Research Article

Two Improved Methods of Generating Adversarial Examples against Faster R-CNNs for Tram Environment Perception Systems

Shize Huang,¹ Xiaowen Liu,² Xiaolu Yang^{1,3}, ,³ Zhaoxin Zhang,² and Lingyu Yang²

¹Shanghai Key Laboratory of Rail Infrastructure Durability and System Safety, Tongji University, Shanghai, China

²The Key Laboratory of Road and Traffic Engineering, Ministry of Education, Tongji University, Shanghai, China

³China Railway Shanghai Group Co., Ltd., Shanghai Signal and Communication Division, Shanghai, China

Correspondence should be addressed to Xiaolu Yang; yangxiaolu@tongji.edu.cn

Received 22 July 2020; Revised 28 August 2020; Accepted 14 September 2020; Published 22 September 2020

Academic Editor: Qiang Chen

Copyright © 2020 Shize Huang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Trams have increasingly deployed object detectors to perceive running conditions, and deep learning networks have been widely adopted by those detectors. Growing neural networks have incurred severe attacks such as adversarial example attacks, imposing threats to tram safety. Only if adversarial attacks are studied thoroughly, researchers can come up with better defence methods against them. However, most existing methods of generating adversarial examples have been devoted to classification, and none of them target tram environment perception systems. In this paper, we propose an improved projected gradient descent (PGD) algorithm and an improved Carlini and Wagner (C&W) algorithm to generate adversarial examples against Faster R-CNN object detectors. Experiments verify that both algorithms can successfully conduct nontargeted and targeted white-box digital attacks when trams are running. We also compare the performance of the two methods, including attack effects, similarity to clean images, and the generating time. The results show that both algorithms can generate adversarial examples within 220 seconds, a much shorter time, without decrease of the success rate.

1. Introduction

Trams have gained popularity because of low cost, high efficiency and environment friendliness [1, 2]. An increasing number of them have even adopted automatic driving, which provokes high requirements for perceiving running environment and driving safety [3]. A tram perception system usually deploys detectors, such as cameras installed on carriages [4], and deep learning methods, such as convolutional neural networks (CNNs), to analyse visions captured by the cameras. The systems are used to detect obstacles and their locations beforehand and take corresponding measures. Neural networks have been adopted widely, such as vehicle license plate recognition [5], system control [6–8], turnout fault diagnosis [9], and immunohistochemistry [10]. The growing perception systems have incited ever-rising adversarial attacks on deep learning algorithms [11–13], imposing hidden threats to tram security.

Unlike subways running in partially enclosed spaces, trams share roads with ground transportation and operate in the OS mode, meaning that trams must adhere to a speed limit and their drivers must be vigilant to circumstances [5]. This imposes drastically high demand on tram drivers' work: they need to determine with their eyes whether there are pedestrians, nonmotor vehicles, motor vehicles, or other obstacles, to confirm safe distances based on speeds and to take proper actions.

To alleviate drivers' burden and to detect hidden dangers that human cannot identify, many researchers have designed tram environment perception systems [14] based on deep learning networks and computer vision technology. Tram running environment perception systems work as the safeguard of trams. If their detected images are replaced by adversarial examples, the systems can no longer correctly recognize objects and potential dangers around, leading to incorrect operations and possible accidents. To the best of

our knowledge, however, there exists no research on adversarial examples that targets tram environment perception systems.

One of the most dangerous and aggressive threats to neural networks is the attack using adversarial examples [15–17]. In 2014, Szegedy et al. [18] proposed the concept of adversarial examples, a kind of inputs that are carefully constructed to deceive neural networks, resulting in misclassified outputs. These samples are slightly modified in a way that is imperceptible to human but can easily confuse a network [19, 20]. Figure 1 displays a typical adversarial example. Current research proved that any network could be attacked by adversarial examples. And Szegedy et al. even found that adversarial examples are universal, which means malicious samples designated to one network can fool all neural networks. Therefore, researchers have veered to the methods of generating adversarial attacks, intending to guide defence measures in the future.

