参赛队员姓名：刘睿臻，钟达之，陈志聪

中学：华南师范大学附属中学

省份：广东省

国家/地区：中国

指导教师姓名：谢晓华，杨晓安

论文题目：Disrupting Deepfakes and Face-Swap Autoencoders with Fine-tuned Adversarial Attacks

本参赛团队声明所提交的论文是在指导老师指导下进行的研究工作和取得的研究成果。尽本团队所知，除了文中特别加以标注和致谢中所罗列的内容以外，论文中不包含其他人已经发表或撰写过的研究成果。若有不实之处，本人愿意承担一切相关责任。

参赛队员： **刘睿臻，钟达之，陈志聪，** **指导老师：** **谢晓华，杨晓安**

2020 年 9 月 15 日

CONTENTS

Disrupting Deepfakes and Face-Swap Autoencoders with Fine-tuned Adversarial Attacks

Ruizhen Liu
International Department,
The Affiliated High School of SCNU
liurz.jason2019@gdhfi.com

Dazhi Zhong
International Department,
The Affiliated High School of SCNU
zhongdz.dazhi2018@gdhfi.com

Zhicong Chen
International Department,
The Affiliated High School of SCNU
chenzc.marin2018@gdhfi.com

***Abstract*—Currently, GAN-based face-swapping methods pose a threat to cybersecurity in fraud, fake news, or explicit content generation. We propose a novel evasion adversarial attacking method to perturb images with imperceptible noise that greatly alter face-swapping results on trained models. This method builds on existing iterative fast gradient sign methods such as Momentum Iterative Fast Gradient Sign Method and Nesterov Iterative Fast Gradient Sign Method, and applies them on deepfake attacks to achieve image tamper security. Because current state-of-the-art deepfake evasion attacks use only vanilla FGSM methods to generate attack examples, we greatly improve on such methods. Our proposed method outperforms current state-of-the-art adversarial attack methods in both white-box and black-box settings in image classification networks and achieves higher output dissimilarity and higher input similarity in image translation networks. Using our proposed method on face-swapping deep neural networks result in poorly generated outputs, proving our efficacy at attacking deepfakes and face-swapping. Our work mainly accomplishes two goals: developing an iterative method that outperforms state-of-the-art methods in adversarial attacks in classification and achieving more effective results in deepfake attacks. This shows potential in AI cybersecurity applications.**

***Index Terms*—AI Security, Adversarial Attacks, Deepfake, Image Translations, Generative Adversarial Networks, Face-Swap**

Fig. 1: **Top row:** an adversarial example crafted by the proposed Adam iterative fast gradient sign method (AI-FGSM) on Inception v3 [1] model. **Middle row:** an adversarial example crafted by the proposed AI-FGSM on StarGAN deepfake model [2]. **Bottom row:** two adversarial images crafted by AI-FGSM on face-swap GAN [3].

I. Introduction

The recent ubiquity of CNN based models has resulted in an upsurge of convolution-based deep learning models [4][5]. These models can be used in both image classification and image generation. GANs, for example, use convolution layers to generate realistic images [6], which can generate faces and other objects [7].These generational methods led to the creation of GANs that can change the facial expressions of a face, add non-existing accessories or hair [2], or swap faces from human to human [8]. Many such methods are in widespread industrial applications, such as TikTok and Snapchat [9][10]. With the ease of using face-swapping methods growing, there comes a risk of generating images with malicious intent or explicit applications [11]. These methods of changing faces on videos have been coined "deepfakes [2]," often with a negative implication. One example may be swapping faces of celebrities on porn stars [11], or creating fake news, photos, or committing fraud [12]. This negative use led to deepfake prevention and detection being a recent boon [13]. One way of countering deepfakes is the use of deepfake detection, often used in videos and images to spot machine-generated specific small differences in deepfake images compared to normal images [14][15]. However, because these methods may rely on imperceptible imperfections or features, they may be unreliable when facing applications they were not trained for, or under compression, which is common in social media apps [16]. Another problem may be that detecting the image's falsehood after it has been used may not lessen as much damage as preventing it from the source. Additionally, deepfake models may adopt adversarial training to avoid detection attacks [17]. Thus, another approach for attack may be to alter the training data of face-swap algorithms, namely "poisoning" attacks,

making them unable to converge, and thus producing low-quality results [18]. However, attackers may not be able to get full access to the models' training data or obtain weights of a model before it is trained. It also suffers from training data that may be variable and changing. One other method may be adding imperceptible perturbations on the input data of face-swap algorithms [19], which would alter output of such algorithms. This would be the same way adversarial attacks may fool image classifiers, even without altering training data. Such methods, such as FGSM, or Fast Gradient Sign Method, could potentially expose model vulnerabilities by maximizing loss on a single image [20]. Potentially, this may be used in profile pictures or social media images, which would then make them tamper-resistant.

Our work focuses on this method of deepfake prevention. In our first part of our essay, we primarily focus on achieving better results than state-of-the-art methods in classical adversarial attack situations, namely, image classification models. Then, in our second part, we focus on using our novel method on deepfake attacks, namely faceswap [8] and StarGAN [21].

In our first part, we show that in image classification attacks, we perform better than baselines and current methods in both white-box and black-box models, and in both singular model attacks and in ensemble attacks. To specify, current adversarial attacks focus on two major fields of attack, white-box attacks and black-box attacks. Under the white-box model, the network's parameters are known to us, so we can calculate gradients in relation to the input image. This gradient would then be the basis of the FGSM algorithm [22], which would be able to perturb the image to maximize loss. On the other hand, the black-box scenario assumes the attacker can only access the input and result, but not parameters or construction of the model. The effectiveness of a white-box surrogate on a black-box target is a huge problem to be solved. The baseline solutions of FGSM and I-FGSM have a major trade-off between the effectiveness of the two scenarios. Among the two, iterative approaches of the FGSM algorithm or I-FGSM often score better in white-box models, but often fail to generalize in black-box models [23]. This is because transferability in adversarial attacks relies on different models making similar decision boundaries around pixels, so gradients on pixels that maximize loss may be transferred [23][24]. Iterative designs may maximize the loss of one model; it may not be so much as using the information on the actual pixels as taking information from the surrogate. They may fall into poor local maxima, which decreases transferability [25]. To counter or lessen the problems with baseline attacks, many current state-of-the-art methods utilize optimization, ensemble attacks [26], and data augmentation [27] [28] algorithms to lower this trade-off, achieving better results. The momentum optimizer and the Nesterov optimizer have been used for this purpose [29] [25], was well Gaussian blurring [19] and image intensity multiplication [25]. Our novel methods with AdaGrad [30], RMSProp [31] and Adam [32] effectively implement adaptive step size to the existing algorithms, ensuring that large gradients would not result in divergence, while small gradients would not result in slowing down. It also increases flexibility in iterative gradient approaches, as previous methods only maintain static step sizes, which increased the number of iterations needed for convergence. Our novel methods also prove effective in increasing transferability in black-box attacks. We show that the AdaGrad, RMSProp, and Adam models achieve better scores than previous methods, while also being able to be combined with image augmentation, image scaling, and ensemble methods to score better in either attack scenarios.

In our second part, we show that similar attacks can be achieved in image translation networks. Though state-of-the-art image classification attacks are not implemented in translation attacks, we implement these attacks in comparison with our attack algorithm. We show that our novel methods, in general, score better than both baseline methods and contemporary methods in image translation attacks. We also show that, in addition to disruption strength, input similarity is also greater than compared to other methods. This further proves the efficacy of our method in a wide range of possible scenarios. We then experiment on autoencoder based networks, which show that Adam achieves significant disruption ability, deforming face-swapped faces. The attacked faces appear to have deformed facial features, as well as paler skin and odd face positions.

In summary, in our work, we defend against face-swapping deepfake models using our state-of-the-art adversarial attack methods, namely AGI-FGSM, RI-FGSM, and AI-FGSM. Our methods outperform existing methods in image classification attacks in both white-box and black-box scenarios. We further implement this algorithm on StarGAN [2] and faceswap GAN [3], two state-of-the-art image translation networks that transfer facial features and swap faces. We show that we achieve major disruptions in these two networks. We then conclude that Adam boosted FGSM is state-of-the-art and can be used in cybersecurity appliances.

Our major contributions are

- We proposed a state-of-the-art method for adversarial attacks in classification scenarios.
- This state-of-the-art method can be used in translation scenarios to form a state-of-the-art deepfake attack.
- This state-of-the-art method can be used in autoencoder based face-swapping deep learning networks with good results.

II. Preliminary

A. Notation

Let x be a benign image with the corresponding label y^{true} and a pre-trained classifier $f(x)$ with loss function $\mathcal{J}$. The purpose of performing an adversarial attack is to perturb x as x^{adv} so as to maximize the loss function $\mathcal{J}(x^{adv}, y^{true})$ with $||x^{adv} - x||_\infty \leqslant \epsilon$, where ϵ is the maximum perturbation allowed.

