IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 20, 2020, accepted February 1, 2020, date of publication February 10, 2020, date of current version February 20, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2973069

Adversarial Attacks for Image Segmentation on Multiple Lightweight Models

XU KANG[1], BIN SONG[1], (Senior Member, IEEE), XIAOJIANG DU[2], (Fellow, IEEE), AND MOHSEN GUIZANI[3], (Fellow, IEEE)

[1]State Key Laboratory of Integrated Services Networks, Xidian University, Xi'an 710071, China
[2]Department of Computer and Information Sciences, Temple University, Philadelphia, PA 19122, USA
[3]Department of Computer Science and Engineering, Qatar University, Doha 2713, Qatar

Corresponding author: Bin Song (bsong@mail.xidian.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61772387, in part by the Fundamental Research Funds of Ministry of Education and China Mobile under Grant MCM20170202, in part by the National Natural Science Foundation of Shaanxi Province under Grant 2019ZDLGY03-03, and in part by the ISN State Key Laboratory.

ABSTRACT Due to the powerful ability of data fitting, deep neural networks have been applied in a wide range of applications in many key areas. However, in recent years, it was found that some adversarial samples easily fool the deep neural networks. These input samples are generated by adding a few small perturbations based on the original sample, making a very significant influence on the decision of the target model in the case of not being perceived. Image segmentation is one of the most important technologies in the medical image and automatic driving field. This paper mainly explores the security of deep neural network models based on the image segmentation tasks. Two lightweight image segmentation models on the embedded device suffered from the white-box attack by using local perturbations and universal perturbations. The perturbations are generated indirectly by a noise function and an intermediate variable so that the gradient of pixels can be propagated unlimitedly. Through experiments, we find that different models have different blind spots, and the adversarial samples trained for a single model have no transferability. In the end, multiple models are attacked by our joint learning. Finally, under the constraint of low perturbation, most of the pixels in the attacked area have been misclassified by both lightweight models. The experimental result shows that the proposed adversary is more likely to affect the performance of the segmentation model compared with the FGSM.

INDEX TERMS Adversarial samples, image segmentation, joint learning, multi-model attack, perturbations.

I. INTRODUCTION

Recently, deep neural networks have been widely applied in various fields, including computer vision, speech recognition, natural language processing and robotics [1]. Deep neural networks are characterized by learning appropriate low-level features from the data rather than relying on handwriting to explicitly program them, which requires less human intervention [2]. Generally, image segmentation technology is the most fun part of computer vision and the basis of all other image processing methods. The quality of image segmentation technology will affect the effect of subsequent processing to a large extent. In the field of computer vision, semantic image segmentation is an essential method of scene understanding that can be used for autonomous driving, video surveillance and robotics [3]. Moreover, in the era of artificial intelligence (AI), most computer vision techniques are based on image segmentation, and research on image segmentation techniques has been underway for decades. From early graphics processing algorithms to deep learning algorithms, thanks to the development of hardware, the improvement of computing capacity, and the generation of massive image data in the information society [4].

With the development of deep learning algorithms, computer vision technology represented by image segmentation has once again entered people's field of vision. Image segmentation algorithms based on deep learning are constantly being proposed [5]. Compared with traditional image segmentation algorithms, deep neural networks exhibit the state-of-the-art performance in image segmentation tasks that rely on large amounts of image data. But it also has some problems. Recent studies found that deep neural networks are

The associate editor coordinating the review of this manuscript and approving it for publication was Kim-Kwang Raymond Choo.

easily attacked by some adversarial samples [6]. Especially in the field of computer vision, well-designed image disturbances can lead to neural network mistakes such as confusing a cat with a computer. This is because the network may not be able to properly classify natural inputs, although this is almost identical to the previously correctly categorized example [1] Adversarial sample raises doubts about the use of DNNs in safety-critical applications. Also, it allows malicious agents to attack systems that use neural networks [7].

Specifically, the adversarial examples enable the network to make arbitrary incorrect predictions by adding intentionally perturbed inputs with small magnitude adversariial perturbation [8]. Therefore, the security issue of the deep neural network model has attracted a lot of attention in the field of safety and security. At present, the research on the adversarial examples mainly focuses on the task of image classification. With the full application in these fields, confrontational attacks have become an essential topic in the study of semantic segmentation systems. This problem has recently attracted a lot of attention and various analyses for understanding adversarial examples have been proposed [9]. For example, it has been suggested that another type of random noise can be added to the input in image preprocessing, and retraining has been proposed to detect adversarial examples when used to defend and classify images [10]. However, these defenses are vulnerable to attacks of other types of attacks or have higher input costs [11].

Therefore, in this paper, we investigate two lightweight image segmentation models on the embedded device that are attacked by using local perturbations and universal perturbations. Generally, the perturbations are generated indirectly by a noise function and an intermediate variable so that the gradient of pixel noise points can be propagated unlimitedly.

Our main contributions are summarized as follows:

1. We first introduced the non-linear adversarial samples generation method avoiding miss gradient from truncating the pixel values in the adversarial images.

2. Then, the perturbations for the local source domain and the universal perturbations on image segmentation are proposed. And a comparison of the advantages and disadvantages of the two methods in the adversarial attack was made.

3. Through experiments, we show that this adversarial learning on the deep neural network for image segmentation task is not transferable, so a kind of adversarial attack method based on multi-model joint learning is proposed.

The remainder of the paper is organized as follows. Section II briefly introduces the related work of image segmentation and adversarial learning. Following that, we expound the non-linear adversarial samples generation method for perturbations to source domain and the universal perturbations on image segmentation in section III. Then we analyze the non-transferability of adversarial samples between different models and propose the joint learning method for multi-model attack in section IV. Section V illustrates and discusses the experimental results. Finally, section VI concludes the paper.

II. RELATED WORK

A. ADVERSARIAL EXAMPLES

Although the performance of deep neural architectures in challenging visual classification benchmarks was impressive, these classifiers were highly susceptible to perturbations. In [12], the authors firstly generated small perturbations on the images in terms of the image classification problem. They made CNNs predict a wrong label with high confidence while these additive perturbations stay almost imperceptible to human eyes. Goodfellow *et al.* [13] and Kurakin *et al.* [14] define these misclassified samples as adversarial examples and explained that they are ''inputs of machine learning models that an attacker has intentionally designed to cause the model to make a mistake''. Such carefully crafted perturbations can be formed by using a gradient-based optimizer to search for a nearby image [10] and estimated by solving an optimization problem [15]. By assuming that the loss function can be linearized around the current data point at each iteration, [16] proposed a simple algorithm to compute the minimal adversarial perturbation. However, without using gradients, the authors in [17] trained a network to generate adversarial examples for a particular target model.

If the adversary succeeds in causing any error at all, the attacks are called untargeted. On the contrary, the attacks are targeted when the adversary succeeds in causing the model to predict a specific incorrect class. The transferability of both untargeted and targeted adversarial examples was studied in [18] and ensemble-based approaches to generate adversarial examples with stronger transferability was proposed. Moosavidezfooli, Seyed Mohsen, et al [19] proposed a systematic algorithm for computing universal perturbations, and show that state-of-the-art deep neural networks are highly vulnerable to such perturbations. The attacks that instead reprogram the target model to perform a task chosen by the attacker—without the attacker needing to specify or compute the desired output for each test-time input was introduced in [20]. Further, [7] showed the adversarial examples for machine learning systems also exist in the physical world. In addition to machine learning, adversarial attacks have contributed to other areas [21]–[23]. Also, recent studies show that adversarial examples can be applied to the real worlds, such as object recognition system [41], controllable voice system [24] and traffic sign recognition system [25]. Zeng et al. proposed a novel audio detection approach to determine whether audio is an adversarial example [26]. Xiao et al. designed a malware detection scheme with Q-learning for a mobile device to derive the optimal offloading rate.