Considerable efforts have been devoted to finding mechanisms of generating adversarial examples. For example, Szegedy et al. [18] proposed an algorithm called L-BFGS to generate adversarial examples by finding the minimal possible disturbance. Goodfellow et al. [21] tried to explain the cause of adversarial examples from liner perspective and designed a method FGSM that seeks the direction where the change of gradient is the greatest and adds disturbance to images in that direction. Moosavi-Dezfooli et al. [22] found an efficient and accurate way called DeepFool, to fool deep neural networks by evaluating the robustness of classifiers and enhancing their performance through proper fine-tuning. This method performs better with less alterations than L-BFGS and FGSM do. Minimization loss function was used to generate adversarial examples by Carlini and Wagner [23], who introduced a set of three adversarial attacks in the wake of defensive distillation against the adversarial perturbations. It is shown that the defensive distillation for targeted networks almost completely fails against these three attacks.

However, given the complicated running environment, trams need information including not only classifications of obstacles but also their specific locations. Classification algorithms are no longer enough. In the last few years, a few researchers have diverted to object detection which have already achieved amazing results in image recognition [24]. Wang et al. [25] presented a method to learn an adversarial network that generates examples with occlusions and deformations. Xie et al. [26] proposed a novel algorithm called dense adversary generation (DAG), which applies a gradient descent algorithm for optimization and extends this problem from classification to object detection. Wei et al. [27] designed an algorithm called united and efficient adversary (UEA) which generates adversarial examples by training GAN with a high-level class loss and a low-level feature loss. Wang et al. [28] used PGD to produce adversarial examples on the total loss of the Faster R-CNN object detector, which achieves a high success rate and is applicable in numerous neural network architectures.

Although these studies of adversarial attacks on object detection provide enough counter examples for neural networks to fight back, they can hardly prepare tram

perception systems. First, most of the studies only conducted nontargeted attacks, which can hardly be full-scale. For example, in Wei et al.'s [27] nontargeted attacks, the training dataset they used was categorized into only 20 classes, and each class differed greatly from each other. Given the complicated and constantly changing environment that tram perception systems process, these nontargeted methods are not enough to defend adversarial examples and targeted attacks must be included. Second, although most studies discussed the accurate results of the adversarial examples, they tended to use a time-costing way and ignored that neural networks might be fooled by a real-time model. For example, Wang et al.'s [28] method spends no less than 35 minutes to generate adversarial examples on different architectures, and a trade-off of time cost is to lower the success rate.

Therefore, we propose an improved projected gradient descent (PGD) algorithm and an improved Carlini and Wagner (C&W) algorithm for Faster R-CNN white-box attacks. This approach is designated to tram environment perception systems to generate adversarial examples, which, to the best of our knowledge, is the first research to do so. It can conduct both targeted and nontargeted attacks with less time and high success rates.

The remainder of this paper is structured as follows. Section 2 introduces the theory of the proposed approach. Section 3 explains the experiments on nontargeted attacks and targeted attacks. Section 4 discusses the attack effects, similarity to clean images, and generating time. Section 5 draws conclusions and provides recommendations for the direction of future work.

2. Materials and Methods

Theoretically, all classification algorithms can be improved such as the method proposed in this paper for adversarial examples to attack object [20]. In this paper, we apply this improvement to PGD and C&W. In this section, the proposed approaches are presented. Some definitions and classifications about adversarial attacks are introduced. Then, the principles of the improved PGD method and the improved C&W method are illustrated.

2.1. Backgrounds of Adversarial Attacks. Adversarial attacks can be assorted to several points from angles of the type of output the attackers desire, the amount of knowledge the attackers have about the target model, and means to put adversarial examples into the target model [29].

