Automated generation of adaptive perturbed images based on GAN for motivated adversaries on deep learning models.

Duy Trung Pham
Department of Information Security -
Academy of Cryptography Techniques
Ha Noi, Vietnam
trungpd@actvn.edu.vn

Cong Thanh Nguyen
Department of Information Security -
Academy of Cryptography Techniques
Ho Chi Minh, Vietnam
thanhnc1212@gmail.com

Phi Ho Truong*
Department of Information Security -
Academy of Cryptography Techniques
Ha Noi, Vietnam
hotp.gvm@actvn.edu.vn

Nhat Hai Nguyen
School of Information and Communication Technology -
Hanoi University of Science and Technology
Ha Noi, Vietnam
hainn@soict.hust.edu.vn

Abstract

Deep learning techniques have achieved great success in many fields, such as computer vision, natural language processing and computer security. However, deep learning models are facing many security risks, in particular motivated adversaries which lead to incorrect predictions or reduce the models' effectiveness. Many previous researches have been investigated on adversarial attacks, aiming to improve the robustness and security of deep learning models. In this article, we propose a method to automatically produce adaptive perturbed images based on GAN for motivated adversaries on deep learning models. The experimental results achieve approximately 60% success rate in evading five state-of-the-art deep learning models for image recognition including ResNet-56, MobileNetV2, VGG19_bn, ShuffleNetV2, RepVGG_a2 on the CIFAR-10 dataset. The proposed method's results are much higher than AIGAN model proposed by Tao Bai et al. achieving 10.17%. The image fidelity of distorted images generated by the proposed method also is positive. The proposed method's PSNR (Peak-Signal-Noise-Ratio) is greater than 40 compared to previous studies such as FGSM, DeepFoll, C&W and AdvGAN, Zhang in which PSNR is less than 30.

Keywords: Deep learning, Generative Adversarial Networks, Adversarial attacks.

ACM Reference Format:
Duy Trung Pham, Cong Thanh Nguyen, Phi Ho Truong, and Nhat Hai Nguyen. 2023. Automated generation of adaptive perturbed images based on GAN for motivated adversaries on deep learning models.. In *Proceedings of The 12th International Symposium on Information and Communication Technology (SOICT'23).* ACM, New York, NY, USA, 8 pages. https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

Adversarial attacks on machine learning are to fool models with perturbed data. Adversarial attacks on machine learning involve generating adversarial examples that can deceive the classifier in a machine learning model. Gartner, an industry-leading market research firm, advises to anticipate and prepare to minimize the potential risks of data corruption, model theft, and adversarial examples [6]. This could be used by an attacker to attack neural networks in object recognition, detecting standard adversarial examples in autonomous (or even semi-autonomous) vehicles, possibly causing catastrophic consequences [4, 28, 40].

Adversarial attacks can make a serious vulnerability in machine learning systems. Adversarial attacks degrade the performance of classifiers on specific tasks. It affects many areas such as medical in the health care system [29], visual classification [13], information security [38, 46], financial forecasting and prediction [2, 35], etc. It is dangerous if an image of a traffic sign is misidentified [19], leading to a malfunction of the autopilot. Or medical diagnostic images have incorrect results due to incorrect prediction and classification models [17].

Adversarial attacks can be divided into three groups: attacks based on the results the attacker wants, attacks based on the amount of knowledge the attacker has about the

*The author is the corresponding author

SOICT'23, December 07-08, 2023, Ho Chi Minh, Vietnam

ACM ISBN 978-1-4503-XXXX-X/18/06...$15.00
https://doi.org/XXXXXXX.XXXXXXX

model, and attacks using how to provide machine learning model input data. Approach toward knowledge about the model, adversarial attacks can be divided two categorises:

- **Black-box Attacks**: The attacker does not have the information, or configuration inside the model, or only grasp the preliminary information about the machine learning model [21, 23, 32].
- **White-box Attacks**: In a white-box attack, the attacker has full knowledge of the model including the model type, model architecture, and the values of all training parameters and weights [21, 45].

Black-box adversarial attacks can be generated on an alternate model and then applied to attack the target model. Hang et al. [16] propose two types of set-based black-box attacks and vulnerability discovery in deep learning systems. In addition, adversarial examples can be produced in many different ways such as Limited-memory BFGS (L-BFGS) [48], Fast Gradient Sign Method (FGSM) [14], Jacobian-based Saliency Map Attack (JSMA) [41], Deepfool Attack [36], Carlini & Wagner (C&W) attack [41], and Zeroth-order optimization attack (ZOO) [7]. Generative Adversarial Networks (GANs) have been used to generate adversarial attacks.

In order to perform a targeted black-box attack to evaluate typical deep learning models such as ResNet-56 [43], MobileNetV2 [33], VGG19 [34], ShuffleNetV2 [25], RepVGG [10]. We propose a method to automatically produce adaptive perturbed images based on GAN for motivated adversaries on deep learning models. The successful attacks rate achieving 60% is higher than previous studies as AIGAN [3], FGSM [14], C&W [5, 24], AdvGAN [42]. The fidelity of adversarial images generated by the proposed method such as PSNR (Peak Signal Noise Ratio) much better than FGSM, DeepFoll, C&W, AdvGAN and proposed by Zhang [32, 49].