B. SEMANTIC SEGMENTATION

Semantic image segmentation denotes a dense prediction task that which requires high-level features to represent each pixel of the image and assign a class label. Deep learning based methods perform best in semantic segmentation tasks [4], [28], [29], [30]. Yu, F. and Koltun, V. developed a new convolutional network module that is specifically

designed for dense prediction, increasing the accuracy of state-of-the-art semantic segmentation systems [30]. In [31], the authors proposed that feature maps from middle or early layers are also used by skip-connections to compensate for the low resolution of high-level features. Encoder-decoders [31]–[33] are another widely used framework. Long et al. introduced FCN-8s for the VGG16 model in [28], which can be divided into an encoder part and a decoder part. The encoder part is used to transform a given image into a low-resolution semantic representation and the decoder part in charge of increasing the localization accuracy and yielding the final semantic segmentation at the resolution of the input image. On the basis of FCN, SegNet [31] introduced a joint encoder-decoder model which is one of the earliest effective segmentation models. Following SegNet, ENet [33] also designed an encoder-decoder model with few layers to reduce computing costs.

A benchmark suite and large-scale dataset to train and test approaches for pixel-level and instance-level semantic labeling were introduced in [34]. Since then, many lightweight image segmentation models for street scene understanding task in autonomous driving road traffic scenes have been proposed [35]–[40], and they are committed to exploiting the model to evaluate road traffic images on embedded devices. Yang et al. proposed to concatenate multiple atrous-convolved features using different dilation rates into a final feature representation in a dense way [37]. Yu et al. proposed a discriminative feature network to handle the intra-class inconsistency problem [38]. A fast and efficient spatial pyramid neural network for semantic segmentation of high resolution images under resource constraints was introduced in [39]. A fast and real time segmentation convolutional neural network on embedded devices with low memory was proposed in [40]. Xie et al. [41] proposed dense adversary generation for segmentation and detection so that the perturbations can be transferred across networks with different training data, based on different architectures, and even for different recognition tasks.

III. METHODOLOGY

The traditional methods of image segmentation [1], [3], [8] are based on the FGSM method [14]. Arnab et al. [1] analyzed the effect of different network architectures, model capacity and multi-scale processing under FGSM. In [3], Static target segmentation and dynamic target segmentation are attacked by image-dependent perturbations and universal perturbations. These works once again proved the effectiveness of FGSM. The researchers explore the potential effects that spatial context information and spatial consistency have on benign and adversarial examples in segmentation models by FGSM in [8]. All of the above studies use FGSM by default because FGSM is an easy method to implement for image segmentation tasks. Although other latest generation methods such as the DeepFool method, JSMA, Carlini and Wagner method have achieved great success in adversarial image classification, it is still a complicated task to apply these methods to the semantic segmentation. In this section, we first analyze some shortcomings of the traditional FGSM method [14] in generating adversarial samples for image segmentation tasks. Then our non-linear generation method for adversarial image is put forward. Afterward, we introduce the perturbations based on the local source domain and the universal perturbations for adversarial attacks, respectively. The analysis and explanation for the comparison of the two methods are displayed in experimental results.

A. NON-LINEAR ADVERSARIAL SAMPLES GENERATION

The traditional iterative updating formula of FGSM is shown in equation (1). Where I indicates the original image while $L\left(f\left(I_t^{adv};\theta\right),\mathrm{y}\right)$ represents the loss function loss function between model output $f\left(I_t^{adv};\theta\right)$ and label y. $clip(I,\epsilon)$ function ensures that the perturbations added iteratively are not too large to cause a large distortion of the image generated in the previous step. Meanwhile, it makes sure that the pixel values of the updated image remain within the domain of definition. Nevertheless, this also brings two disadvantages. It's the $clip(I,\epsilon)$ function that results in the inadequate learning of the pixels whose values near the minimum (0) and the maximum (255). This situation is somewhat similar to the RELU activation function in neural networks, where the back propagation of the gradient is blocked in some regions of the activation function.

$$\begin{aligned} &I_0^{adv} = I \\ &I_{t+1}^{adv} = clip(I_t^{adv} + \alpha \cdot \mathrm{sign}(\nabla_{I_t^{adv}} L\left(f\left(I_t^{adv};\theta\right),\mathrm{y}\right)),\epsilon) \end{aligned} \quad (1)$$

Goodfellow *et al.* [13] found that the linear models also show obvious vulnerability to the adversarial samples. The linearity in the high dimensional space is enough to cause the adversarial sample. The converse also applies: using linear functions to construct adversarial samples is not sufficient to find all the blind spots of the target deep learning model. The use of non-linear functions to change the sample may help these highly over-fitted models explore the sample space that was not involved during their training stage.

On account of the analysis and thinking above, we try to attack the deep learning model with non-linear function for the generation of adversarial samples. The shape of tanh function in the interval $[-1.5, 1.5]$, as shown in Fig. 1. Most of the function values close to the origin present an approximately linear form. The values of the functions from both sides away from the original show non-linear properties. No matter how the input changes, the output always stays within the range $[-1, 1]$. Any perturbations will not affect the next iteration, when the original image is normalized to $[-1,1]$.

Our non-linear adversarial sample generation and iteration method is shown in equation (2).

$$\begin{aligned} I_0^{adv} &= I \\ I_{t+1}^{adv} &= tanh(W_t \odot I_t^{adv} + \xi_t) \\ \epsilon_{t+1} &= I_{t+1}^{adv} - I \end{aligned} \quad (2)$$

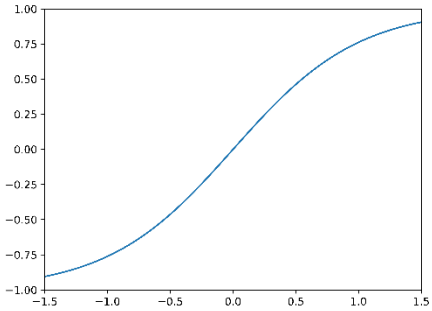

FIGURE 1. The shape of *tanh* function in the interval [−1.5, 1.5]. Most of the function values close to the origin presents an approximate linear form. The values of the functions from both sides away from the origin show non-linear properties. No matter how the input changes, the output always stays within the range [−1, 1].

where W_t indicates the scaling transformation matrix for pixel values while ξ_t represents the offset of the pixel values in $t'th$ step iteration. ϵ_t is the purterbations between adversarial image and original image. Our perturbations bring about from non-linear function with the intermediate variables W_t and ξ_t. $\odot$ denotes the element-wise multiplication between the scaling matrix and the adversarial image. Therefore, W_t and ξ_t have the same size as I. In each iteration, W and ξ are calculated based on the adversarial samples generated in the previous step:

$$\begin{aligned} W_{t+1} &= W_t - \alpha \cdot \nabla_{W_t} L(f\left(I_t^{adv}; \theta\right), y^{adv}) \\ \xi_{t+1} &= \xi_t - \alpha \cdot \nabla_{\xi_t} L(f\left(I_t^{adv}; \theta\right), y^{adv}) \end{aligned} \tag{3}$$

In Eq. (3), θ is the parameters of the attacking model which kept constant during adversarial learning. L is the loss function which minimize the predicted value of the ground-truth class and maximize the predicted value of the a class. L is composed of two terms in our attacking task, one is the cross entropy between the attack class and the adversarial class, the other is the amplitude of the perturbations.

$$\begin{aligned} L\left(f\left(I^{adv}; \theta\right), y^{adv}\right) &= C\left(f\left(I^{adv}; \theta\right), y^{adv}, \omega\right) + \lambda \left\|\epsilon_t\right\|_2 \\ \omega_i &= (p_i^{tar} - \min_c p_i^c)/(\max_c p_i^c - \min_c p_i^c) \end{aligned} \tag{4}$$

where C denotes the cross entropy between predicted value $f\left(I^{adv}; \theta\right)$ and the adversarial label y^{adv}. ω is a weight matrix for the attacking regions, where ω_{i} is defined as the weight of pixel i which is determined by the proportion of the predicted value of the target class with respect to the predicted value of other classes. Higher ω_{i} indicate that the pixel is hard to be attacked.

