Adversarial Attack on Scene Text Recognition Based on Adversarial Networks

Yanju Liu[1], Xinhai Yi[2(✉)], Yange Li[2], Bing Wang[1], Huiyu Zhang[2], and Yanzhong Liu[2]

[1] Nanjing Normal University of Special Education, Nanjing 210038, China
yanjuliu@qqhru.edu.cn
[2] Qiqihar University, Qiqihar 161000, China
yixinhai000@163.com

Abstract. Deep learning further improves the recognition performance of scene text recognition technology, but it also faces many problems, such as complex lighting, blurring, and so on. The vulnerability of deep learning models to subtle noise has been proven. However, the problems faced by the above scene text recognition technology are likely to become a adversarial sample leading to text recognition model recognition errors. An effective measure is to add adversarial samples to the training set to train the model, so studying adversarial attacks is very meaningful. Current attack models mostly rely on manual design parameters. When generating adversarial samples, continuous gradient calculation is required on the original samples. Most of them are for non-sequential tasks such as classification tasks. Few attack models are for sequential tasks such as scene text recognition. This paper reduces the time complexity of generating adversarial samples to O(1) level by using the Adversarial network to semi-white box attack on the scene text recognition model. And a new objective function for sequence model is proposed. The attack success rates of the adversarial samples on the IC03 and IC13 datasets were 85.28% and 86.98% respectively, while ensuring a structural similarity of over 90% between the original samples and the adversarial samples.

Keywords: Deep learning · Text recognition · AdvGAN · Adversarial examples · Natural scene

1 Introduction

Writing is a tool for human beings to record information and transmit it to civilizations for a long time. As time goes by, a large number of texts need to be stored digitally. Optical Character Recognition (OCR) [1] technology meets the needs of human social development by translating textual information in pictures into computer text. Scene Text Recognition (STR) has become a hot research problem as a subproblem of OCR, which is to recognize the text information in natural scene pictures and convert them into string form [2]. Deep learning networks have recently taken center stage in the field

B. Wang et al. (Eds.): ICMTEL 2023, LNICST 535, pp. 53–63, 2024.
https://doi.org/10.1007/978-3-031-50580-5_5

of scene text recognition, and as deep learning has advanced, so too has the technology for scene text recognition, making it more accurate and able to handle complicated images. However, Szegedy et al. [3] found that in image classification tasks, adding small-magnitude perturbations to the input samples may lead to incorrect classification by deep learning models. Scene text is more complex than ordinary text images, so while trying to improve recognition accuracy, it is important to investigate whether scene text recognition models can lead to incorrect recognition due to subtle interference [4]. If the issue of generalization of recognition models is not considered, then security risks may arise in practical applications.

At present, the decoders used in scene text recognition technology mainly use two technologies, one is the Connectionist Temporal Classification (CTC) [5] mechanism and the other is the attention mechanism [6]. The attention mechanism have been mainly studied in recent years, so the study of methods to attack attentional mechanisms allows for better generalization of attack models [7]. Most current attack methods are based on optimization equations and simple matrix calculations in pixel space, but these methods usually rely on manual parameter design, which makes the generated interference inflexible and time-consuming in generating adversarial examples through multiple gradient calculations. Xiao et al. [8] used Generative Adversarial Network (GAN) [9] to generate adversarial examples (AdvGAN). However, AdvGAN has limited ability to attack the scene text. The scene text recognition model based on attention decoder is used as the target model. And the attack ability and quality of adversarial samples are both optimized as objectives Once the discriminator and the generator are trained, the generation of adversarial examples can be done quickly with a single query. The following two aspects are mostly covered by the research for this article.

In order to optimize the generator, we restructure the network structure in this study and create a new objective function.

(1) An generative adversarial network is designed to generate adversarial examples for scene text recognition. The generated adversarial examples are closer to the original samples and have higher attack success rate than those generated by the manual parameter design method.
(2) The scene text recognition model based on attention decoder is used as the target model. And the attack ability and quality of adversarial samples are both optimized as objectives. It makes the loss function of the adversarial network easier to optimize and accelerates the convergence speed of the generative network.

