

Adversarial Perturbations Fool Deepfake Detectors

Apurva Gandhi and Shomik Jain

SPONSORS:

What Are Deepfakes?

Source



Target



Deepfake

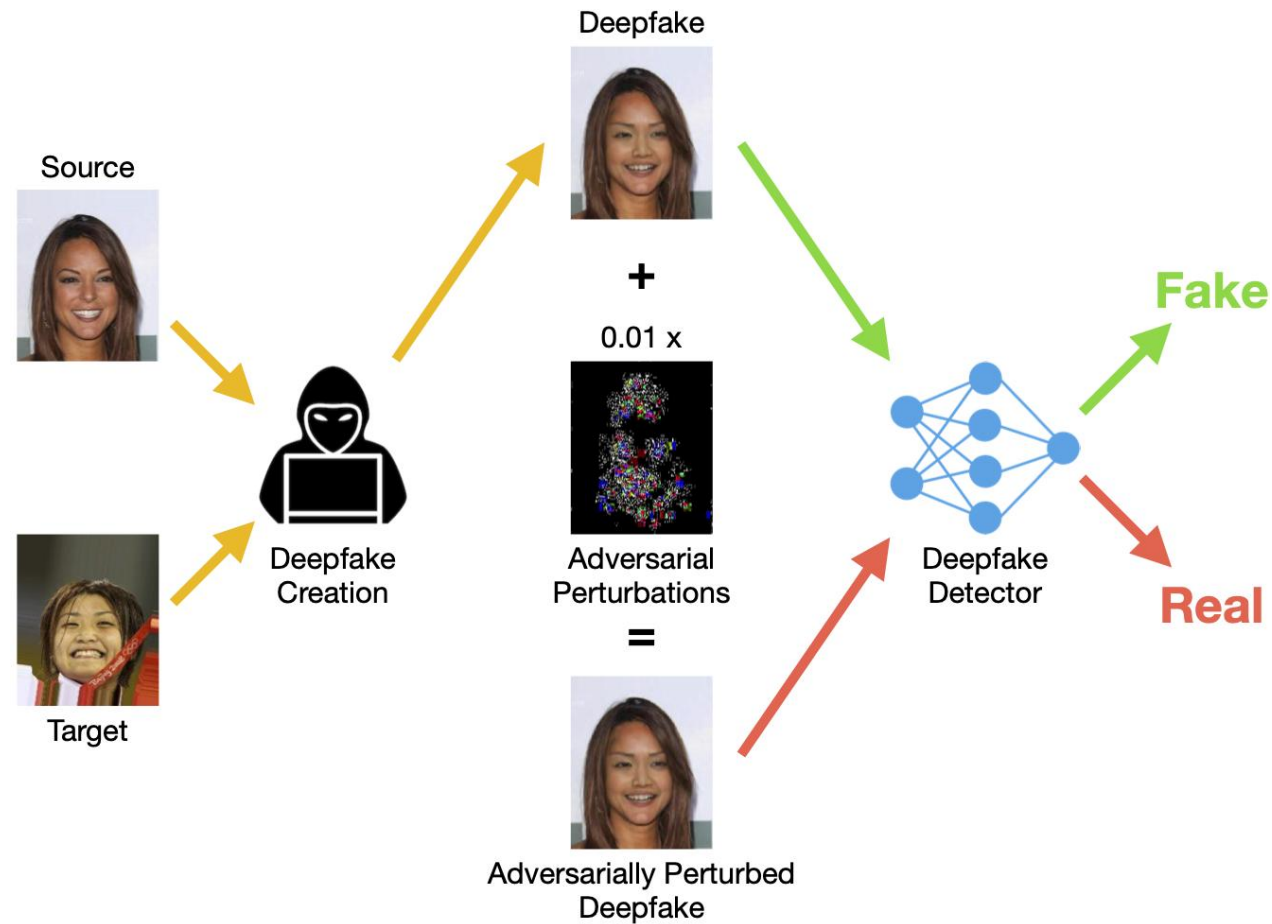


Malicious applications of deepfakes make detection imperative, especially in **pornography**, **fake news**, and **social media**.

