An Adversarial Attack Based on Incremental Learning Techniques for Unmanned in 6G Scenes

Huanhuan Lv, Mi Wen, *Member, IEEE*, Rongxing Lu, *Fellow, IEEE*, Jinguo Li, *Member, IEEE*

***Abstract*—With the development of artificial intelligence(AI), unmanned vehicles can relieve traffic jamming and decrease the risk of traffic accidents, where deep neural networks (DNNs) play an important role and have become one of the most critical technologies. Nevertheless, DNNs are still susceptible to adversarial examples. Even worse, they also show severe performance degradation when the system needs DNNs to learn new knowledge without forgetting the old one. As unmanned vehicles travel on the road, they need to frequently learn new categories and different representations. Learning all data after the new sample arrives will expend a lot of time and space. As a result, it will affect the deployment of artificial intelligence in unmanned scenes. In recent years, it has been observed that incremental learning technology can solve the above challenges. However, previously reported works mainly focused on batch learning. It is not clear how much impact the adversarial attack will have on the deep learning model when performing incremental learning tasks. This issue exposes the hidden safety risks of unmanned driving and increases discuss opportunities. Therefore, we propose an adversarial attack based on incremental learning techniques for unmanned scenes in this paper. Specifically, it can retain information previously learned by the model. At the same time, it can renew the old model to learn new model, thereby continually adding small perturbation to legitimate examples. A couple of experiments on the Pascal VOC 2012 dataset has been conducted, and the experiment results show that the adversarial attack based on incremental learning techniques has a higher attack success rate. Further, it can improve the successful attack rate by 8.43%.**

***Index Terms*—Unmanned, adversarial examples, catastrophic forgetting, incremental learning.**

I. Introduction

With the development and popularization of mobile communications, ultra-low latency 6G can significantly improve the technology of Internet of Vehicles (IOV) and make unmanned more perfect [1]–[3]. The most important part of IOV is information transmission between the car and its surrounding environment and cloud platform [4]–[6]. In addition, edge computing allows more applications to run on the edge, reducing the delay caused by data transmission speed and bandwidth limitations [7], [8]. In unmanned scenes, deploying an edge computing platform on the access network can make information transmission faster, more stable, and safer. At present, unmanned vehicles are composed of multiple subsystems, which interact with each other. As is known, unmanned has become a scientific research technology with both study potential and practical value. On the one hand, compared to humans, it can analyze and evaluate real-time road conditions through more powerful information collection and processing capabilities, and select low-risk operating instructions to ensure traffic safety. On the other hand, it can exchange real-time information with each other through IOV to plan travel roads in real time and alleviate traffic congestion. Recently, DNNs have brought excellent performance on many vision tasks of unmanned vehicles, such as image segmentation [9], lane line detection [10], [11], and image recognition [12]. Nevertheless, due to the immaturity of deep learning models, they still have two important problems: susceptibility to adversarial examples and catastrophic forgetting.

In recent years, Szegedy *et al.* [13] found that DNNs are susceptible to adversarial examples. Adding subtle interference to legitimate examples may influence the classification accuracy. Many works [14]–[18] have presented various adversarial example attack algorithms, which have achieved good attack results. Adversarial example attacks have gradually become a major safety hazard in deep learning, and it also makes unmanned scene recognition face security problems. As shown in Fig.1, although the image is indistinguishable from the human eyes, the automatic recognition system will misjudge it as a passable sign. When unmanned vehicles and human drivers are driving on the road at the same time, it will cause catastrophic consequences. In Florida, the United States, a Tesla car hit a white truck, resulted in the world's first fatal traffic accident for an autonomous driving system [19]. We all know that Tesla is equipped with today's top autopilot technology. However, the artificial intelligence here cannot correctly distinguish between a white cloud and a white truck. After Tesla, the unmanned vehicle developed by Google also suffered a serious accident. Therefore, adversarial attacks can be used in the IOV system, which can easily lead to traffic accidents and may injure human safety [20], [21]. With the upcoming 6G technology, we need to consider the security of using deep learning algorithms in IOV systems [22]–[25].

In unmanned scenes, it is usually impossible for the system

Manuscript received XXX, XX, 2020; revised XXX, XX, 2020. This work was supported by the National Natural Science Foundation of China under Grant No.61872230, No.61802248, U1936213, No.61802249 and No.61702321. (Corresponding author: Mi Wen.)

H. Lv is with the College of Computer Science and Technology, Shanghai University of Electric Power, Shanghai 200090, China (e-mail: huanhuanlv@mail.shiep.edu.cn).

M. Wen is with the College of Computer Science and Technology, Shanghai University of Electric Power, Shanghai 200090, China (e-mail: miwen@shiep.edu.cn).

R. Lu is with the Faculty of Computer Science, University of New Brunswick, Fredericton, Canada E3B 5A3 (e-mail: rlu1@unb.ca).

J. Li is with the College of Computer Science and Technology, Shanghai University of Electric Power, Shanghai 200090, China (e-mail: lijg@shiep.edu.cn).

to get all training samples at once, but gradually get them over time. Therefore, despite the great success that DNNS have achieved in many unmanned vision tasks, they still need to solve the problem of incremental learning. This batch learning will expend a lot of time and space when we learn all the data after the new sample arrives [26]. These problems will affect the deployment of vehicular networks. The method of fine-tuning on the new class can ameliorate these problems, but it seriously reduces the performance of the old classes. This issue is called catastrophic interference or forgetting [27], as shown in Fig.2. Recently, many works [28]–[30] show that incremental learning can solve these challenges.

Fig. 1. Normal traffic stop sign (left) and its adversarial example (right).

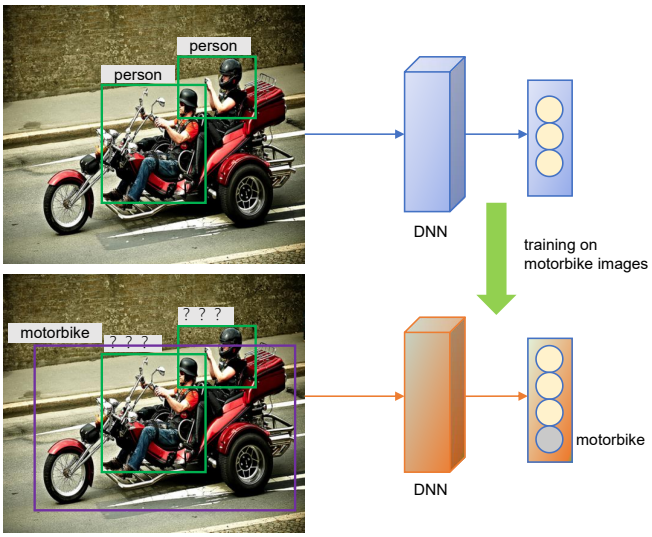

Fig. 2. Catastrophic forgetting. DNN originally trained for three classes, including *person*, detects the rider (top). When the network is retrained with images of the new class motorbike, it detects the motorbike in the test image, but fails to localize the rider (bottom).

Semantic segmentation is an essential part of image understanding in unmanned vision tasks [31]. Unmanned vehicles can identify which part of the current scene is the drivable area by inferring relevant knowledge or semantics from the image. It is particularly crucial for unmanned vehicles. In 2018, Arnab *et al.* [32] evaluated the robustness of adversarial attacks against semantic segmentation models. In addition, our previous research [33] found that adversarial examples have a well attack effect on unmanned scenes recognition. These explain that the example attack has huge hidden danger in the unmanned driving system. In 2019, Michieli *et al.* [29] proposed the research on incremental learning in semantic segmentation, which can work in the real world without retaining previously seen images. This explains that incremental learning is becoming more and more widely among semantic segmentation. However, the previous research on adversarial examples mainly focused on batch learning. It is not clear how they affect the performance of deep learning models when performing incremental learning tasks in unmanned scenes. This uncertainty has potential security risks. In this specific scene, it is a challenge to study the impact of incremental adversarial examples on the performance of deep learning models.