2 Related Work

2.1 Scene Text Recognition Method

Shi et al. [10] fed the learned feature maps into a stacked Bidirectional Long Short Term Memory (Bi LSTM) network [11], and connected the CTC decoder to the end of the Bi LSTM network to achieve text recognition. Jaderberg et al. [12] first corrects the irregular text to a horizontal direction, and then performs routine recognition. Shi et al. [13] have further extended the paper [12] by using a bidirectional decoder. The “attentional drift problem” is found to correct the center of attention by focusing the attentional network [14]. Also for this problem, a multi-directional non-local self-attention module

is proposed [15]. Litman et al. [16] combined CTC and Attention mechanisms and designed a cascaded Attention selective attention decoder with the aid of CTC training.

2.2 Attack Method for Generating Adversarial Examples

The attack problem can be defined as the model f recognizing errors when interference δ is added to the original image, i.e., $f(x+\delta) \neq f(x)$. Szegedy et al. [3] proposed the Fast Gradient Sign Method (FGSM) by computing the fastest descent of δ infinite norm, which obtains the adversarial sample by

$$x + \epsilon \cdot \text{sign}(\nabla x \cdot \text{Loss}(f(x), l)) \tag{1}$$

Kurakin et al. [17] improves the FGSM algorithm by shortening the step size. This can be better approximated by smaller steps and more iterations, which makes it more aggressive than FGSM. It generates the adversarial sample with an objective function of

$$x_{t-1} + \alpha \cdot \text{sign}(\nabla x \cdot \text{Loss}(f(x_{t-1}), l)) \tag{2}$$

Madry et al. [18] is a method for generating disturbances by multiple iterations. Since the target model is mostly nonlinear, if only one iteration is performed, the perturbation direction of Loss is unclear and it is difficult to attack successfully in one calculation. Project Gradient Descent (PGD) is also smaller step size, multiple iterations. In each iteration, the generated perturbations are controlled within the specified range, and finally an adversarial sample that can be attacked successfully is generated. Its objective function for each iteration is

$$x_t{}' = \prod\nolimits_{\epsilon} (x_{t-1} + \alpha \cdot \text{sign}(\nabla x \cdot \text{Loss}(f(x_{t-1}), l)) \tag{3}$$

Goodfellow et al. first proposed to generate samples using two networks constrained against each other. Isola et al. [19] further improves the quality of synthetic images. AdvGAN uses the idea of GAN to generate adversarial examples. However, in the backbone network of its generator and discriminator, the multi-scale features of scene text cannot be well learned. In this study, we modified the network structure of the attack network and design a loss function more suitable for scene text to optimize the generator, and successfully integrate the scene text recognition model into the generative adversarial network to attack it to generate adversarial examples of scene text images.

3 AdvGAN-Based Scene Text Recognition Attack Method

3.1 Structure of the Attack Model

Figure 1 depicts the general structure of the confrontation model, which is made up of three primary components: a discriminator, a generator, and a target network (model for scene text recognition). The target network is an scene text recognition model that has already been trained. The generator constantly perturbs the real samples during the training process, and the discriminator separates the disturbed instances from the true data. This is done in order to get the generator's samples closer and closer to the genuine samples until the discriminator is unable to tell them apart. The end result is an adversarial example that can deceive the target network while also being somewhat close to the genuine sample.