According to the type of output the attackers desire, adversarial attacks can be divided into two categories:

- (1) Nontargeted attack: the attackers only intend to lead the classifier to any wrong classifications
- (2) Targeted attack: the attackers intend to lead the classifier to a specific wrong classification

According to the amount of knowledge the attackers have about the target model, adversarial attacks can be divided into two categories:

FIGURE 1: An adversarial image generated by FGSM [21]. Left: a clean image of a panda. Middle: perturbation. Right: one sample adversarial image, classified as a gibbon.

- (1) White-box attack: attackers are fully aware of the targeted model, including its type, architecture, and all parameters
- (2) Black-box attack: the attackers merely know limited information about the target model or no information at all

2.2. Improved PGD Algorithm. The improved PGD algorithm proposed in this paper is a kind of white-box attack that only targets Faster R-CNNs. Therefore, before the attack, all parameters of the Faster R-CNN are required in order to generate adversarial examples.

A typical Faster R-CNN architecture for tram detection systems consists of two networks: the region proposal network (RPN) and the object-detecting network (OCD), as shown in Figure 2. The two networks undertake two jobs: first, RPN outputs several region proposals; second, OCD conducts classification in each region proposal.

An object detector is defined as

$$F: [-1, 1]^{h \times w \times 3} \longrightarrow (R^{N \times K}, R^{N \times 4}), \quad (1)$$

where the image with the height of h and the width of w is input; the outputs $(R^{N \times K}, R^{N \times 4})$ are N kinds of objects that are detected. The output of each detector includes the probability distribution of K predefined categories, and the four coordinates described position of the detected object.

The number of objects the detector can identify changes with different input images. For the sake of simplicity, we choose the first n detected objects sorted by confidences.

In our improved PGD algorithm, the loss function of every region proposal is calculated and then added during each iteration. The loss function can be defined as

$$L = \frac{1}{m} \sum_{r_i \in \text{rpn}(x)} L_{F_i}(x, y'), \quad (2)$$

where $\text{rpn}(x) = \{r_1, \dots, r_m\}$, m is the number of region proposals, r_i refers to a region proposal, and y' means the wrong classification that needs to be attacked.

Thus, the following equation can be obtained:

$$L_{F_i}(x, y') = L(F(x_{r_i}), y'), \quad (3)$$

where x_r means the subimage of region r and $L(F(x_r), y')$ means the loss function of the distance between the output of the model $F(x_r)$ and the incorrect classification y' .

In this method, the cross entropy loss function is adopted and the final loss function can be expressed as

$$L = \frac{1}{m} \sum_{r_i \in \text{rpn}(x)} L(F(x_{r_i}), y'), \quad (4)$$

where y' is an n -dimensional vector where each dimension represents the confidence of the classification.

The procedure of the improved PGD algorithm is illustrated in Figure 3. In the second step, the region proposals of the image are obtained by RPN. We perform forward transference on RPN and fix the modified region proposals as constants in the second step of each iteration. Thus, the optimization problem can be solved by removing useless region proposals. In the final step, when counter-propagation is performed, the adversarial examples are iteratively updated until the iteration times reach a specified number.

2.3. Improved C&W Algorithm. In this section, an improved C&W algorithm is proposed to attack the object detector in Faster R-CNNs. This algorithm is also a kind of white-box attack which only targets Faster R-CNNs, meaning that all parameters are required.

An object detector is defined as

$$E: [-1, 1]^{h \times w \times 3} \longrightarrow (R^{N \times P}, R^{N \times 4}), \quad (5)$$

where h means the height of the input image and w means the width; $(R^{N \times P}, R^{N \times 4})$ represents the output; $R^{N \times P}$ refers to N kinds of categories which are detected with their corresponding probability distribution P ; $R^{N \times 4}$ refers to the corresponding coordinates of the N kinds of detected categories in the input clean image.