The rest of the paper is organized as follows. We first review the related work in Section 2. Section 3 presents the proposed GAN-based model for generating motivated adversaries. Section 4 mentions to experiment and results. We conclude the paper with a conclusion and future works in Section 5.

2 Related work

Some possible survey adversarial attack methods are listed below. L-BFGS method [48] is a non-linear gradient-based numerical optimization algorithm that minimizes the amount of perturbation added to the image. L-BFGS method is effective in creating adversarial examples, but it has complex calculations because of constraints, making it time-consuming and impractical. The C&W attack is based on the L-BFGS attack without the box constraint and different objective functions [12, 44]. This makes the method more effective in generating counterexamples; it has been proven to be able to defeat modern defense systems. C&W attack is also cost due to calculation intensive. Goodfellow et al. propose the FGSM method [14] not only changes our perspective of how machine learning works but also exposes real-world issues affecting promising applications. The FGSM method is used to generate adversarial examples to minimize the maximum amount of perturbation added to any pixel of the image to cause misclassification. It calculates relatively quickly, but it may introduce high turbulence due to its aggressive nature.

The Deepfool Attack [36] is used to minimize the Euclidean distance between the disturbed samples and the original samples. Decision boundaries between classes are estimated and perturbations are added iteratively. However, it takes time to calculate due to the repetitive process. To address the limitations of previous methods that introduced perturbations to images using predefined algorithms, GAN-based approach for generating adversarial images through automatic data generation were studied. However, traditional GAN models do not offer explicit control over the generated data. In other words, data generation is unsupervised, and the classification of the data is entirely random.

To overcome these limitations, conditional GANs was introduced by Mirza and colleagues [26]. In conditional GANs, additional information, such as class labels, is incorporated into the training process to produce predictable outputs. Subsequent advancements like infoGAN [8] and ACGAN [30] have further refined GANs, allowing for fine-grained control over the generated data using complex structures. Today, modern GANs have improved significantly compared to basic GANs. Zhang et al. proposed Stack Generative Adversarial Network (StackGAN) to generate realistic images of size 256×256 based on text description [47]. In this method, they used multi-stage Gan to improve image quality from low to high resolution. Tao Bai et al. introduced a new framework called Attack Inspired GAN (AI-GAN), in which the creator, the discriminator, and the attacker are trained together[3]. However, the successful attack results as announced are only above 10%. By directly generating adversarial examples from given input images, Zhang's method [49] generates perturbations that better match the edges and underlying shapes present in the input, but the images are evaluated fairly differently through the indicators in the research.

By employing conditional GANs, our approach to creates adversarial images with greater control and more predictability. The generated images will be used as adversarial examples to invade state-of-art classification models.

3 The proposed GAN-based model for generating motivated adversaries.

We create adaptive perturbation and combine it with original image at a scale of k ratio into adversarial image:

$$x_{adv} = x \oplus (k \times pert) \quad (1)$$

where x_{adv} is an adversarial image, x is origin image, $pert$ is the noise mask generated by proposed GAN-based model with generator and discriminator. This adversarial image is used to fool the target model in order to misidentify image class with false label as Figure 1.

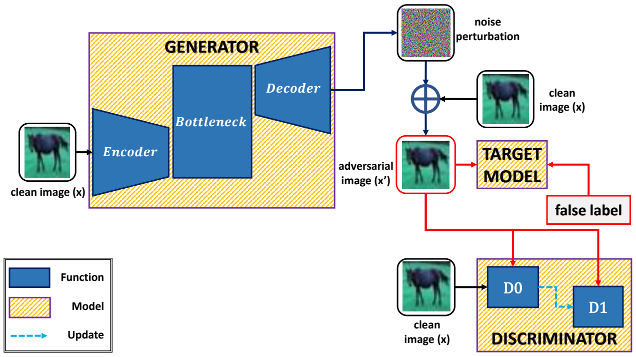


Figure 1. The GAN-based model for generating motivated adversaries.

3.1 The proposed Generator

The generator network has three main components: Encoder, Bottleneck, and Decoder as Figure 2.