To address above mentioned challenges, we present an adversarial attack based on incremental learning techniques for unmanned scenes, which has a higher attack success rate. The main contributions of our work can be summarized as:

- First, in this work, we employ Deeplab v2 network as segmentation model. Then we combine the incremental learning technology of knowledge distillation, using FGSM, Deepfool and MI-FGSM algorithms to attack under different disturbance values ϵ, respectively. Moreover, we compare the attack success rate of whether to use incremental learning technology. And we analyze the reason why the adversarial attack based on incremental learning techniques has a higher success rate.
- Second, a couple of experiment results illustrate that the adversarial attack based on incremental learning techniques not only has a higher attack success rate but also solves the problem of deep learning architecture catastrophic forgetting. When adding the last class, we can find that the first incremental learning method (L'_D) has better effects on robustness than the second incremental learning method ($E_q L'_D$). However, L'_D has better effects on robustness than $E_q L'_D$ when adding the last five classes.

The remainder of this paper is organized as follows. Related works are discussed in Section II. We present baseline attack algorithms in Section III. Then, we introduce the proposed adversarial attack based on incremental learning techniques in unmanned scenes in Section IV. In Section V, we conduct a series of algorithm performance evaluation and experimental results analysis. In section VI, we illustrate the conclusion of this work.

II. Related Works

A. Adversarial Attacks

According to the effect of the attack, adversarial examples contain two types, namely non-targeted attacks and targeted attacks. For targeted attack, it will set the target of the attack before the attack, which means that the effect after the attack is certain. For non-targeted attacks, there is no need to set an attack target, just change the recognition result after the attack. In addition, according to the attack cost, adversarial examples consists of three types as below.

1) White-box Attack: The premise of white-box attack is that the architecture of the model can be fully obtained, including its parameter values, and the composition of the model [34], [35]. Its advantage is that the calculation speed is relatively fast, but the gradient information of the target network is required. White-box attack algorithm mainly include the fast gradient symbol method (FGSM) [14], DeepFool algorithm [16], C&W algorithm [36] and the strongest first-order method PGD [37] etc. algorithm. In this work, we mainly discuss this type of attack algorithms.

2) Black-box Attack: Black-box attack conduct the next attack by comparing input and output feedback. It is not clear about the structure of the model. They conduct the next attack by comparing input and output feedback. One-pixel attack proposed by Su *et al.* [38], which employs differential evolution (DE) to generate adversarial examples only by changing a one pixel. Hu *et al.* [39] proposed a malware adversarial examples generation method based on MalGAN. In addition, Sarkar *et al.* [40] proposed UPSET, which generates universal disturbances by training a generative neural network G. At the same time, they proposed ANGRI. However, it generates disturbance is not universal.

3) Physical Attack: Physical attacks do not understand the structure of the model and have weak control over the input. Kurakin *et al.* [41] verified the existence of adversarial attacks in physical-world scenario. Attacking the system in a real environment, the target model is a landing service model, and the attack method is operative. Eykholt *et al.* [42] proposed the RP2 attack algorithm to deceive the road sign classification model in unmanned scenes, which employs stickers and other methods to process road signs on real roads. AdvPatch [43] allows larger disturbances and is not affected by scaling or rotation. Moreover, Liu *et al.* [44] improved AdvPatch, and proposed a perceptual-sensitive generative adversarial networks (PS-GAN) for the enhancement of attack power and the improvement of fidelity.

B. Incremental Learning

Under normal circumstances, DNNs have achieved very successful results on many computer vision problems. However, they take a high training time to learn the models. Furthermore, when faced with new categories in the dataset, the neural network will forget the previously learned knowledge when adapting to the new categories, which is called catastrophic interference or forgetting [27]. Recently, researchers have been studying ways to mitigate this effect [30]. In 2017, elastic weight consolidation (EWC) [45] algorithm allows AI to retain previous knowledge through machine learning, and it can reuse old knowledge to solve new problems. After the learning without forgetting (LwF) only employs new knowledge to train while retaining the former features [46]. Zhou *et al.* [47] proposed that the current model extracts information from cropped versions of previous model.

C. Semantic Segmentation Model

Semantic segmentation is a pixel classification issue with semantic marks [48], [49]. In 2015, fully convolutional network (FCN) [50] has opened up a new path for the semantic segmentation task. The SegNet model proposed by Badrinarayanan *et al.* [51] is a symmetrical structure, which is more similar to an auto-encoder in form. DeepLab V1 architecture was proposed by the Google team [52]. The model achieved a high accuracy of 71.6% in the semantic segmentation competition, which greatly improved the model's ability to capture low-level details. DeepLab V2 [53] network made the following two improvements based on DeepLab V1: first, the VGG16 network of the feature extraction part was replaced with ResNet. Second, the hollow space pyramid pooling was proposed Atrous spatial pyramid pooling (ASPP) module. In addition, some more successful methods are U-Net [54], RefineNet [55], PSPNet [56] and DeepLab V3 [57].

III. Baseline Attack Algorithms

In this work, we employ three classic adversarial example attack algorithms. These algorithms are white-box non-target attacks. We choose them due to their advantages as follows: the calculation of FGSM is small, and the process of generating adversarial examples is very fast [20]. DeepFool algorithm generated adversarial examples are highly like the original examples [34]. MI-FGSM algorithm has good attack results under the white-box condition. Moreover, for untargeted attacks, the generated adversarial examples have better migration capabilities [58].

A. FGSM Algorithm

In 2014, Goodfellow *et al.* [14] proposed the FGSM algorithm used the gradient of a neural network to create adversarial examples. It employs the gradient of the loss relative to the original image x, and then creates a new image x^* that maximizes the loss. It can be denoted as

$$x^* = x + \epsilon * sign(\nabla_x J(\theta, x, y)) \tag{1}$$

where y denotes the original image label. ϵ denotes an adjustment coefficient. $J(\theta, x, y)$ denotes the loss function. y' denotes the target label. By reducing the gradient of the loss function $J(\theta, x, y')$, FGSM algorithm can be extended to a targeted attack. The targeted FGSM can be denoted as

$$x^* = x - \epsilon * sign(\nabla_x J(\theta, x, y')) \tag{2}$$

Moreover, it is worth pointing out that the FGSM effect will be worse when the decision function is nonlinear. This method can effectively generate the required adversarial examples for various models.

B. Deepfool Algorithm

In 2016, Moosavi-Dezfooli *et al.* [16] proposed Deepfool algorithm, which is based on hyperplane classification. Assuming the classification function of the classifier is $f(x) = w^T x + b$, according to the classification function, it can be known that its affine plane is $f = x : w^T + b = 0$. The smallest perturbation that changes the decision of the classifier is $\Delta(x_0, f)$, and its direction is perpendicular to Γ. It can be defined as

$$\begin{aligned} \rho_*(x_0) &:= \arg\min \|\rho\|_2 \\ s.t.\ sign(f(x_0+\rho)) &\neq sign(f(x_0)) \\ &= -\frac{f(x_0)}{\|w\|_2^2} \end{aligned} \tag{3}$$

In the overall iterative process, the adversarial examples are generated, which can be obtained as

$$\arg\min_{r_*} \|r_*\|_2 \ s.t.\ f(x_i) + \nabla f(x_i)^T r_* = 0 \tag{4}$$

Specifically, it employs a hyperplane to separate each category from other categories to establish. In this way, the optimal solution to this problem can be derived, and adversarial examples can be generated. After finding a real adversarial example, the search will terminate.