In our method, the role of *tanh* function is similar to that of non-linear activation function in neural networks. The RELU activation function in convolutional neural network is to reduce the amount of computation and increase the sparsity of the network, thus ensuring the generalization ability of the neural network and avoiding over-fitting on training data. There is no over-fitting problem in our adversarial-attack task, and it will be fatal to the target model even if only one adversarial sample exists. Moreover, the computational complexity of the adversarial learning process is insignificant compared with that of neural network. Therefore, the use of non-linear activation function and intermediate variables W and ξ can enrich our exploration of adversarial sample space without affecting the computational complexity.

B. ADVERSARIAL PERTURBATIONS TO LOCAL SOURCE DOMAIN

Image segmentation is to divide the image into several specific regions and to predict the proposals of interest. At the same time, each pixel in the image is expected to be correctly classified. The purpose of our attack is to fool the neural network model with the adversarial samples, so that some regions which are originally classified correctly can be mistakenly identified as another class. The image segmentation based on convolutional neural network labels every pixel during training, and the predicted feature map has the same size as the original image.

When attacking a model, the first step is to determine the target class to be attacked and the adversarial class, and then add the perturbations on the local source domain ϕ_t inferred from the model. The Eq. (5) tells the procedure. ϕ_t is a matrix of the same size as the original image where the values of the target regions are 1 otherwise 0. It is used to select the target regions in the original image and the predicted feature map. $\bar{\phi}_t$ denotes logical non-operation of each element in ϕ_t.

$$I_{t+1}^{adv} = tanh\left(W_t \odot I_t^{adv} + \xi_t\right) \odot \phi_t + I_t^{adv} \odot \bar{\phi}_t \tag{5}$$

The purpose of the local domain attack is to minimize the cross-entropy between the predicted probability of the ground-truth class and the adversarial class while maximizing the cross-entropy between the predicted probability of the attacked class and the ground-truth class on the local source domain pixels. As described in Eq. (6), $f^d(I^{adv}; \theta)$ represents the predictions on the source domain. y^{gt} and y^{adv} are the labels of the ground-truth and the adversarial images. The ultimate optimization objective can be inferred as the difference between the log-probability of the target class and the adversarial class.

$$\begin{aligned} C &= C\left(f^d\left(I^{adv}; \theta\right), y^{adv}\right) - C\left(f^d\left(I^{adv}; \theta\right), y^{gt}\right) \\ &= \sum\nolimits_i \omega_i log \frac{p_i^{tar}\left(I^{adv}; \theta\right)}{p_i^{adv}\left(I^{adv}; \theta\right)} \end{aligned} \tag{6}$$

The pseudo-code of adversarial perturbations to the local source domain is described in Algorithm 1. Each iteration does not update all the perturbation matrices, because the variables of the source domain variable ϕ_t is different at each step. Even so, the noise variables of the overlapping source domains will be updated emphatically due to their robustness to perturbations than other domains.

Algorithm 1 Adversarial Perturbations to Local Source Domain

Input: An image $I_0^{adv} = I$ from the dataset
Output: Final adversarial image I_T^{adv}

begin
  Initialization:
    Normalize the values of I_0^{adv} to $[-1, 1]$
    Init $W_0 \sim u(-1.01, 1.01)$, $\xi_0 \sim u(-0.01, 0.01)$
  For $t = 0, 1, \ldots, T-1$, **do**
    Infer $f(I_t^{adv}; \theta)$
    Infer the source domain ϕ_t from $f(I_t^{adv}; \theta)$
    Update the adversarial image with Eq. (2)
    Infer $f(I_{t+1}^{adv}; \theta)$
    Compute loss L with Eq. (4) and Eq. (6)
    Update W_t and ξ_t with Eq. (3)
  End
  Update the adversarial image I_t^{adv} with Eq. (2)
  Maps the pixel values in I_T^{adv} to the range [0, 255]
end

C. UNIVERSAL PERTURBATIONS ON IMAGE SEGMENTATIONS

Fig. 2 shows a presentation of two sets of local source domain attacks. The two images on the left are respectively the segmentation results of the original image by FastSCNN [40] and ESPNet (The predictions of ESPNet [39] includes the background class but FastSCNN does not). The two images on the right are respectively the segmentation results of the adversarial images by these two models. In the above two pictures, we attack cars (blue) as road (pink) while the pedestrians (red) are attacked as the vegetation (green) in the below.

FIGURE 2. This is a presentation of two sets of local source domain attacks. In the above two pictures, we attack cars (blue) as road (pink) while the pedestrians (red) are attacked as the vegetation (green) in the below. When the source domain is a convex set, the perturbations added is easy to attack the model. But when the source domain is irregularly shaped, this attack affects the classification of adjacent pixels, especially where there are other non-target classes embedded in.

When the source domain is a convex set, the perturbations added is easy to attack the model. But when the source domain is irregularly shaped, this attack affects the classification of adjacent pixels, especially where there are other non-target classes embedded in. This result is actually due to the fact that image segmentation based on a convolutional neural network is not a pixel to pixel task. Each neuron in a convolutional neural network has a receptive field. Each pixel on the predicted feature map is mapped from this particular region of the original image. When perturbations are added to the source domain, it affects not only the predicted value of the target region, but also that of the adjacent region.

In image convolution, the pixels on the next feature map are obtained by the interaction between the convolution kernel and the region of the same size. With the increase of the depth of convolutional layers, the corresponding receptive field of the pixels on the feature map of each layer also increases. The change of pixels in the receptive field of each feature point will affect the predicted value of it. The pixels in the source domain and the pixels in the non-source domain may belong to the same receptive field.

In order to make the source and non-target regions more cohesive in the attack result and not have a great impact on the boundary, we try to use universal perturbations to attack the model. Universal perturbations can learn the relationship between pixels, making the prediction in the source domain more false and the prediction in the non-source domain more real.

In the universal perturbations attack, Eq. (2) is still used to update the adversarial samples. In addition, we focus on the overall cross-entropy between the adversarial target and the whole image rather than the cross-entropy of the local source domain. We use Eq. (4) directly to update our loss function, where the part of the source domain in y^{adv} is changed from the ground-truth y^{gt}. Although we are running a white box attack, suppose that we cannot access the structure of the model and the ground truth labels, by default we cannot get the model structure and ground-truth labels. Here, we use the network prediction of the original sample image I_0^{adv} as ground-truth y^{gt}, which is the confidence result of the model. On this basis, we can find the place where the model is easy to attack. Eq. (7) illustrates the relationship between i'th feature and y^{gt}. Eq. (8) demonstrates the relationship between y^{adv} and y^{gt}

$$y^{gt} = \underset{c}{\operatorname{argmax}} f_c\left(I_0^{adv}; \theta\right) \tag{7}$$

$$y_{i,j}^{adv} = \begin{cases} cls^{adv}, & if\ (i,j) \in \varphi_0 \\ y_{i,j}^{gt}, & ohterwise \end{cases} \tag{8}$$

The procedure of our universal perturbations to attack the image segmentation task is illustrated in Algorithm 2.

