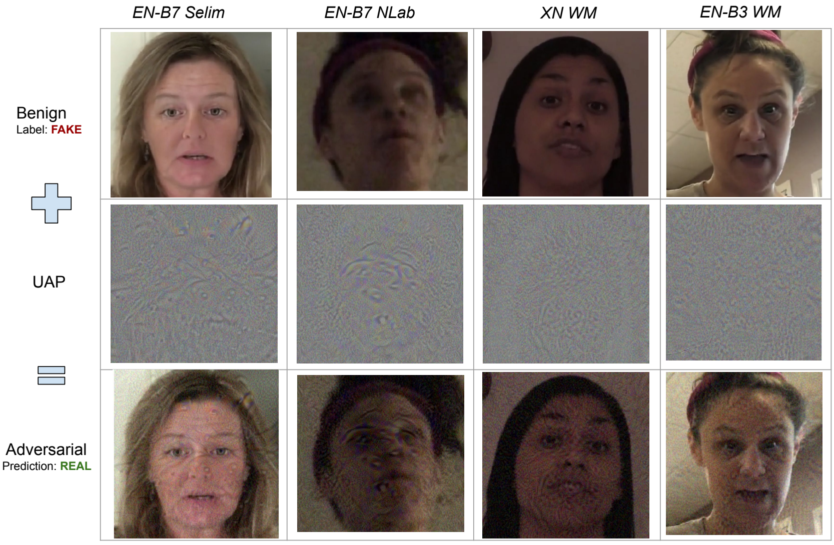


Fig. 6. Visualization of universal adversarial perturbations trained on different Deepfake detection models at $\epsilon = 0.156$.

Visually, the universal perturbations at $\epsilon = 0.156$ are more perceptible than our per-frame attacks discussed in the sections above. Figure 6 shows examples of universal adversarial perturbations trained on different Deepfake detectors and the resulting adversarial images obtained after adding the perturbation to the face-crop of the benign frame.

We perform an additional experiment to study the effectiveness of universal adversarial perturbations at different magnitudes of added perturbations. We choose *EN-B7 NLab* as the victim model and perform our universal attack at different values of ϵ. The attack success rates across different models are shown in Figure 7. Figure 7 also shows what a perturbed image looks like at different values of ϵ. At $\epsilon < 0.1$, the perturbation is fairly imperceptible but can still achieve high success rates on various test models.

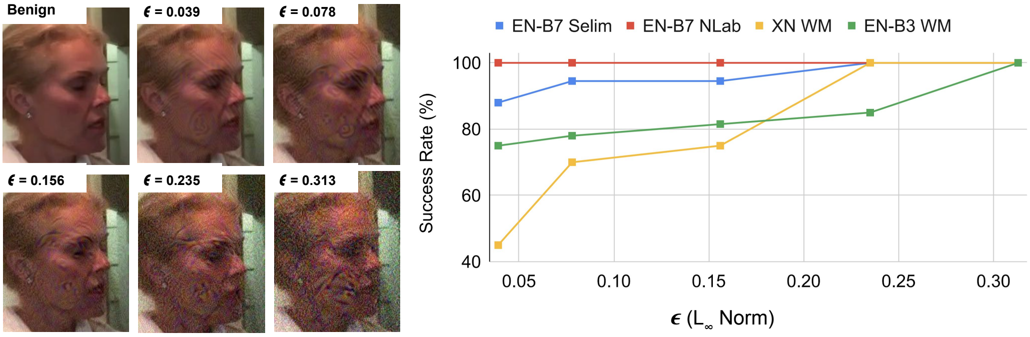


Fig. 7. *Left:* Visualization of the perturbed images using different magnitudes (ϵ) of universal adversarial perturbations trained on *EN-B7 NLab*. *Right:* Attack success rates of the universal attacks (Section 3.7) on different victim models and their transferability to unseen detectors (test models).

	3D CNN Sequence Model			
Attack Type	L_∞	**SR - U**	**SR - C**	**Acc. - C%**
None	-	-	-	91.74
Simple White-box (Section 3.3)	0.037	100.00	77.67	22.33
Robust and Transferable (Section 3.4)	0.059	100.00	100.00	0.00
Query based Black-box (Section 3.5)	0.061	87.99	24.43	75.57
Query based robust Black-box (Section 3.5)	0.062	88.21	51.02	48.98

Table 7. 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.