The proposed C&W algorithm in this paper is improved based on the most important feature of Faster R-CNNs, that is, RPN. A Faster R-CNN extracts feature maps of the input images via convolutional layers, generates region proposals by RPN, and then completes classification in each region proposal. In the improved C&W algorithm, the loss function of each region proposal is superposed and calculated, and

FIGURE 2: Faster R-CNN structure.

FIGURE 3: The shared procedure of the improved PGD method and the improved C&W method. The only difference between the two methods is the gradient calculation formula.

sum of the loss functions is optimized after each iteration. Here, the cross entropy loss function is adopted.

The loss function of this improved C&W algorithm can be defined as

$$\frac{1}{n} \sum_{r_i \in \text{rpn}(x)} f_{E_i} \left[\frac{1}{2} (\tan h(\omega) + 1), y' \right] + c \cdot \left\| \frac{1}{2} (\tan h(\omega) + 1) - x \right\|_2^2, \quad (6)$$

where $\text{rpn}(x) = \{r_1, \dots, r_n\}$ represents n region proposals and c is an adjustable parameter. Some studies proved that bigger c could lead to a higher success rate of the attack and a larger average size of the disturbance [23]. $f_{E_i}[(1/2)(\tan h(\omega) + 1), y']$ represents the value of the loss function of the i^{th} region proposal r_i , which shows the difference of the classification result and the incorrect classification y' , that is, the attack target.

The improved C&W algorithm shares the same procedure with that of the improved PGD algorithm, as shown in Figure 3. The only difference between them is the formula of the loss function.

In this method, the loss function is calculated after removing unnecessary region proposals in order to reach optimized convergence.

3. Experiment

3.1. Experiment Data. The clean images in this paper were captured by cameras installed in trams in Shenzhen, China, with a size of 960×540 . We chose four representative images during a day as the target images, as shown in Figure 4. The first four images covered common scenarios that the tram may meet every day, such as vehicles passing by, pedestrians waiting aside, and traffic lights on and off. The following experiments were all conducted on these four images. The computer we used was Nvidia Tesla K80.

Since the Faster R-CNNs require input images to be square, we enlarged the images' width from 540 to 960 by adding a 50% neutral grey picture in the widened parts. Thus, the demand of Faster R-CNNs was satisfied without imposing any noise on the clean images.

3.2. Adversarial Attacks Using the Improved PGD Algorithm. In this section, we performed both nontargeted attacks and targeted attacks on Faster R-CNNs with the use of our improved PGD algorithm.

Normally, a recognition result is considered as a success only when the confidence of one classification is greater than 50% (sometimes even 80%) while those results with confidences less than 50% are discounted by neural networks and considered as a failed detection.

3.2.1. Nontargeted Attacks. We defined a successful nontargeted attack in this paper as the one when the Faster R-CNN cannot correctly recognize objects such as pedestrians and vehicles in adversarial examples.

The values of all dimensions in Equation (2) in the improved PGD method were set as 0. Adversarial examples of the four images in Figures 4(a)–4(d) were generated on TensorFlow, and the results are shown in Figures 4(e)–4(h). It is hardly to tell the differences between the clean images and their adversarial examples with human eyes.

We took the first image in Figure 4 and its corresponding adversarial example as an example to proceed detailed analysis.

The detected results of the clean image and its adversarial example by the Faster R-CNNs are displayed in Figure 5. Figure 5(a) shows the result of the clean image, where detected objects include vehicles, pedestrians, and traffic signs with high confidences of approximately 95%. Figure 5(b) shows the result of the adversarial example, where the detected objects with confidences from 0% to 100% are all retained.

For further comparison, Figures 6(a) and 6(b) display the confidences of the two images. In Figure 6(a), the confidences of the detected results in the clean image are well distributed

while the confidences of the adversarial example in Figure 6(b) are extremely low, all below 11%. Given the common standard that a success confidence is no less than 50%, this result indicated that the nontargeted attack initiated by the improved PGD method against the Faster R-CNN succeeded.