Image Source: <https://www.youtube.com/watch?v=rvF5IA7HNKc>

SPONSORS:

Fooling Deepfake Detectors



SPONSORS:

Our Work

1. Deepfake Dataset and Detection
2. Adversarial Perturbations
3. Defending Against Adversarial Perturbations

SPONSORS:



Deepfake Dataset and Detection

SPONSORS:



Deepfake Dataset

Real
Images



*CelebA Dataset
(Liu et al., 2015)*

Fake
Images



*Fewshot Face
Translation GAN
(Shaoanlu, 2019)*

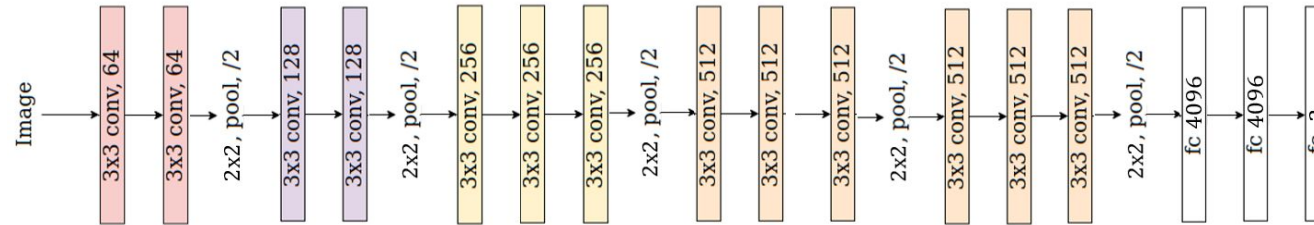
10,000 Images: 5,000 Real and 5,000 Fake.

SPONSORS:

Deepfake Detection

VGG-16

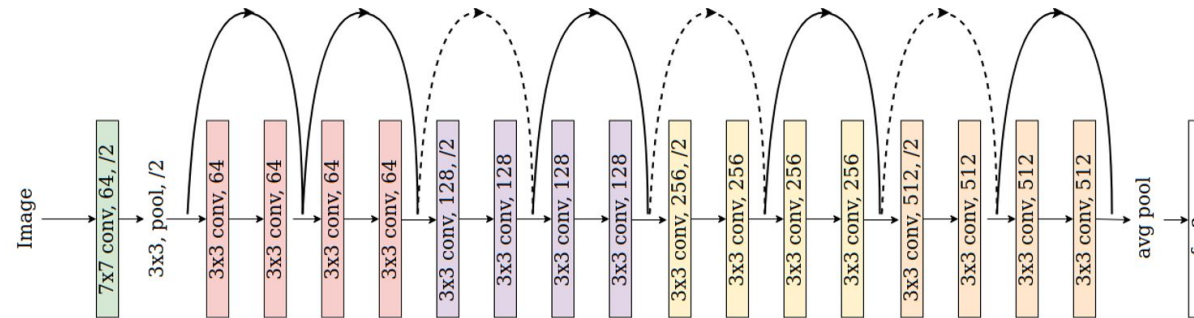
Very Deep Convolutional Networks for Large Scale Image Recognition (Simonyan and Zisserman, 2015)



Accuracy	99.7 %	AUROC	99.9%
Fake Recall	99.7 %	Real Recall	99.8%

ResNet-18

Deep Residual Learning for Image Recognition (He et al., 2015)



Accuracy	93.2%	AUROC	97.9%
Fake Recall	95.4 %	Real Recall	91.1%

SPONSORS:

Adversarial Perturbations of Deepfakes

SPONSORS:



Fast Gradient Sign Method (FGSM)

Explaining and Harnessing Adversarial Examples (Goodfellow, Shlens and Szegedy, 2015)

$$\mathbf{x}_{adv} = \mathbf{x} + \epsilon \operatorname{sign}(\nabla_x J(\mathbf{x}, \mathbf{y}, \theta))$$

Why? Consider the linear approximation of the loss function:

$$\begin{aligned} J(\mathbf{x}_{adv}, \mathbf{y}, \theta) &\approx J(\mathbf{x}, \mathbf{y}, \theta) \\ &\quad + \underbrace{\epsilon \nabla_x J(\mathbf{x}, \mathbf{y}, \theta)^T \operatorname{sign}(\nabla_x J(\mathbf{x}, \mathbf{y}, \theta))}_{\text{non-negative}} \end{aligned}$$

A **single step** of gradient ascent fools an accurate classifier with simply the sign of the gradient.

SPONSORS:

Carlini and Wagner L_2 Norm Attack (CW- L_2)

Towards Evaluating the Robustness of Neural Networks (Carlini and Wagner, 2016)

CW- L_2 minimizes an objective function with 2 components:

$$\mathbf{x}_{adv} = \arg \min_{\mathbf{x}'} \left\{ \underbrace{\|\mathbf{x}' - \mathbf{x}\|_2^2}_{\text{similarity}} + c \underbrace{\max\{\mathbf{Z}(\mathbf{x}')_{fake} - \mathbf{Z}(\mathbf{x}')_{real}, -\kappa\}}_{\text{misclassification}} \right\}$$

pre-softmax outputs (logits)

A change of variables to enforces a box constraint (pixel values in range $[0, 1]$):

$$\mathbf{x}' = \frac{1}{2}(\tanh(\omega) + 1)$$

CW- L_2 optimizes for both a misclassification and similarity to original image.