C. MI-FGSM Algorithm

In 2018, Dong *et al.* [17] proposed the MI-FGSM algorithm. During the iterative process, momentum method accumulates the velocity vector along the gradient direction of the loss function. By recording the gradient of the previous step to assist in crossing troughs, small peaks and poor local minimums or maximums. Using this method to generate adversarial examples can achieve better results than traditional algorithms. To solves the L_∞ norm boundary, a non-targeted adversarial example x^* can be generated from the correct example x that satisfies constraint optimization problem. Update g_{t+1} by

$$g_{t+1} = \mu \cdot g_t + \frac{\nabla_x J(x_t^*, y)}{\|\nabla_x J(x_t^*, y)\|_1} \tag{5}$$

Among them, g_{t+1} is updated according to the direction of gradient descent. μ is decay factor. Update x_{t+1}^* by applying the sign gradient as

$$x_{t+1}^* = x_t^* + \alpha \cdot sign(g_{t+1}) \tag{6}$$

To find adversarial examples for gradient-based methods by solving constrained optimization problems

$$\arg_{x^*} max J(x^*, y),\ s.t. \|x^* - x\|_\infty \leq \varepsilon \tag{7}$$

IV. Proposed Adversarial Attack Approach

To help understand, we discuss adversarial attacks based on incremental learning techniques in detail. Fig.3 illustrates the core framework of the unmanned vehicle system, including perception, planning and control. Among them, the perception layer employs sensors in the vehicle to discover semantic areas (such as cars, pedestrians or roads) in street scenes and provide them with information about the surrounding environment, which is essential for the security of unmanned vehicles [59]. At present, there are four main ways to attack by adversarial examples in the unmanned vehicle system. First, when the image is transmitted from the sensor to the data processing system, the attacker can attack by tampering or mixing adversarial examples. Second, the attacker can steal important information of the vehicle by entering the unmanned vehicle operating system, causing the system to crash and lead to parking. Third, the attacker can also enter the unmanned vehicle control system to manipulate mechanical components, hijacking the unmanned vehicle to injure people. This situation is very dangerous. Fourth, connect multiple unmanned vehicles and cloud platform systems through IOV to hijack their communication systems, thus confusing communication between unmanned vehicles. We mainly focuses on the first and fourth cases in this work. In addition, we analyze the robustness of the attack based on incremental learning in the last part of this section.

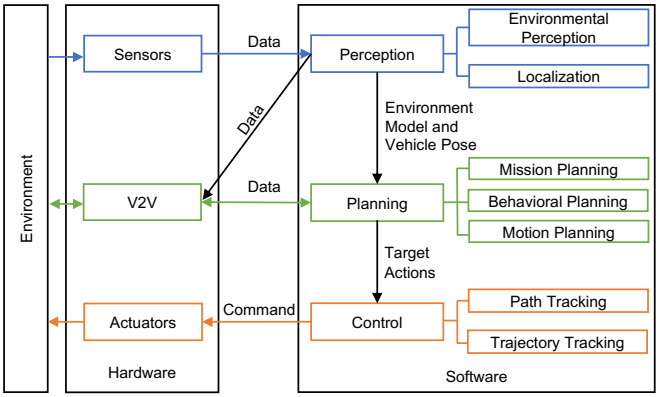

Fig. 3. Unmanned vehicle system architecture. The blue box is the task of perception layer. The green box is the task of planning layer. The orange box is the task of control layer.

A. The Proposed Adversarial Attack Based on Incremental Learning Techniques

We propose the adversarial attack based on incremental learning techniques for unmanned scenes, which has a higher attack success rate. This scheme mainly includes three parts: incremental learning techniques, incremental to obtain segmentation maps, and adversarial attack based on incremental learning.

*1) **Incremental Learning Techniques**:* We employ Michieli *et al.* [29] proposed two incremental learning methods: the first is to perform knowledge distillation on the output layer to obtain the distillation loss L'_D. The second is to freeze the encoder while performing knowledge distillation on the output layer to obtain the distillation loss $E_q L'_D$ when the encoder is frozen. Knowledge distillation is the transfer of knowledge learned from a complex model or multiple models to another lightweight model [60]. Compared with many existing incremental methods [27], [45]–[47], these two methods are the most challenging settings in which images from old mission are not saved and employed to assist the incremental process. This is particularly suitable for the unmanned vehicle system, because the system has privacy issues and limited storage budgets. Fig.4 shows the general overview of this method.

L'_D is the cross-entropy loss masked by the logarithm between the output of the softmax layer of the former model M_{k-1} and the latter model M_k (assuming it is in the $k-th$ incremental step at this time). This is because we want to

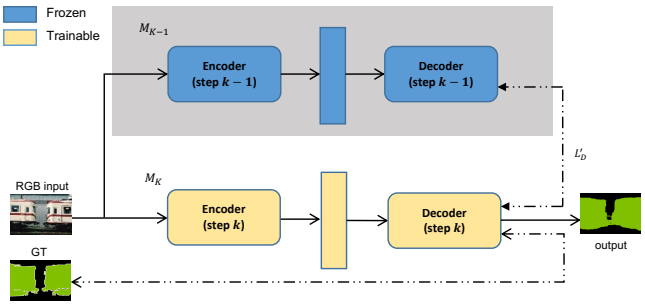

Fig. 4. Overview of the $k-th$ incremental step of our learning framework for semantic segmentation of RGB images.

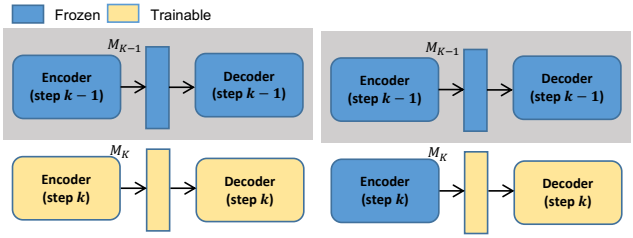

Fig. 5. The freezing scheme of the encoder in the $k-th$ incremental step. M_{K-1} is the entire model of the previous step.

preserve them by guiding the learning process, so the cross-entropy is shielded, which is good for considering the classes that have been seen. The method to evaluate whether there is a distillation loss can be denoted as

$$L'_D = -\frac{1}{|D_k^{tr}|}\sum_{X_n \epsilon D_k^{tr}}\sum_{c \epsilon S_{k-1}} M_{K-1}(X_n)[c] \cdot log(M_k(X_n[c])) \tag{8}$$

where D_k^{tr} refers to a new training sample at each step. $k = 1, 2, \cdots$ are incremental steps of indexing so that the model learns a new set of classes every time. $M_k(X_n[c])$ reflects the evaluation score for class c. S_{k-1} is the union of all classes learned before.

$E_q L'_D$ is modified based on the basis of the first L'_D. Encoder aims to extract some intermediate feature representations, which modification is based on this point. This method allows the network to be restricted to learning new categories only through the decoder. Compared with the previous training stage, it retains the same feature extraction capabilities, as shown in Fig.5.

*2) **Incremental to Obtain Segmentation Maps**:* In 2018, Arnab *et al.* [32] found that attacks created from the basic DeepLab v2 model adopting FGSM had good results. Therefore, we choose the classic image semantic segmentation method Deeplab v2 network as feature extractor. The main task of this part is to gain the semantic segmentation map through the semantic segmentation model. DeepLab v2 model include: atrous (dilated) convolution, ASPP and conditional random field (CRF) [61]. The DeepLab v2 model obtains approximate segmentation results through DCNN (based on ResNet-101) that is use atrous convolution. Fig.6 shows the structure of DeepLab v2. ResNet structure consists of building blocks and bottleneck, as shown in Fig.7.