B. Attack Methods

Fast Gradient Sign Method (FGSM) [22]. FGSM generates an adversarial example x^{adv} in one iteration so as to maximize the loss $J(x^{adv}, y^{true})$ as

$$x^{adv} = x + \epsilon \cdot \text{sign}(\nabla_x \mathcal{J}(\boldsymbol{x}^{adv}, y^{true}). \quad (1)$$

Iterative Fast Gradient Sign Method (I-FGSM) [33] is the iterative version of FGSM. It updates the adversarial example x^{adv} multiple times as

$$x_0^{adv} = x,$$
$$x_{t+1}^{adv} = \text{Clip}_x^{\epsilon}\{x_t^{adv} + \alpha \cdot \text{sign}(\nabla_x \mathcal{J}(\boldsymbol{x}^{adv}, y^{true})\} \quad (2)$$

where $\text{Clip}_x^{\epsilon}(\cdot)$ bounds the adversarial examples within the maximum perturbation ϵ.

Projected Gradient Descent (PGD) [34], instead of picking x itself, initializes x^{adv} with a random noise within the ϵ bound and update x^{adv} with I-FGSM.

Momentum Iterative Fast Gradient Sign Method (MI-FGSM) [29] updates the adversarial examples x^{adv} in I-FGSM, using the update rules in the momentum optimizer. This leaves the update rules for x_t^{adv} as

$$g_{t+1} = \mu \cdot g_t + \frac{\nabla_x \mathcal{J}(x_t^{adv}, y^{true})}{\|\nabla_x \mathcal{J}(x_t^{adv}, y^{true})\|_1}$$
$$x_{t+1}^{adv} = \text{Clip}_x^{\epsilon}\{x_t^{adv} + \alpha \cdot \text{sign}(g_{t+1})\} \quad (3)$$

Nesterov Iterative Fast Gradient Sign Method (NI-FGSM) [25] optimizes adversarial attacks using Nesterov Accelerated Gradient (NAG), which is improved version of Momentum method. It is discovered that NAG can not only stabilize the update directions of x^{adv}, but also correct the previously accumulated gradients so as to provide a lookahead property. The update rule is as follows:

$$x_t^{nes} = x_t^{adv} + \alpha \cdot \mu \cdot g_t$$
$$g_{t+1} = \mu \cdot g_t + \frac{\nabla_x \mathcal{J}(x_t^{nes}, y^{true})}{\|\nabla_x \mathcal{J}(x_t^{nes}, y^{true})\|_1}$$
$$x_{t+1}^{adv} = \text{Clip}_x^{\epsilon}\{x_t^{adv} + \alpha \cdot \text{sign}(g_{t+1})\} \quad (4)$$

Diverse Input Method (DIM) [35] performs resizing and padding for input images to boost up adversarial attacks. DIM can be seamlessly integrated into any gradient-based methods mentioned above.

Translation-Invariant Method (TIM) [28] uses a set of translated images to optimize adversarial attacks with the gradient being calculated by convolving the gradient with a pre-defined kernel $\boldsymbol{W}$. The method is also available for integration into gradient-based methods, with the following update rule,

$$x_{t+1}^{adv} = x_t^{adv} + \alpha \cdot \text{sign}(\boldsymbol{W} \cdot \nabla_x \mathcal{J}(\boldsymbol{x}_t^{adv}, y^{true}). \quad (5)$$

TIM and DIM can be combined as TI-DIM, a strong black-box attack method.

Carlini & Wagner Attack (C&W) [36] directly search for a x^{adv} so as to minimize the distance the between an adversarial example and its real example by solving:

$$\underset{x^{adv}}{\text{argmin}} \, ||x^{adv} - x||_\infty - J(x^{adv}, y) \quad (6)$$

However, this method lacks transferability in black-box scenario.

Scale-Invariant Property (SIM) [25] generates adversarial perturbations based on a set of m scaled copies of the input image:

$$\underset{x^{adv}}{\text{argmax}} \frac{1}{m} \sum_{i=1}^{m} \mathcal{J}(S_i(x_t^{adv}), y^{true}), \quad (7)$$
$$\text{s.t. } ||x^{adv} - x||_\infty \leqslant \epsilon,$$

where $S_i(x) = x/2^i$ scales the input image x by a factor of $1/2^i$.

C. Deepfakes Models

Generative Adversarial Networks (GAN) [37] comprised most of the current Deepfake models. GAN consists of a discriminator D and a generator G: the former aims to identify whether an input image x is fake or not, and the discriminator to generate a fake image x^{adv} to fool the generator G into misclassifying x^{adv} as real. Current GANs for deepfake include pix2pixHD [38], CycleGAN [39], GANimation [40] and StarGAN [2]. Among these methods, StarGAN generates images with the highest quality [2]. It can integrate several datasets with different sets of labels and generate a fake example with any label in it.

D. Image Translation Disruption on deepfakes

Attempts have been made to disrupt deepfake models previously [19][41]. Similar to performing adversarial attack, we need to add a perturbation noise η to the input image to generate the disrupted image x^{adv} as

$$x^{adv} = x + \eta. \quad (8)$$

When fed into the deepfake generator $\boldsymbol{G}$, $\boldsymbol{y}$ and $\boldsymbol{y}^{adv}$ are the translated output images with the mappings $\boldsymbol{G}(\boldsymbol{x})$ and $\boldsymbol{G}(\boldsymbol{x}^{adv})$ respectively.

We want to create an adversarial example x^{adv} such that the alteration by deepfakes can be obvious for human beings, meaning to maximize the distortion and evaluate perturbation using the L^1, L^2 or L^∞ norm. If applied as I-FGSM, the update rule for x^{adv} can be formulated as

$$x_{t+1}^{adv} = \text{Clip}_x^{\epsilon}\{x_t^{adv} + \alpha \cdot \text{sign}(\nabla_{x^{adv}} L(G(x^{adv}), r))\}, \quad (9)$$

where α is the step size and ϵ is the bound for translation.

E. Autoencoder-based Faceswap Models

The autoencoder neural network [42][43], in essence, compresses and decompresses data using an "encoder" and a "decoder". The encoder takes a high dimensional input x and makes a low dimensional representation of them, z, commonly referred to as the latent representation of x. Then, the decoder takes the latent low dimensional representation of x, to reconstruct $\hat{x}$. For encoder ϕ and decoder ψ we have equations

$$z = \phi(x) \tag{10}$$

$$\hat{x} = \psi(z) \tag{11}$$

The loss function then, is to minimize the difference between x and $\hat{x}$, which autoencoders often use the mean average error of all pixels as a loss to minimize. We have equations

$$\phi, \psi = \underset{\phi,\psi}{\operatorname{argmin}} \|X - (\phi \circ \psi)X\| \tag{12}$$

In the faceswap model, the two decoders for the faces that need to be swapped share the same encoder, which are then trained on warped faces of the original. That is, taking the original face x, the model adds warps to the face to form x', which is then fed into the autoencoder product output $\hat{x}$, which is then trained to minimize the MAE loss between x and $\hat{x}$. Additionally, this model uses PixelShuffle, a method to increase output resolution [44]. When trained with two decoders, the two faces will then share a latent representation, which would then allow the autoencoder to swap faces. When facing new faces, the warping training essentially allows the encoder to generalize among more variations of faces, which would unsure face-swapping ability when taking an untrained face as an input [8].

F. Metrics

L1 loss and L2 loss (MSE loss). The L1 error metric that we use measure the mean absolute error between pixels in the input images K and I [45]. The L2, or mean square error, metric measures the squared difference between pixel values of the input images K and I [46]. The two formulas are

$$L1 = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}|I(i,j) - K(i,j)| \tag{13}$$

$$L2 = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[I(i,j) - K(i,j)]^2 \tag{14}$$

For the two equations above, the two images are denoted as K, and I, a pixel in one images is denoted K(i,j), and the width and height of the two images are m and n, respectively.

PSNR, Peak Signal-to-Noise Ratio. The PSNR metric [47] is another per pixel image similarity metric that measures the ratio between the largest possible value of an image to the values of a corrupting noise, hence the name, Peak Signal-to-Noise Ratio. The formula of PSNR is

$$PSNR = 10 \cdot \log_{10}(\frac{MAX_I^2}{L2}) \tag{15}$$

The MAX is the largest pixel value of an image, while the L2 is just the calculated L2 norm. We use a logarithmic scale because pixels can have a larger dynamic range.