SPONSORS:

Adversarially Perturbed Deepfakes

Unperturbed



FGSM



CW-L₂



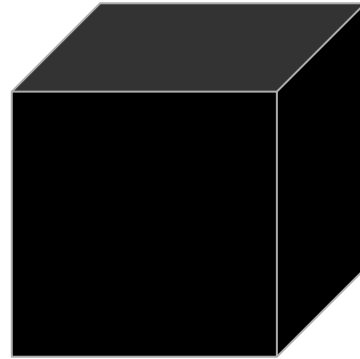
Perturbed deepfakes look similar to unperturbed deepfakes but fool detectors.

SPONSORS:

Types of Adversarial Attacks

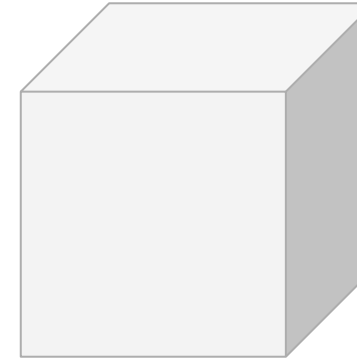
Blackbox Attack


Adversary has **limited** access to the detector (e.g. training dataset)



Whitebox Attack

Adversary has **complete** access to the detector (e.g. model parameters)

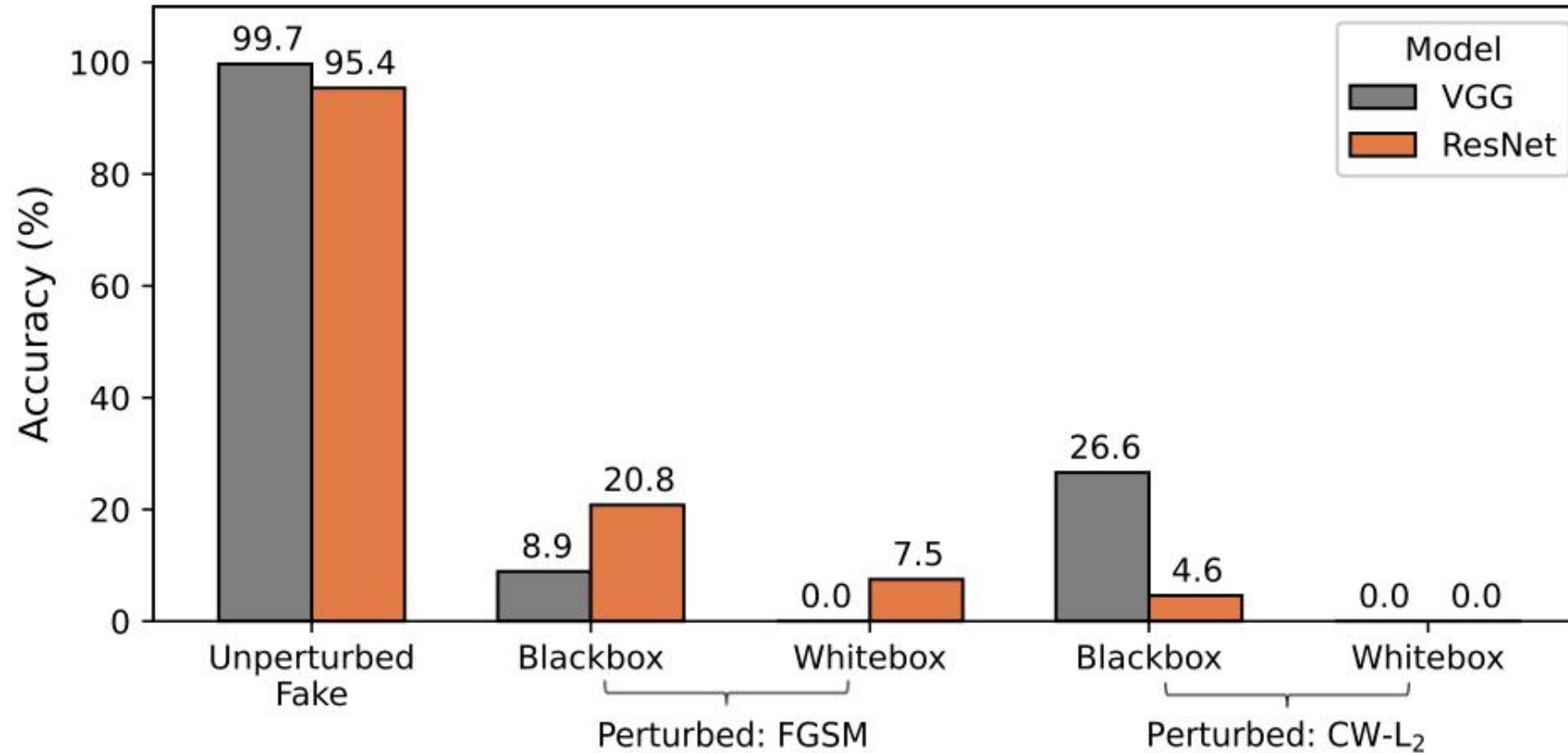


Low  High
Adversary's Knowledge

Blackbox attacks are more practical. Whitebox attacks are more effective.

SPONSORS:

Adversarial Attack Results