By that analogy, we conducted nontargeted attacks on the other three adversarial examples in Figures 4(f)–4(h). Confidences of the detected results of Figure 4(f) were all lower than 7%, those of Figure 4(g) were all below 7%, and those of Figure 4(h) were all below 8%. These low confidences meant that the Faster R-CNN could not detect any of the objects; that is, it could not detect all the adversarial examples. Thus, our improved PGD method could generate adversarial examples to successfully attack Faster R-CNNs.

3.2.2. Targeted Attacks. We defined a successful targeted attack as the one when images are detected to be the classification which is specified up front.

Our improved PGD method can also be used to conduct targeted attacks on images of trams' running environment, and the targeted categories can be any objects that can be detected by the well-trained Faster R-CNN. In this experiment, we chose "dog" as the targeted class. The value of the corresponding dimension of "dog" in Equation (2) was set as 1 while others as 0. The corresponding adversarial examples for targeted attacks were generated on TensorFlow. Also, it was almost impossible to tell the difference between clean images and their corresponding adversarial examples by human eyes.

As before, we took the first clean image in Figure 4(a) and its corresponding adversarial example as an example to elaborate the attack mechanism.

Figure 5(d) displays the detected results of the adversarial example by the Faster R-CNN. Although there were no actual dogs in the picture, the objects that the Faster R-CNN identified were all "dog," and all of the results were achieved with extremely high confidences ranging from 97% to 99%.

The comparison of the confidences between the clean image and its adversarial example in Figure 6 can further explain the targeted attack. As we can see, there was no recognition of "dog" in the clean image while the recognition result was nothing but "dog" in the adversarial example.

We conducted targeted attacks on the other three adversarial examples, and the results are shown in Figures 5(e)–5(g). The objects could not be detected correctly, such as the result of the first image. Worsely, they were all recognized as "dog" with high confidences.

This experiment proved that the improved PGD method could launch effective targeted attacks against Faster R-CNNs.

3.3. Adversarial Attacks Using the Improved C&W Algorithm. In this part, we started both nontargeted attacks and targeted attacks on the four images using the improved C&W algorithm.

3.3.1. Nontargeted Attacks. A successful nontargeted attack here conformed the same standard as that in the experiment by the improved PGD.

FIGURE 4: Four clean images for experiments and their corresponding adversarial examples generated by the improved PGD method for nontargeted attacks. (a–d) All images shot from cameras installed in a tram driver's cab, and each of them consists of multiple objects such as cars, pedestrians, and traffic lights.

FIGURE 5: Detection results of Faster R-CNN using the improved PGD method. (a) Detection result on the clean image in Figure 4(a). (b) Detection result on the adversarial example for nontargeted attacks with confidence of from 0% to 100% remaining. (c) Detection result on the adversarial example for nontargeted attacks after filtering objects with the confidence of less than 50%. (d) Detection result on the adversarial example for targeted attacks. (e–g) Detection results on the adversarial examples for targeted attacks generated from Figures 4(b)–4(d), respectively.

In this experiment, values of every dimensions of y' in Equation (6) were set as 0. Adversarial examples of the four clean images were generated on TensorFlow.

Figures 7(a) and 7(b) exhibit the detected results of the first clean image and its corresponding adversarial example. Objects such as banana, chair, bottle, and bird were detected

FIGURE 6: Comparison of confidences between the clean image and the adversarial examples by the improved PGD method. We take the confidence as the horizontal axis and the number as the vertical axis. (a) Confidence distribution of detection results except for “dog” on the clean image in Figures 4(a). (b) Confidence distribution of detection results on the adversarial example for nontargeted attacks. (c) Confidence distribution of detection results which are all detected as “dog” on the adversarial example for targeted attacks.

by the Faster R-CNNs with low confidences no more than 27%. Similarly, confidences were compared between the clean image and its adversarial example, as explained in Figure 8. The confidences of the clean image were well distributed while most confidences of the adversarial example were lower than 5%, and all of them were below 27%.