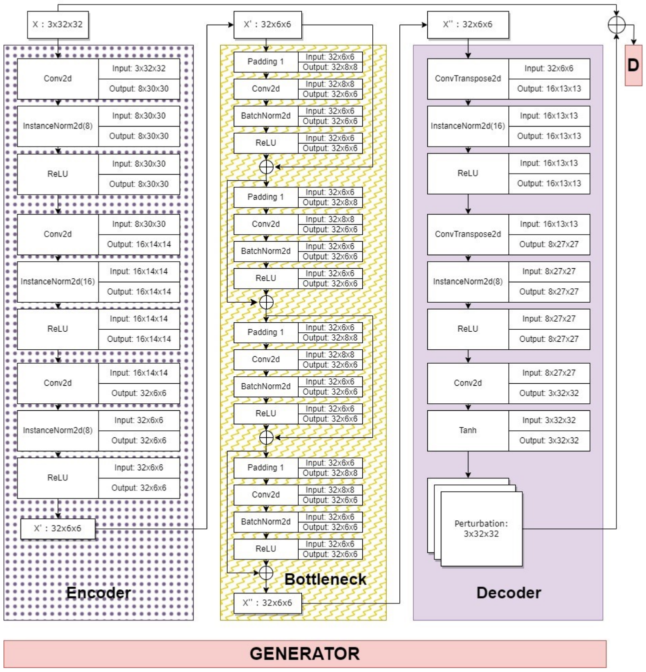

X : 3x32x32
Conv2d Input: 3x32x32 Output: 8x30x30
InstanceNorm2d(8) Input: 8x30x30 Output: 8x30x30
ReLU Input: 8x30x30 Output: 8x30x30
Conv2d Input: 8x30x30 Output: 16x14x14
InstanceNorm2d(16) Input: 16x14x14 Output: 16x14x14
ReLU Input: 16x14x14 Output: 16x14x14
Conv2d Input: 16x14x14 Output: 32x6x6
InstanceNorm2d(8) Input: 32x6x6 Output: 32x6x6
ReLU Input: 32x6x6 Output: 32x6x6
X' : 32x6x6
Encoder
X' : 32x6x6
Padding 1 Input: 32x6x6 Output: 32x8x8
Conv2d Input: 32x8x8 Output: 32x6x6
BatchNorm2d Input: 32x6x6 Output: 32x6x6
ReLU Input: 32x6x6 Output: 32x6x6
Padding 1 Input: 32x6x6 Output: 32x8x8
Conv2d Input: 32x8x8 Output: 32x6x6
BatchNorm2d Input: 32x6x6 Output: 32x6x6
ReLU Input: 32x6x6 Output: 32x6x6
Padding 1 Input: 32x6x6 Output: 32x8x8
Conv2d Input: 32x8x8 Output: 32x6x6
BatchNorm2d Input: 32x6x6 Output: 32x6x6
ReLU Input: 32x6x6 Output: 32x6x6
Padding 1 Input: 32x6x6 Output: 32x8x8
Conv2d Input: 32x8x8 Output: 32x6x6
BatchNorm2d Input: 32x6x6 Output: 32x6x6
ReLU Input: 32x6x6 Output: 32x6x6
Bottleneck
X'' : 32x6x6
X'' : 32x6x6
ConvTranspose2d Input: 32x6x6 Output: 16x13x13
InstanceNorm2d(16) Input: 16x13x13 Output: 16x13x13
ReLU Input: 16x13x13 Output: 16x13x13
ConvTranspose2d Input: 16x13x13 Output: 8x27x27
InstanceNorm2d(8) Input: 8x27x27 Output: 8x27x27
ReLU Input: 8x27x27 Output: 8x27x27
Conv2d Input: 8x27x27 Output: 3x32x32
Tanh Input: 3x32x32 Output: 3x32x32
Perturbation: 3x32x32
Decoder
D
GENERATOR

Figure 2. The proposed Generator architecture.

The loss function in training of network G consists of three parts:

1. $\mathcal{L}_{Generator\_pert}$ is the loss of the noise mask. The function $\mathcal{L}_{Generator\_pert}$ are calculated as Equation (2):

$$\mathcal{L}_{Generator\_pert} = \text{Average}(\text{Max}_{0 \leq i \leq n}(pert(i) - \lambda; 0)) \quad (2)$$

where λ is the chaos threshold in model training process.

2. $\mathcal{L}_{Discriminator\_fake} = \mathcal{L}_{D1}$ (Equation (7)) is output of Discriminator
3. $\mathcal{L}_{adv}$ is loss score between true targets and adversarial prediction, and $\mathcal{L}_{adv}$ calculated as Equation (3) (The sum of x_{adv} maximum value case i minus x case i compared to 0).

$$\mathcal{L}_{adv} = \sum(\text{Max}_{0 \leq i \leq n}(x_{adv} - x; 0)) \quad (3)$$

The goal of the Generator is to minimize:

$$\mathcal{L}_G = \gamma\mathcal{L}_{Generator\_pert} + \alpha\mathcal{L}_{adv} + \beta\mathcal{L}_{Discriminator\_fake} \quad (4)$$

With γ, α, β are parameters that represent the importance of the loss function corresponding to it. We tune γ, α, β according to research by Tongxin et al. [18].

3.2 The proposed Discriminator

The Discriminator network has two main components are $D0$ and $D1$. $D0$ has the function to update the parameters of the Convolutional Neural Network structure including Convtranspose2d, InstanceNorm2d, Conv2d (see in [15]) of $D1$ as Figure 3.