Adversarial attacks compromise detector accuracy in all cases.

SPONSORS:

Defending Against Adversarial Perturbations

SPONSORS:



Lipschitz Regularization

*Adversarial explanations for understanding image classification decisions and improved neural network robustness
(Woods, Chen and Teuscher, 2019)*

Augment the loss function to regularize the norms of the logit gradients with respect to the input:

$$J_{aug}(\mathbf{x}, \mathbf{y}, \theta) = J(\mathbf{x}, \mathbf{y}, \theta) + \frac{\lambda}{CN} \sum_{i=1}^C \|\nabla_x \mathbf{Z}(\mathbf{x})_i\|_2^2$$

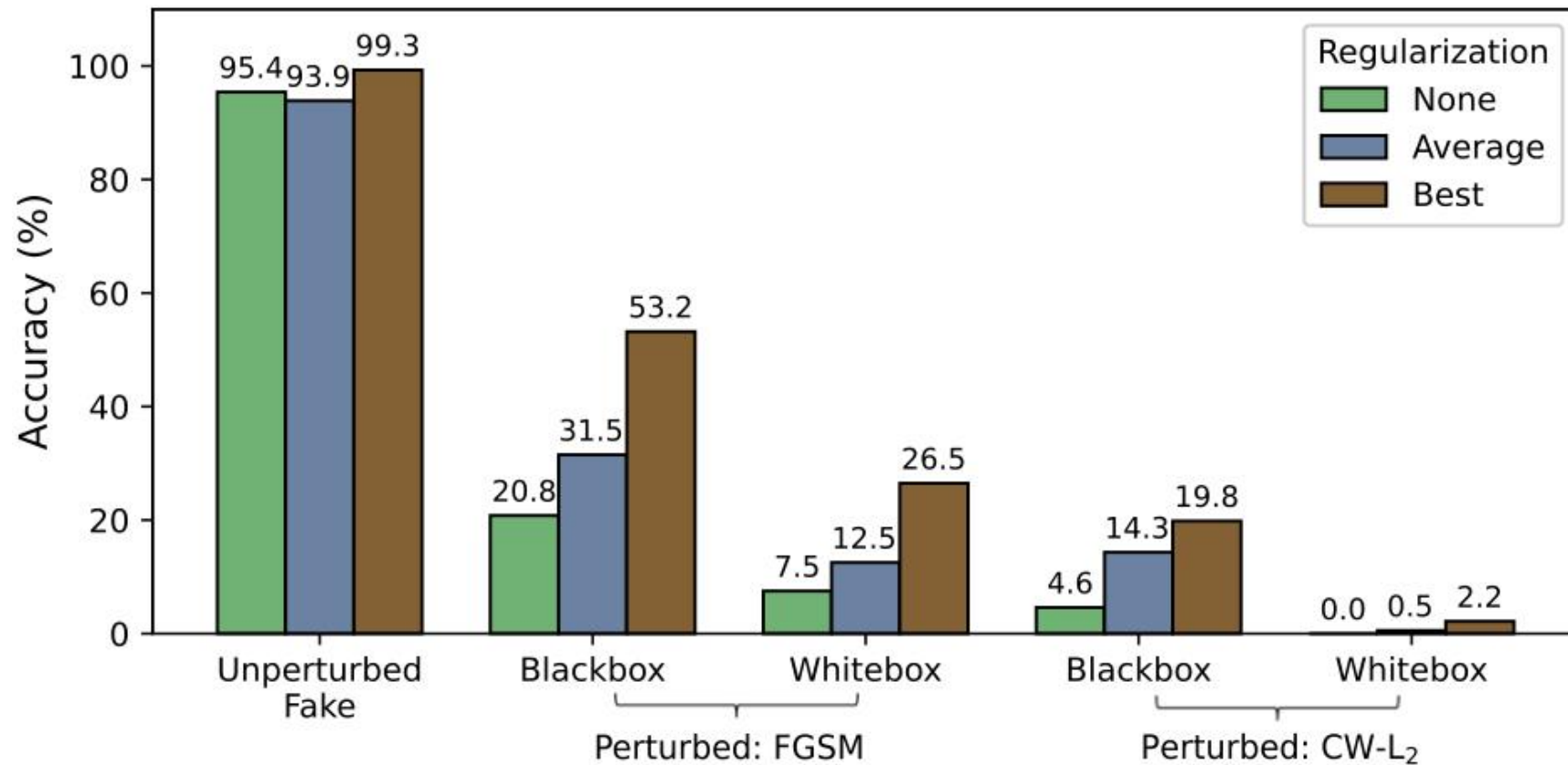
Why? Consider the linear approximation of the loss function:

$$\begin{aligned} J(\mathbf{x}_{adv}, \mathbf{y}, \theta) &\approx J(\mathbf{x}, \mathbf{y}, \theta) + \nabla_x J(\mathbf{x}, \mathbf{y}, \theta)^T (\mathbf{x}_{adv} - \mathbf{x}) \\ &= J(\mathbf{x}, \mathbf{y}, \theta) + \sum_{i=1}^C \frac{\partial J}{\partial \mathbf{Z}_i} \boxed{\nabla_x \mathbf{Z}(\mathbf{x})_i^T} (\mathbf{x}_{adv} - \mathbf{x}). \end{aligned}$$

Regularize the gradient norms to make the detector less sensitive to small perturbations.

SPONSORS:

Regularization Results



Regularization improved detectors on average but accuracy remains low.

SPONSORS:

Deep Image Prior (DIP)

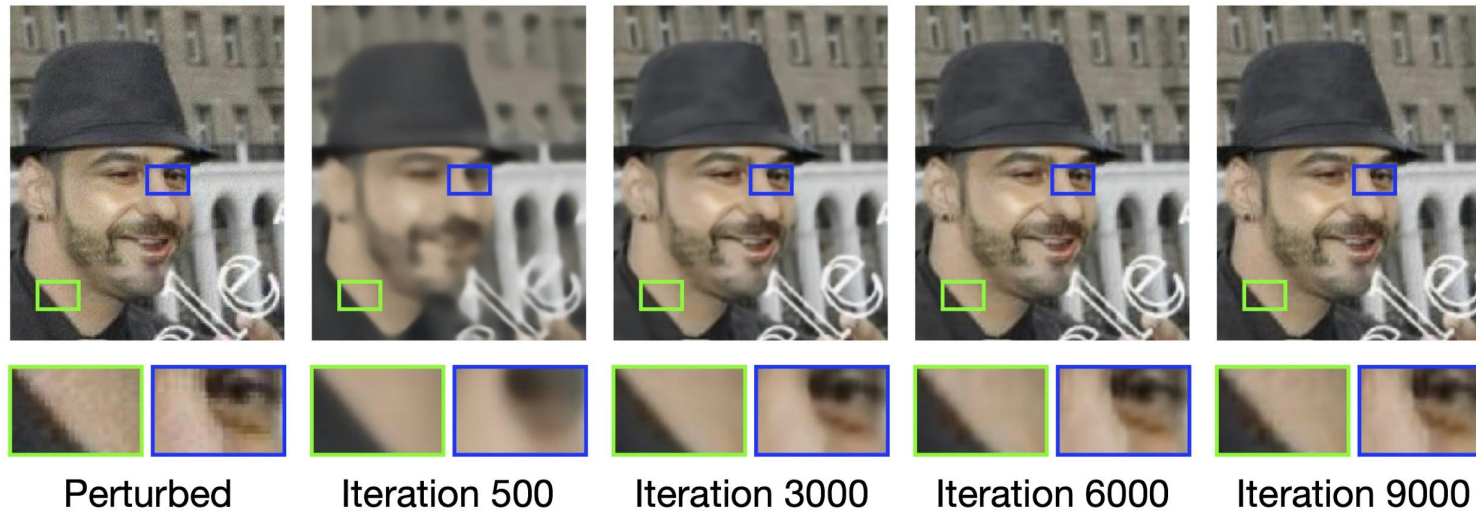
Deep Image Prior (Ulyanov, Lempitsky and Vedaldi, 2017)

Removing Perturbations from Deepfakes

Let $\mathbf{f}(\theta, \mathbf{z})$ represent a generative CNN and \mathbf{x}_{adv} be a perturbed deepfake.