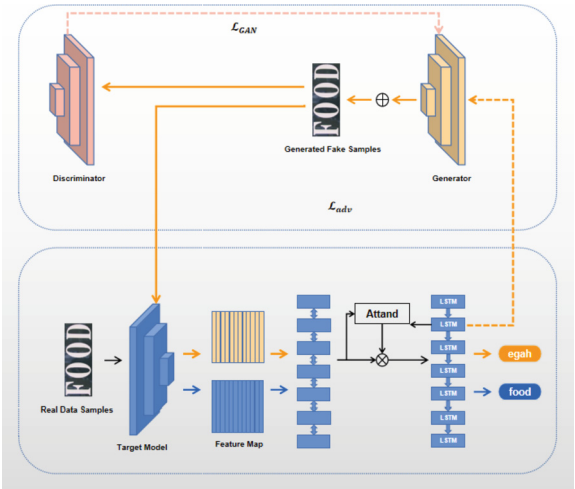


Fig. 1. The architecture of Recognition model and GAN

3.2 Structure of the Generative Adversarial Network

In the network model design of the generative adversarial network, the step size and kernel size of the 3rd and 4th pooling layers of the discriminator network are set to (1, 2) instead of the traditional 2 steps, which can make the width of the image faster convolution to be computed to 1, and finally a binary classification result can be obtained. The step length of the pooling layer in the middle of the generator network is also adjusted and set to (2, 1), so that a longer feature sequence can be obtained. Since the width of a scene text image is usually much larger than the height, getting a longer feature sequence can reduce the loss of features during the convolution calculation. The structure of the generation network and the discriminative network is shown in Fig. 2 and Fig. 3.

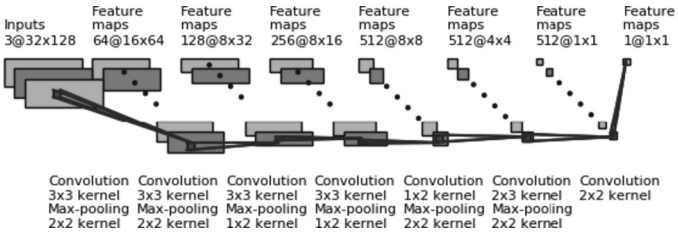


Fig. 2. The architecture of discriminator

3.3 Scene Text Recognition Model Based on Attention Mechanism

Attentional Scene Text Recognizer (ASTER) is a classical model in the study of scene text recognition techniques, which uses a bidirectional decoding mechanism based on

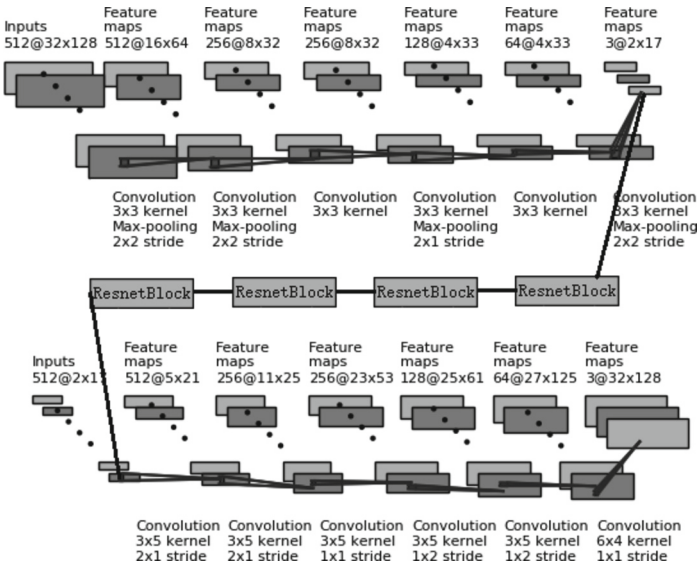


Fig. 3. The architecture of Generator

Attention to train a left-to-right decoder and a right-to-left decoder. The two decoders output recognition results from two directions, and then the one with higher confidence is selected as the final recognition result.

The Attention decoder decodes the feature $\boldsymbol{H}$ from the encoder directly into the target sequence $\{l_1, \ldots, l_n\}$, and the maximum time step set by the decoder is T. The decoding stops when the terminator "EOS" is encountered, and the output at the t th time step is

$$\begin{aligned} p(y_t) &= \text{softmax}(\mathrm{W_{out}s_t} + \mathrm{b_{out}}) \\ y_t &\sim p(l_t) \end{aligned} \tag{4}$$

where W_{out} and b_{out} are learnable parameters. Softmax$(\cdot)$ guarantees that $0 \leq p(y_t) \leq 1$ and $\sum_{i=1}^{T} p(y_t) = 1$. s_t is the hidden state at the t time step, and s_t is calculated as follows.