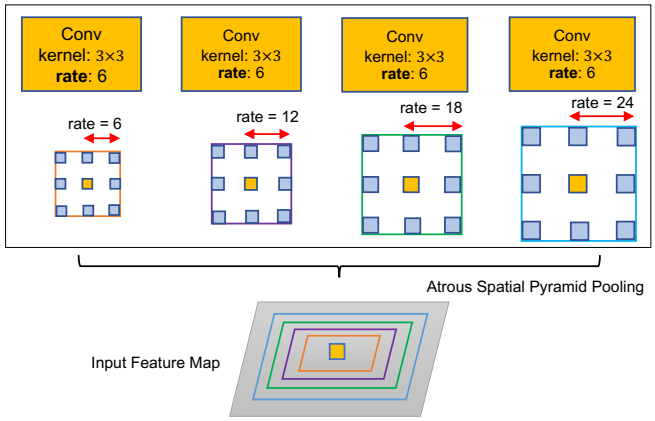

Fig. 6. Atrous Spatial Pyramid Pooling (ASPP). To classify the center pixel (yellow), ASPP utilizes multi-scale features by using multiple line filters at different rates. The valid Fields-Of-View are displayed in different colors.

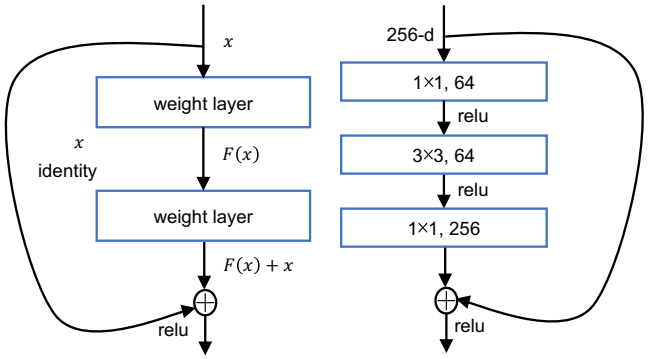

Fig. 7. Left is a building block. Right is a bottleneck.

*3) **Adversarial Attack Based on Incremental Learning**:* First, we apply the DeepLab v2 network as feature extractor to incremental obtain the semantic segmentation map. We employ the Pascal VOC 2012 dataset in experimental, which is divided into 21 classes (including background). Specifically, here we first employ the DeeLab v2 network to learn 21 classes. Then we use three attack algorithms to attack under different perturbation constraint to obtain the attack success rate, respectively. This is the case of no incremental learning. To see the impact of incremental learning on the attack success rate, we choose two options: add one class and add five classes. We employ the above two incremental methods L'_D and $E_q L'_D$ conduct experiments for these two options, respectively. Specifically, first, we employ the DeepLab v2 to incrementally learn 20 classes, then learn the last class. Similarly, we employ the three attack algorithms to attack under different perturbation constraints to get the attack success rate of the added one class. Second, we employ the DeepLab v2 network to incrementally learn 16 classes and then learn the last five classes (here we are adding five classes at once). In the same way, we use the three attack algorithms to attack under different perturbation

constraint to get the attack success rate of the added five classes.

B. Robustness Analysis of Incremental attack

Incremental learning is mainly reflected in two aspects: on the one hand, there is no need to reconstruct all the knowledge bases when every time new tasks are added. This will reduce the storage space occupation. On the other hand, incremental learning uses the results of the original knowledge base in the current sample training, and we only update the learning model for changes caused by new tasks. This will significantly reduce time for follow up training. In the growth process, people learn gradually carried out. Therefore, incremental learning is like to the model of human learning, and they usually do not forget the knowledge previously learned.

Incremental learning is to acquire new knowledge from new samples based on retaining the learning from old samples. In other words, these new samples have not been learned before incremental learning. For adversarial example, it actually adds subtle interference to the original data. And these new samples themselves are equivalent to adversarial examples. By continuously adding new samples for adversarial learning, the model is more robust.

The immaturity of deep learning models mainly depends the inexplicability of DNNs. Neural networks are easily deceived. Adversarial examples can fool the neural network through different adversarial example generation strategies, which treat unrecognizable images as images of known classes. This reveals the blind spots of deep learning algorithms, and also shows that there are hidden features and blind spots in the process of DNN learning through back-propagation. In addition, in [62] also shows that adversarial examples are not bugs but meaningful data distribution characteristics. The transferability of adversarial examples is the reason for adversarial example attacks. The transferability means that they are misclassified by the A_1 model, and can also be misclassified by the A_2 model. This illustrates that the attacker can misclassify examples without contacting the basic model. In recent years, many works [13], [17], [63]–[65] have also employed the migration of adversarial examples to achieve attacks. Therefore, when we perform incremental learning tasks in vehicular networks system, we need to consider the accuracy and robustness of the model.

V. Performance Evaluation

To evaluate the practical performance of our approach that proposed in section IV, this part start to introduce experimental setup. Then we introduce incremental learning of adversarial example attack experiment. The experimental scheme of incremental learning follows the proposed in [27] [29]. We conduct two experiments in alphabetical order, specifically adding the last class and adding five classes together.

A. Experimental Setup

We employ the Pascal VOC 2012 dataset [66] for experiments, which is divided into 21 different classes (including background). It contains 10582 training images and 1449 verification images. In addition, it contains six classes (bicycle, bus, car, motorbike, person, train) common unmanned scenes dataset. Therefore, the dataset not only can be employed to evaluate the performance of adversarial example attack based on incremental technologies in unmanned scenes but also is more general.

In this work, our training network is simulated on a computer with the experimental settings shown in Table I. First, we employ the semantic segmentation method DeepLab v2 network based on ResNet-101. Next, we employ the previously introduced baseline attack algorithms to attack classification models. We compare the attack success rate of whether to use incremental learning technology. Furthermore, we set different perturbation constraints ϵ when evaluating the attack success rate. In this way, we can better evaluate the impact of incremental learning.

TABLE I
Experimental Settings

Item	Settings
CPU	Intel Core i7-7800X
GPU	NVIDIA RTX2080 Ti
Hard Disk	500G
Operating System	Ubuntu 16.04
Programming Language	Python 3.6
Deep Learning Frame	Tensorflow, Keras

B. Add One Class of Attack Success Rate

First, in alphabetical order, we add the last class to the network. Specifically, we divide the 21 classes of dataset into two groups. The first group contains the top 20 classes, and the second group only contains the last class (tvmonitor). We call the model trained on the dataset M, and the model learned from 21 classes is called $M(0-20)$. We use two incremental learning methods L'_D and $E_qL'_D$ to illustrate the effect of an adversarial example attack based on incremental learning techniques. As shown in Fig.8, examples of semantic segmentation for the two incremental learning methods when we add the last class. The IOU values of the semantic segmentation results obtained by the two incremental learning methods, as shown in Table II.

Semantic segmentation map obtained according to the semantic segmentation model DeepLab V2 network. We employ three attack algorithms to incrementally attack the classification model, respectively. This is because the nature of the attack is an attack classification model. For the incremental learning of added the last class, we employ three attack algorithms to attack the semantic segmentation maps obtained

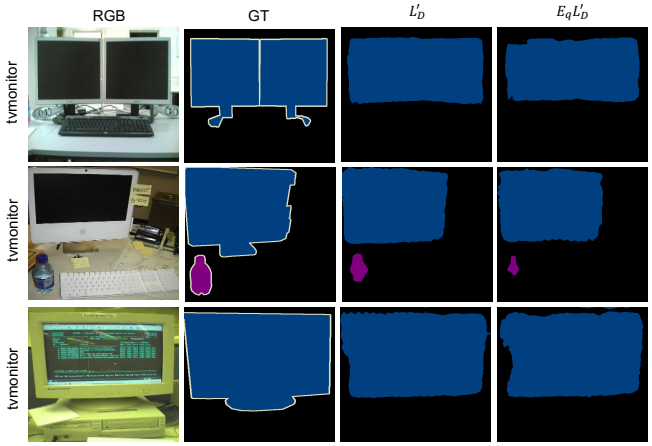

Fig. 8. Examples of semantic segmentation maps when added one class. Blue represents the tvmonitor, and purple represents the bottle.