SSIM, Structural Similarity Index Measure. The SSIM metric [38] in attacks is a perceptual metric that measures the degradation of an image. It uses perceptual phenomena such as luminance masking and contrasting masking to compare images. It uses measures in structural information in an image to achieve a stronger measurement than mean squared error or PSNR methods. This is because MSE or PSNR measure absolute errors and lose structural information. While SSIM measures local structural information, which is based on the assumption that closer pixels form stronger dependencies. The SSIM is calculated by

$$\text{SSIM}(x,y) = \frac{(2\mu_x\mu_y + C_1) + (2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \tag{16}$$

which is calculated on various windows of an image. The equation takes two windows x and y of an image with the same size of $N \times N$ as an input. In it, μ_x is the average of x and μ_y is the average of y. Calculated by

$$\mu_x = \frac{1}{N}\sum_{i=0}^{N} x_i \tag{17}$$

σ_x and σ_y are the variance of x and, respectively, calculated by

$$\sigma_x = (\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \mu_x)^2) \tag{18}$$

σ_{xy} is the covariance of x and y, calculated by

$$\sigma_{xy} = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \mu_x)(y_i - \mu_y) \tag{19}$$

$c_1 = (k_1 L)^2$ $c_2 = (k_2 L)^2$ which is used to stabilize division. L is the dynamic range of the pixel values, and $k_1 = 0.01$ and $k_2 = 0.03$ as a default.

III. Methodology

In this paper, we introduce **AdaGrad iterative gradient-based methods**, **RMSProp iterative gradient-based methods** and **Adam iterative gradient-based methods** to generate adversarial examples for deepfakes, which have seen better performances in perturbing images in both image classification attacks and image translation attacks than the standard FGSM and I-FGSM methods.

We first illustrate how AdaGrad, RMSProp and Adam are integrated into iterative FGSM respectively. This induces three attack methods generating adversarial examples satisfying the L_∞ bound: AdaGrad iterative fast gradient sign method (AGI-FGSM), RMSProp iterative gradient sign method (RI-FGSM) and Adam iterative gradient sign method (AI-FGSM). Then we integrate them with TI-DIM and SIM to attack an ensemble of pretrained models with equal ensemble weights. Finally we extend these methods to yield a broad class of attacks on deepfakes.

A. AdaGrad Iterative Fast Gradient Sign Method

AdaGrad [30], or Adaptive Sub-gradient Method, is an optimizer that reduces the learning rate at every update. This allows a larger initial learning rate and enable the parameters to converge more quickly to their optimum. The parameters update rule of AdaGrad can be formulated as:

$$\begin{aligned} \boldsymbol{s}_{t+1} &= \boldsymbol{s}_t + \nabla_x \mathcal{J}(\boldsymbol{\theta}_t) \ \odot \nabla_x \mathcal{J}(\boldsymbol{\theta}_t) \\ \boldsymbol{\theta}_{t+1} &= \boldsymbol{\theta}_t - \frac{\eta}{\sqrt{s_{t+1} + \psi}} \odot \nabla_x \mathcal{J}(\boldsymbol{\theta}_t) \end{aligned} \tag{20}$$

where η is the learning rate and ψ is a constant that prevents the learning rate from being divided by 0.

Normal gradient-based iterative attacks (e.g. I-FGSM) are prone to fall into local maxima, which demonstrates less transferability than the adversarial examples created by one-step methods. In contrast to Momentum and Nesterov optimizer, which stabilize the update direction by accumulating gradients, AdaGrad uses the accumulated gradients to adjust the step size. Such adaptive step size property would make the step size sufficiently large with small gradients so as to produce larger step sizes, but also sufficiently small with large gradients so that the it will not diverge. AdaGrad works at its best with sparse gradients [32].

We integrate AdaGrad into the iterative gradient-based so as to leverage adaptive step size property of AdaGrad and construct an adversarial attack method. We refer to it as AGI-FGSM, namely AdaGrad Iterative Fast Gradient Sign Method. Specifically, the method divides the step size with the accumulation of all previous gradients in each iteration. With s_0 being initialized to 0, the update of an adversarial example x^{adv} follows:

$$\begin{aligned} g_t &= \nabla_x \mathcal{J}(x_t^{adv}, y^{true}) \\ s_{t+1} &= s_t + g_t \odot g_t, \\ x_{t+1}^{adv} &= \text{Clip}_x^{\epsilon}\{x_t^{adv} + \frac{\alpha}{\sqrt{s_{t+1} + \psi}} \odot \text{sign}(\frac{g_t}{\|g_t\|_1})\}, \end{aligned} \tag{21}$$

where α is the initial step size, ϵ is the maximum perturbation allowed and ψ is a constant that stabilize prevent α from being divided by 0.

B. RMSProp Iterative Fast Gradient Method

RMSProp [31] is another optimizer with an adaptive learning rate. This is an improved version of AdaGrad, in which s_t is normalized by a leaky average with hyper-parameter $0 < \gamma < 1$:

$$\begin{aligned} g_t &= \nabla_x \mathcal{J}(\boldsymbol{\theta}_t) \\ \mathbf{s}_{t+1} &= \gamma \mathbf{s}_t + (1-\gamma) g_t \odot g_t, \\ \theta_{t+1} &= \theta_t - \frac{\eta}{\sqrt{\mathbf{s}_t + \psi}} \odot \nabla_x \mathcal{J}(\boldsymbol{\theta}_t) \ . \end{aligned} \tag{22}$$

In the AdaGrad method, the vector s_t keeps increasing without bound, because it sums up all the squares of gradients at every iteration. This may reduce the step sizes to values too small to approach to their optimum [31]. RMSProp solves the problem by applying leaky average to the accumulated gradients s_t so that s_t will not constantly increase and step sizes will not constantly reduce. In contrast to AdaGrad, RMSProp works at its best with on-line and non-stationary objects [32].

To avoid linear convergences of s_t in AdaGrad attacks, we also adopted RMSProp into attacks. With s_0 also being initialized to 0, the update of an adversarial example x^{adv} follows:

$$\begin{aligned} g_t &= \nabla_x \mathcal{J}(x_t^{adv}, y^{true}) \\ s_{t+1} &= \gamma s_t + (1-\gamma) g_t \odot g_t \\ x_{t+1}^{adv} &= \text{Clip}_x^{\epsilon}\{x_t^{adv} + \frac{\alpha}{\sqrt{s_{t+1} + \psi}} \odot \text{sign}(\frac{g_t}{\|g_t\|_1})\} \end{aligned} \tag{23}$$

where γ is the distribution rate of the leaky average, α is the initial step size, ϵ is the maximum perturbation allowed and ψ is a constant that stabilize prevent α from being divided by 0.

C. Adam Iterative Fast Gradient Method

Adam [32] is the current state-of-the-art optimizer, which combines the advantages of both AdaGrad and RMSProp. It also adopt adopts momentum with leaky average. The parameter update rule of Adam can be expressed as:

$$\begin{aligned} \mathbf{v}_{t+1} &= \beta_1 \mathbf{v}_t + (1-\beta_1) \nabla_x \mathcal{J}(\boldsymbol{\theta}_t), \\ \mathbf{s}_{t+1} &= \beta_2 \mathbf{s}_t + (1-\beta_2) \nabla_x \mathcal{J}(\boldsymbol{\theta}_t) \ \odot \nabla_x \mathcal{J}(\boldsymbol{\theta}_t) \\ \hat{\mathbf{v}}_{t+1} &= \frac{\mathbf{v}_{t+1}}{1-\beta_1^{t+1}}, \\ \hat{\mathbf{s}}_{t+1} &= \frac{\mathbf{s}_{t+1}}{1-\beta_2^{t+1}}, \\ \theta_{t+1} &= \theta_t - \frac{\eta}{\sqrt{\hat{\mathbf{s}}_{t+1}} + \psi} \odot \hat{\mathbf{v}}_{t+1}, \end{aligned} \tag{24}$$

where both $\beta_1, \beta_2 \in (0,1)$ are both constants for leaky averages, and the suggested choices for them are $\beta_1 = 0.9$ and $\beta_2 = 0.999$ [32]. This inevitably decelerates the updates of parameters θ_t with small iteration epochs, so Eq.(6) normalizes the two state vectors v_t, s_t to correct such bias.

For the best attack performances towards both simple images with sparse gradients and videos with non-stationary gradients, and for the looking ahead property presented by Momentum and Nesterov attack methods, we integrated Adam into I-FGSM as AI-FGSM, namely Adam Iterative Fast Gradient Sign Method. With state vectors g_0' and s_0 being initialize to 0, AI-FGSM updates the adversarial example x^{adv} by:

$$g_t = \nabla_x \mathcal{J}(x_t^{adv}, y^{true}) \tag{25}$$

$$\begin{aligned} g'_{t+1} &= \beta_1 g'_t + (1-\beta_1) \frac{g_t}{\|g_t\|_1}, \\ s_{t+1} &= \beta_2 s_t + (1-\beta_2) g_t \odot g_t, \end{aligned} \tag{26}$$

$$\hat{\boldsymbol{g}'}_{t+1} = \frac{g'_{t+1}}{1-\beta_1^t}, \hat{\mathbf{s}}_{t+1} = \frac{s_{t+1}}{1-\beta_2^t}, \tag{27}$$

$$x_{t+1}^{adv} = \text{Clip}_x^{\psi}\{x_t^{adv} + \frac{\alpha}{\sqrt{\hat{\boldsymbol{s}}_{t+1}} + \psi} \odot \text{sign}(\hat{\boldsymbol{g}'}_{t+1})\}, \quad (28)$$

where β_1, β_2 are the distribution rate of the leaky average, α is the initial step size, ϵ is the maximum perturbation allowed and ψ is a constant that stabilize prevent α from being divided by 0. The detail of this attacking algorithm can be found in Algorithm1.