$$s_t = \text{rnn}(s_{t-1}, (g_t, f(l_{t-1})) \tag{5}$$

Instead of decoding a result based on a particular feature in the decoding process, the Attention mechanism first calculates an attention weight vector $\boldsymbol{\alpha}$, weighted and summed over all features according to the weights, and obtains a feature g_t with context. g_t and $\boldsymbol{\alpha}$ are denoted as

$$g_t = \sum\nolimits_{\mathrm{i}=1}^{\mathrm{n}} \alpha_{\mathrm{t,i}} \mathrm{h_i} \tag{6}$$

$$\begin{aligned} \alpha_{t,i} &= \exp(e_{t,i}) / \textstyle\sum_{\mathrm{i'}=1}^{\mathrm{n}} \exp(e_{t,i'}) \\ e_{t,i} &= \omega^{\mathrm{T}} \cdot \tanh(\mathrm{Ws_{t-1}} + \mathrm{Vh_i} + \mathrm{b}) \end{aligned} \tag{7}$$

where ω^T, W, V, b are all learnable parameters. s_{t-1} is the hidden state of the previous time step.

3.4 Loss Function

The process of constructing an adversarial example is a continuous optimization process. The generator G generates a fine perturbation $G(x)$ after receiving a given input image x. The true label sequence corresponding to x is then $l = \{l_1, \ldots, l_n\}$, and the adversarial example $x' = (x + G(x))$ is formed after that. When non-targeted assaults occur, the adversarial example x' is fed into the scene text recognition model f to obtain the wrong output sequence $l' = \{l'_1, \ldots, l'_n\}$. This problem can be expressed as follows.

$$\begin{aligned} &\min_{\delta} \mathcal{D}(x, x') \\ &\text{s.t. } f(x) = l \\ &f\left(x'\right) = l' \\ &x' \in [-1, 1] \end{aligned} \tag{8}$$

where $\mathcal{D}(\cdot, \cdot)$ denotes the distance between the original image and the adversarial example. Formula 6 simultaneously targets the quality and attack ability of adversarial samples as training objectives However, due to the highly nonlinear nature of $f\left(x'\right) = l'$, it is not easy to optimize in the actual solution process, so a function $g(\cdot)$ is defined that satisfies the condition $f\left(x'\right) = l'$ [20] if and only if $g(x') \leq 0$. $g\left(x'\right)$ is expressed as the following equation.

$$\mathrm{g}\left(x'\right) = \log(1 + \exp(\mathrm{Z}\left(x'\right)_{\mathrm{l}} - \max_{\mathrm{l} \neq \mathrm{l}'} \mathrm{Z}(x')_{\mathrm{l}'})) - \log(2) \tag{9}$$

where $Z(x')$ is the output feature of the recognition model without the softmax operation and $Z(x) = Wx + b$ (see Eq. (4)). This way the objective function becomes more linear and it becomes more favorable for optimization. Thus the loss function for generating the adversarial example is

$$\begin{aligned} &\min \mathcal{D}\left(x, x'\right) + c \cdot g(x') \\ &\text{s.t. } x' \in [-1, 1] \end{aligned} \tag{10}$$

where $c (c > 0)$ is a constant to measure the relative importance of two terms in the objective function. Taking $\mathcal{D}(\cdot, \cdot)$ as the ℓ_2 parametrization, the final loss of the attack target model f is obtained as

$$\mathcal{L}_{\mathrm{adv}} = \mathbb{E}_{\mathrm{x}} \left\| x, x' \right\| + \mathrm{c} \cdot \mathbb{E}_{\mathrm{x}} \sum\nolimits_{\mathrm{i}=1}^{\mathrm{T}} (\log\left(1 + \exp\left(\mathrm{Z}\left(x'\right)_{\mathrm{l,i}} - \max_{\mathrm{l} \neq \mathrm{l}'} \mathrm{Z}(x)_{\mathrm{l}',\mathrm{i}}\right)\right) - \log(2)) \tag{11}$$