Nontargeted attacks on the other three adversarial examples achieved similar results. Confidences of the detected results of the second image were all lower than 15%, those of the third were all below 10%, and those of the fourth were all below 15%. The Faster R-CNNs could not detect any of the objects effectively; that is, it could not detect all the adversarial examples.

These results demonstrated that our improved C&W method can launch successful nontargeted digital attacks against Faster R-CNNs.

3.3.2. Targeted Attacks. A successful targeted attack here conformed the same standard as that in the experiment by the improved PGD.

Similarly, we chose “dog” as the targeted category and the value of the corresponding dimension of “dog” in Equation (6) was set as 1 while others as 0. Adversarial examples of the four images in Figure 4 were generated by the improved C&W method.

Figure 7 displays the detected results of the adversarial examples by the Faster R-CNN. Despite there are no actual “dogs” in these pictures, the Faster R-CNN only identified “dog” with high confidences. It proved that our improved C&W method could launch successful and efficient targeted digital attacks against Faster R-CNNs.

4. Results and Discussions

In this section, we evaluate performances, such as attack effects, similarity to clean images, and the generating time of our algorithms.

4.1. Attack Effect. Success rate of attacks is one of the most significant indicators to illustrate the performance of

adversarial examples. The two experiments demonstrated that both the improved PGD method and the improved C&W method can effectively conduct nontargeted attacks and targeted attacks against Faster R-CNNs. The average confidences of the two methods in both nontargeted attacks and “dog” targeted attacks were calculated, and the results are listed in Table 1.

For nontargeted attacks, the average confidence of the improved PGD method was smaller than that of the improved C&W method, which means that the improved PGD method has better attack effects than the improved C&W method. As for targeted attacks of category “dog,” the average confidence of the improved PGD method was higher than that of the improved C&W method, which again proves that the improved PGD method has better attack effects than the improved C&W method. Therefore, the improved PGD method performs a little better on both nontargeted attacks and targeted attacks than the improved C&W method does.

4.2. Similarity to Clean Images. Differences between the clean images and their corresponding adversarial examples generated by the improved PGD method and the improved C&W method could barely be perceived by human eyes. Therefore, we adopted Euclidean distance, the most widely used distance, to measure the differences. The results are shown in Figure 9. For both targeted attacks and nontargeted attacks, the Euclidean distances between the clean images and the adversarial examples generated by the improved C&W method were shorter than those by the improved PGD method, which means that adversarial examples generated by the improved C&W method are more similar to the clean images.

4.3. Operation Time. Generating time is also a crucial indicator. Figure 10 shows the comparison of generating times between the improved PGD method and the improved C&W method. The generating times of the two were basically the same, no more than 220 seconds, and much

FIGURE 7: Detection results of Faster R-CNN using the improved C&W method. (a) Detection result on the adversarial example for nontargeted attacks with confidence of from 0% to 100% remaining. (b) Detection result on the adversarial example for nontargeted attacks after filtering objects with the confidence of less than 50%. (c–f) Detection results on the adversarial examples for targeted attacks generated from Figures 4(a)–4(d), respectively.

FIGURE 8: Comparison of confidences between the clean image and the adversarial examples by the improved C&W method. We take the confidence as the horizontal axis and the number as the vertical axis. (a) Confidence distribution of detection results on the clean image in Figure 4(a). (b) Confidence distribution of detection results on the adversarial example for nontargeted attacks. (c) Confidence distribution of detection results which are all detected as “dog” on the adversarial example for targeted attacks.

TABLE 1: Comparison of average confidence among different methods.

Method	Attack type	Average confidence
Improved PGD	Nontargeted	0.0370
Improved PGD	Targeted	0.9669
Improved C&W	Nontargeted	0.0442
Improved C&W	Targeted	0.8857

FIGURE 9: Comparison of Euclidean distances between the clean images and the adversarial examples generated by the improved PGD algorithm and those generated by the improved C&W method.