TABLE II
IOU ON VOC2012 DATASET WHEN ADDING ONE CLASS

M	IOU[%]	IOU of authors[%]
M(0-19)(L'_D)	66.2	68.4
M(0-19)+M(20)($E_F L'_D$)	68.7	71.5
M(0-20)	70.5	73.6

by the two incremental methods, respectively. We compare the attack success rate of whether to employ incremental learning technology. In addition, we set different perturbation constraints ϵ when evaluating the attack success rate. To compare the results of incremental learning of the last class, we choose to attack the top 20 classes when the perturbation constraint ϵ is 0.3. We perform experiment on the three attack algorithms when we employ two incremental learning methods to add the last class(tvmonitor), and the results are shown in Table IV. We can perceive that the first incremental method L'_D can achieve better attack effects when we add the last class.

As shown in Fig.10, first, we choose the perturbation constraint $\epsilon = 0.3$ and FGSM algorithm for detailed analysis. When we employ the first incremental learning method L'_D, the attack success rate can reach 94.55%. When we employ the second incremental learning method $E_qL'_D$, the attack success rate can reach 92.10%. Without using incremental learning methods, the attack success rate is only 86.12%, which increases the successful attack rate by 8.43%. From the results of attack on only 20 classes, it is can perceive that the attack success rate after incremental learning is indeed improved. Similarly, for the DeepFool algorithm, we analyze the situation when the perturbation constraint $\epsilon = 0.2$. When we adopt incremental learning method L'_D, the attack success rate can reach 83.71%. The attack success rate can reach 81.52% when we use $E_qL'_D$. If the incremental learning method is not used, the attack success rate is only 80.18%. It can increase the successful rate by 3.53% when perturbation constraint $\epsilon = 0.2$. For the MI-FGSM algorithm, it can be observed that the incremental learning method L'_D can increase the successful attack rate of adversarial examples by 2.59% when the perturbation constraint is $\epsilon = 0.3$. Besides, $E_qL'_D$ also increase a certain attack success rate, but not as much as L'_D. So we can conclude that the second incremental learning method $E_qL'_D$ has better effects on robustness than L'_D. To sum up, adversarial attacks based on incremental learning technology not only have a higher attack success rate, but also can solve the catastrophic forgetting of deep learning architecture.

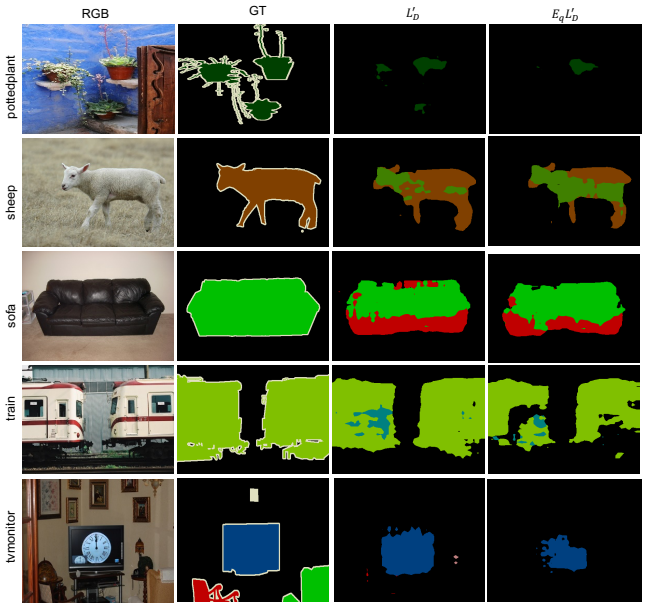

Fig. 9. Examples of semantic segmentation maps when added five classes.

TABLE III
IOU ON VOC2012 DATASET WHEN ADDING FIVE CLASSES

M	IOU[%]	IOU of authors[%]
M(0-15)+M(16-20)(L'_D)	62.5	65.7
M(0-15)+M(16-20)($E_F L'_D$)	62.4	64.2
M(0-20)	70.5	73.6

C. Add Five Classes of Attack Success Rate

We divide the 21 classes of dataset into two groups to see the impact of incremental learning technology. The difference from the previous is the first group contains the top 16 classes, and the second group contains the last five classes (potted-plant, sheep, sofa, train, tvmonitor). Fig.9 shows examples of semantic segmentation for two incremental learning methods.

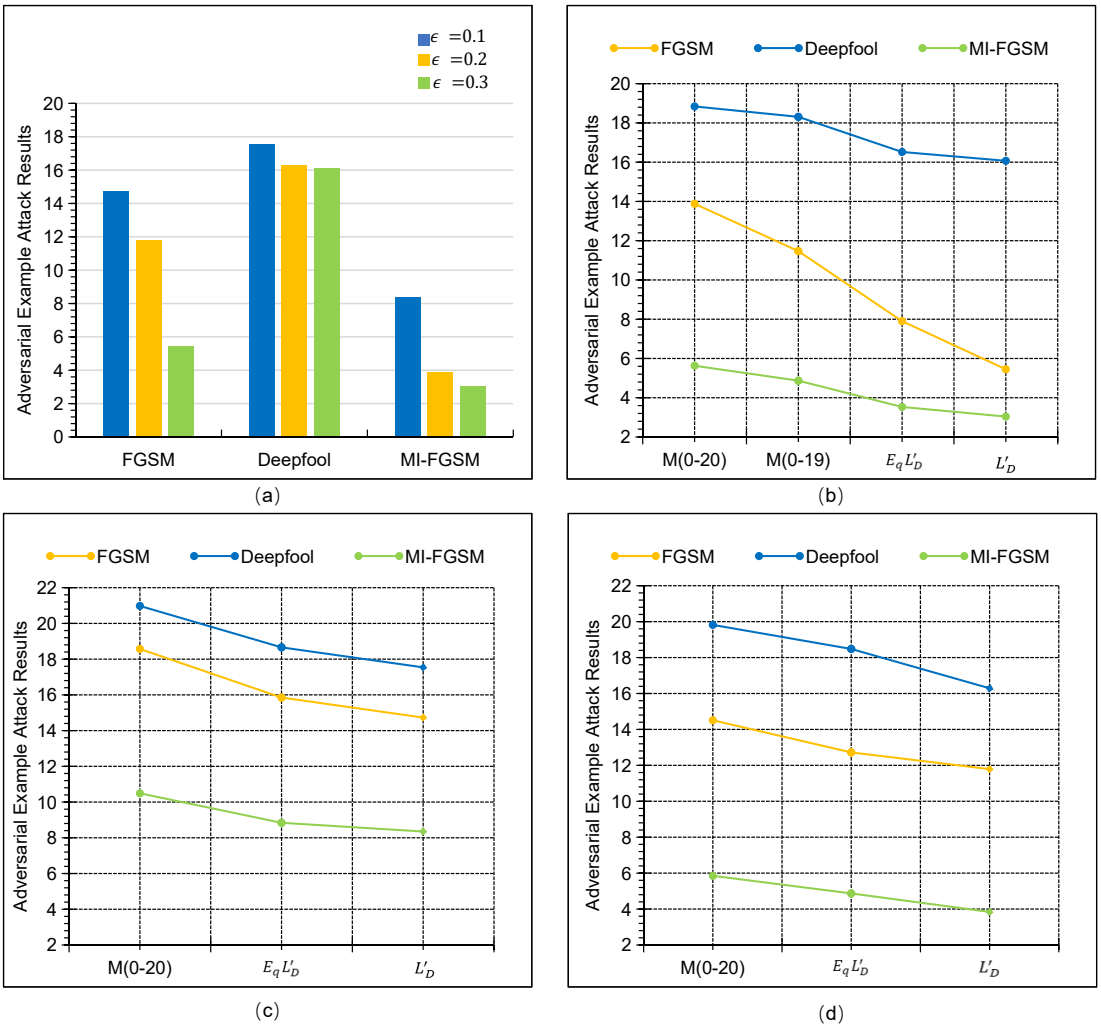

Fig. 10. (a) shows the classification results of the three algorithms attacking the model under different disturbances, when we apply the first incremental learning technique L'_D. (b) shows the classification result of incremental attack when the perturbation constraint $\epsilon = 0.3$. (c) shows the classification result of incremental attack when the perturbation constraint $\epsilon = 0.1$. (d) shows the classification result of incremental attack when the perturbation constraint $\epsilon = 0.2$.