Solve Optimization Problem:

$$\min_{\theta} \{ \text{MSE}(\mathbf{f}(\theta, \mathbf{z}), \mathbf{x}_{adv}) \}$$



An **unperturbed** image lies along the optimization path.

SPONSORS:

DIP Generative CNN

Deep Image Prior (Ulyanov, Lempitsky and Vedaldi, 2017)

Downsampling

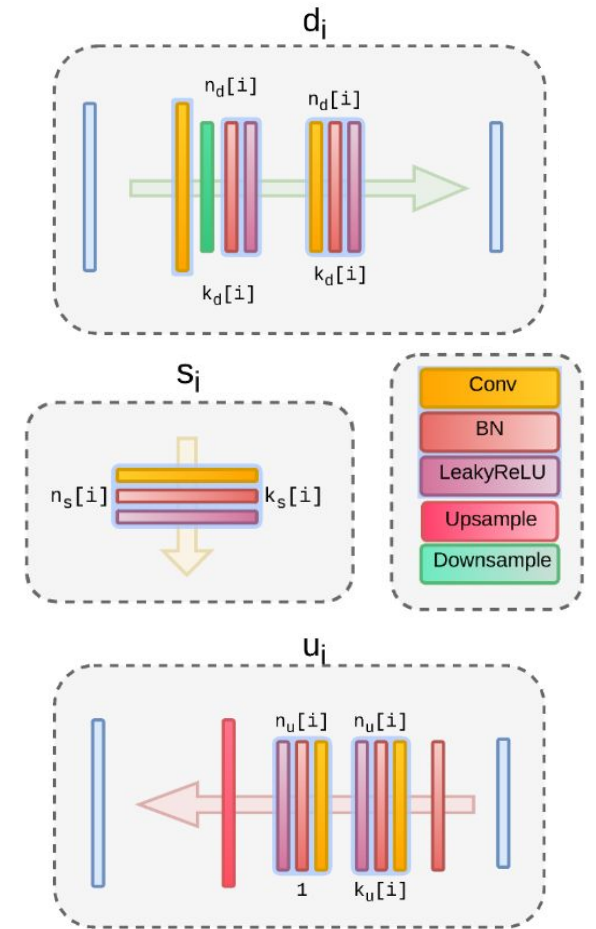
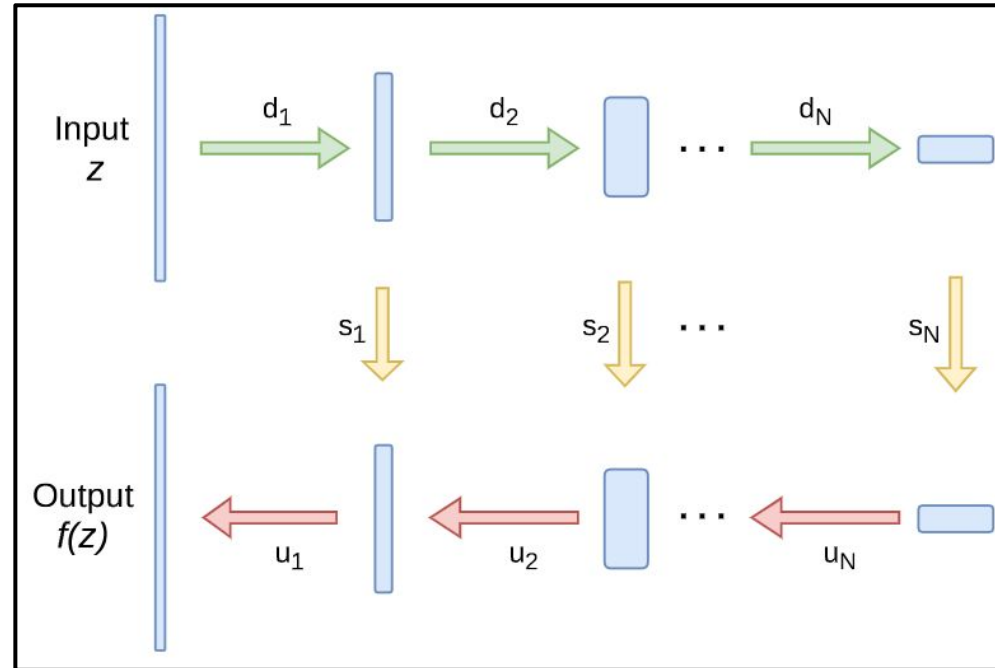
$n_d = [8, 16, 32, 64, 128]$
 $k_d = [3, 3, 3, 3, 3]$