IV. JOINT LEARNING FOR MULTI-MODEL ATTACK

Fig. 3 shows an illustration of the non-transferability of the adversarial attack methods. The image on the top left shows the result of an adversarial attack on the FastSCNN model (pedestrians to vegetation) using the universal perturbations method. The image on the top right is the result of the same adversarial attack (the same adversarial image input) on the ESPNet model. The image on the bottom left shows the result of an adversarial attack on the ESPNet model (cars to the road) using the universal perturbations method. The image

Algorithm 2 Universal Perturbations on Image Segmentations

Input: An image $I_0^{adv} = I$ from the dataset
Output: Final adversarial image I_T^{adv}

begin
  Initialization:
    Read an image $I_0^{adv} = I$ from the dataset
    Normalize the pixel values to $[-1, 1]$
    Init $W_0 \sim u(-1.01, 1.01)$, $\xi_0 \sim u(-0.01, 0.01)$
    Infer $f(I_0^{adv}; \theta)$
    Infer the source domain ϕ_0 from $f(I_0^{adv}; \theta)$
    Infer y^{gt} and y^{adv} from ϕ_0
  For $t = 1, \ldots, T$, **do**
    Update the adversarial image with Eq. (5)
    Infer $f(I_t^{adv}; \theta)$
    Compute loss L with Eq. (4) and Eq. (8)
    Update W_t and ξ_t with Eq. (3)
  End
  Update the adversarial image I_t^{adv} with Eq. (2)
  Maps the pixel values in I_T^{adv} to the range [0, 255]
End

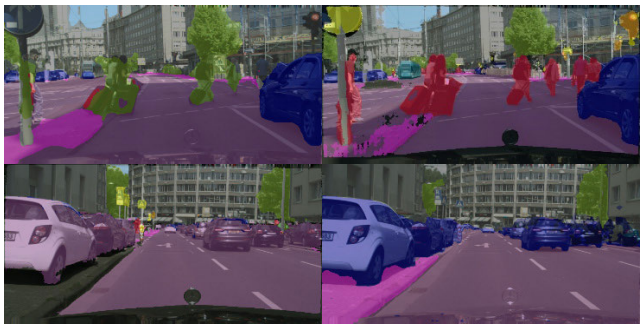

FIGURE 3. This is an illustration of the non-transferability of the adversarial attack methods. The two images above respectively are the segmentation results of the FastSCNN model and ESPNet model with the same adversarial image for FastSCNN as input. The two images below respectively are the segmentation results of the ESPNet model and FastSCNN model with the same adversarial image for ESPNet as input.

on the bottomright is the result of the same adversarial attack on the FastSCNN model. It can be concluded from this set of crossover experiments that adversarial samples inferred from a single model are likely to be an ineffective attack on another model.

Due to the difference in the network structure, even with the same output, it is difficult to ensure that the calculation process is consistent when generating adversarial samples. A comparison of the two outputs of the ESPNet and FastSCNN with the same noise image input is shown in Fig. 4. The noise image is the same size as the original image, with the pixel value set to 0 (−1 for model input) for all locations except the 100×100 pixels around the center point set to 255 (1 for model input). Interestingly, ESPNet and FastSCNN respond differently to such a noisy image, even though they

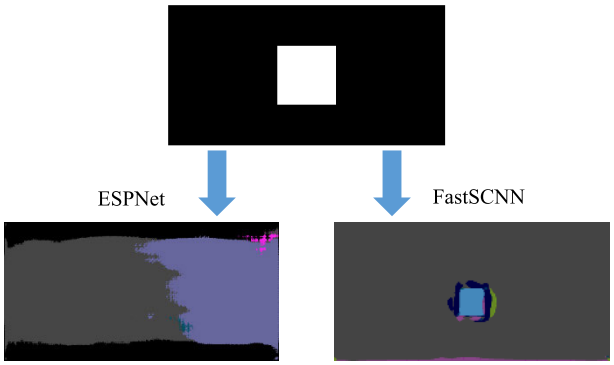

FIGURE 4. This is a comparison about the two outputs of the ESPNet and FastSCNN with the same noise image input. Compared with ESPNet, FastSCNN can still maintain relatively complete local features.

can correctly classify the same original image in the dataset. The prediction of FastSCNN seems to be consistent with the appearance of the pixels around the center point in the noise image, while the prediction of ESPNet seem to be irregular that it is difficult to directly find its relationship to the pixels in noise image.

To unlock the secrets of adversarial learning in white box attacks, a comparison of the gradient of the variable ξ and the perturbations ϵ between ESPNet and FastSCNN during the adversarial learning stage will be discussed. The gradients and perturbations in the training process are visualized with their relative magnitude:

$$M = (M - min(M))/(max(M) - min(M)) \times 255 \quad (9)$$

where M indicates the matrix of grads or perturbations.

Fig. 5 visualizes the gradients and perturbations from attacking the pedestrians and bicycles to vegetation. The gradient images are generated at step 30 of the adversarial learning, while the perturbation images are generated at the last step. The gradients or perturbations at the higher brightness is higher than that at the lower brightness. It is obvious that the perturbations learned from FastSCNN are more concentrated in the specific target region than that from ESPNet. It's hard to discriminate the exact shape of the perturbations generated

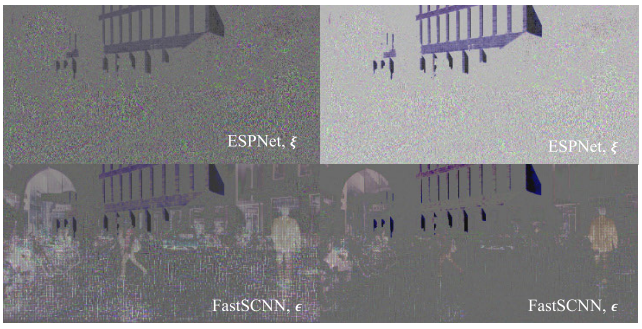

FIGURE 5. This is a comparison about the gradient of the variable ξ and the perturbations ϵ of ESPNet and FastSCNN during the adversarial learning stage. It is obvious that the perturbations learned from FastSCNN are more concentrated in a specific target region than that from ESPNet.