Additionally, Diverse Input Method (DIM) [35], Translation-Invariant Method (TIM) [29], and Scale-Invariant Method (SIM) [25] can be integrated into AI-FGSM as AI-DIM, AI-SIM and AI-TIM, respectively. By combining the three we end up with a high-performance SI-AI-TI-DIM. The details of this attack algorithm can be found in Appendix A.

Algorithm 1: AI-FGSM

Input: A clean example x with ground-truth label y^{true};a classifier f with loss function J; Hyper-parameters: Perturbation size ϵ, maximum iterations T; decay factor β_1, β_2

Output: An adversarial example x^{adv}

$\alpha = \epsilon/T$;

$s_0 = 0, g'_0 = 0, x_0^{adv} = x$;

for $t = 0$ *to* $T-1$ **do**

  Fetch the gradients g_t by Eq.25;

  Update s_{t+1} and g'_{t+1} by $g'_{t+1} = \beta_1 g'_t + (1-\beta_1)\frac{g_t}{\|g_t\|_1}$, ; $s_{t+1} = \beta_2 s_t + (1-\beta_2) g_t \odot g_t$,

  Perform bias correction using $\hat{\boldsymbol{g}'}_{t+1} = \frac{g'_{t+1}}{1-\beta_1^t}, \hat{\boldsymbol{s}}_{t+1} = \frac{s_{t+1}}{1-\beta_2^t}$,;

  Update x_{t+1}^{adv} by Eq.28;

end

return $x^{adv} = x_T^{adv}$

IV. Experimental Results

In this section, we demonstrate that our proposed methods can generate effective adversarial examples to attack image classification models as well as Deepfake models. Firstly, we introduce our experimental settings in Section IV-A. Next, in Section IV-B and IV-C, we compare baseline methods and our proposed methods on their attacking results on both regularly trained and adversarially trained models [1]. We also compare our methods with the classic gradient-based attacks methods and the current start-of-the-art method, namely SI-NI-TI-DIM , in Section IV-D.

Beyond that,we prove that the our proposed methods works well in image translation attacks as well, and compare it against the state-of-the-art methods on translation networks. We test it on white-box attacks on StarGAN, and compare results in IV-E [2]. Finally, we use this model to attack face-swapping deep neural networks using GAN architecture and autoencoders in IV-F [3].

A. Experimental Setup

Models. For the study of the robustness of different adversarial attack methods, we use eight image classification models. Four of them are regularly trained models: Inception v3 (Inc-v3) [1], Inception-v4 (Inc-v4), Inception Resnet v2 (IncRes-v2) [48], Resnet v2-101 (Res-v2) [49]; and the remaining four are adversarially trained models: Inc-v3$_{ens3}$, Inc-v3$_{ens4}$, IncRes-v2$_{ens}$ and Inc-v3$_{adv}$ [50].

In our experiment in image translation, we attack StarGAN. We use the pre-trained 128×128 model from using the CelebA dataset. The faceswap model is from [3], which we use the pre-trained model at 200,000 iterations on the trump cage dataset from [8].

Dataset. For attacks on image classification models, we randomly select 100 images belonging to the 1000 categories from ILSVRC 2012 validation set, most of which can be correctly classified by our chosen Inception models. For attacks on image translation models we select the CelebA (Large-scale Celeb Faces Attributes) dataset, which we use choose randomly 50 images and 5 attributes for a total of 250 images. The images used in faceswap models are randomly chosen 100 images from the trump-cage dataset from [8].

Baselines. In our attacks we consider 7 algorithms to compare. Two are baseline methods of attack, namely the FGSM and I-FGSM attacks, these provide comparisons to the current methods. From [29] we have MI-FGSM, that improves on these baselines with gradients using momentum as an accelerator to escape poor local maxima. From [25] we also have NI-FGSM, which improves on both baseline and MI-FGSM. Finally, we finally include three of our proposed methods, AdaGrad-I-FGSM, Adam-I-FGSM, and RMSProp-I-FGSM. For further attacks on image classification models, we combine AI-FGSM with the existing image augmentation methods to validate our AI-FGSM's compatibility and improvement upon these methods. Currently image augmentation methods include TIM [28], DIM [35] and SIM [25]. Denote AI-FGSM integrated with them respectively as AI-TIM, AI-DIM, SI-AI-FGSM; and jointly as SI-AI-TI-DIM. In image translation attacks, we compare our methods similarly with previous mentioned baselines. The comparison is made between vanilla methods of attack, optimized, and our methods. Optimized methods were not previously introduced in image translation attacks, only using FGSM and I-FGSM for attacks.

Metrics. The two main metrics that we focus on are the similarity of the images to the original, and the attack success rate, in image classification attacks this means the rate of misclassification of the target model, in the image translation attacks this means the dissimilarity of the generated image

[1]https://github.com/jasonliuuu/SI-AI-FGSM

[2]https://github.com/DazhiZhong/disrupting-deepfakesforcodeofattackingdeepfake

[3]https://github.com/DazhiZhong/deepfakes_faceswapforcodeofattackingfaceswapGAN

TABLE I: **Comparison Between Attack Success Rates(%) of TIM and AI-TIM**

Model	Attack	Inc-v3	Inc-v4	IncRes-v2	Res-v2	Inc-v3$_{ens3}$	Inc-v3$_{ens4}$	IncRes-v2$_{ens}$	Inc-v3$_{adv}$	**Average**
Inc-v3	TIM	**100.0***	14.0	16.0	14.0	**10.0**	6.0	**8.0**	6.0	21.8
	AI-TIM(**Ours**)	**100.0***	**36.0**	**30.0**	**20.0**	8.0	**18.0**	4.0	**20.0**	**29.5**
Inc-v4	TIM	22.0	**100.0***	14.0	14.0	6.0	4.0	2.0	2.0	20.5
	AI-TIM(**Ours**)	**44.0**	**100.0***	**28.0**	**16.0**	**8.0**	**8.0**	**4.0**	**16.0**	**28.0**
IncRes-v2	TIM	30.0	28.0	90.0*	26.0	16.0	12.0	6.0	8.0	27.0
	AI-TIM(**Ours**)	**56.0**	**60.0**	**100.0***	**40.0**	**12.0**	**10.0**	**10.0**	**18.0**	**38.3**

TABLE II: **Comparison Between Attack Success Rates(%) of DIM and AI-DIM**

Model	Attack	Inc-v3	Inc-v4	IncRes-v2	Res-v2	Inc-v3$_{ens3}$	Inc-v3$_{ens4}$	IncRes-v2$_{ens}$	Inc-v3$_{adv}$	**Average**
Inc-v3	DIM	94.0*	44.0	32.0	38.0	**18.0**	**16.0**	**8.0**	14.0	33.0
	AI-DIM(**Ours**)	**98.0***	**58.0**	**48.0**	**42.0**	8.0	**16.0**	**8.0**	**18.0**	**37.0**
Inc-v4	DIM	44.0	**100.0***	24.0	36.0	12.0	**8.0**	2.0	4.0	28.8
	AI-DIM(**Ours**)	**54.0**	**100.0***	**42.0**	**42.0**	**6.0**	4.0	**4.0**	**8.0**	**32.5**
IncRes-v2	DIM	42.0	44.0	96.0*	24.0	**14.0**	4.0	**10.0**	18.0	32.0
	AI-DIM(**Ours**)	**64.0**	**54.0**	**100.0***	**48.0**	8.0	**6.0**	**10.0**	**20.0**	**38.8**

TABLE III: **Comparison Between Attack Success Rates(%) of SIM and SI-AI-FGSM**

Model	Attack	Inc-v3	Inc-v4	IncRes-v2	Res-v2	Inc-v3$_{ens3}$	Inc-v3$_{ens4}$	IncRes-v2$_{ens}$	Inc-v3$_{adv}$	**Average**
Inc-v3	SI-FGSM	**100.0***	28.0	26.0	20.0	**16.0**	10.0	**12.0**	16.0	28.5
	SI-AI-FGSM(**Ours**)	**100.0***	**54.0**	**54.0**	**42.0**	10.0	**12.0**	6.0	**26.0**	**38.0**
Inc-v4	SI-FGSM	36.0	**100.0***	22.0	28.0	10.0	8.0	**6.0**	10.0	27.5
	SI-AI-FGSM(**Ours**)	**56.0**	**100.0***	**44.0**	**48.0**	**14.0**	**10.0**	4.0	**22.0**	**37.3**
IncRes-v2	SI-FGSM	38.0	28.0	98.0*	18.0	8.0	8.0	6.0	16.0	27.5
	SI-AI-FGSM(**Ours**)	**60.0**	**46.0**	**100.0***	**42.0**	**10.0**	**10.0**	**8.0**	**24.0**	**37.5**

TABLE IV: **Comparison Between Attack Success Rates(%) of SI-TI-DIM and SI-AI-TI-DIM**

Attack	Inc-v3*	Inc-v4*	IncRes-v2*	Res-v2	Inc-v3$_{ens3}$	Inc-v3$_{ens4}$	IncRes-v2$_{ens}$	Inc-v3$_{adv}$	**Average**
SI-TI-DIM	98.0	98.0	94.0	68.0	48.0	42.0	32.0	44.0	65.5
SI-AI-TI-DIM(**Ours**)	**100.0**	**100.0**	**100.0**	**78.0**	**56.0**	**52.0**	**44.0**	**46.0**	**72.0**

with attack and the generated image without attack. The similarity of original and adversarial images will be measured by SSIM, or structural similarity metric, which is an index of the perceptual difference between two images [38]. The success rate for classification attacks would be the percentage of amount of attack success images in all images. When stating a method scores better in similarity, it refers to a higher SSIM or PSNR scores of the input, while stating a method scores better in attack effect means lower SSIM and PSNR score of the generated result image, or a lower percentage of accurate classification of a target mode.