FIGURE 10: Comparison of generating time(s) between the improved PGD algorithm and the improved C&W method.

shorter than the 35-minute longer operation time in [28]. The two proposed algorithms can present efficient performance.

5. Conclusions

In this paper, we propose an improved PGD and an improved C&W to generate adversarial examples and attack a Faster R-CNN object detector for tram environment perception. The two methods can successfully conduct nontargeted attacks and targeted attacks, and the comparisons of different indicators demonstrate that they satisfy the requirement of real-time under the premise of high accuracy and are superior to the methods in references.

Future work will concentrate on further shortening the generating time without a large drop of the success rate, applying this kind of improvement to a third classification algorithm to find a pattern, and looking for more effective defence measures. It is also worth researching on black-box attacks and applying these two methods to real world.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This research was supported by the National Key R&D Program of China (2016YFB1200402), the National Natural Science Foundation of China (grant no. 61703308), and Sichuan Province Science and Technology Program (2019YFG0040).

References

- [1] T. Zhang, R. Zhao, E. E. F. Ballantyne, and D. A. Stone, "Increasing urban tram system efficiency, with battery storage and electric vehicle charging," *Transportation Research Part D: Transport and Environment*, vol. 80, 2020.
- [2] A. Z. Zeng, C. F. Durach, and Y. Fang, "Collaboration decisions on disruption recovery service in urban public tram systems," *Transportation Research Part E: Logistics and Transportation Review*, vol. 48, no. 3, pp. 578–590, 2012.
- [3] J. Na, Y. Huang, X. Wu, S.-F. Su, and G. Li, "Adaptive finite-time fuzzy control of nonlinear active suspension systems with input delay," *IEEE Transactions on Cybernetics*, vol. 50, no. 6, pp. 2639–2650, 2020.
- [4] S. Huang, Y. Zhai, M. Zhang, and X. Hou, "Arc detection and recognition in pantograph-catenary system based on convolutional neural network," *Information Sciences*, vol. 501, pp. 363–376, 2019.
- [5] Y. Qian, D. Ma, B. Wang et al., "Spot evasion attacks: adversarial examples for license plate recognition systems with convolutional neural networks," *Computers & Security*, vol. 95, 2020.
- [6] Q. Chen, S. Xie, M. Sun, and X. He, "Adaptive nonsingular fixed-time attitude stabilization of uncertain spacecraft," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 54, no. 6, 2018.
- [7] Q. Chen, X. Yu, M. Sun, C. Wu, and Z. Fu, "Adaptive repetitive learning control of PMSM servo systems with bounded nonparametric uncertainties: theory and experiments," *IEEE Transactions on Industrial Electronics*, 2020.
- [8] Q. Chen, H. Shi, and M. Sun, "Echo state network-based backstepping adaptive iterative learning control for stric-