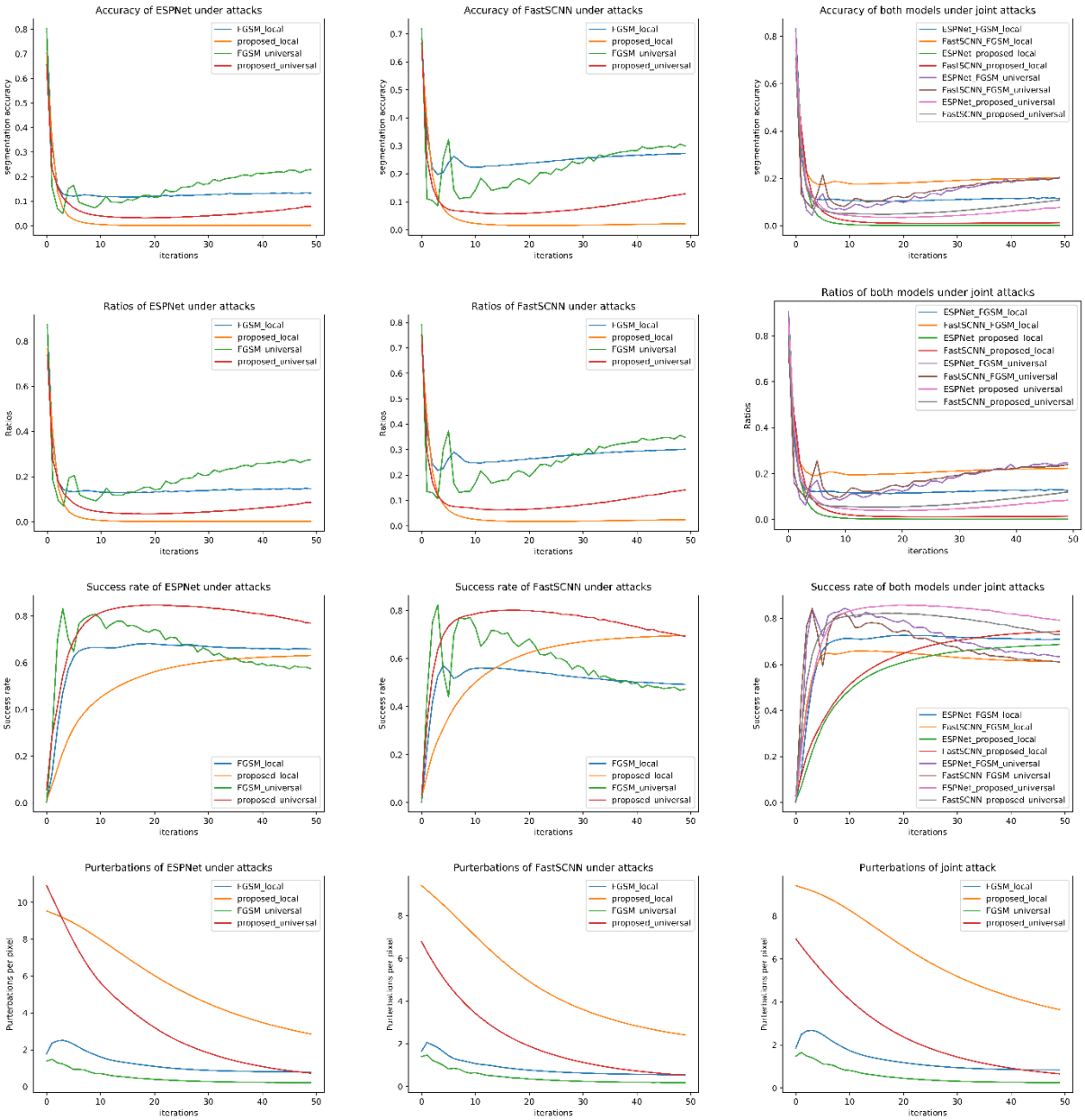

FIGURE 6. A comparison between proposed method and FGSM under local source domain attacks and universal attacks on two models in the 50-step iterative generation of adversarial images. We attacked ESPNet and FastSCNN separately, and then did a joint attack on both models. Segmentation accuracy, ratios of segmentation accuracy, attacking accuracy and perturbations per pixel are used to measure the performance of the two methods. The figures of the first row to the fourth row are respectively the performance on segmentation accuracy, ratios of segmentation accuracy, attacking accuracy and perturbations per pixel. The figures of the first column to the third column respectively indicate the attacks on ESPNet, the attacks one FastSCNN and joint attacks on both models.

for ESPNet, which seems to focus on the overall features of the image.

The main reason for this difference is that the two networks have completely different architectures. Massively dilated convolutions are used in ESPNet and the maximum dilation rate is 16. This leads to a rapid increase in receptive field. In addition to this, the combination of a multi dilation rate cause a gradient on the predicted feature map to be propagated back to regions of different scales on the original image. FastSCNN did not use dilated convolution, and it only fused the features of two resolutions. Compared with ESPNet, it paid more attention to the local features of the image.

In order to attack both models simultaneously, we use joint optimization to narrow the gap between them. For the same original picture and the output of two different models, a unified ground truth map should be formulated. The ground-truth is determined by two predictions at first, as can be seen in Eq. 10:

$$y^{gt} = \underset{c}{\text{argmax}}(\max(f_c^1\left(I_0^{adv};\theta^1\right), f_c^2\left(I_0^{adv};\theta^2\right))) \quad (10)$$

TABLE 1. A comparison between proposed method and FGSM under local source domain attacks and universal attacks on two models of the final adversarial images in test set.

Segmentation accuracy	Local				Universal			
	ESPNet	FastSCNN	ESPNet (Joint)	FastSCNN (Joint)	ESPNet	FastSCNN	ESPNet (Joint)	FastSCNN (Joint)
FGSM	0.133	0.274	0.115	0.202	0.229	0.3	0.204	0.202
proposed	0.002	0.022	0.001	0.012	0.008	0.129	0.078	0.109
Ratios of accuracy	Local				Universal			
	ESPNet	FastSCNN	ESPNet (Joint)	FastSCNN (Joint)	ESPNet	FastSCNN	ESPNet (Joint)	FastSCNN (Joint)
FGSM	0.146	0.302	0.126	0.223	0.275	0.348	0.247	0.238
proposed	0.002	0.025	0.001	0.014	0.086	0.142	0.084	0.119
Success rate	Local				Universal			
	ESPNet	FastSCNN	ESPNet (Joint)	FastSCNN (Joint)	ESPNet	FastSCNN	ESPNet (Joint)	FastSCNN (Joint)
FGSM	0.659	0.493	0.71	0.611	0.575	0.473	0.635	0.611
proposed	0.632	0.696	0.687	0.744	0.77	0.693	0.792	0.73
Purterbations per pixel	Local				Universal			
	ESPNet	FastSCNN	Joint		ESPNet	FastSCNN	Joint	
FGSM	0.792	0.543	0.844		0.207	0.171	0.244	
proposed	1.635	1.097	2.216		0.724	0.518	0.657	

The label of the adversarial sample is still modified by ground-truth label. The pixels are set to the classes whose predicted value is the maximum of both predictions. The adversarial labels y^{adv} are still changed from y^{gt} as Eq. 8. The cross entropy in Eq. 4 consists of two predictions corresponding to the two models, as shown in Eq. 11:

$$\begin{aligned} C &= C\left(f^1\left(I^{adv};\theta\right), y^{adv}, \omega\right) + C\left(f^2\left(I^{adv};\theta\right), y^{adv}, \omega\right) \\ \omega_i &= (p_i^{tar} - \min_c M_i^c)/(\max_c M_i^c - \min_c M_i^c) \\ M_i^c &= \max\left(p_i^{1,c}, p_i^{2,c}\right) \end{aligned} \tag{11}$$

where $p_i^{1,c}$ and $p_i^{2,c}$ respectively indicate the two predictions of FastSCNN and ESPNet at the i'th pixel and the c'th channel.

In the previous section, the adversarial attacking methods based on the local source domain and universal perturbations were introduced. Both of these methods can be combined with the joint learning method proposed in this section to carry out a joint local attack and joint universal attack.

V. EXPERIMENTS

We evaluated our proposed attacking method on the validation set of the Cityscapes dataset [34], and report its performance in this section. Cityscapes dataset is comprised of a large, diverse set of stereo video sequences recorded in streets from 50 different cities. The main subset of it consists of a training set with 2990 images, a validation set with 500 images and a test set with 1525 images. We mainly attacked 500 urban images in the validation set. White box attacks are used by default, so that there is no access to the model weights and ground truth labels.

Actually, three attacks were implemented, respectively: misclassifying pedestrians, riders, motorcycles and bicycles as the vegetation, misclassifying cars, truck, buses and trains as the road, as well as misclassifying the cars, trucks, buses and trains to the building. We did the attacks using the proposed method compared with the FGSM method [14] on the FastSCNN [40], ESPNet [39] and a combination of them. The performance of the two methods is evaluated by the attacking accuracy and the perturbations per pixel. The attacking accuracy is defined as the ratio of the number of pixels in the target source domain misclassified as the adversarial class to the number of the total number of pixels in the target source domain. The perturbations per pixel is defined as the average value of the absolute value of the pixel difference between the adversarial image and the original image at all pixels in the image. Towards the attacks on the local source domain, the average is calculated only for the pixels in the source domain.

During the adversarial learning, each image was input with the size 512×1024. When doing the FGSM attack, the pixels are normalized to [0,1] and then be preprocessed to the formats of the model input. When attacking with our nonlinear method, the image pixels are all first normalized to $[-1,1]$ to fit the range of the tanh function. Our experimental simulation platform is a desktop with a NVIDIA GTX-2080 GPU (2944 CUDA cores, 8 GB Total Memory). Adam optimizer was used to minimize the cost function mentioned in Sec. III. Each image was updated with 50 steps. And the learning rate to each intermediate variable is 0.01.