Hyper-parameters. We consider the configuration in [29]'s experiments and alter them so as to approach to Adam's default hyper-parameters settings [32] (i.e. the maximum perturbation $\epsilon = 10$, number of iterations $T = 10$, decay factor $\mu = 1.0$, and step size $\alpha = 0.1$). We also configure TIM's kernel as Guassian kernel of shape 7×7, DIM's transform probability as 0.5, and SIM's number of scale copies as 5. In translation attacks, images are standardized to $0 - 1$, with the step size $\alpha = 0.005$, and epsilon the max perturbation amount as 0.05, we run this on $k = 100$ iterations. SSIM kernel size is a default of 11×11 .

B. Attacking A Single Model

We integrate AI-FGSM into TIM [28], DIM [35], SIM [25], and their combination SI-TI-DIM, respectively. As shown in I, II and III,we use the baselines and our proposed methods to trick Inception v3, Inception v4 and Inception Resnet v2, respectively, into misclassifying images from ILSVRC 2012 dataset, and we transfer them to attack the rest of our selected models in black-box manner. We observe that, in most cases, our methods outperform the baseline by 10$\sim$20% in black-box scenario, and generally achieve 100% success rates for white-box attacks. Overall, the aforementioned experiments have shown that AI-FGSM is compatible with major image augmentation methods and can consistently improve the transferability of adversarial examples.

An adversarial example generated AI-FGSM on Inception v3 is visualized in Fig.1.

C. Attacking An Ensemble of Models

We validate the robustness of our methods by attacking an ensemble of models. We choose to simultaneously attack Inception v3, Inception v4 and Inception Resnet v2 with SI-TI-DIM and SI-AI-TI-DIM respectively. From Table IV, it is

noticeable that SI-AI-TI-DIM generally improves SI-TI-DIM the baseline by 5$\sim$ 20% in black-box scenario and achieves an attack success rate of 100% to all white-box models. Overall, SI-AI-TI-DIM yields an average attack success rate of as high as 72%. The experiment demonstrates that, for ensemble attacks, AI-FGSM significantly improves the performance of the baseline method, and thus adversarially trained models are by no means robust under the attack of SI-AI-TI-DIM. We also visualize the attack results in Appendix B

D. Further Analysis on Image Classification Model Attacks

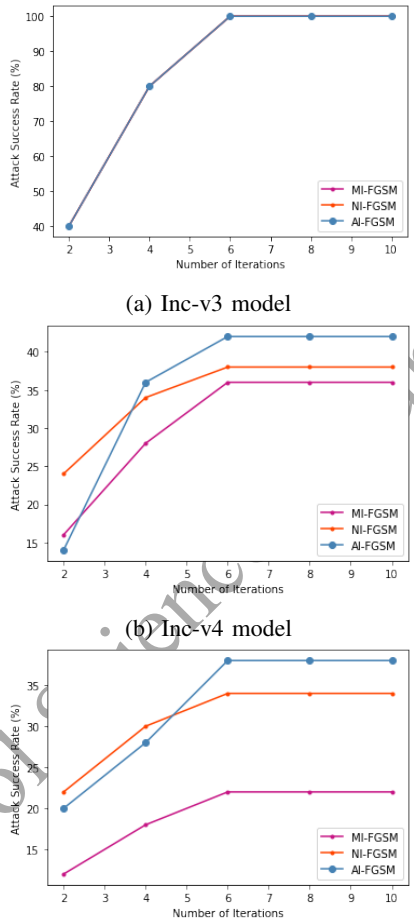
MI-FGSM vs. NI-FGSM vs. AI-FGSM. We also draw comparisons among MI-FGSM [29], NI-FGSM [25], and AI-FGSM. The adversarial examples are created and updated on Inception v3 model for ten iterations. They are then transferred to attack Inception v4 and Inception Resnet v2 every two iterations. The Fig.2 shows that, in terms of transfer-based attacks, AI-FGSM yields the highest success rate under equal numbers of iterations, which means that AI-FGSM is the least prone to fall into local maxima.

RI-FGSM and AI-FGSM compared to other gradient-based attacks. We also compare RI-FGSM and AI-FGSM to the current classical gradient-based attacks, see Table V. We choose Inception v3 as the white-box model and transfer the adversarial examples to the rest of the models. The result shows that our methods yield the first and second highest attack success rates among MI-FGSM, NI-FGSM and the other classical attacks. Specifically, AI-FGSM crafts the strongest adversarial attacks on every model. This illustrates that our methods outperform other gradient-based methods in both white-box and black-box settings.

E. Attacking Deepfake Models

Attack standards. In attacks on StarGAN, we evaluate models on 6 metrics. Two metrics measure the similarity of the attacked images to the original, while four measure the difference between the generated images. The former two, Input-SSIM, which is the Structural similarity metric of the input, and Input-PSNR, which is the peak signal-to-noise ratio of the input, should be maximized so that the perturbations are minimal. Gen-SSIM, and Gen-PSNR, the similarity measures of the output, should be minimized to achieve the highest attack effect. Finally, the L_1, and L_2 differences, which measure the difference between the two images in a straightforward, pixel-by-pixel way, should be maximized.

Attack results. We show our attack results in Table VI. We see that for the similarity metrics of the input, the vanilla iterative attack scores best, followed by our proposed AGI-FGSM and RI-FGSM. Our Adam optimized method (AI-FGSM), however, only scored mid-range. However, its scores are better than state-of-the-art methods, namely, momentum and Nesterov accelerated FGSM updates. This may be because the inclusion of a variable step size results in a smaller step size in later perturbations, which then results in less input dissimilarity, while the reason for I-FGSM to generate more



(a) Inc-v3 model

(b) Inc-v4 model

(c) IncRes-v2 model

Fig. 2: **Changes of Attack success rates (%) of NI-FGSM, MI-FGSM and AI-FGSM as the number of iterations increases.** The adversarial examples are generated on Inc-v3 model against (a) Inc-v3 model, (b) Inc-v4 model and (c) IncRes-v2 model.

similar images may be that it is limited by the strength of its attack.

The three proposed methods, AGI-FGSM, RI-FGSM, and AI-FGSM, are have a very similar update rule. Adam, however, slightly improves upon the attack effect because of its bias correction [32]. This may be because the gradients are sparse in this particular task, as the model to attack is trained to high complexity, so a bias correction magnifies the calculated gradients and step sizes. We display one adversarial examples crafted by AI-FGSM on StarGAN and its corresponding outputs in Fig.5.

Other methods hold a static step size, and because their gradients are all signed, their perturbation amount is inevitably larger over higher iterations, while the non-iterative FGSM attack is bound by the epsilon hyper-parameter, or the max perturbation amount, which it reaches in its one iteration. Vanilla I-FGSM results in a more similar input may be because it falls into poor local maxima.

Attack	Inc-v3*	Inc-v4	IncRes-v2	Res-v2	Inc-v3$_{ens3}$	Inc-v3$_{ens4}$	IncRes-v2$_{ens}$	Inc-v3$_{adv}$	**Average**
FGSM	70.0	6.0	6.0	6.0	6.0	4.0	0.0	0.0	12.3
I-FGSM	**100.0**	12.0	16.0	16.0	8.0	6.0	4.0	2.0	20.5
MI-FGSM	**100.0**	36.0	22.0	26.0	16.0	12.0	12.0	20.0	30.5
NI-FGSM	**100.0**	38.0	34.0	28.0	8.0	10.0	8.0	16.0	30.3
RI-FGSM (**Ours**)	**100.0**	32.0	32.0	30.0	**16.0**	8.0	**18.0**	16.0	31.5
AI-FGSM (**Ours**)	**100.0**	**42.0**	**38.0**	**48.0**	8.0	**14.0**	2.0	**26.0**	**35.3**

TABLE V: **The success rates (%) of non-targeted adversarial attacks against seven models, with the $*$ symbol indicating white-box attacks.** We use Inc-v3 to create our adversarial examples using FGSM, I-FGSM, MI-FGSM, NI-FGSM, AGI-FGSM, RI-FGSM, AI-FGSM.