5.4 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 the DFDC [21] train set) released by the NTech-Lab team [17]. 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 7.

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).

6 ETHICS AND PRACTICAL IMPACT

The threat posed by Deepfake videos is already apparent - there are malicious users using such videos to defame famous personalities, spread disinformation, influence elections and polarize citizens. With more convincing and accessible Deepfake video synthesis techniques, this threat has become even bigger in magnitude. Since the intent of Deepfake generation can be malicious, their detection is a practical security concern. Deepfake synthesis and detection, closely follow the virus and anti-virus dynamic and is often considered an arms race with no end. It is important to thoroughly test the reliability of such detectors in practice since the task of Deepfake detection is inherently adversarial. 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. We demonstrate that adversarial videos designed using our robust attacks can significantly transfer across various detection models. This poses a practical security threat in a black-box setting since the attacker may attack a surrogate open-source model and use the adversarial videos to bypass alternate detection models in production. More alarmingly, we show that it is possible to craft universal adversarial perturbations which can be added to all the frames of any given video to fool a number of seen and unseen detection models. Deploying such attacks, therefore require minimal computational overhead and makes them more effective for real-time scenarios. Since such a perturbation can be easily shared amongst attackers, it makes these attacks more accessible and escalates the vulnerabilities of state-of-the-art Deepfake detection systems. We demonstrate that the current works on Deepfake detectors [1, 12, 21, 29, 57, 65] have been majorly ignoring the issue of adversarial inputs and can be fairly easily bypassed.

Our work therefore motivates the need to incorporate defenses to adversarial examples while training Deepfake detection systems. By actively incorporating adversarial images and videos while training the detection systems, it is possible to train more robust classification models. We recommend approaches similar to Adversarial Training [28] 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.

7 CONCLUSION

In this work, we study the vulnerabilities of various DNN based Deepfake systems. We consider both per-frame and sequence-based Deepfake detection models and demonstrate that they can be bypassed under various attack settings and attacker capabilities. We first design an attack pipeline to bypass Deepfake detectors in a white-box attack setting and propose techniques to increase the robustness of such attacks to video compression codecs. Next, we demonstrate that adversarial videos crafted using our robust attacks can fool alternate models to a significant extent thereby posing a real-world threat in a black-box attack setting. Finally, we demonstrate the existence of universal adversarial perturbations which pose a more practical threat since they can be easily shared amongst attackers and applied to any video in real-time. The threats proposed in this work motivate the need for training Deepfake detection systems that are robust to adversarial examples.

REFERENCES

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

[2] Irene Amerini, Leonardo Galteri, Roberto Caldelli, and Alberto Del Bimbo. 2019. Deepfake Video Detection through Optical Flow Based CNN. In *The IEEE International Conference on Computer Vision (ICCV) Workshops*.

[3] Anish Athalye, Nicholas Carlini, and David Wagner. 2018. 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. 2018. Synthesizing Robust Adversarial Examples. In *Proceedings of the 35th International Conference on Machine Learning.*

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

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

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

[8] Melika Behjati, Seyed-Mohsen Moosavi-Dezfooli, Mahdieh Soleymani Baghshah, and Pascal Frossard. 2019. Universal adversarial attacks on text classifiers. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP).* 7345–7349.

[9] Yonatan Belinkov and Yonatan Bisk. 2018. Synthetic and Natural Noise Both Break Neural Machine Translation. In *International Conference on Learning Representations.*

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

[11] R Bohme and M Kirchner. 2013. Digital Image Forensics: There is More to a Picture Than Meets the Eye, chapter Counter-forensics: Attacking Image Forensics.

[12] Nicolo Bonettini, Edoardo Daniele Cannas, Sara Mandelli, Luca Bondi, Paolo Bestagini, and Stefano Tubaro. 2020. Video face manipulation detection through ensemble of cnns. *arXiv preprint arXiv:2004.07676* (2020).