Fig. 6 shows a comparison between the proposed method and FGSM under local source domain attacks and universal attacks on two models in the iterative generation of adversarial images. We attacked ESPNet and FastSCNN

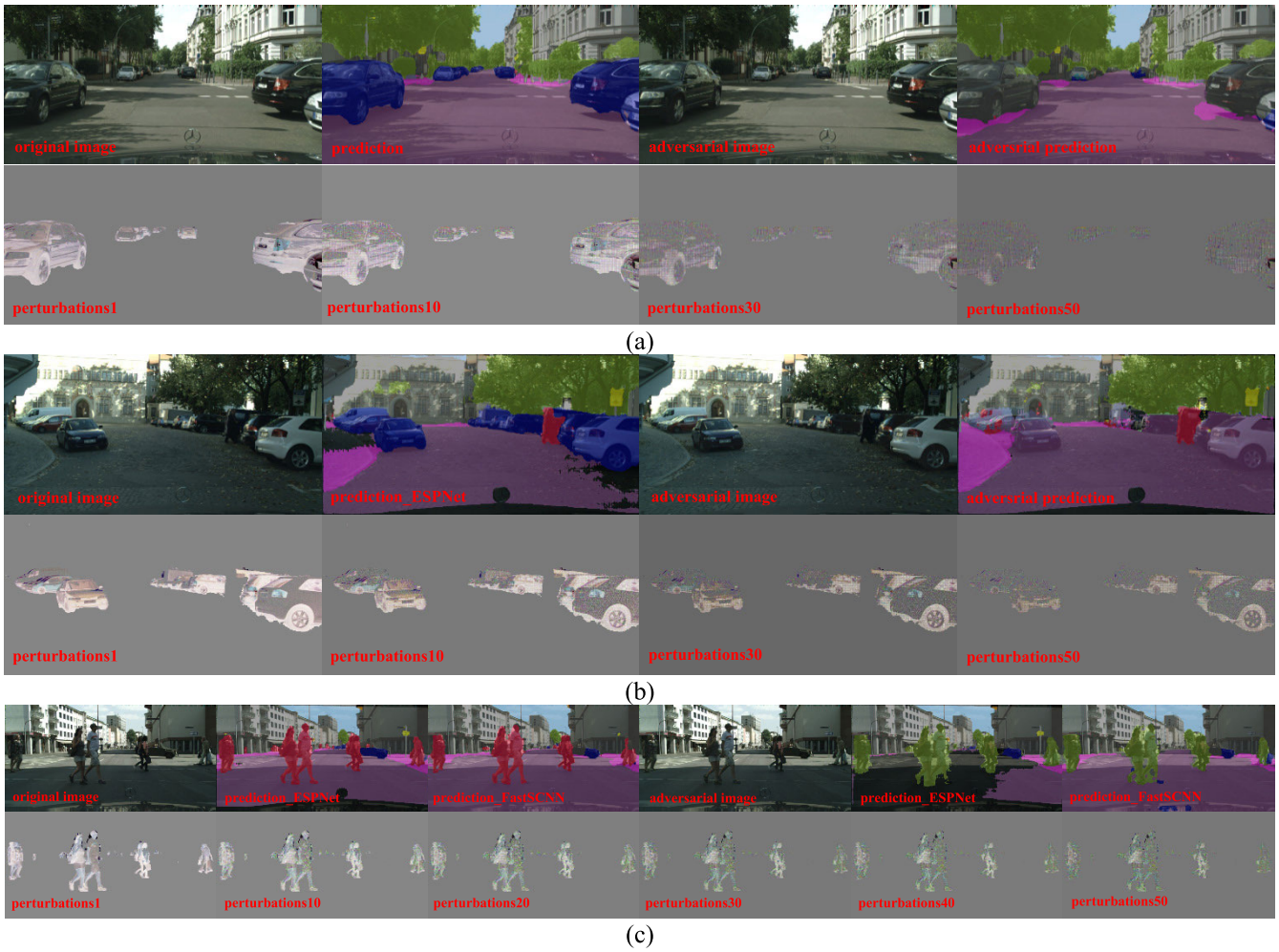

FIGURE 7. Adversarial attacks on local source domain: (a) FastSCNN: Attacks of car, truck, bus and train to road; (b) ESPNet: Attacks of car, truck, bus and train to building; (c) Both models: Attacks of person, rider, motorcycle and bicycle to vegetation.

separately, and then did a joint attack on both models. Segmentation accuracy, ratios of segmentation accuracy, attacking accuracy and perturbations per pixel are used to measure the performance of the two methods. In each case we carried out three attacks: attacks of person, rider, motorcycle and bicycle to vegetation, attacks of car, truck, bus and train to road, and attacks of car, truck, bus and train to the building. Segmentation accuracy refers to the accuracy of the target class under the ground truth labels. Its value is the ratio of the number of correct classification in the original class pixels to the pixel value of the original class pixels in the ground truth. For an adversarial attack task, the lower segmentation accuracy reflects the superior performance of the adversary. It is not enough to evaluate the adversarial performance only by the segmentation accuracy, because some adversaries perform very well at the target pixels, but at the same time will affect the classification accuracy at the non-target pixels, resulting in the overall decrease in the global segmentation accuracy. In order to make the evaluation system more perfect, we introduced ratios of accuracy which indicate the segmentation accuracy of the target pixels to the segmentation accuracy of the non-target pixels. If an adversary reduces the classification accuracy of the adversarial image at the target pixel, but keeps the classification accuracy at the non-target pixel higher, the adversary is efficient. In addition, the success rate is also an important criterion for the performance of the adversary. It is the ratio of the number of pixels successfully attacked as the target class to the total number of pixels in the attacked area. The perturbations per pixel indicates the average of the absolute value of the perturbation attached to each pixel in the target area and the values range from 0 to 255.

The values of all the curves in Fig. 6 are obtained by averaging over all the test images within a 50-step iteration of the attack. The values in Tab. 1 show the final results after iterations corresponding to the Fig.8. From the perspective of segmentation accuracy, whether attacking the ESPNet, FastSCNN or the joint model, the proposed method reduces the segmentation accuracy more than the FGSM method. which is the same in both local and universal attacking modes.