Attacks	Gen-SSIM	Input-SSIM	Gen-PSNR	Iuput-PSNR	L1	L2
FGSM	0.100	0.875	3.698	27.80	0.547	0.450
I-FGSM	-0.227	**0.910**	-0.942	**29.03**	1.024	1.289
MI-FGSM	-0.253	0.846	-1.717	26.38	1.146	1.557
NI-FGSM	-0.240	0.863	-1.079	26.79	1.039	1.317
AGI-FGSM(**Ours**)	-0.255	0.908	-1.620	28.83	1.118	1.495
RI-FGSM(**Ours**)	-0.242	0.908	-1.251	28.93	1.064	1.374
AI-FGSM(**Ours**)	**-0.275**	0.878	**-1.990**	27.26	**1.181**	**1.627**
AI-FGM(**Ours**)	-0.264	0.894	-1.800	28.07	1.158	1.578

TABLE VI: **The metrics for input similarity and output dissimilarity for image translation attacks on StarGAN deepfake.** The metrics include SSIM, PSNR, L1, and L2 diff. Our novel attack methods are compared with baselines and state-of-the-art attack algorithms.

	FGSM	I-FGSM	MI-FGSM	NI-FGSM	AI-FGSM(**Ours**)	AI-FGM(**Ours**)
Gen SSIM	0.716	0.583	0.576	0.692	**0.524**	0.537
Input SSIM	**0.746**	0.687	0.694	0.799	0.648	0.659

TABLE VII: **The metrics for input similarity and output dissimilarity for image translation attacks on face-swap.** The metrics include SSIM, PSNR, L1, and L2 diff. Our novel attack methods are compared with baselines and state-of-the-art attack algorithms.

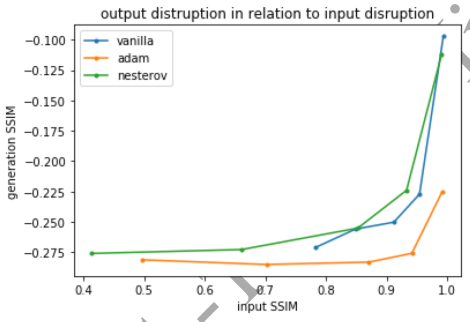


Fig. 3: **Output image distortion rates with respect to the perturbation rates on input image using Vanilla, NI-FGSM and AI-FGSM, respectively.** The horizontal axis refers to input images' SSIM with respect to the original image, and the vertical axis refers to the output's SSIM with respect to the original image.

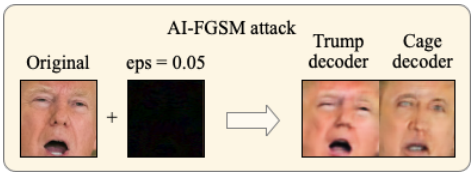


Fig. 4: **An adversarial example successfully defended by the perturbation of AI-FGSM from faceswap autoencoder.** We can see that there is noticeable distortion towards the eyes and mouths.

For more adversarial attacks on faceswap GAN from AI-FGSM, see Appendix C

Further experiments When increasing or deceasing the max perturbation amount, ϵ, we get Fig. 3,where we plot the relationship between input dissimilarity and generation dissimilarity. The range of ϵ that we choose were 0.01, 0.03, 0.05, 0.1, and0.2. For the potted lines, the lower the better, indicating that for a lower perturbation amount, it can disrupt the output more. From it, we see that the AI- FGSM exceeds both I-FGSM [22] (blue) and NI-FGSM [25] (green) methods in attack efficiency across perturbation amounts. Though previous experiments show that under the same maximum, I-FGSM scores better, we can see that under the same input perturbation, AI-FGSM disrupts output generation more, and for the same output disruption, AI-FGSM has the higher input similarity.

F. Disrupting Face-Swapping Neural Networks

Attack results. In Table VII, we see that in the face-swap attack results, comparing to previous methods, AI-FGSM resulted in a higher attack perturbation under the same epsilon value, however, we can see that this comes at the expense of input perturbation amount, which we would further discuss. Overall, under the same max perturbation amount, alpha value, and iteration number, we see that AI-FGSM achieved the best results of 0.524 SSIM for generated image similarity, lowest of all. For generated results, we see that the output quality becomes much lower with small perturbations. Many generated images achieve dis-formed eyes and mouths, while the worst results achieved paler skin and inaccurate facial expressions. One pair of the results is visualized in Fig. 4 This proves effective in discerning generated images with non-attacked images. We see that compared to StarGAN, this model's attack effusiveness is much lower. StarGAN models may achieve results that completely change the output color to black or white, but this only results in a change in facial generation. We suspect this is because of three reasons. Primarily, the training methods are trained to recreate original images from trained altered or warped images. Thus, because the training data was augmented, it is possible that this type of model would show resistance against perturbed results. Secondly, because it is an autoencoder model, the encoder results may reduce the effects of noise on the final generation, and changes within the encoder output may not affect the generator to produce output. Thirdly, because this model was trained extensively, sparse data may be a problem.

V. Conclusion

In our work, we propose three novel algorithms of image disruption to attack image classification models and image translation models. In these models, namely AdaGrad Iterative Fast Gradient Sign Method, RMSProp Iterative Fast Gradient Sign Method, and Adam Iterative Fast Gradient Sign Method, incorporate variable step size in Iterative Fast Gradient Sign Methods, a novel approach compared to previous attacks. We show that the variable step size feature in our novel methods increases attack efficacy by decreasing the number of iterations to converge, increasing generated image disruption amount or misclassification rate, and decreasing input image perturbation. Our extensive experiments in image classification attacks show stronger results than the previous state-of-the-art methods in white-box, black-box, and ensemble attacks. We show that with or without incorporating image augmentation methods, our model consistently outperforms vanilla and previous methods. Extensive experiments in image translation attacks show that our algorithm achieves similar efficacy on face changing algorithms. We show that, in image translation attacks, our algorithm of attack disrupts image generation more than previous state-of-the-art methods, while maintaining a higher similarity of the input to the original. Finally, we show that our novel algorithm achieves significant results when attacking deepfake models, showing potential in AI security defenses against malicious facial manipulation.

Acknowledgment

We would like to express Prof. Xie for offering his advice on using SSIM and PSNR as an evaluation matrix. Our appreciations are also extended to the Office of Academic Affairs at Affiliated High School of South China Normal University, which provides backups for our research. We would also like to extend our thanks to our fellow schoolmate Yudi Lu, who has talked us through the process of the entire competition based on his own experience. Moreover, we immensely appreciate our effort in this competition. We researched and read splendid and complicated papers during the competition period, learned new knowledge, and checked our duplication of other documents.

Additionally, we need to thank Prof. Chen for offering us her GPU servers to run our models and attacks on it, and Miss Wu for informing specific information of the competition and organizing the schedule to remind us to keep working hard on our paper. To the best of the team's acknowledgment, the paper does not contain research published or written by others, except as specified in the citation and acknowledgments. If there is any error, each of our team members is willing to assume all relevant responsibilities.

References

[1] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," *arXiv:1512.00567 [cs]*, Dec. 11, 2015. arXiv: 1512.00567. [Online]. Available: http://arxiv.org/abs/1512.00567 (visited on 09/12/2020).

[2] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo, "StarGAN: Unified generative adversarial networks for multi-domain image-to-image translation," *arXiv:1711.09020 [cs]*, Sep. 21, 2018. arXiv: 1711.09020. [Online]. Available: http://arxiv.org/abs/1711.09020 (visited on 08/31/2020).

[3] joshua-wu, *Joshua-wu/deepfakes_faceswap*, original-date: 2017-12-15T11:45:52Z, Sep. 13, 2020. [Online]. Available: https://github.com/joshua-wu/deepfakes_faceswap (visited on 09/13/2020).

[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, May 24, 2017, ISSN: 0001-0782, 1557-7317. DOI: 10.1145/3065386. [Online]. Available: https://dl.acm.org/doi/10.1145/3065386 (visited on 09/13/2020).

[5] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, L. Wang, G. Wang, J. Cai, and T. Chen, "Recent advances in convolutional neural networks," *arXiv:1512.07108 [cs]*, Oct. 19, 2017. arXiv: 1512.07108. [Online]. Available: http://arxiv.org/abs/1512.07108 (visited on 09/13/2020).

[6] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *arXiv:1406.2661 [cs, stat]*, Jun. 10, 2014. arXiv: 1406.2661. [Online]. Available: http://arxiv.org/abs/1406.2661 (visited on 09/13/2020).

[7] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, "Analyzing and improving the image quality of StyleGAN," *arXiv:1912.04958 [cs, eess, stat]*, Mar. 23, 2020. arXiv: 1912.04958. [Online]. Available: http://arxiv.org/abs/1912.04958 (visited on 09/13/2020).

[8] (). "Shaoanlu/faceswap-GAN: A denoising autoencoder + adversarial losses and attention mechanisms for face swapping.," [Online]. Available: https://github.com/shaoanlu/faceswap-GAN (visited on 09/13/2020).