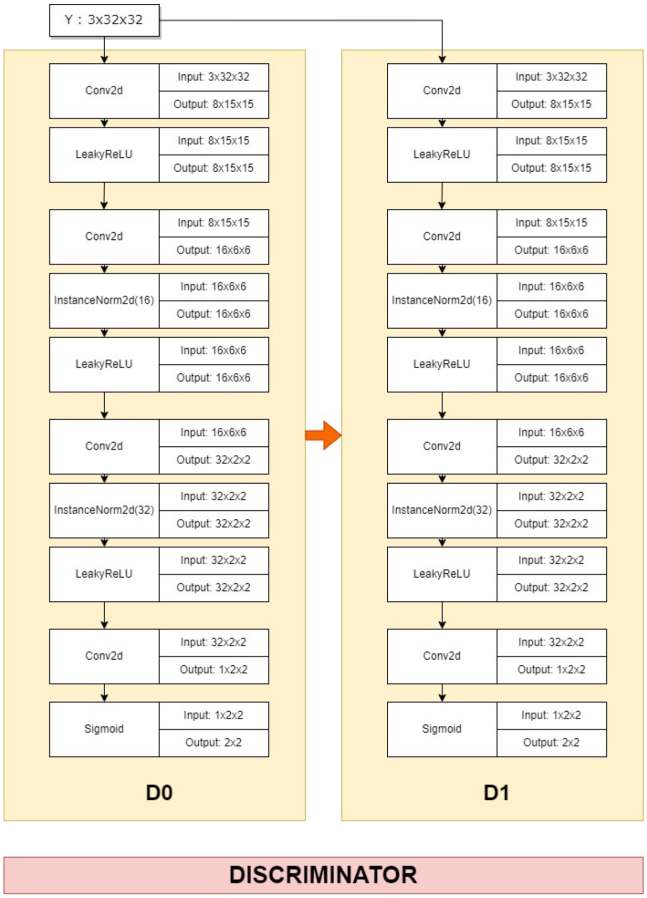

Y : 3x32x32
Conv2d Input: 3x32x32 Output: 8x15x15
LeakyReLU Input: 8x15x15 Output: 8x15x15
Conv2d Input: 8x15x15 Output: 16x6x6
InstanceNorm2d(16) Input: 16x6x6 Output: 16x6x6
LeakyReLU Input: 16x6x6 Output: 16x6x6
Conv2d Input: 16x6x6 Output: 32x2x2
InstanceNorm2d(32) Input: 32x2x2 Output: 32x2x2
LeakyReLU Input: 32x2x2 Output: 32x2x2
Conv2d Input: 32x2x2 Output: 1x2x2
Sigmoid Input: 1x2x2 Output: 2x2
D0
Conv2d Input: 3x32x32 Output: 8x15x15
LeakyReLU Input: 8x15x15 Output: 8x15x15
Conv2d Input: 8x15x15 Output: 16x6x6
InstanceNorm2d(16) Input: 16x6x6 Output: 16x6x6
LeakyReLU Input: 16x6x6 Output: 16x6x6
Conv2d Input: 16x6x6 Output: 32x2x2
InstanceNorm2d(32) Input: 32x2x2 Output: 32x2x2
LeakyReLU Input: 32x2x2 Output: 32x2x2
Conv2d Input: 32x2x2 Output: 1x2x2
Sigmoid Input: 1x2x2 Output: 2x2
D1
DISCRIMINATOR

Figure 3. The proposed Discriminator architecture.

The loss function in training of network D consists of three parts:

1. $\mathcal{L}_D$ is real/noisy image distinction score, S for discriminating real/perturbed images, $\mathcal{L}_D$ calculated according to Equation (5):

$$\mathcal{L}_D = E[\log P(S = real \mid x_{real})] + k \times E[\log P(S = pert \mid x_{pert})] \tag{5}$$

2. $\mathcal{L}_{D0}$ is value to distinguish real image and noise mask, $\mathcal{L}_{D0}$ calculated according to Equation (6):

$$\mathcal{L}_{D0} = E[\log P(C = y \mid x_{pert})] \tag{6}$$

where y represents the original label, and a target classifier C.

3. $\mathcal{L}_{D1}$ is value of classification on the noise masks generated by the attacker and the G network. Set d_fake is result of the model $D0$, the vector displays the results of 128 images. d_fake is a vector 1×128, which includes the values 0 is label of adversarial image and 1 is label of clean image (for example: d_fake has form [1 0 1...1 1 0]). $\mathcal{L}_{D1}$ is the Euclidean distance of d_fake to the vector 1×128 contains all value of 1:

$$\mathcal{L}_{D1} = \sqrt{(\text{d\_fake} - [1\ 1\ 1...1\ 1\ 1])^2} \tag{7}$$

The goal of Discriminator is to maximize$(\mathcal{L}_D + \mathcal{L}_{D0})$ and minimize $\mathcal{L}_{D1}$.

3.3 Adversarial example generation

Each interaction with 128 images input into Generator produces the noise mask called $pert$. We combine noise mask with the original image (according to equation (1)) to create adversarial example. Adversarial example is caculated with Discriminator to update state of Discriminator and Generator. With the cost T is equal total images used for adversarial image generation divided 128, we use Algorithm 1 to generate an adversarial example.

Algorithm 1: Adversarial example generation

```
Input: $x$
Output: AE (Adversarial example)
$T$ = Total images / 128
$iter$ = 1;
while $(iter < T)$ do
    $pert$ = Generator(iter) works through 3 steps (as Figure 1):
            1. Encoder(iter);
            2. Bottleneck(Encoder);
            3. Decoder(Bottleneck);
    $AE = x_{adv} = x \oplus (k \times pert)$ ;
    if $(Discriminator(AE, x) == 0)$ then
        Update Discriminator
    end
    if $(Discriminator(x, AE) == 1)$ then
        Update Discriminator
    end
end
```

4 Experiments and Results

4.1 Experiments

We perform experiments to evaluate the proposed model on CIFAR-10 dataset [1, 20], which consists of 60,000 color images size at 32× 32 pixels in 10 classes. The proposed model is implemented by Anaconda Python 3.11.3, Pytorch 1.8 framework, and CUDA 10.1 library on the computer with CPU configuration I9 9900, 64GB RAM and GPU RTX2080 8GB VRAM. Our proposed model produces adversarial images in attacking to five state-of-the-art deep learning models for image recognition including ResNet-56 [43], MobileNetV2 [33], VGG19_bn [34], ShuffleNetV2 [25], RepVGG_a2 [10].