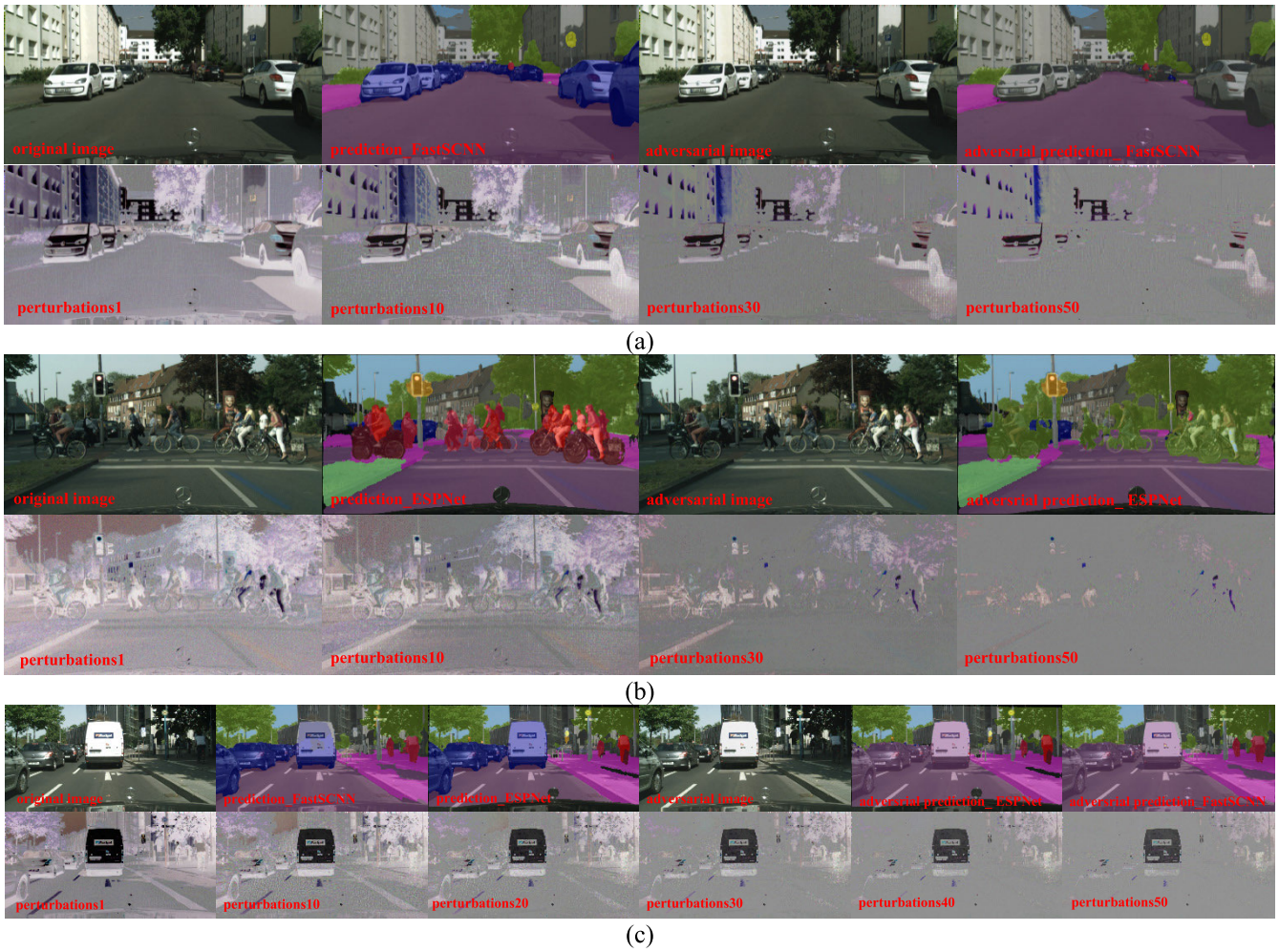

FIGURE 8. Adversarial attacks with universal perturbations: (a) FastSCNN: Attacks of car, truck, bus and train to building; (b) ESPNet: Attacks of person, rider, motorcycle and bicycle to vegetation; (c) Both models: Attacks of car, truck, bus and train to road.

The conclusion is true in both local attacks and universal attack cases. Joint attacks have a greater impact on model accuracy than separate attacks. From the view of the ratios of accuracy, the curves are almost identical to those of segmentation accuracy, even for the values. This indicates that neither the FGSM nor the proposed method has much effects on the accuracy of the non-target area. From the pixel success rate, except that the proposed method ended up about two percentage points lower than FGSM in the case of the local attacks on the ESPNet, the proposed method is superior than FGSM in most cases (different models, different attack modes). For the comparison of the perturbations, the final perturbations obtained by both the proposed method and FGSM converge to very small values in universal attacks. In the case of a local attack, the final perturbation value of the proposed method is 1.4 pixels higher than that of FGSM. It can be seen from the trend of the curve that there should be room for the perturbation value to decline after subsequent learning. One or two pixels are almost imperceptible to the human eyes. It can be concluded from the overall analysis that the proposed adversary is more likely to affect the performance of the segmentation model, compared with the FGSM method.

Fig. 7 and Fig. 8 respectively show the adversarial images and generation process of perturbed image by the proposed adversary. The perturbed image is normalized to between 0 and 255, so the very dark (negative perturbation) and the very bright (positive perturbation) pixels in the image are where the perturbation is large.

VI. CONCLUSION

Recent studies found that deep neural networks are easily attacked by some adversarial samples. Image segmentation is the most basic part of computer vision and the basis of all other image processing methods. The attack on the image segmentation task causes us to think about the potential security problems in the deep learning system. This paper introduced the non-linear adversarial samples generation method avoiding miss gradient from truncating the pixel values in the adversarial images. Then, the perturbations for

the local source domain and the universal perturbations on image segmentations a joint learning method are proposed for the multi-model attack. Due to the different network structures, the two models have different sizes of the receptive fields, resulting in different blind spots and weaknesses of the model. Experimental results show that the proposed method can attack FastSCNN model and ESPNet model effectively. The transferable attacks based on multiple models are still the focus of future research.

However, the proposed method has some shortcomings. Because the proposed method needs to calculate the values of each perturbation matrix and then map back to the pixel space, the time-consuming process can lead to a slow or even non-convergence of the adversarial images when the model encounters a complex graphic structure. Further, the proposed approach is limited to white box attacks. During a joint attack, the gradient of each model must be accessible. Therefore, in future research, it is necessary to explore the common structure between the image and the model, which is of great significance not only for the attacks but also for the defense of the models. It is also necessary to apply other latest generation methods such as the DeepFool method, JSMA, Carlini and Wagner method to adversarial image segmentation tasks. It is possible to change the values of just a few pixels to make a big difference in the structure of the predicted map.

REFERENCES

[1] A. Arnab, O. Miksik, and P. H. S. Torr, "On the robustness of semantic segmentation models to adversarial attacks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, Jun. 2018, pp. 888–897.

[2] A. M. Zador, "A critique of pure learning and what artificial neural networks can learn from animal brains," *Nature Commun.*, vol. 10, no. 1, pp. 1–7, 2019.

[3] J. H. Metzen, M. C. Kumar, T. Brox, and V. Fischer, "Universal adversarial perturbations against semantic image segmentation," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Venice, Italy, Oct. 2017, pp. 2774–2783.

[4] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, Apr. 2018.

[5] P. Luc, C. Couprie, S. Chintala, and J. Verbeek, "Semantic segmentation using adversarial networks," 2016, *arXiv:1611.08408*. [Online]. Available: http://arxiv.org/abs/1611.08408

[6] R. Feinman, R. R. Curtin, S. Shintre, and A. B. Gardner, "Detecting adversarial samples from artifacts," 2017, *arXiv:1703.00410*. [Online]. Available: http://arxiv.org/abs/1703.00410

[7] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," 2016, *arXiv:1607.02533*. [Online]. Available: http://arxiv.org/abs/1607.02533

[8] P. Y. Chen, Y. Sharma, H. Zhang, J. Yi, and C. J. Hsieh, "EAD: Elastic-net attacks to deep neural networks via adversarial examples," in *Proc. 22nd AAAI Conf. Artif. Intell.*, 2018, pp. 10–17.

[9] C. Xiao, R. Deng, B. Li, F. Yu, M. Liu, and D. Song, "Characterizing adversarial examples based on spatial consistency information for semantic segmentation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 217–234.

[10] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *Proc. IEEE Symp. Secur. Privacy (SP)*, San Jose, CA, USA, May 2017, pp. 39–57.

[11] K. Chalupka, P. Perona, and F. Eberhardt, "Visual causal feature learning," 2014, *arXiv:1412.2309*. [Online]. Available: http://arxiv.org/abs/1412.2309

[12] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," 2013, *arXiv:1312.6199*. [Online]. Available: https://arxiv.org/abs/1312.6199

[13] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," 2014, *arXiv:1412.6572*. [Online]. Available: http://arxiv.org/abs/1412.6572

[14] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial machine learning at scale," 2016, *arXiv:1611.01236*. [Online]. Available: http://arxiv.org/abs/1611.01236

[15] O. Bastani, Y. Ioannou, L. Lampropoulos, D. Vytiniotis, A. Nori, and A. Criminisi, "Measuring neural net robustness with constraints," in *Proc. 30th Annu. Conf. Neural Inf. Process. Syst. (NIPS)*, 2016.

[16] M. Dezfooli, S. Mohsen, A. Fawzi, and P. Frossard, "DeepFool: A simple and accurate method to fool deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2574–2582.