[9] M. Nuñez. (). "Snapchat and TikTok embrace 'deepfake' video technology even as facebook shuns it," Forbes, Section: Innovation, [Online]. Available: https://www.forbes.com/sites/mnunez/2020/01/08/snapchat-and-tiktok-embrace-deepfake-video-technology-even-as-facebook-shuns-it/ (visited on 09/14/2020).

[10] (). "Snapchat cameos edit your face into videos," TechCrunch, [Online]. Available: https://social.techcrunch.com/2019/12/08/snapchat-cameo-edits-your-face-into-videos/ (visited on 09/14/2020).

[11] C. Öhman, "Introducing the pervert's dilemma: A contribution to the critique of deepfake pornography," *Ethics and Information Technology*, vol. 22, no. 2, pp. 133–140, Jun. 2020, ISSN: 1388-1957, 1572-8439. DOI: 10.1007/s10676-019-09522-1. [Online]. Available: http://link.springer.com/10.1007/s10676-019-09522-1 (visited on 09/13/2020).

[12] "'deepfake' app causes fraud and privacy fears in china," *BBC News*, Sep. 4, 2019. [Online]. Available: https://www.bbc.com/news/technology-49570418 (visited on 09/15/2020).

[13] R. Tolosana, R. Vera-Rodriguez, J. Fierrez, A. Morales, and J. Ortega-Garcia, "DeepFakes and beyond: A survey of face manipulation and fake detection," *arXiv:2001.00179 [cs]*, Jun. 18, 2020. arXiv: 2001.00179. [Online]. Available: http://arxiv.org/abs/2001.00179 (visited on 09/13/2020).

[14] D. Güera and E. J. Delp, "Deepfake video detection using recurrent neural networks," in *2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, Nov. 2018, pp. 1–6. DOI: 10.1109/AVSS.2018.8639163.

[15] Y. Li and S. Lyu, "Exposing DeepFake videos by detecting face warping artifacts," p. 7,

[16] R. Tolosana, R. Vera-Rodriguez, J. Fierrez, A. Morales, and J. Ortega-Garcia, "DeepFakes and beyond: A survey of face manipulation and fake detection," *arXiv:2001.00179 [cs]*, Jun. 18, 2020, version: 3. arXiv: 2001.00179. [Online]. Available: http://arxiv.org/abs/2001.00179 (visited on 08/30/2020).

[17] P. Neekhara, S. Hussain, M. Jere, F. Koushanfar, and J. McAuley, "Adversarial deepfakes: Evaluating vulnerability of deepfake detectors to adversarial examples," *arXiv:2002.12749 [cs]*, Mar. 13, 2020. arXiv: 2002.12749. [Online]. Available: http://arxiv.org/abs/2002.12749 (visited on 09/13/2020).

[18] C. Yang, L. Ding, Y. Chen, and H. Li, "Defending against GAN-based deepfake attacks via transformation-aware adversarial faces," *arXiv:2006.07421 [cs, eess]*, Jun. 12, 2020. arXiv: 2006.07421. [Online]. Available: http://arxiv.org/abs/2006.07421 (visited on 09/13/2020).

[19] N. Ruiz, S. A. Bargal, and S. Sclaroff, "Disrupting deepfakes: Adversarial attacks against conditional image translation networks and facial manipulation systems," *arXiv:2003.01279 [cs]*, Apr. 27, 2020. arXiv: 2003.01279. [Online]. Available: http://arxiv.org/abs/2003.01279 (visited on 09/02/2020).

[20] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv:1412.6572 [cs, stat]*, Mar. 20, 2015. arXiv: 1412.6572. [Online]. Available: http://arxiv.org/abs/1412.6572 (visited on 09/13/2020).

[21] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo, "StarGAN: Unified generative adversarial networks for multi-domain image-to-image translation," *arXiv:1711.09020 [cs]*, Sep. 21, 2018. arXiv: 1711.

09020. [Online]. Available: http://arxiv.org/abs/1711.09020 (visited on 09/13/2020).

[22] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv:1412.6572 [cs, stat]*, Mar. 20, 2015. arXiv: 1412.6572. [Online]. Available: http://arxiv.org/abs/1412.6572 (visited on 09/12/2020).

[23] A. Demontis, M. Melis, M. Pintor, M. Jagielski, B. Biggio, A. Oprea, C. Nita-Rotaru, and F. Roli, "Why do adversarial attacks transfer? explaining transferability of evasion and poisoning attacks," *arXiv:1809.02861 [cs, stat]*, Jun. 13, 2019. arXiv: 1809.02861. [Online]. Available: http://arxiv.org/abs/1809.02861 (visited on 09/13/2020).

[24] S. Bhambri, S. Muku, A. Tulasi, and A. B. Buduru, "A survey of black-box adversarial attacks on computer vision models," *arXiv:1912.01667 [cs, stat]*, Feb. 7, 2020. arXiv: 1912.01667. [Online]. Available: http://arxiv.org/abs/1912.01667 (visited on 09/13/2020).

[25] J. Lin, C. Song, K. He, L. Wang, and J. E. Hopcroft, "Nesterov accelerated gradient and scale invariance for adversarial attacks," Aug. 17, 2019. [Online]. Available: https://arxiv.org/abs/1908.06281v5 (visited on 08/30/2020).

[26] J. Liu, "Iterative ensemble adversarial attack," p. 5,

[27] M. A. A. Milton, "Evaluation of momentum diverse input iterative fast gradient sign method (m-DI2-FGSM) based attack method on MCS 2018 adversarial attacks on black box face recognition system," *arXiv:1806.08970 [cs]*, Jun. 23, 2018. arXiv: 1806.08970. [Online]. Available: http://arxiv.org/abs/1806.08970 (visited on 09/13/2020).

[28] Y. Dong, T. Pang, H. Su, and J. Zhu, "Evading defenses to transferable adversarial examples by translation-invariant attacks," *arXiv:1904.02884 [cs]*, Apr. 5, 2019. arXiv: 1904.02884. [Online]. Available: http://arxiv.org/abs/1904.02884 (visited on 08/30/2020).

[29] Y. Dong, F. Liao, T. Pang, H. Su, J. Zhu, X. Hu, and J. Li, "Boosting adversarial attacks with momentum," *arXiv:1710.06081 [cs, stat]*, Mar. 22, 2018. arXiv: 1710.06081. [Online]. Available: http://arxiv.org/abs/1710.06081 (visited on 09/02/2020).

[30] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *The Journal of Machine Learning Research*, vol. 12, pp. 2121–2159, null Jul. 1, 2011, ISSN: 1532-4435.

[31] T. Tieleman and G. Hinton, "Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude," *COURSERA: Neural networks for machine learning*, vol. 4, no. 2, pp. 26–31, 2012.

[32] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv:1412.6980 [cs]*, Jan. 29, 2017. arXiv: 1412.6980. [Online]. Available: http://arxiv.org/abs/1412.6980 (visited on 09/12/2020).

[33] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," *arXiv:1607.02533 [cs, stat]*, Feb. 10, 2017. arXiv: 1607.02533. [Online]. Available: http://arxiv.org/abs/1607.02533 (visited on 09/12/2020).

[34] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," *arXiv:1706.06083 [cs, stat]*, Sep. 4, 2019. arXiv: 1706.06083. [Online]. Available: http://arxiv.org/abs/1706.06083 (visited on 09/12/2020).

[35] C. Xie, Z. Zhang, Y. Zhou, S. Bai, J. Wang, Z. Ren, and A. Yuille, "Improving transferability of adversarial examples with input diversity," *arXiv:1803.06978 [cs, stat]*, Jun. 1, 2019. arXiv: 1803.06978. [Online]. Available: http://arxiv.org/abs/1803.06978 (visited on 09/12/2020).

[36] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," *arXiv:1608.04644 [cs]*, Mar. 22, 2017. arXiv: 1608.04644. [Online]. Available: http://arxiv.org/abs/1608.04644 (visited on 08/30/2020).

[37] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds., Curran Associates, Inc., 2014, pp. 2672–2680. [Online]. Available: http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf (visited on 09/13/2020).

[38] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro, "High-resolution image synthesis and semantic manipulation with conditional GANs," *arXiv:1711.11585 [cs]*, Aug. 20, 2018. arXiv: 1711.11585. [Online]. Available: http://arxiv.org/abs/1711.11585 (visited on 09/13/2020).

[39] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," Mar. 30, 2017. [Online]. Available: https://arxiv.org/abs/1703.10593v7 (visited on 08/31/2020).

[40] A. Pumarola, A. Agudo, A. M. Martinez, A. Sanfeliu, and F. Moreno-Noguer, "GANimation: Anatomically-aware facial animation from a single image," *arXiv:1807.09251 [cs]*, Aug. 28, 2018. arXiv: 1807.09251. [Online]. Available: http://arxiv.org/abs/1807.09251 (visited on 09/12/2020).

[41] C.-Y. Yeh, H.-W. Chen, S.-L. Tsai, and S.-D. Wang, "Disrupting image-translation-based DeepFake algorithms with adversarial attacks," in *Proceedings of the IEEE winter conference on applications of computer vision workshops*, 2020, pp. 53–62.