TABLE IV
ADVERSARIAL EXAMPLE ATTACK RESULTS WHEN ADDING ONE CLASS. THE CLASSIFICATION ACCURACY OF THE THREE ATTACK ALGORITHMS AFTER ATTACK UNDER DIFFERENT DISTURBANCE VALUES ϵ.

		Adversarial Attack Algorithms		
M	ϵ	FGSM	DeepFool	MI-FGSM
M(0-19)	$\epsilon = 0.3$	**11.47%**	**18.31%**	**4.87%**
M(0-19)+M(20) (L'_D)	$\epsilon = 0.1$	14.73%	17.54%	8.35%
	$\epsilon = 0.2$	11.79%	16.29%	3.84%
	$\epsilon = 0.3$	**5.45%**	**16.07%**	**3.04%**
M(0-19)+M(20) ($E_q L'_D$)	$\epsilon = 0.1$	15.85%	18.66%	8.84%
	$\epsilon = 0.2$	12.72%	18.48%	4.87%
	$\epsilon = 0.3$	**7.90%**	**16.52%**	**3.53%**
M(0-20)	$\epsilon = 0.1$	18.57%	20.98%	10.49%
	$\epsilon = 0.2$	14.51%	19.82%	5.85%
	$\epsilon = 0.3$	**13.88%**	**18.84%**	**5.63%**

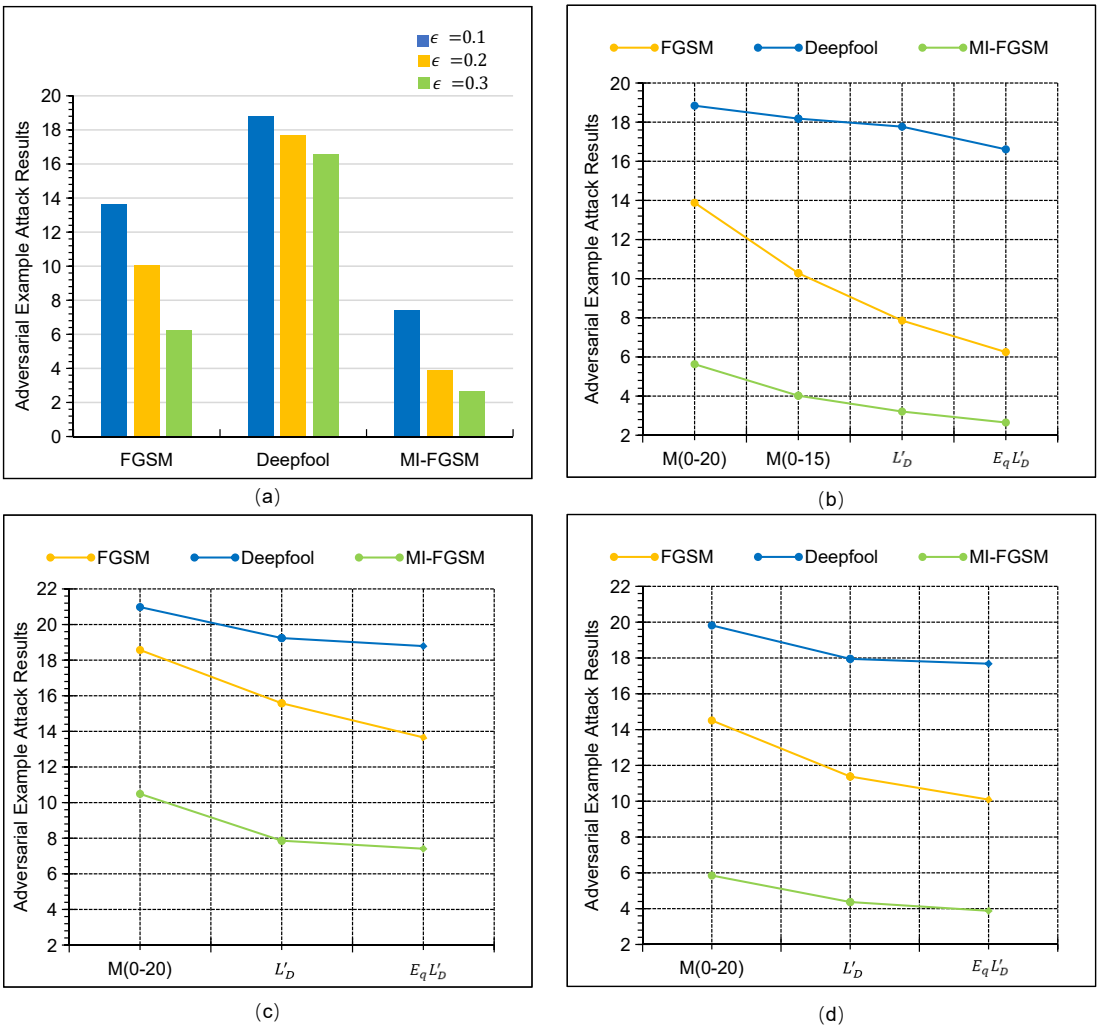

Fig. 11. (a) shows the classification results of the three algorithms attacking the model under different disturbances, when we apply the second incremental learning technique $E_qL'_D$. (b) shows the classification result of incremental attack when the perturbation constraint $\epsilon = 0.3$. (c) shows the classification result of incremental attack when the perturbation constraint $\epsilon = 0.1$. (d) shows the classification result of incremental attack when the perturbation constraint $\epsilon = 0.2$.

TABLE V
ADVERSARIAL EXAMPLE ATTACK RESULTS WHEN ADDING FIVE CLASSES. THE CLASSIFICATION ACCURACY OF THE THREE ATTACK ALGORITHMS AFTER ATTACK UNDER DIFFERENT DISTURBANCE VALUES ϵ.

		Adversarial Attack Algorithms		
M	ϵ	FGSM	DeepFool	MI-FGSM
M(0-15)	$\epsilon = 0.3$	**10.28%**	**18.18%**	**4.02%**
M(0-15)+M(16-20) (L'_D)	$\epsilon = 0.1$	15.58%	19.24%	7.86%
	$\epsilon = 0.2$	11.38%	17.95%	4.37%
	$\epsilon = 0.3$	**7.86%**	**17.77%**	**3.21%**
M(0-15)+M(16-20) ($E_qL'_D$)	$\epsilon = 0.1$	13.66%	18.79%	7.41%
	$\epsilon = 0.2$	10.09%	17.68%	3.88%
	$\epsilon = 0.3$	**6.25%**	**16.61%**	**2.65%**
M(0-20)	$\epsilon = 0.1$	18.57%	20.98%	10.49%
	$\epsilon = 0.2$	14.51%	19.82%	5.85%
	$\epsilon = 0.3$	**13.88%**	**18.84%**	**5.63%**

The IOU values of the semantic segmentation results obtained by the two incremental learning methods, as shown in Table III.

For the incremental learning of the added five classes, we also use three attack algorithms to attack the semantic segmentation maps obtained by the two incremental methods, respectively. We also compare the attack success rate of whether to use incremental learning technology. Furthermore, to compare the results of incremental learning of the last five classes, we choose to attack the top 16 classes when the perturbation constraint $\epsilon = 0.3$. We perform experiment on the three attack algorithms when we employ two incremental learning methods to add the last five class (pottedplant, sheep, sofa, train, tvmonitor), as shown in Table V.