In this paper, generative adversarial networks are used to generate adversarial examples, and the adversarial loss proposed by GAN is defined as follows.

$$\mathcal{L}_{\mathrm{GAN}} = \mathbb{E}_x \log(\mathrm{D}(x)) - \mathbb{E}_x \log(1 - \mathrm{D}(x + \mathrm{G}(x))) \tag{12}$$

where $D(x)$ represents the probability that the discriminator D determines whether x is a real picture (since x itself is real, the closer this value is to 1 for $D(x)$, , the better). And $D(x + G(x))$ is the probability that the discriminator judges whether the generated

picture by the generator is true or not, and the closer this value is to 1 the better for the generator as well. The two networks play each other, and finally the generator generates the sample closest to the real picture.

In attacking the target model, the generator may generate more obvious disturbances in order to attack successfully. In this paper, soft hinge loss [21] is used on the ℓ_2 paradigm to limit the perturbation range.

$$\mathcal{L}_{\text{hinge}} = \mathbb{E}_{\text{x}}\max(0, \|\text{G}(\text{x})\| - \text{c}) \tag{13}$$

where c is a manually set parameter that stabilizes the training of the GAN.

As a result, the objective function of our entire generative adversarial network is

$$\mathcal{L} = \mathcal{L}_{\text{GAN}} + \alpha\mathcal{L}_{\text{adv}} + \beta\mathcal{L}_{\text{hinge}} \tag{14}$$

where $\alpha = 1$ and $\beta = 0.2$ to denote the weight of each sub-objective. Finally, the discriminator and generator are continuously optimized by $arg \min_{G} \max_{D} \mathcal{L}$.

4 Experiments and Analysis

4.1 Datasets

In this paper, the adversarial samples are generated on the basis of ICDAR2003 (IC03) [22] and ICDAR2013 (IC13) [23] datasets, and the image content is horizontal text.

4.2 Scene Text Recognition Model

ASTER is the classical recognition model based on Attention mechanism in scene text recognition. Since the test datasets are all horizontal text, the rectification network of ASTER is removed in the experiments and the bidirectional decoder module is retained. The experiments are implemented based on cuda11.1, PyTorch1.9.0 framework, and the generative adversarial network is trained on an NVIDIA RTX 3050 graphics card with a batch size of 64, and all the optimizers use Adam optimizer. The test results of ASTER on IC03 and IC13 datasets in this experimental environment are shown in Table 1.

Table 1. Recognition accuracy of ASTER on different datasets.

Model	IC03	IC13
ASTER	0.8941	0.8811

4.3 Generation of Adversarial Examples

A high attack success rate is usually accompanied by a higher distortion rate of the adversarial samples. Since this paper limits the range of perturbations generated by the generator, and the generator always finds a balance between generating more noise and seeking a higher attack success rate, it is difficult for the model to achieve an attack success rate close to 100%, but it can be seen from Table 2 that the model generates a higher attack success rate for the adversarial examples on dataset IC03 than the FGSM, BIM, and PGD methods by 60.36%, 9.85%, and 5.23%, respectively. It is 55.56%, 7.11%, and 4.32% higher on dataset IC13, respectively. And in the case of the same attack success rate, the model generated adversarial samples with lower ℓ_2 distance and higher SSIM values. This demonstrates that, when the attack success rate is guaranteed, the adversarial instances are more challenging to tell apart with the naked eye, indicating that they are more realistic. The scene text recognition technique's attack success rate (ASR) is represented as

$$\mathrm{ASR} = \frac{\mathrm{num}(\mathrm{lower}(\mathrm{f}(\mathrm{x}')) \neq \mathrm{lowerf}((\mathrm{x})))}{\mathrm{num}(\mathrm{x})} \tag{15}$$

where lower$(\cdot)$ is the conversion of the characters of the model recognition result to lowercase form. The ASR can be interpreted as the number of images that can make the model misclassify divided by the total number of images.