[42] (). "Deep learning," [Online]. Available: http://www.deeplearningbook.org/ (visited on 09/15/2020).

[43] G. E. Hinton and R. S. Zemel, "Autoencoders, minimum description length and helmholtz free energy," p. 8,

[44] W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang, "Real-time single image and video super-resolution using an efficient subpixel convolutional neural network," *arXiv:1609.05158 [cs, stat]*, Sep. 23, 2016. arXiv: 1609.05158. [Online]. Available: http://arxiv.org/abs/1609.05158 (visited on 09/15/2020).

[45] E. F. Krause, *Taxicab Geometry: an adventure in non-Euclidean geometry*. New York: Dover Publications, 1987, 88 pp., ISBN: 978-0-486-25202-5.

[46] E. L. Lehmann and G. Casella, *Theory of point estimation*, 2nd ed, ser. Springer texts in statistics. New York: Springer, 1998, 589 pp., ISBN: 978-0-387-98502-2.

[47] A. Hore and D. Ziou, "Image quality metrics: PSNR vs. SSIM," in *2010 20th international conference on pattern recognition*, tex.organization: IEEE, 2010, pp. 2366–2369.

[48] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inception-v4, inception-ResNet and the impact of residual connections on learning," *arXiv:1602.07261 [cs]*, Aug. 23, 2016. arXiv: 1602.07261. [Online]. Available: http://arxiv.org/abs/1602.07261 (visited on 09/12/2020).

[49] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," *arXiv:1603.05027 [cs]*, Jul. 25, 2016. arXiv: 1603.05027. [Online]. Available: http://arxiv.org/abs/1603.05027 (visited on 09/12/2020).

[50] F. Tramèr, A. Kurakin, N. Papernot, I. Goodfellow, D. Boneh, and P. McDaniel, "Ensemble adversarial training: Attacks and defenses," *arXiv:1705.07204 [cs, stat]*, Apr. 26, 2020. arXiv: 1705.07204. [Online]. Available: http://arxiv.org/abs/1705.07204 (visited on 09/12/2020).

APPENDIX

A. Detailed Algorithm

Here, we summarize SI-AI-TI-DIM method in Algorithm 2. Removing Step 10 and $S_i(\cdot)$ in Step 6 and 7 turns it into AI-DIM attack, removing Step 10 and $Di(\cdot;p)$ in Step 6 and 7 turns it into SI-AI=FGSM attack, and replacing Step 6 and 7 with Eq.25 turns it into AI-TIM attack.

Algorithm 2: SI-AI-TI-DIM

Input: A clean example x with ground-truth label y^{true};a classifier f with loss function J; Hyper-parameters: Perturbation size ϵ, maximum iterations T; diverse input translator Di with probability of apply random padding p; number of scale copies m; a gaussian kernel $\mathbf{W}$; decay factor β_1, β_2

Output: An adversarial example x^{adv}

1 $\alpha = \epsilon/T$;
2 $s_0 = 0, g'_0 = 0, x_0^{adv} = x$;
3 **for** $t = 0$ *to* $T-1$ **do**
4 $g_t = 0$;
5 **for** $i = 0$ *to* $m-1$ **do**
6 Fetch gradients by $\nabla_x \mathcal{J}(Di(S_i(x_t^{adv});p), y^{true})$;
7 Sum all the gradients as $g_t = g_t + \nabla_x \mathcal{J}(Di(S_i(x_t^{adv});p), y^{true})$;
8 **end**
9 Average the gradients as $g_t = \frac{g_t}{m}$;
10 Convolve the gradients by $g_t = \mathbf{W} * g_t$;
11 Update s_{t+1} and g'_{t+1} by $g'_{t+1} = \beta_1 g'_t + (1-\beta_1)\frac{g_t}{\|g_t\|_1}$, ; $s_{t+1} = \beta_2 s_t + (1-\beta_2) g_t \odot g_t$,
12 Perform bias correction using $\hat{\boldsymbol{g}'}_{t+1} = \frac{g'_{t+1}}{1-\beta_1^t}, \hat{\boldsymbol{s}}_{t+1} = \frac{s_{t+1}}{1-\beta_2^t}$,;
13 Update x_{t+1}^{adv} by Eq.28;
14 **end**
15 Return $x^{adv} = x_T^{adv}$

B. More Visualization of Attacking StarGAN and Faceswap GAN

Here, we randomly select and showcase a group of original input images, their corresponding adversarial examples, their disrupted outputs and their undisturbed outputs in Fig.6. The adversarial images are crafted on face-swap autoencoder using AI-FGSM. We see that there is humanly perceptible distortion towards the individual's eyes.

C. More Visualization of Attacking Image Classification Models

We also randomly select 4 original input images, and their adversarial examples in Fig.7, which are generated on an ensemble of models using SI-AI-TI-DIM. We see that the perturbations made by AI-FGSM are imperceptible.

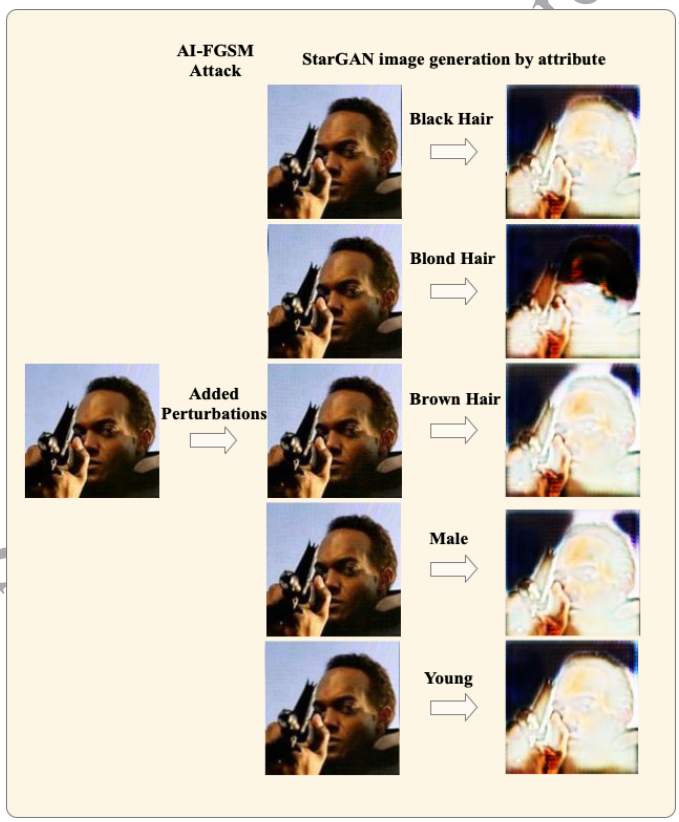


Fig. 5: **An adversarial example successfully defended by the perturbation of AI-FGSM from StarGAN.** The left column is the original image, the middle column is the adversarial input images, and the righted column is the disrupted output images. We see that there is imperceptible perturbation on the adversarial inputs, but obvious distortions in the generated images.

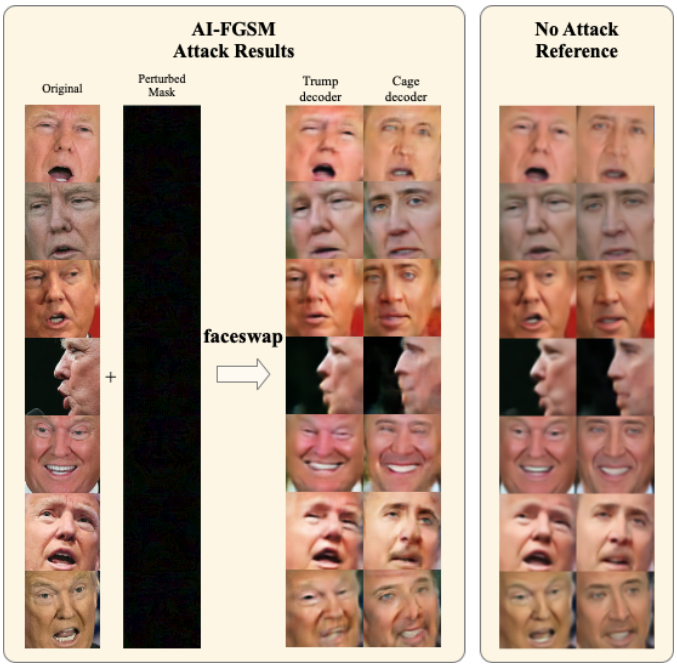


Fig. 6: **Randomly selected adversarial examples on faceswap autoencoder using AI-FGSM.** The first column is the original input images, the second column is the perturbed masks that are added to the original, the third and fourth are the disturbed outputs, and the last two are the undisturbed outputs, as a reference.

Fig. 7: **Randomly selected adversarial examples on an ensemble of models using SI-AI-TI-DIM.** The upper row refers to the original images, and the lower row refers to the adversarial images.