Similarly, as shown in Fig.11, we choose the perturbation constraint $\epsilon = 0.3$ and FGSM algorithm for detailed analysis. When we use the first incremental learning method L'_D, the attack success rate can reach 92.14%. When we apply $E_q L'_D$, the attack success rate can reach 93.75%. Without using incremental learning methods, the attack success rate is only 86.12%. From the results of attack on only 16 classes, we also can discover that the attack success rate after incremental learning is indeed improved. For the DeepFool algorithm, we also analyze the situation when the perturbation constraint $\epsilon = 0.3$. When we apply incremental learning method L'_D, the attack success rate can reach 82.23%. It can reach 83.39% when we use $E_q L'_D$. If the incremental learning method is not used, the attack success rate is only 81.16%. The incremental learning method $E_q L'_D$ can increase the successful attack rate by 2.23%. For the MI-FGSM algorithm, it can be found that the incremental learning method $E_q L'_D$ can increase the successful attack rate of adversarial examples by 3.08% when the perturbation constraint is $\epsilon = 0.1$. Therefore, when adding five classes, we can also draw the same conclusion as adding one class. Adversarial attacks based on incremental learning technology have a higher attack success rate. However, in this case, the first incremental learning method L'_D has better effects on robustness than $E_q L'_D$.

VI. Conclusion And Future Work

In this work, we have presented an adversarial attack based on incremental learning techniques for unmanned in 6G scenes. The development of 6G-based Internet of Vehicles systems can make future unmanned technologies develop faster and safer. Compared with the three baseline attack algorithms, we conclude that the adversarial attack based on incremental learning technology has a good attack effect. The system model in this paper can learn new knowledge without storing images of old tasks, which can reduce the waste of time and space, and can also solve the catastrophic forgetting problem of deep learning architecture.

In the future work, we will try a dataset consisting of only unmanned scene classes, and we can use other incremental learning methods for better evaluation. We will also try to analyze the impact of adversarial attacks based on incremental learning technology on unmanned object detection tasks. If conditions permit, we will try to combine with vehicular networks in the era of 6G.

References

[1] T. K. Rodrigues, K. Suto, and N. Kato, "Edge cloud server deployment with transmission power control through machine learning for 6g internet of things," *IEEE Transactions on Emerging Topics in Computing*, 2019.

[2] W. Saad, M. Bennis, and M. Chen, "A vision of 6g wireless systems: Applications, trends, technologies, and open research problems," *IEEE network*, vol. 34, no. 3, pp. 134–142, 2019.

[3] S. Zhang, H. Zhang, and L. Song, "Beyond d2d: Full dimension uav-to-everything communications in 6g," *IEEE Transactions on Vehicular Technology*, 2020.

[4] J. Liu, J. Ren, W. Dai, D. Zhang, P. Zhou, Y. Zhang, G. Min, and N. Najjari, "Online Multi-Workflow Scheduling under Uncertain Task Execution Time in IaaS Clouds," *IEEE Transactions on Cloud Computing*, pp. 1–1, 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8669862/

[5] Z. Ning, J. Huang, X. Wang, J. J. P. C. Rodrigues, and L. Guo, "Mobile edge computing-enabled internet of vehicles: Toward energy-efficient scheduling," *IEEE Network*, vol. 33, no. 5, pp. 198–205, 2019.

[6] J. Ren, D. Zhang, S. He, Y. Zhang, and T. Li, "A Survey on End-Edge-Cloud Orchestrated Network Computing Paradigms: Transparent Computing, Mobile Edge Computing, Fog Computing, and Cloudlet," *ACM Computing Surveys*, vol. 52, no. 6, pp. 1–36, Jan. 2020. [Online]. Available: https://dl.acm.org/doi/10.1145/3362031

[7] H. Yang, X. Xie, and M. Kadoch, "Intelligent resource management based on reinforcement learning for ultra-reliable and low-latency iov communication networks," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 5, pp. 4157–4169, 2019.

[8] F. Lyu, J. Ren, N. Cheng, P. Yang, M. Li, Y. Zhang, and X. Shen, "LEAD: Large-Scale Edge Cache Deployment Based on Spatio-Temporal WiFi Traffic Statistics," *IEEE Transactions on Mobile Computing*, pp. 1–1, 2020. [Online]. Available: https://ieeexplore.ieee.org/document/9055080/

[9] A. Kirillov, R. Girshick, K. He, and P. Dollár, "Panoptic feature pyramid networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 6399–6408.

[10] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[11] B. Liu, H. Liu, and J. Yuan, "Lane Line Detection based on Mask R-CNN," in *Proceedings of the 3rd International Conference on Mechatronics Engineering and Information Technology (ICMEIT 2019)*. Dalian, China: Atlantis Press, 2019. [Online]. Available: https://www.atlantis-press.com/article/55917250

[12] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[13] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *arXiv preprint arXiv:1312.6199*, 2013.

[14] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.

[15] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The limitations of deep learning in adversarial settings," in *2016 IEEE European symposium on security and privacy (EuroS&P)*. IEEE, 2016, pp. 372–387.

[16] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "Deepfool: a simple and accurate method to fool deep neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2574–2582.

[17] Y. Dong, F. Liao, T. Pang, H. Su, J. Zhu, X. Hu, and J. Li, "Boosting adversarial attacks with momentum," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 9185–9193.

[18] Y. Deng, X. Zheng, T. Zhang, C. Chen, G. Lou, and M. Kim, "An analysis of adversarial attacks and defenses on autonomous driving models," *arXiv: Signal Processing*, 2020.

[19] M. A. Alcorn, Q. Li, Z. Gong, C. Wang, L. Mai, W.-S. Ku, and A. Nguyen, "Strike (with) a pose: Neural networks are easily fooled by strange poses of familiar objects," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4845–4854.

[20] A. Chakraborty, M. Alam, V. Dey, A. Chattopadhyay, and D. Mukhopadhyay, "Adversarial attacks and defences: A survey," *arXiv preprint arXiv:1810.00069*, 2018.

[21] R. Duan, X. Ma, Y. Wang, J. Bailey, A. K. Qin, and Y. Yang, "Adversarial camouflage: Hiding physical-world attacks with natural

styles," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 1000–1008.

[22] F. Tang, Y. Kawamoto, N. Kato, and J. Liu, "Future intelligent and secure vehicular network toward 6g: Machine-learning approaches," *Proceedings of the IEEE*, vol. 108, no. 2, pp. 292–307, 2019.

[23] N. Kato, B. Mao, F. Tang, Y. Kawamoto, and J. Liu, "Ten challenges in advancing machine learning technologies toward 6g," *IEEE Wireless Communications*, 2020.

[24] Y. Sun, J. Liu, J. Wang, Y. Cao, and N. Kato, "When machine learning meets privacy in 6g: A survey," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 4, pp. 2694–2724, 2020.

[25] K. Wang, P. Xu, C.-M. Chen, S. Kumari, M. Shojafar, and M. Alazab, "Neural architecture search for robust networks in 6g-enabled massive iot domain," *IEEE Internet of Things Journal*, 2020.

[26] J.-M. Perez-Rua, X. Zhu, T. M. Hospedales, and T. Xiang, "Incremental few-shot object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 13 846–13 855.

[27] K. Shmelkov, C. Schmid, and K. Alahari, "Incremental learning of object detectors without catastrophic forgetting," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 3400–3409.

[28] V. Losing, B. Hammer, and H. Wersing, "Incremental on-line learning: A review and comparison of state of the art algorithms," *Neurocomputing*, vol. 275, pp. 1261–1274, 2018.

[29] U. Michieli and P. Zanuttigh, "Incremental learning techniques for semantic segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, Oct 2019.

[30] J. Zhang, J. Zhang, S. Ghosh, D. Li, S. Tasci, L. Heck, H. Zhang, and C.-C. J. Kuo, "Class-incremental learning via deep model consolidation," in *The IEEE Winter Conference on Applications of Computer Vision*, 2020, pp. 1131–1140.