To generate adversarial examples, the model under consideration underwent training with a batch size of 128 and input images sized at 32×32. The model was equipped with a chaos threshold (λ) set at 0.5, aiming to restrict the disturbance in the loss function during and after each $iter = 128$. After tunnings, we choose $\alpha = 0.2$, $\beta = 0.6$, $\gamma = 0.2$ as the best value for our model. The model converges after 600 epochs as Figure 4. In testing phase, the proposed Generator produces adversarial examples with ratio k in range 0.1 to 1.

To examine the effect of proposed model, we use the rate successes of adversarial images to relabel the original images with target labels, P_s (%), the average of time in which the proposed model successfully creates an adversarial image T_{avg} (seconds). P_s is calculated as Equation (8):

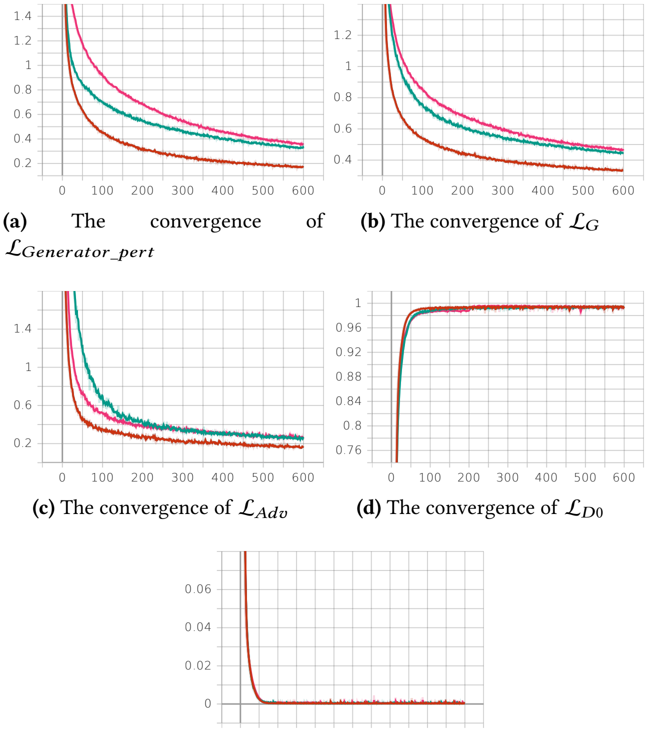
$$P_s = N_{in}/N_s \times 100 \tag{8}$$

where N_{in} is the number of images as input, and N_s is the number of adversarial images relabelling successfully the original images with target labels.

In addition, PSNR in Equation (9) is also investigated to evaluate the fidelity of adversarial images. A larger PSNR value indicates that the adversarial image more closely resembles its original image, meaning that the adversarial image has better imperceptibility. If the value of PSNR is large, it indicates that the noise or distortion due to the adversary is very small. Minimum PSNR of 40-50 dB is advised by Chen and Ramabadran [50].

$$PSNR = \frac{20 \log_{10} \max(x)}{\sqrt{\frac{1}{N} \sum_{n=1}^{N} (x - x')^2}} \tag{9}$$

where x is original image, x' is the transformed image. The results will be presented in section 4.2.



(a) The convergence of $\mathcal{L}_{Generator\_pert}$

(b) The convergence of $\mathcal{L}_G$

(c) The convergence of $\mathcal{L}_{Adv}$

(d) The convergence of $\mathcal{L}_{D0}$

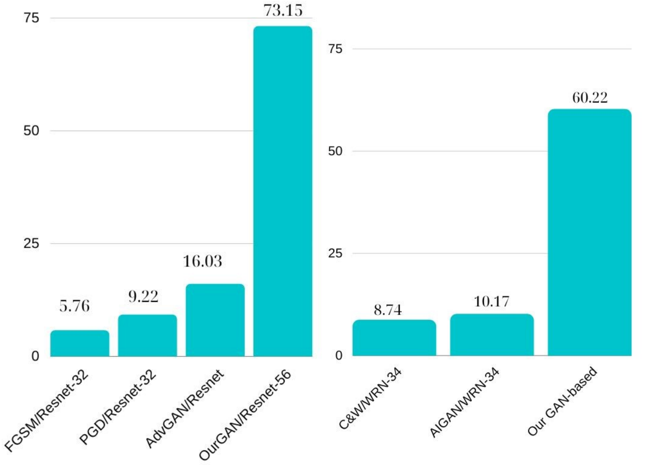
(e) The convergence of $\mathcal{L}_{D1}$

Figure 4. Visualization of the loss function convergence of the model after 600 epochs.

4.2 Results

Table 1 shows the success rates (%) of generating adversarial images by the proposed GAN model for 10 classes in the CIFAR-10 dataset. The average success rate for generating adversarial images achieved 73.15% is the highest rate for testing on the ResNet-56 model, 43.80% is the lowest rate for VGG19_bn. The average rate of experiments achieved 60.22% on five models. The authors highlight the lowest success rate values for generation of adversarial images for specific image classifiers on each model. It can be seen that the adversarial images generated results for the Frogs class are the lowest for the two VGG19_bn (22.58%) and ShuffleNetV2 (21.43%) models. This makes the rate of successful adversarial images generated for this model low. VGG19_bn and ShuffleNetV2 modes are more robust than the other 3 models in the attack experiment conducted by the authors.