[17] S. Baluja and I. Fischer, "Adversarial transformation networks: Learning to generate adversarial examples," 2017, *arXiv:1703.09387*. [Online]. Available: http://arxiv.org/abs/1703.09387

[18] Y. Liu, X. Chen, C. Liu, and D. Song, "Delving into transferable adversarial examples and black-box attacks," 2016, *arXiv:1611.02770*. [Online]. Available: http://arxiv.org/abs/1611.02770

[19] S. M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard, "Universal adversarial perturbations," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1765–1773.

[20] G. F. Elsayed, I. Goodfellow, and J. Sohl-Dickstein, "Adversarial reprogramming of neural networks," 2018, *arXiv:1806.11146*. [Online]. Available: http://arxiv.org/abs/1806.11146

[21] S. Huang, N. Papernot, I. Goodfellow, Y. Duan, and P. Abbeel, "Adversarial attacks on neural network policies," 2017, *arXiv:1702.02284*. [Online]. Available: http://arxiv.org/abs/1702.02284

[22] A. Ghorbani, A. Abid, and J. Zou, "Interpretation of neural networks is fragile," 2017, *arXiv:1710.10547*. [Online]. Available: http://arxiv.org/abs/1710.10547

[23] J. Kos, I. Fischer, and D. Song, "Adversarial examples for generative models," in *Proc. IEEE Secur. Privacy Workshops (SPW)*, San Francisco, CA, USA, May 2018, pp. 36–42.

[24] G. Zhang, C. Yan, X. Ji, T. Zhang, T. Zhang, and W. Xu, "DolphinAtack: Inaudible voice commands," 2017, *arXiv:1708.09537*. [Online]. Available: https://arxiv.org/abs/1708.09537

[25] 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 models," 2017, *arXiv:1707.08945*. [Online]. Available: http://arxiv.org/abs/1707.08945

[26] Q. Zeng, J. Su, C. Fu, G. Kayas, L. Luo, X. Du, and J. Wu, "A multiversion programming inspired approach to detecting audio adversarial examples," in *Proc. 49th Annu. IEEE/IFIP Int. Conf. Dependable Syst. Netw. (DSN)*, Portland, OR, USA, Jun. 2019, pp. 39–51.

[27] L. Xiao, Y. Li, X. Huang, and X. Du, "Cloud-based malware detection game for mobile devices with offloading," *IEEE Trans. Mobile Comput.*, vol. 16, no. 10, pp. 2742–2750, Oct. 2017.

[28] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 4, pp. 640–651, Feb. 2014.

[29] L. C. Chen, Y. Yang, J. Wang, W. Xu, and A. L. Yuille, "Attention to scale: Scale-aware semantic image segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 3640–3649.

[30] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," 2015, *arXiv:1511.07122*. [Online]. Available: http://arxiv.org/abs/1511.07122

[31] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2481–2495, Dec. 2017.

[32] E. Romera, J. M. Alvarez, L. M. Bergasa, and R. Arroyo, "ERFNet: Efficient residual factorized convnet for real-time semantic segmentation," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 1, pp. 263–272, Jan. 2018.

[33] A. Paszke, A. Chaurasia, S. Kim, and E. Culurciello, "ENet: A deep neural network architecture for real-time semantic segmentation," 2016, *arXiv:1606.02147*. [Online]. Available: http://arxiv.org/abs/1606.02147

[34] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 3213–3223.

[35] J. Guo, B. Song, F. R. Yu, Y. Chi, and C. Yuen, "Fast video frame correlation analysis for vehicular networks by using CVS–CNN," *IEEE Trans. Veh. Technol.*, vol. 68, no. 7, pp. 6286–6292, Jul. 2019.
[36] S. Chen, B. Song, L. Fan, X. Du, and M. Guizani, "Multi-modal data semantic localization with relationship dependencies for efficient signal processing in EH CRNs," *IEEE Trans. Cogn. Commun. Netw.*, vol. 5, no. 2, pp. 347–357, Jun. 2019.
[37] M. Yang, K. Yu, C. Zhang, Z. Li, and K. Yang, "DenseASPP for semantic segmentation in street scenes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 3684–3692.
[38] C. Yu, J. Wang, C. Peng, C. Gao, G. Yu, and N. Sang, "Learning a discriminative feature network for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1857–1866.
[39] S. Mehta, M. Rastegari, A. Caspi, L. Shapiro, and H. Hajishirzi, "ESPNet: Efficient spatial pyramid of dilated convolutions for semantic segmentation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 552–568.
[40] R. P K Poudel, S. Liwicki, and R. Cipolla, "Fast-SCNN: Fast semantic segmentation network," 2019, *arXiv:1902.04502*. [Online]. Available: http://arxiv.org/abs/1902.04502
[41] C. Xie, J. Wang, Z. Zhang, Y. Zhou, L. Xie, and A. Yuille, "Adversarial examples for semantic segmentation and object detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 1378–1387.

XU KANG received the B.S. and M.S. degrees in communication and information systems from Xidian University, Xi'an, China, in 2015 and 2018, respectively, where he is currently pursuing the Ph.D. degree. His research interests include image recognition, multimedia, deep learning, and big data.

BIN SONG (Senior Member, IEEE) received the B.S., M.S., and Ph.D. degrees in communication and information systems from Xidian University, Xi'an, China, in 1996, 1999, and 2002, respectively. He is currently a Professor with Xidian University. He has authored more than 60 journal articles or conference papers and 30 patents. His research interests are in distributed video coding, compressed sensing-based video coding, content-based image recognition, machine learning, deep reinforcement learning, the Internet of Things, and big data.

XIAOJIANG (JAMES) DU (Fellow, IEEE) received the B.S. and M.S. degrees in electrical engineering from Tsinghua University, Beijing, China, in 1996 and 1998, respectively, and the M.S. and Ph.D. degrees in electrical engineering from the University of Maryland at College Park, in 2002 and 2003, respectively. He is currently a tenured Professor with the Department of Computer and Information Sciences, Temple University, Philadelphia, USA. He has authored more than 300 journal and conference papers in these areas and a book published by Springer. He has been awarded more than 6 million U.S. dollars research grants from the US National Science Foundation (NSF), Army Research Office, Air Force Research Laboratory, NASA, Qatar, the State of Pennsylvania, and Amazon. His research interests are security, wireless networks, and systems. He is a Life Member of ACM. He received the Best Paper Award at IEEE GLOBECOM 2014 and the Best Poster Runner-Up Award at ACM MobiHoc 2014. He serves on the editorial boards of three international journals.

MOHSEN GUIZANI (Fellow, IEEE) received the B.S. (Hons.) and M.S. degrees in electrical engineering and the M.S. and Ph.D. degrees in computer engineering from Syracuse University, Syracuse, NY, USA, in 1984, 1986, 1987, and 1990, respectively. He served in different academic and administrative positions at the University of Idaho, Western Michigan University, the University of West Florida, the University of Missouri–Kansas City, the University of Colorado at Boulder, and Syracuse University. He is currently a Professor with the CSE Department, Qatar University, Qatar. His research interests include wireless communications and mobile computing, computer networks, mobile cloud computing, security, and smart grid. He has authored 9 books and more than 500 publications in refereed journals and conferences. He is a Senior Member of ACM. He also served as a member, chair, and general chair of a number of international conferences. Throughout his career, he received three teaching awards and four research awards. He also received the 2017 IEEE Communications Society WTC Recognition Award and the 2018 AdHoc Technical Committee Recognition Award for his contribution to outstanding research in wireless communications and Ad-Hoc Sensor networks. He was the Chair of the IEEE Communications Society Wireless Technical Committee and the Chair of the TAOS Technical Committee. He served as the IEEE Computer Society Distinguished Speaker. He is currently the IEEE ComSoc Distinguished Lecturer. He guest edited a number of special issues in IEEE journals and magazines. He is currently the Editor-in-Chief of the *IEEE Network Magazine*, serves on the editorial boards of several international technical journals, and the Founder and Editor-in-Chief of *Wireless Communications and Mobile Computing* (Wiley).