Structural similarity (SSIM) is one of the important indicators for evaluating the similarity of two images. The higher the value of SSIM, the more similar the two images are.

$$\begin{gathered}
\mathrm{l}(\mathrm{X},\mathrm{Y}) = (2\mu_\mathrm{x}\mu_\mathrm{y} + \mathrm{A}_1)/(\mu_\mathrm{x}^2 + \mu_\mathrm{y}^2 + \mathrm{A}_1) \\
\mathrm{c}(\mathrm{X},\mathrm{Y}) = (2\sigma_\mathrm{x}\sigma_\mathrm{y} + \mathrm{A}_2)/(\sigma_\mathrm{x}^2 + \sigma_\mathrm{y}^2 + \mathrm{A}_2) \\
\mathrm{s}(\mathrm{X},\mathrm{Y}) = (\sigma_\mathrm{XY} + \mathrm{A}_3)/(\sigma_\mathrm{X}\sigma_\mathrm{Y} + \mathrm{A}_3) \\
\mu_\mathrm{x} = \frac{1}{\mathrm{H}\times\mathrm{W}}\sum_{\mathrm{i}=1}^{\mathrm{H}}\sum_{\mathrm{j}=1}^{\mathrm{W}}\mathrm{X}_{(\mathrm{i},\mathrm{j})} \\
\sigma_\mathrm{x}^2 = \frac{1}{\mathrm{H}\times\mathrm{W}}\sum_{\mathrm{i}=1}^{\mathrm{H}}\sum_{\mathrm{j}=1}^{\mathrm{W}}(\mathrm{X}_{(\mathrm{i},\mathrm{j})} - \mu_\mathrm{x})^2 \\
\sigma_\mathrm{x}^2 = \frac{1}{\mathrm{H}\times\mathrm{W}}\sum_{\mathrm{i}=1}^{\mathrm{H}}\sum_{\mathrm{j}=1}^{\mathrm{W}}(\mathrm{X}_{(\mathrm{i},\mathrm{j})} - \mu_\mathrm{x})^2
\end{gathered} \tag{16}$$

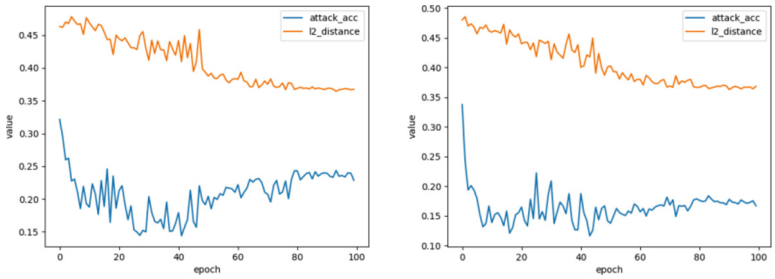
where μ_x, σ_x^2 and σ_{XY} denote the mean, variance and covariance of the images, respectively. A_1, A_2 and A_3 are constants, generally taken as 1% to 3% of the maximum pixel value, to constrain the formula and prevent the calculation of zero. $l(\cdot,\cdot)$, $c(\cdot,\cdot)$ and $s(\cdot,\cdot)$ calculate the brightness, contrast and structure of the two images, respectively. The final structural similarity of the two images is obtained by multiplying the three metrics together as follows.

$$\mathrm{SSIM} = \mathrm{l}(\mathrm{X},\mathrm{Y}) \cdot \mathrm{c}(\mathrm{X},\mathrm{Y}) \cdot \mathrm{s}(\mathrm{X},\mathrm{Y}) \tag{17}$$

The effect of the $\mathcal{L}_{hinge}$ method on the attack success rate and ℓ_2 distance on the IC03 datasets is shown in Fig. 4. By comparing the two pictures, it is obvious that the training of the model is more stable when using $\mathcal{L}_{hinge}$, and stronger adversarial examples can be generated with smaller ℓ_2 distance.