We compare the average success rates (%) of adversarial images generated on each target model with AIGAN[3], AdvGAN [42], FGSM [14], C&W [5], and PGD [24] methods, the results are shown visually through the chart in Figure 5. Figure 5a shows that our proposed method's attack success rate on the Resnet model is higher than previous method as FGSM, PGD, AdvGAN, AIGAN. The average performance on other models is also better (show in Figure 5b).



(a) Attack success rates on ResNet model

(b) Attack success rates on other model

Figure 5. Visualization of attack success rates (%) of GAN-based on CIFAR-100 compares with previous method [3, 5, 14, 24, 42]. Our GAN-base looks lighter.

The authors chose value $k = 0.8$ for ShuffleNetV2, $k = 0.7$ for the remaining 4 models is the best value for balance between P_s, T_{avg} and PSNR as desired. We call k=0.7 and k=0.8 are ideal k. The results of Table 2 represent indices such as N_{in}, N_s, P_s, T_{avg}, $PSNR$ respectively k=0.1, ideal k, $k = 1$.

Results in Table 2 through the indicators identified above show that the proposed method has promising effectiveness. In particular, the number of adversarial samples generated on a test dataset is used to fool the original models such as ResNet-56, MobileNetV2, and RepVGG_a2 when choosing ideal k achieving higher than 60%. The difference between the generated adversarial images and the original image is relatively small as indicated by the $PSNR$ in Table 2. We also compare the average time to successfully generate an adversarial image, PSNR with previous studies such as FGSM, DeepFoll, C&W and AdvGAN, Zhang. The comparison results are shown in Table 3.

The results in Table 3 show average time successfully create an adversarial image and average success rate of adversarial images generated, PSNR is higher than other methods such as FGSM [14], AdvGAN [42], DeepFool [27], C&W[5, 31], and the model proposed by Zhang [49]. The average time to successfully generate an adversarial image is higher than previous studies, but our PSNR value is better.

5 Conclusion and Future works

In this study, we propose a GAN-based model to generate adversarial images that cause the target model to misclassify.

Table 1. The success rates (%) of adversarial images generated by proposed GAN-based on the dataset CIFAR-10 specifically in 10 classes with idea k. The lowest successful adversarial image generation rate is highlighted in bold.

Classification	**ResNet-56** [43]	**MobileNetV2** [33]	**VGG19_bn** [34]	**ShuffleNetV2** [25]	**RepVGG_a2** [10]
Airplanes	87.67	81.86	44.05	59.91	80.00
Cars	74.20	61.88	46.15	50.00	77.93
Birds	70.64	60.81	36.23	39.39	51.45
Cats	44.99	**24.60**	42.33	24.07	**31.79**
Deer	71.06	55.45	38.33	57.14	72.94
Dogs	85.80	89.29	60.99	62.05	74.77
Frogs	**62.62**	56.31	**22.58**	**21.43**	65.94
Horses	84.73	88.98	57.78	68.81	75.82
Ships	66.58	61.07	60.13	61.83	76.33
Trucks	83.16	68.48	29.41	56.25	84.86
Average	**73.15**	**64.87**	**43.80**	**50.09**	**69.18**

Table 2. Experiment results of the proposed GAN model with three k values on each selected model.

Target models	k	N_{in}	N_s	P_s	T_{avg}	PSNR
ResNet-56 [43]	0.1	5222	160	3.06	1.58	47.44
	0.7	5222	3691	**70.68**	**0.08**	**42.96**
	1	5222	4666	**89.35**	0.09	41.88
MobileNetV2 [33]	0.1	3691	65	1.76	2.54	47.20
	0.7	3691	2322	**62.91**	**0.08**	**43.12**
	1	3691	3060	**82.90**	0.07	42.09
VGG19_bn [34]	0.1	2886	45	1.56	2.44	46.37
	0.7	2886	1365	**47.30**	**0.08**	41.21
	1	2886	2129	**73.77**	0.06	40.02
ShuffleNetV2 [25]	0.1	3072	42	1.37	3.19	46.75
	0.8	3072	1041	**45.61**	**0.11**	**42.17**
	1	3072	1900	**61.85**	0.08	41.44
RepVGG_a2 [10]	0.1	4261	97	2.28	1.56	46.90
	0.7	4261	2812	**65.99**	**0.06**	**42.66**
	1	4261	3707	**87.00**	0.05	41.47

Table 3. Comparison of average time successfully creates an adversarial image (T_{avg}), PSNR, average success rate of adversarial images generated (P_s).

Method	T_{avg}	PSNR	P_s
FGSM [14]	0.00951	15.252	5.76%
DeepFool [27]	0.108	42.019	31%
C&W[5, 31]	3.98	32.955	8.74%
AdvGAN [42]	0.00502	15.252	15.96%
The model proposed by Zhang [49]	**0.00499**	28.775	**67**%
Average value of our GAN-based model	**0.082**	**42.424**	**60.22**%

The proposed model is trained and fine-tuned to perform targeted adversarial attacks as per our intent, achieving significant effectiveness while maintaining image quality. We compare our GAN-based approach with several black-box attack methods implemented in AIGAN, FGSM, C&W, PGD, AdvGAN, DeepFool, and model proposed by Zhang. With the architecture and training regimen, GAN-based outperforms the compared methods on the CIFAR-10 dataset. However, the selection of the number of epochs to conclude training and the process of model selection did not yield the expected results, primarily due to the limited number of training iterations resulting from time constraints and the extensive experimentation on multiple models. The authors aim to conduct training using a significantly larger number of epochs and with datasets containing larger image sizes.

Our future development direction involves working with larger image datasets, such as those with sizes of 64x64, 128x128, and 256x256 pixels. We also plan to expand our experiments to various datasets like ImageNET [9], MS-COCO [22], and evaluate the performance of the proposed model on different models like YOLO [37, 39], Inception-V3 [11] with the goal of assessing its effectiveness. Furthermore, we intend to train the model for longer durations and experiment with various parameters to create more effective adversarial data generation models than the previous results. This approach will allow us to continue improving and adapting our model to a wider range of applications and challenges in the future.

...

References

[1] Yehya Abouelnaga, Ola S Ali, Hager Rady, and Mohamed Moustafa. 2016. Cifar-10: Knn-based ensemble of classifiers. In *2016 International Conference on Computational Science and Computational Intelligence (CSCI)*. IEEE, 1192–1195.

[2] John Alberg and Zachary C Lipton. 2017. Improving factor-based quantitative investing by forecasting company fundamentals. *arXiv preprint arXiv:1711.04837* (2017).

[3] Tao Bai, Jun Zhao, Jinlin Zhu, Shoudong Han, Jiefeng Chen, Bo Li, and Alex Kot. 2021. Ai-gan: Attack-inspired generation of adversarial examples. In *2021 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2543–2547.

[4] Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, et al. 2016. End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316* (2016).

[5] 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.

[6] Pin-Yu Chen and Sijia Liu. 2023. Holistic adversarial robustness of deep learning models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 37. 15411–15420.

[7] Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, and Cho-Jui Hsieh. 2017. Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *Proceedings of the 10th ACM workshop on artificial intelligence and security*. 15–26.

[8] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. 2016. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. *Advances in neural information processing systems* 29 (2016).

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 248–255.

[10] Xiaohan Ding, Xiangyu Zhang, Ningning Ma, Jungong Han, Guiguang Ding, and Jian Sun. 2021. Repvgg: Making vgg-style convnets great again. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 13733–13742.

[11] Na Dong, Li Zhao, Chun-Ho Wu, and Jian-Fang Chang. 2020. Inception v3 based cervical cell classification combined with artificially extracted features. *Applied Soft Computing* 93 (2020), 106311.

[12] Chuan Du, Chaoying Huo, Lei Zhang, Bo Chen, and Yijun Yuan. 2021. Fast C&W: A fast adversarial attack algorithm to fool SAR target recognition with deep convolutional neural networks. *IEEE Geoscience and Remote Sensing Letters* 19 (2021), 1–5.

[13] Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, and Dawn Song. 2018. Robust physical-world attacks on deep learning visual classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1625–1634.

[14] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572* (2014).

[15] Jiuxiang Gu, Zhenhua Wang, Jason Kuen, Lianyang Ma, Amir Shahroudy, Bing Shuai, Ting Liu, Xingxing Wang, Gang Wang, Jianfei Cai, et al. 2018. Recent advances in convolutional neural networks. *Pattern recognition* 77 (2018), 354–377.

[16] Jie Hang, Keji Han, Hui Chen, and Yun Li. 2020. Ensemble adversarial black-box attacks against deep learning systems. *Pattern Recognition* 101 (2020), 107184.

[17] Hokuto Hirano, Akinori Minagi, and Kazuhiro Takemoto. 2021. Universal adversarial attacks on deep neural networks for medical image classification. *BMC medical imaging* 21 (2021), 1–13.

[18] Tongxin Hu, Vasileios Iosifidis, Wentong Liao, Hang Zhang, Michael Ying Yang, Eirini Ntoutsi, and Bodo Rosenhahn. 2020. Fairnn-conjoint learning of fair representations for fair decisions. In *Discovery Science: 23rd International Conference, DS 2020, Thessaloniki, Greece, October 19–21, 2020, Proceedings 23*. Springer, 581–595.

[19] Xiaowei Huang, Daniel Kroening, Marta Kwiatkowska, Wenjie Ruan, Youcheng Sun, Emese Thamo, Min Wu, and Xinping Yi. 2019. Safety and Trustworthiness of Deep Neural Networks: A Survey (2019). *arXiv preprint arXiv:1812.08342* (2019).

[20] Alex Krizhevsky, Geoffrey Hinton, et al. 2009. Learning multiple layers of features from tiny images. (2009).

[21] Alexey Kurakin, Ian Goodfellow, Samy Bengio, Yinpeng Dong, Fangzhou Liao, Ming Liang, Tianyu Pang, Jun Zhu, Xiaolin Hu, Cihang Xie, et al. 2018. Adversarial attacks and defences competition. In *The NIPS'17 Competition: Building Intelligent Systems*. Springer, 195–231.

[22] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*. Springer, 740–755.