[31] S. Minaee, Y. Boykov, F. Porikli, A. Plaza, N. Kehtarnavaz, and D. Terzopoulos, "Image segmentation using deep learning: A survey," *arXiv preprint arXiv:2001.05566*, 2020.

[32] A. Arnab, O. Miksik, and P. H. Torr, "On the robustness of semantic segmentation models to adversarial attacks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 888–897.

[33] X. Wang, M. Wen, J. Li, Z. Fu, R. Lu, and K. Chen, "Adversarial attack against scene recognition system for unmanned vehicles," in *Proceedings of the ACM Turing Celebration Conference-China*, 2019, pp. 1–6.

[34] X. Yuan, P. He, Q. Zhu, and X. Li, "Adversarial Examples: Attacks and Defenses for Deep Learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 9, pp. 2805–2824, Sep. 2019.

[35] M. Ozdag, "Adversarial attacks and defenses against deep neural networks: a survey," *Procedia Computer Science*, vol. 140, pp. 152–161, 2018.

[36] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *2017 ieee symposium on security and privacy (sp)*. IEEE, 2017, pp. 39–57.

[37] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," *arXiv preprint arXiv:1706.06083*, 2017.

[38] J. Su, D. V. Vargas, and K. Sakurai, "One pixel attack for fooling deep neural networks," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 5, pp. 828–841, 2019.

[39] W. Hu and Y. Tan, "Generating adversarial malware examples for black-box attacks based on gan," *arXiv preprint arXiv:1702.05983*, 2017.

[40] S. Sarkar, A. Bansal, U. Mahbub, and R. Chellappa, "Upset and angri: Breaking high performance image classifiers," *arXiv preprint arXiv:1707.01159*, 2017.

[41] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," *arXiv preprint arXiv:1607.02533*, 2016.

[42] K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, C. Xiao, A. Prakash, T. Kohno, and D. Song, "Robust physical-world attacks on deep learning visual classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[43] T. B. Brown, D. Mané, A. Roy, M. Abadi, and J. Gilmer, "Adversarial patch," *arXiv preprint arXiv:1712.09665*, 2017.

[44] A. Liu, X. Liu, J. Fan, Y. Ma, A. Zhang, H. Xie, and D. Tao, "Perceptual-sensitive gan for generating adversarial patches," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 1028–1035.

[45] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska *et al.*, "Overcoming catastrophic forgetting in neural networks," *Proceedings of the national academy of sciences*, vol. 114, no. 13, pp. 3521–3526, 2017.

[46] Z. Li and D. Hoiem, "Learning without forgetting," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 12, pp. 2935–2947, 2017.

[47] P. Zhou, L. Mai, J. Zhang, N. Xu, Z. Wu, and L. S. Davis, "M2kd: Multi-model and multi-level knowledge distillation for incremental learning," *arXiv preprint arXiv:1904.01769*, 2019.

[48] T. He, C. Shen, Z. Tian, D. Gong, C. Sun, and Y. Yan, "Knowledge adaptation for efficient semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 578–587.

[49] D. Guo, Y. Pei, K. Zheng, H. Yu, Y. Lu, and S. Wang, "Degraded image semantic segmentation with dense-gram networks," *IEEE Transactions on Image Processing*, vol. 29, pp. 782–795, 2020.

[50] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[51] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.

[52] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Semantic image segmentation with deep convolutional nets and fully connected crfs," *arXiv preprint arXiv:1412.7062*, 2014.

[53] ——, "DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 4, pp. 834–848, Apr. 2018. [Online]. Available: http://ieeexplore.ieee.org/document/7913730/

[54] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[55] G. Lin, A. Milan, C. Shen, and I. Reid, "Refinenet: Multi-path refinement networks for high-resolution semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1925–1934.

[56] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2881–2890.

[57] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 801–818.

[58] K. Ren, T. Zheng, Z. Qin, and X. Liu, "Adversarial attacks and defenses in deep learning," *Engineering*, 2020.

[59] C. Badue, R. Guidolini, R. V. Carneiro, P. Azevedo, V. B. Cardoso, A. Forechi, L. Jesus, R. Berriel, T. Paixão, F. Mutz *et al.*, "Self-driving cars: A survey," *arXiv preprint arXiv:1901.04407*, 2019.

[60] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.

[61] Z. Gong, Y. Yang, and L. Ma, "Pedestrian Parsing by Joint Learning from Wholes and Parts," in *2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*. Taipei, Taiwan: IEEE, Sep. 2019, pp. 1–8. [Online]. Available: https://ieeexplore.ieee.org/document/8909822/

[62] A. Ilyas, S. Santurkar, D. Tsipras, L. Engstrom, B. Tran, and A. Madry, "Adversarial examples are not bugs, they are features," in *Advances in Neural Information Processing Systems*, 2019, pp. 125–136.

[63] Y. Liu, X. Chen, C. Liu, and D. Song, "Delving into transferable adversarial examples and black-box attacks," *arXiv preprint arXiv:1611.02770*, 2016.

[64] F. Liao, M. Liang, Y. Dong, T. Pang, X. Hu, and J. Zhu, "Defense against adversarial attacks using high-level representation guided denoiser," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1778–1787.

[65] D. Wu, Y. Wang, S.-T. Xia, J. Bailey, and X. Ma, "Skip connections matter: On the transferability of adversarial examples generated with resnets," *arXiv preprint arXiv:2002.05990*, 2020.

[66] M. Everingham, L. Van Gool, C. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes challenge 2012 (voc2012)," *Results*, 2012.

Huanhuan Lv received the Bachelor's degree in automation from Hohai University Wentian College, China, in 2018. She is currently pursuing the M.S. degree with the College of Computer Science and Technology, Shanghai University of Electric Power, China. Her current research interests include artificial intelligence security and adversarial attacks and defenses.

Rongxing Lu (S'09-M'11-SM'15-F'21) is an associate professor at the Faculty of Computer Science (FCS), University of New Brunswick (UNB), Canada. Before that, he worked as an assistant professor at the School of Electrical and Electronic Engineering, Nanyang Technological University (NTU), Singapore from April 2013 to August 2016. Rongxing Lu worked as a Postdoctoral Fellow at the University of Waterloo from May 2012 to April 2013. He was awarded the most prestigious "Governor General's Gold Medal", when he received his PhD degree from the Department of Electrical & Computer Engineering, University of Waterloo, Canada, in 2012; and won the 8th IEEE Communications Society (ComSoc) Asia Pacific (AP) Outstanding Young Researcher Award, in 2013. He is presently an IEEE Fellow. Dr. Lu currently serves as the Vice-Chair (Conferences) of IEEE ComSoc CIS-TC. Dr.Lu is the Winner of 2016-17 Excellence in Teaching Award, FCS, UNB.

Mi Wen (M'10) received the M.S. degree from the University of Electronic Science and Technology of China, Chengdu, China, in 2005, and the Ph.D. degree from Shanghai Jiao Tong University, Shanghai, China, in 2008, both in computer science. She is currently a Professor of the College of Computer Science and Technology with Shanghai University of Electric Power, Shanghai, China. From May 2012 to May 2013, she was a Visiting Scholar at the University of Waterloo, Waterloo, ON, Canada. In 2016, she was awared as Shanghai Municipal dawn scholar. Her research interests include privacy preserving in wireless networks, big data, smart grid, etc. She is an Associate Editor of Peer-to- Peer Networking and Applications (Springer). She is presently an director of Shanghai Computer Society. She acts as the track chairs of many conferences such as the IEEE VTC et al.

Jinguo Li (M'16) received the B.S. degree in information security and the Ph.D. degree in computer science and technology from Hunan University, China, in 2007 and 2014, respectively. He is currently an Associate Professor with the College of Computer Science and Technology, Shanghai University of Electric Power. His research interests include information security and privacy, applied cryptography, and cloud computing.