Table 2. Compare the quality and attack ability of FGSM, BIM, PGD and the model proposed in this paper

Method/Datasets		IC03				IC13			
		ASR	ℓ_2Dis	SSIM	Iterator	ASR	ℓ_2Dis	SSIM	Iterator
FGSM	$\varepsilon = 0.5$	50.92	22.52	0.32	-	31.42	17.84	0.51	-
BIM	$\varepsilon = 0.5$	75.43	2.92	0.71	20	79.87	2.67	0.74	20
PGD	$\varepsilon = 0.5$	80.05	1.64	0.80	20	82.66	1.75	0.82	20
Ours	-	85.28	0.38	0.89	-	86.98	0.35	0.92	-



(a) **(b)**

Fig. 4. **(a).** Attack success rate and ℓ_2 distance on IC03 dataset without $\mathcal{L}_{hinge}$. **(b).** Attack success rate and ℓ_2 distance on IC03 dataset with $\mathcal{L}_{hinge}$.

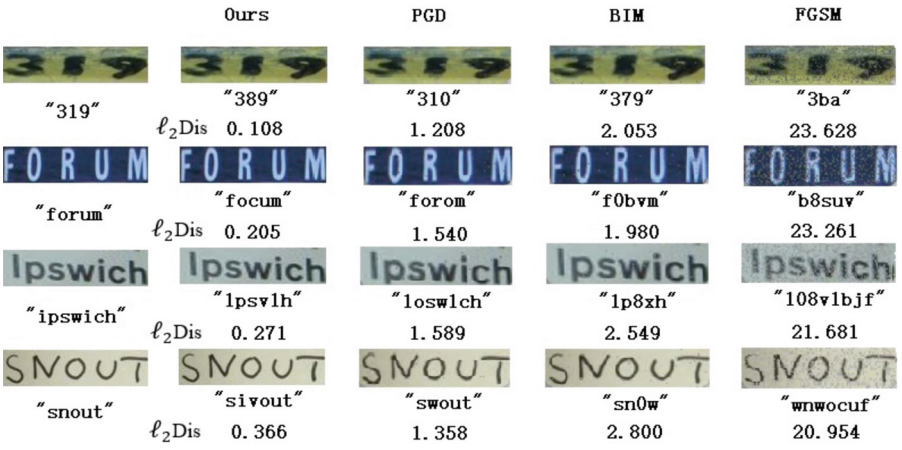


Fig. 5. Comparison of the adversarial examples generated by different attack methods

The adversarial sample instances generated by different attack algorithms are shown in Fig. 5. Due to FGSM only performing one gradient calculation, which is equivalent

to only adding noise to the original sample once, the generated noise is minimal, and it appears to the naked eye that there is no significant difference between the original sample and the adversarial sample. The scene character recognition model is relatively complex. If only one iteration is carried out, the disturbance direction for the loss function is not clear, and it is difficult to attack successfully in one calculation. BIM and PGD conducts multiple iterations, and adds noise on the original samples to improve the ability to generate attacks against samples. It can be seen that as the disturbance coefficient and iteration number increase, the generated noise becomes more and more obvious.

5 Summary

In this paper, an generation adversarial network is used to generate adversarial examples to avoid the limitation of manually setting parameters, and when the training of the network is completed, each sample only needs one calculation to generate the adversarial examples, which greatly reduces the generation time. In the training generation process, a new objective function optimization generator is proposed to reduce the number of optimization iterations and make the generated adversarial samples closer to the real samples and more powerful in attack. In a future study, we will target an attack with another commonly used CTC decoder in scene text recognition, making the attack target coverage more comprehensive for the attack model. Generating high-quality confrontation samples is only a prerequisite to improve the robustness of the recognition model, and the confrontation defense of the attention mechanism based scene text recognition model will be investigated in the future to reinforce the robustness of the recognition model.

Funding. This research was funded by Qiqihar University Graduate Innovative Research Project (Grant No. YJSCX2021079), Jiangsu Province College Student Innovation and Entrepreneurship Project (Grant No. 202212048052Y), Jiangsu Higher Education Association Project (Grant No. 2022JDKT133) and Education and Teaching Reform Project of Nanjing Normal University of Special Education (Grant No. 2022XJJG015).