Skip-Connections

$n_s = [0, 0, 0, 4, 4]$
 $k_s = [\text{NA}, \text{NA}, \text{NA}, 1, 1]$

Upsampling

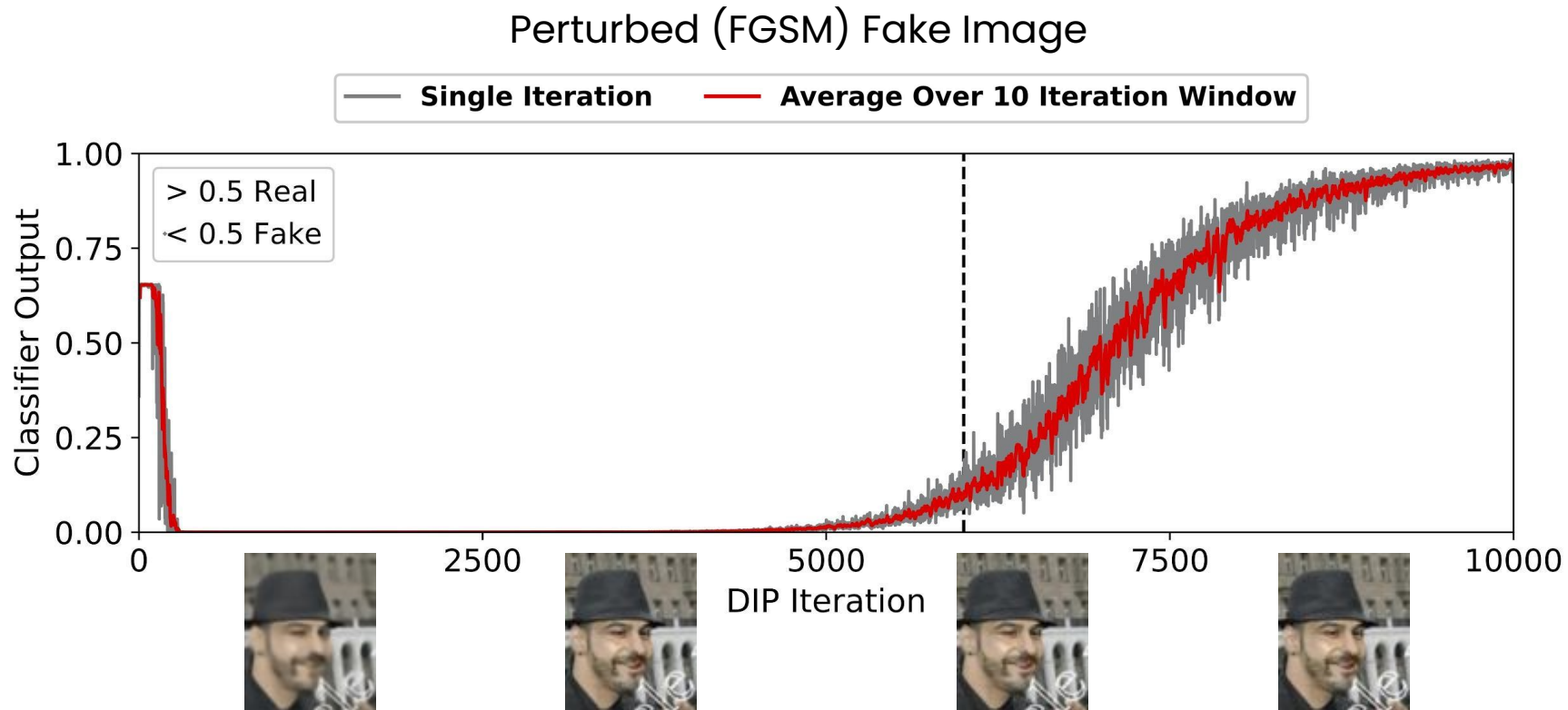
$n_u = [8, 16, 32, 64, 128]$
 $k_u = [3, 3, 3, 3, 3]$



DIP optimizes a randomly initialized encoder-decoder architecture for image generation.

SPONSORS:

DIP Optimization Path

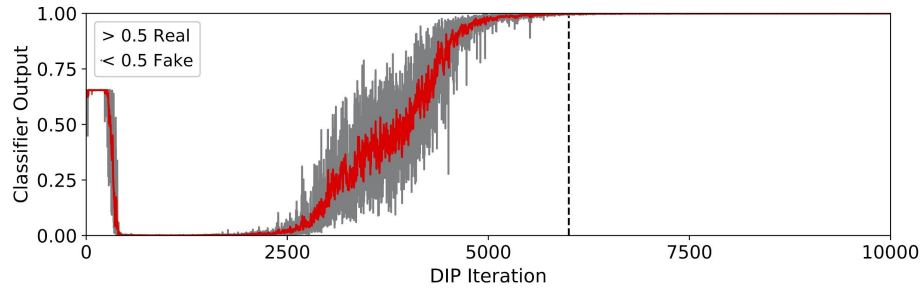


An **accurate prediction** lies along the optimization path before the generative CNN learns perturbations.