- feedback systems: an error-tracking approach,” *IEEE Transactions on Cybernetics*, vol. 50, no. 7, 2020.
- [9] S. Huang, L. Yang, F. Zhang, W. Chen, and Z. Wu, “turnout fault diagnosis based on CNNs with self-generated samples,” *Journal of Transportation Engineering, Part A: Systems*, vol. 146, no. 9, Article ID 04020105, 2020.
  - [10] D. Fan, L. Wei, and M. Cao, “Extraction of target region in lung immunohistochemical image based on artificial neural network,” *Multimedia Tools and Applications*, vol. 75, no. 19, pp. 12227–12244, 2016.
  - [11] C. Gou, K. Wang, Y. Yao, and Z. Li, “Vehicle license plate recognition based on extremal regions and restricted Boltzmann machines,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 4, pp. 1096–1107, 2016.
  - [12] S. Khan, H. Rahmani, S. A. A. Shah, and M. Bennamoun, “A guide to convolutional neural networks for computer vision,” *Synthesis Lectures on Computer Vision*, vol. 8, no. 1, pp. 1–207, 2018.
  - [13] S. Li, C. Wang, and S. Li, “Technical standards for tramway operational safety and signaling system,” *Urban Mass Transit*, vol. 22, no. 4, pp. 115–118, 2019.
  - [14] M. Yu, P. Yang, and S. Wei, “Railway obstacle detection algorithm using neural network,” in *Proceedings of the AIP 1967*, Busan, Republic of Korea, May 2018.
  - [15] M. Pawlicki, M. Choraś, and R. Kozik, “Defending network intrusion detection systems against adversarial evasion attacks,” *Future Generation Computer Systems*, vol. 110, pp. 148–154, 2020.
  - [16] X. Wang, J. Li, X. Kuang, Y.-a. Tan, and J. Li, “The security of machine learning in an adversarial setting: a survey,” *Journal of Parallel and Distributed Computing*, vol. 130, pp. 12–23, 2019.
  - [17] T. Huang, Y. Chen, B. Yao, B. Yang, X. Wang, and Y. Li, “Adversarial attacks on deep-learning-based radar range profile target recognition,” *Information Sciences*, vol. 531, pp. 159–176, 2020.
  - [18] C. Szegedy, W. Zaremba, I. Sutskever et al., “Intriguing properties of neural networks,” 2013, <https://arxiv.org/abs/1312.6199>.
  - [19] B. Kolosnjaji, A. Demontis, and B. Biggio et al., “Adversarial malware binaries: evading deep learning for malware detection in executables,” in *Proceedings of the 2018 26th European Signal Processing Conference (EUSIPCO)*, IEEE, Rome, Italy, pp. 533–537, September 2018.
  - [20] S. T. Chen, C. Cornelius, J. Martin et al., “ShapeShifter: robust physical adversarial attack on faster R-CNN object detector,” in *Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, Springer, Cham, Switzerland, pp. 52–68, September 2018.
  - [21] I. J. Goodfellow, J. Shlens, and C. Szegedy, “Explaining and harnessing adversarial examples,” in *Proceedings of the ICLR*, pp. 1–11, San Diego, CA, USA, May 2015.
  - [22] S. M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, “A simple and accurate method to fool deep neural networks,” in *Proceedings of the CVPR*, pp. 2574–2582, Las Vegas, NV, USA, 2016.
  - [23] N. Carlini and D. Wagner, “Towards evaluating the robustness of neural networks,” in *Proceedings of the SP*, pp. 39–57, San Jose, CA, USA, May 2017.
  - [24] T. Tao, D. Dong, S. Huang, and W. Chen, “Gap detection of switch machines in complex environment based on object detection and image processing,” *Journal of Transportation Engineering*, vol. 146, no. 8, 2020.
  - [25] X. Wang, A. Shrivastava, and A. Gupta, “A-Fast-RCNN: hard positive generation via adversary for object detection,” in *Proceedings of the CVPR*, pp. 3039–3048, Honolulu, Hawaii, July 2017.
  - [26] C. Xie, J. Wang, Z. Zhang, Y. Zhou, L. Xie, and A. Yuille, “Adversarial examples for semantic segmentation and object detection,” in *Proceedings of the CV*, pp. 1378–1387, Andhra Pradesh, India, May 2017.
  - [27] X. Wei, S. Liang, X. Cao, and J. Zhu, “Transferable adversarial attacks for image and video object detection,” in *Proceedings of the IJCAI*, pp. 954–960, Macao, China, May 2019.
  - [28] Y. Wang, K. Wang, Z. Zhu, and F.-Y. Wang, “Adversarial attacks on faster R-CNN object detector,” *Neurocomputing*, vol. 382, pp. 87–95, 2020.
  - [29] Y. Liu, “Generation and application of image adversarial example for neural networks,” M. S. thesis, University of Science and Technology of China, Hefei, China, 2019.