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

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

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

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

[17] Azat Davletshin. 2020. https://github.com/NTech-Lab/deepfake-detection-challenge.

[18] DeepFakes. 2017. https://github.com/deepfakes/faceswap.

[19] Jiankang Deng, Jia Guo, Evangelos Ververas, Irene Kotsia, and Stefanos Zafeiriou. [n.d.]. RetinaFace: Single-Shot Multi-Level Face Localisation in the Wild. In *Conference on Computer Vision and Pattern Recognition (CVPR).*

[20] Brian Dolhansky, Joanna Bitton, Ben Pflaum, Jikuo Lu, Russ Howes, Menglin Wang, and Cristian Canton Ferrer. 2020. The DeepFake Detection Challenge (DFDC) dataset. *arXiv preprint arXiv:2006.07397* (2020).

[21] Brian Dolhansky, Russ Howes, Ben Pflaum, Nicole Baram, and Cristian Canton Ferrer. 2020. The Deepfake Detection Challenge (DFDC) Dataset. *arXiv preprint arXiv:2006.07397* (2020).

[22] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. 2018. Boosting adversarial attacks with momentum. In *Proceedings of the IEEE conference on computer vision and pattern recognition.* 9185–9193.

[23] Yinpeng Dong, Tianyu Pang, Hang Su, and Jun Zhu. 2019. Evading defenses to transferable adversarial examples by translation-invariant attacks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.*

[24] Amanda Duarte, Francisco Roldan, Miquel Tubau, Janna Escur, Santiago Pascual, Amaia Salvador, Eva Mohedano, Kevin McGuinness, Jordi Torres, and Xavier Giro-i Nieto. 2019. Wav2Pix: speech-conditioned face generation using generative adversarial networks. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),* Vol. 3.

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

[26] Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. 2018. 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).*

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

[28] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and Harnessing Adversarial Examples. *stat* (2015).

[29] Luca Guarnera, Oliver Giudice, and Sebastiano Battiato. 2020. DeepFake Detection by Analyzing Convolutional Traces. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops.*

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

[31] Bin Guo, Yasan Ding, Lina Yao, Yunji Liang, and Zhiwen Yu. 2020. The Future of False Information Detection on Social Media: New Perspectives and Trends. *ACM Comput. Surv.* 53, 4 (2020). https://doi.org/10.1145/3393880

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

[33] Cui Hao. 2020. https://github.com/cuihaoleo/kaggle-dfdc.

[34] Shehzeen Hussain, Paarth Neekhara, Malhar Jere, Farinaz Koushanfar, and Julian McAuley. 2021. Adversarial Deepfakes: Evaluating Vulnerability of Deepfake Detectors to Adversarial Examples. *WACV 2021* (2021).

[35] Andrew Ilyas, Logan Engstrom, Anish Athalye, and Jessy Lin. 2018. Black-box Adversarial Attacks with Limited Queries and Information. In *International Conference on Machine Learning*. 2137–2146.

[36] Z. Jin, J. Cao, Han Guo, Yongdong Zhang, Y. Wang, and Jiebo Luo. 2017. Detection and Analysis of 2016 US Presidential Election Related Rumors on Twitter. In *SBP-BRiMS*.

[37] Tero Karras, Samuli Laine, and Timo Aila. 2019. A style-based generator architecture for generative adversarial networks. In *Conference on Computer Vision and Pattern Recognition (CVPR)*. 4401–4410.

[38] Marek Kowalski. 2018. FaceSwap https://github.com/MarekKowalski/FaceSwap/.

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

[40] Jian Li, Yabiao Wang, Changan Wang, Ying Tai, Jianjun Qian, Jian Yang, Chengjie Wang, Ji-Lin Li, and Feiyue Huang. [n.d.]. DSFD: Dual Shot Face Detector. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

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

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

[43] Yuezun Li and Siwei Lyu. 2019. Exposing DeepFake Videos By Detecting Face Warping Artifacts. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 46–52.

[44] Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. 2016. Delving into transferable adversarial examples and black-box attacks. *arXiv preprint arXiv:1611.02770* (2016).