[23] 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).

[24] Aleksander Mądry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2017. Towards deep learning models resistant to adversarial attacks. *stat* 1050 (2017), 9.

[25] Yoanna Martindez-Diaz, Luis S Luevano, Heydi Mendez-Vazquez, Miguel Nicolas-Diaz, Leonardo Chang, and Miguel Gonzalez-Mendoza. 2019. Shufflefacenet: A lightweight face architecture for efficient and highly-accurate face recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*. 0–0.

[26] Mehdi Mirza and Simon Osindero. 2014. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784* (2014).

[27] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. 2016. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2574–2582.

[28] Ben Nassi, Dudi Nassi, Raz Ben-Netanel, Yisroel Mirsky, Oleg Drokin, and Yuval Elovici. 2020. Phantom of the adas: Phantom attacks on

driver-assistance systems. *Cryptology ePrint Archive* (2020).

[29] AKM Iqtidar Newaz, Nur Imtiazul Haque, Amit Kumar Sikder, Mohammad Ashiqur Rahman, and A Selcuk Uluagac. 2020. Adversarial attacks to machine learning-based smart healthcare systems. In *GLOBECOM 2020-2020 IEEE Global Communications Conference*. IEEE, 1–6.

[30] Augustus Odena, Christopher Olah, and Jonathon Shlens. 2017. Conditional image synthesis with auxiliary classifier gans. In *International conference on machine learning*. PMLR, 2642–2651.

[31] Tianyu Pang, Chao Du, Yinpeng Dong, and Jun Zhu. 2018. Towards robust detection of adversarial examples. *Advances in neural information processing systems* 31 (2018).

[32] Nicholas Papernot, P McDaniel, I Goodfellow, S Jha, Z Berkay Celik, and A Swami. [n. d.]. Practical Black-Box Attacks against Deep Learning Systems using Adversarial Examples (2016). *ArXiv e-prints* ([n. d.]).

[33] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. 2018. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4510–4520.

[34] Manali Shaha and Meenakshi Pawar. 2018. Transfer Learning for Image Classification. In *2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA)*. 656–660. https://doi.org/10.1109/ICECA.2018.8474802

[35] Aaron Smalter Hall and Thomas R Cook. 2017. Macroeconomic indicator forecasting with deep neural networks. *Federal Reserve Bank of Kansas City Working Paper* 17-11 (2017).

[36] Jiawei Su, Danilo Vasconcellos Vargas, and Kouichi Sakurai. 2019. One pixel attack for fooling deep neural networks. *IEEE Transactions on Evolutionary Computation* 23, 5 (2019), 828–841.

[37] Juan Terven and Diana Cordova-Esparza. 2023. A comprehensive review of YOLO: From YOLOv1 to YOLOv8 and beyond. *arXiv preprint arXiv:2304.00501* (2023).

[38] Stefan Thaler, Vlado Menkovski, and Milan Petkovic. 2018. Deep learning in information security. *arXiv preprint arXiv:1809.04332* (2018).

[39] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. 2023. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 7464–7475.

[40] Jiakai Wang. 2021. Adversarial Examples in Physical World.. In *IJCAI*. 4925–4926.

[41] Rey Wiyatno and Anqi Xu. 2018. Maximal jacobian-based saliency map attack. *arXiv preprint arXiv:1808.07945* (2018).

[42] Chaowei Xiao, Bo Li, Jun-Yan Zhu, Warren He, Mingyan Liu, and Dawn Song. 2018. Generating adversarial examples with adversarial networks. *arXiv preprint arXiv:1801.02610* (2018).

[43] Zhonghui You, Kun Yan, Jinmian Ye, Meng Ma, and Ping Wang. 2019. Gate decorator: Global filter pruning method for accelerating deep convolutional neural networks. *Advances in neural information processing systems* 32 (2019).

[44] Xiaoyong Yuan, Pan He, Xiaolin Lit, and Dapeng Wu. 2020. Adaptive adversarial attack on scene text recognition. In *IEEE INFOCOM 2020-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 358–363.

[45] Xiaoyong Yuan, Pan He, Qile Zhu, and Xiaolin Li. 2019. Adversarial examples: Attacks and defenses for deep learning. *IEEE transactions on neural networks and learning systems* 30, 9 (2019), 2805–2824.

[46] Zhenlong Yuan, Yongqiang Lu, and Yibo Xue. 2016. Droiddetector: android malware characterization and detection using deep learning. *Tsinghua Science and Technology* 21, 1 (2016), 114–123.

[47] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris N Metaxas. 2017. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *Proceedings of the IEEE international conference on computer vision*. 5907–5915.

[48] Jiebao Zhang, Wenhua Qian, Rencan Nie, Jinde Cao, and Dan Xu. 2022. LP-BFGS attack: An adversarial attack based on the Hessian with limited pixels. *arXiv preprint arXiv:2210.15446* (2022).

[49] Weijia Zhang. 2019. Generating adversarial examples in one shot with image-to-image translation GAN. *IEEE Access* 7 (2019), 151103–151119.

[50] Sheng Zhong, Qing-yun Shi, and Min-Teh Cheng. 1994. Adaptive hierarchical vector quantization for image coding. *Pattern recognition letters* 15, 12 (1994), 1171–1175.

Received 12 September 2023