References

1. Radwan, M.A., Khalil, M.I., Abbas, H.M.: Neural networks pipeline for offline machine printed Arabic OCR. Neural Process. Lett. **48**(2), 769–787 (2018). http://www.springer.com/lncs. Accessed 21 Nov 2016
2. Jin, L.W., Zhong, Z.Y., Yang, Z., et al.: Applications of deep learning for handwritten Chinese character recognition: a review. Acta Autom. Sin. **42**(8), 1125–1141 (2016)
3. Szegedy, C., Zaremba, W., Sutskever, I., et al.: Intriguing properties of neural networks. arXiv preprint arXiv:1312.6199 (2013)
4. Yuan, X., He, P., Lit, X., et al.: Adaptive adversarial attack on scene text recognition. In: IEEE INFOCOM 2020-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), pp. 358–363. IEEE (2020)
5. Graves, A., Fernández, S., Gomez, F., et al.: Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In: Proceedings of the 23rd International Conference on Machine Learning, pp. 369–376 (2006)

6. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473 (2014)
7. Yang, M., Zheng, H., Bai, X., et al.: Cost-effective adversarial attacks against scene text recognition. In: 2020 25th International Conference on Pattern Recognition (ICPR), pp. 2368–2374. IEEE (2021)
8. Xiao, C., Li, B., Zhu, J.Y., et al.: Generating adversarial examples with adversarial networks. arXiv preprint arXiv:1801.02610 (2018)
9. Goodfellow, I., Pouget-Abadie, J., Mirza, M., et al.: Generative adversarial nets. In: Advances in Neural Information Processing Systems, 27 (2014)
10. Shi, B., Bai, X., Yao, C.: An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. IEEE Trans. Pattern Anal. Mach. Intell. **39**(11), 2298–2304 (2016)
11. Graves, A., Liwicki, M., Fernández, S., et al.: A novel connectionist system for unconstrained handwriting recognition. IEEE Trans. Pattern Anal. Mach. Intell. **31**(5), 855–868 (2008)
12. Jaderberg, M., Simonyan, K., Vedaldi, A., et al.: Deep structured output learning for unconstrained text recognition. arXiv preprint arXiv:1412.5903 (2014)
13. Shi, B., Yang, M., Wang, X., et al.: ASTER: an attentional scene text recognizer with flexible rectification. IEEE Trans. Pattern Anal. Mach. Intell. **41**(9), 2035–2048 (2018)
14. Cheng, Z., Bai, F., Xu, Y., et al.: Focusing attention: towards accurate text recognition in natural images. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 5076–5084 (2017)
15. Lu, N., Yu, W., Qi, X., et al.: Master: multi-aspect non-local network for scene text recognition. Pattern Recognit. **117**, 107980 (2021)
16. Litman, R., Anschel, O., Tsiper, S., et al.: Scatter: selective context attentional scene text recognizer. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 11962–11972 (2020)
17. Kurakin, A., Goodfellow, I., Bengio, S.: Adversarial examples in the physical world (2016)
18. Madry, A., Makelov, A., Schmidt, L., et al.: Towards deep learning models resistant to adversarial attacks. arXiv preprint arXiv:1706.06083 (2017)
19. Isola, P., Zhu, J.Y., Zhou, T., et al.: Image-to-image translation with conditional adversarial networks. In: CVPR (2017)
20. Carlini, N., Wagner, D.: Towards evaluating the robustness of neural networks. In: 2017 IEEE Symposium on Security and Privacy (SP), pp. 39–57. IEEE (2017)
21. Liu, Y., Chen, X., Liu, C., et al.: Delving into transferable adversarial examples and black-box attacks. arXiv preprint arXiv:1611.02770 (2016)
22. Lucas, S.M., Panaretos, A., Sosa, L., et al.: ICDAR 2003 robust reading competitions: entries, results, and future directions. Int. J. Doc. Anal. Recognit. (IJDAR) **7**(2–3), 105–122 (2005)
23. Karatzas, D., Shafait, F., Uchida, S., et al.: ICDAR 2013 robust reading competition. In: 2013 12th International Conference on Document Analysis and Recognition, pp. 1484–1493. IEEE (2013)