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

[46] Konda Reddy Mopuri, Utsav Garg, and R Venkatesh Babu. 2017. Fast feature fool: A data independent approach to universal adversarial perturbations. *arXiv preprint arXiv:1707.05572* (2017).

[47] Paarth Neekhara, Shehzeen Hussain, Shlomo Dubnov, and Farinaz Koushanfar. 2019. 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. https://doi.org/10.18653/v1/D19-1525

[48] Paarth Neekhara, Shehzeen Hussain, Prakhar Pandey, Shlomo Dubnov, Julian McAuley, and Farinaz Koushanfar. 2019. Universal Adversarial Perturbations for Speech Recognition Systems. In *Proc. Interspeech 2019*.

[49] CBS News. 2019. *Doctored Nancy Pelosi video highlights threat of "deepfake" tech.* https://www.cbsnews.com/news/doctored-nancy-pelosi-video-highlights-threat-of-deepfake-tech-2019-05-25/

[50] Yuval Nirkin, Yosi Keller, and Tal Hassner. 2019. FSGAN: Subject agnostic face swapping and reenactment. In *Conference on Computer Vision and Pattern Recognition (CVPR)*. 7184–7193.

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

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

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

[54] Yao Qin, Nicholas Carlini, Garrison Cottrell, Ian Goodfellow, and Colin Raffel. 2019. Imperceptible, Robust, and Targeted Adversarial Examples for Automatic Speech Recognition. In *Proceedings of the 36th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 97)*, Kamalika Chaudhuri and Ruslan Salakhutdinov (Eds.). PMLR, Long Beach, California, USA, 5231–5240. http://proceedings.mlr.press/v97/qin19a.html

[55] Ramachandra Raghavendra, Kiran B Raja, Sushma Venkatesh, and Christoph Busch. 2017. 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)*. IEEE, 1822–1830.

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

[57] Andreas Rossler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Niessner. 2019. FaceForensics++: Learning to Detect Manipulated Facial Images. In *The IEEE International Conference on Computer Vision (ICCV)*.

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

[59] Tim Salimans, Jonathan Ho, Xi Chen, Szymon Sidor, and Ilya Sutskever. 2017. Evolution Strategies as a Scalable Alternative to Reinforcement Learning. arXiv:1703.03864 http://arxiv.org/abs/1703.03864

[60] Selim Seferbekov. 2020. https://github.com/selimsef/dfdc_deepfake-_challenge.

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

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

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

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

[65] Mingxing Tan and Quoc Le. 2019. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. In *International Conference on Machine Learning*. 6105–6114.

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

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

[68] Cristian Vaccari and Andrew Chadwick. 2020. Deepfakes and disinformation: exploring the impact of synthetic political video on deception, uncertainty, and trust in news. *Social Media+ Society* (2020).

[69] Luisa Verdoliva. 2020. Media Forensics and DeepFakes: an overview. *arXiv preprint arXiv:2001.06564* (2020).

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

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

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

[73] Cihang Xie, Jianyu Wang, Zhishuai Zhang, Zhou Ren, and Alan Yuille. 2018. Mitigating Adversarial Effects Through Randomization. In *International Conference on Learning Representations*.

[74] Cihang Xie, Zhishuai Zhang, Jianyu Wang, Yuyin Zhou, Zhou Ren, and A. Yuille. 2019. Improving Transferability of Adversarial Examples With Input Diversity. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2019), 2725–2734.

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

[76] Egor Zakharov, Aliaksandra Shysheya, Egor Burkov, and Victor Lempitsky. 2019. Few-shot adversarial learning of realistic neural talking head models. In *International Conference on Computer Vision (ICCV)*. 9459–9468.

[77] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao. 2016. Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks. *IEEE Signal Processing Letters* (2016).

[78] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao. 2016. Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks. *IEEE Signal Processing Letters* 23, 10 (2016), 1499–1503. https://doi.org/10.1109/LSP.2016.2603342

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

[80] Wen Zhou, X. Hou, Y. Chen, Mengyun Tang, Xiangqi Huang, X. Gan, and Yong Yang. 2018. Transferable Adversarial Perturbations. In *ECCV*.