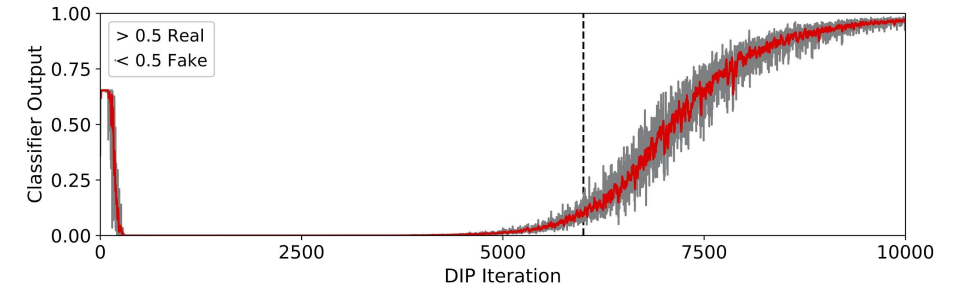
SPONSORS:

DIP Optimization Path

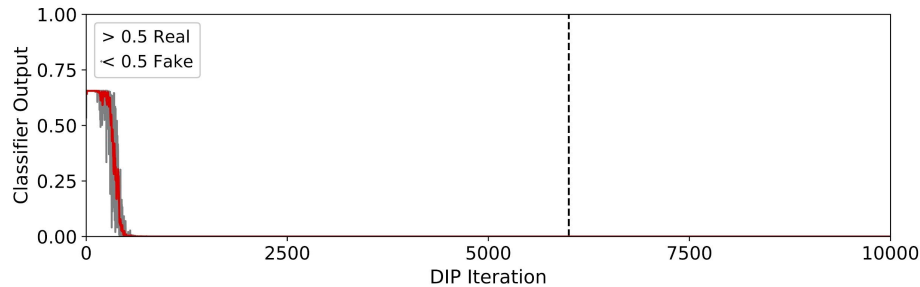
Unperturbed Real Image



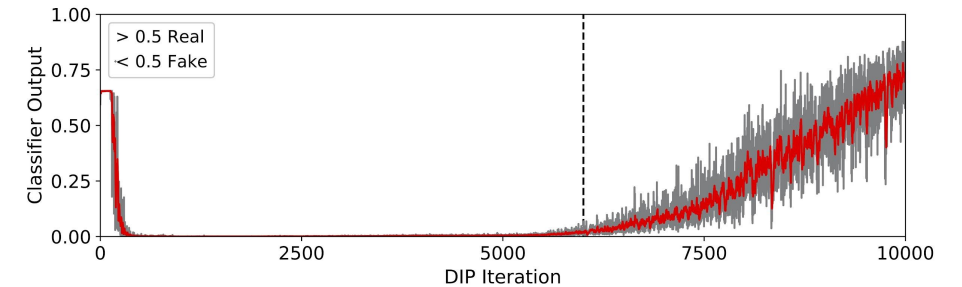
Perturbed (FGSM) Fake Image



Unperturbed Fake Image



Perturbed (CW- L_2) Fake Image



In all cases, an accurate prediction lies along the optimization path at iteration 6,000.

SPONSORS:

DIP Defense

DIP Experiments: 100 Image Subsample

Real	Unperturbed	Blackbox	FGSM	Correct
Fake	Perturbed	Whitebox	CW-L ₂	Incorrect

DIP Results

Model	Accuracy	AUROC	Fake		Real	
			Precision	Recall	Precision	Recall
DIP Defense	97.0%	99.2%	98.9%	97.8%	81.8%	90.0%
Original Detector	60.0%	41.9%	100.0%	55.5%	20.0%	100.0%

The DIP defense achieved 95% accuracy on perturbed deepfakes that fooled the original detector.

SPONSORS:

Conclusions and Limitations

1. Adversarial perturbations of deepfakes fool common CNN-based detectors.
2. Lipschitz regularization marginally improved detection of perturbed deepfakes.
3. The DIP defense successfully removes perturbations but has high computational cost.

Future work involves finding more efficient methods to improve deepfake detector robustness to adversarial perturbations.

SPONSORS: