International Journal of Computational Intelligence Systems
Vol. **14(1)**, 2021, *pp.* 1823–1830
DOI: https://doi.org/10.2991/ijcis.d.210620.002; ISSN: 1875-6891; eISSN: 1875-6883
https://www.atlantis-press.com/journals/ijcis/

Research Article

Fast Category-Hidden Adversarial Attack Against Semantic Image Segmentation

Yinghui Zhu, Yuzhen Jiang*, Zhongxing Peng, Wei Huang

School of Computer and Information Engineering, Hanshan Normal University, Qiaodong Street, Xiangqiao District Chaozhou, 521041, P. R. China

ARTICLE INFO

Article History

Received 23 Feb 2021
Accepted 09 Jun 2021

Keywords

Adversarial example
Deep neural networks (DNNs)
Semantic segmentation
Category-hidden adversarial attack (CHAA)
Logits map

ABSTRACT

In semantic segmentation, category-hidden attack is a malicious adversarial attack which manipulates a specific category without affecting the recognition of other objects. A popular method is the nearest-neighbor algorithm, which modifies the segmentation map by replacing a target category with other categories close to it. Nearest-neighbor method aims to restrict the strength of perturbation noise that is imperceptive to both human eyes and segmentation algorithms. However, its spatial search adds lots of computational burden. In this paper, we propose two fast methods, dot-based method and line-based method, which are able to quickly complete the category transfers in logits maps without spatial search. The advantages of our two methods result from generating the logits maps by modifying the probability distribution of the category channels. Both of our methods are global, and the location and size of objects to hide are not cared, so their processing speed is very fast. The dot-based algorithm takes the pixel as the unit of calculation, and the line-based algorithm combines the category distribution characteristics of the horizontal direction to calculate. Experiments verify the effectiveness and efficiency compared with nearest-neighbor method. Specifically, in the segmentation map modification step, our methods are 5 times and 65 times faster than nearest-neighbor, respectively. In the small perturbation attack experiment, dot-based method gets the fastest speed, while different datasets and different setting experiments indicate that the line-based method is able to achieve faster and better adversarial segmentation results in most cases.

1. INTRODUCTION

The applications of deep neural networks (DNNs) have significantly reshaped the landscape of computer vision. However, despite of its potential to overtake human beings in many different tasks, DNNs were founded vulnerable to adversarial attacks [1–3]. In those attacks, tiny perturbations imperceptible to human eyes can easily fool algorithms to output wrong predictions. Adversarial examples were mainly crafted to cheat the image classifiers [1,4–7] at the beginning, then were used to attack other computer vision tasks such as object detection and semantic segmentation [8–11]. There are many methods to attack semantic segmentation due to different recognition tasks and different attack intentions [9]. Similar to the attacks on image classifiers, an important and interesting problem is how to construct some imperceptible perturbations on a benign image which can be hardly detected by DNNs models. Worth to mention that, the existing attacking algorithms usually come with the cost of either low speed or much obvious noise or both of them [9,12,13]. Therefore it is still a challenge to propose an efficient way to attack semantic segmentation with unnoticeable noise. In this paper, we propose two fast category-hidden attacks against semantic image segmentation which are imperceptible to the human eye but able to fool the segmentation networks. The motivation of our research is to explore more efficient and deceptive category-hidden algorithms of adversarial attack, and to reveal the potential security issues of segmentation application system.

There are two types of semantic segmentation attack algorithms: nontargeted and targeted. Nontargeted attacks simply mislead the model to result in meaningless maps that are either unrecognizable or useless. Figure 1 shows the output maps of two kinds of nontargeted attacks: messy segmentation map [12] and strip segmentation map [14]. Obviously, neither the messy map nor strip map looks related to the input images. Different from the meaningless maps generated by nontargeted attacks, the output maps in a targeted attack look meaningful and useful even though they have covert destruction intent. Figure 2 shows three patterns of targeted attacks. In the first one shown in the first row, the black double lines are carefully drawn on the original image to fool the segmentation algorithm to predict a wrong road direction [13]. The second row shows the second pattern example, which achieves an image-independent map almost unchanged for any input image [15]. In the third row, the third pattern example achieves an intentional segmentation map where "pedestrians" are unrecognizable [15]. All these targeted attack algorithms are able to mislead the segmentation algorithms at some level, however, perceptive additives or almost unchanged output map is unable to cheat people eyes.

*Corresponding author. E-mail: jyz366@163.com

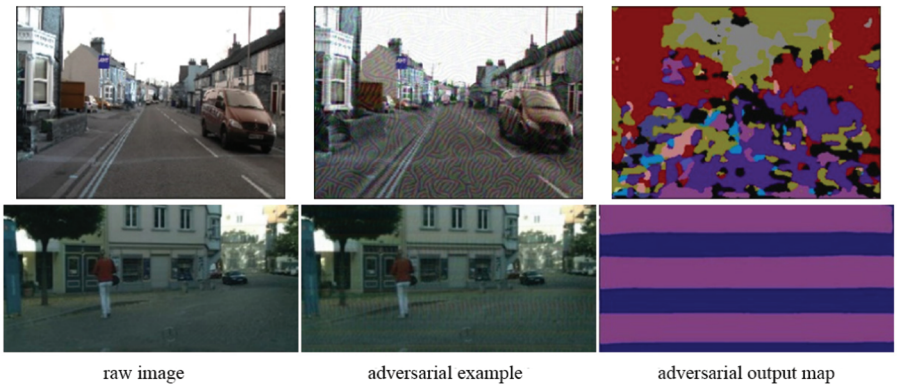

Figure 1 | Nontargeted adversarial attack patterns against semantic segmentation.

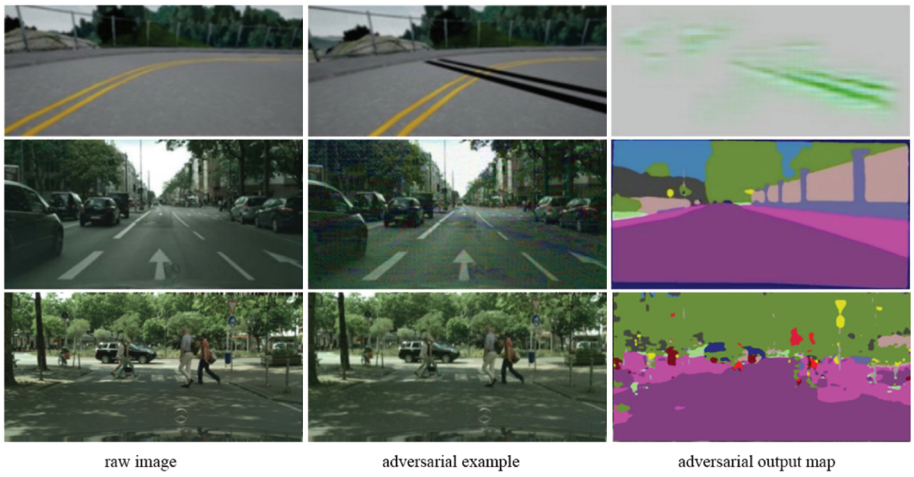

Figure 2 | Targeted adversarial attack patterns against semantic segmentation.

Meanwhile, the universal perturbation in the third attack pattern can't hide all designated information completely (not all red areas can be removed).

Beyond segmentation map, perturbation noise is another important criterion of adversarial example. Many adversarial examples show too much noise (e.g., the first row example in Figure 1 and the second row example in Figure 2). Visible noise can easily be noticed by people or measured by the detection system which will lead to a fail attack. In this paper, we study the category-hidden adversarial examples and propose two novel methods which are able to launch fast and imperceptible attacks with perfect hiding effect.

The contributions of this paper include the following:

- We study fast and deceptive category-hidden adversarial attack (CHAA) algorithms, then we propose two new and effective algorithms based on targeted-attacking. We verify the proposed algorithms on both Pascal VOC2012 and Cityscapes datasets, and reveal the potential security risks of semantic segmentation system.
- Different from the conventional spatial processing method, we use the category channels to obtain the logits maps, which are proven to be better than the neighborhood methods in various ways, such as with high speed, less noise adversarial examples, and higher visual deception of prediction output map.
- Through different experiments, we show that, in addition to algorithm selection, the perturbation coefficient and the size of attack area are also important factors affecting the efficiency and effectiveness of the category-hidden algorithm.

In the following sections, we first discuss a vanilla category-hidden adversarial algorithm called NN, whose segmentation reference map is generated based on the spatial domain method, that is, all the pixels of designated category are replaced with the nearest other category. Next, we propose two new algorithms with more efficiency and effectiveness: dot-based and line-based methods. In our new methods, the logits maps are obtained by modifying the category channels individually. The dot-based algorithm is processed by pixel by pixel, and the line-based algorithm is processed by line by line. Then, Section 3 shows the experimental results carried on two different datasets: Pascal VOC2012 and Cityscapes. In both datasets, our proposed methods are proved to be more efficient. It is worth noting that the attacking effect of line-based algorithm shows especially natural and aggressive.

2. IMPERCEPTIBLE CHAA

CHAA aims to prevent segmentation algorithms from detecting a special category without affecting the detection of other categories. Thus, it becomes more difficult for segmentation algorithms to discover the attack. For example, the algorithms never see "pedestrian," but it can still see all other objects [15,16]. Except the "pedestrian," all other segmentations look unchanged and correct, so the whole segmentation looks correct and reasonable. Furthermore, to make the attack more successful, the perturbation noise should be as few and invisible as possible.

2.1. Vanilla Method

The vanilla category-hidden adversarial method is the nearest-neighbor (NN) algorithm mentioned in [15,16]. Although many category-hidden algorithms are proposed, most of them only concern about eliminating the attacked category, but ignore the quality of adversarial examples and prediction maps. For example, [17] replaces the attacked category with any other categories. Without considering the accuracy of the segmentation area of the prediction map, its results are weak [18] only deals with the two class segmentation, which can convert the target category to background category by a small patch of explicit noise. However, the unnatural and obvious noise in this method can easily expose its attack signs. In [15], a proposed method aims to hiding a certain category by using a universal perturbation, but it can't hide all the target objects. In contrast, NN algorithm combines neighboring categories to implement hiding, and no any new category will appear in its prediction map, so it is more effective than the above algorithms.

NN has the following steps: (i) get the normal segmentation map; (ii) find out all areas of the target class; (iii) for every dot of the intentional category, find the nearest other category to replace it; (iv) use this modified segmentation map to generate the adversarial example. Figure 3 shows the workflow of NN in CHAA. Specifically, a raw image is predicted by the semantic segmentation network with a multi-channel logits map, which indicates the probability of various categories at every pixel. The logits map is then merged into a single-channel segmentation map using the argmax function. Then the target category is replaced with the NN category pixel-wise. As a result, a reference category-hidden map (modified map) is got. To apply the gradient to the adversarial perturbation, the adversarial map must be split into multiple channels by one-hot process. Finally, the adversarial example is generated by the iterative gradient calculation of the loss function.

The most important step of the vanilla method is the NN search. Let P_{target} be the target category position set, P_{other} be the position set of other categories.

$$P_{target} = \{(i,j) \,|y_{i,j} = L_{target}\} \tag{1}$$

$$P_{other} = \{(i,j) \,|y_{i,j} = L_{other}\} \tag{2}$$

where y is the normal final segmentation map, $y = f_\theta(I)$. L_{target} and L_{other} represent the label values of the target category and other categories, respectively.

Let y^* represent the target category-hidden map:

$$\begin{cases} y^*_{i,j} = y_{i,j} \quad s.t.\ (i,j) \in P_{other} \\ y^*_{i,j} = y_{i',j'} \ s.t.\ (i,j) \in P_{target} \,\&\, (i',j') \in P_{other} \end{cases} \tag{3}$$

where i', j' are searched for as follows:

$$i', j' = \arg\min (i' - i)^2 + (j' - j)^2 \tag{4}$$

Obviously, applying NN in CHAA is computational intensive, because of the spatial search of the nearest category, as well as the combination and decomposition of channels to obtain adversarial examples.

To achieve a fast algorithm for CHAA, we propose a novel scheme without locating the target category on the logits map as well as using spatial search. The workflow of the proposed scheme is descripted in Figure 4. We apply two different methods of target category transfer in our scheme: dot-based transfer and line-based transfer.

2.2. Dot-Based Target Category Transfer

Here, we use Z to represent the normal logits map, Z' to represent the adversarial logits map, and C_{target} to represent the target channel. To remove the target category, one potential way is to set all C_{target} in Z' to be 0. However, this way makes it hard to converge and to generate an adversarial example. Thus, in our scheme, rather than setting directly to zero, we distribute the value of C_{target} into other channels.

Firstly, let $Z' = Z$ in the target category channel. Then, the value of C_{target} is added to the value of C_{embed}, which has the maximum value and locates at the same pixel position of C_{target} but on different channels. Finally, we set 0 to $Z'_{i,j,C_{target}}$.

$$Z'_{i,j,C_{embed}} = Z_{i,j,C_{embed}} + Z_{i,j,C_{target}} \tag{5}$$

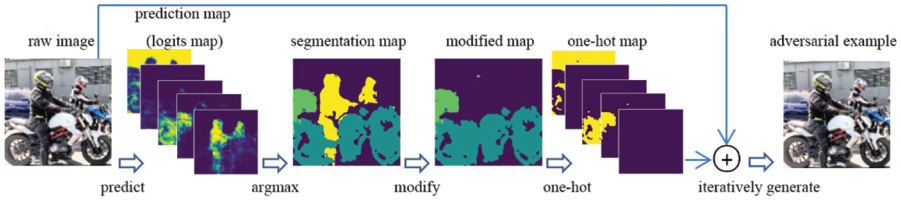

Figure 3 Workflow of NN in CHAA.

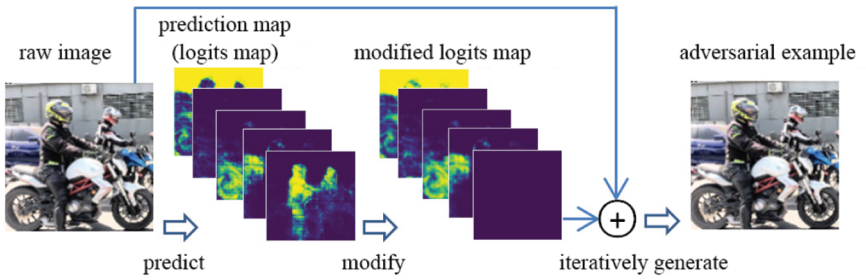

Figure 4 Workflow of the proposed scheme in CHAA.

$$Z'_{i,j,C_{target}} = 0 \tag{6}$$

where C_{embed} is the special channel of the (i,j) pixel to be added to hiding information.

$$C_{embed} = \text{argmax}\, Z'_{i,j,C} \quad s.t. \quad C \neq C_{target} \tag{7}$$

2.3. Line-Based Target Category Transfer

To make our scheme faster, we try to modify our dot-based target category transfer. We observe that in scene images, it is highly possible that the objects often stand vertically, such as mountains, human, and cars. In such images, the scenery has a characteristic of horizontal line similarity, because the category of every pixel is usually similar to the ones at its left and right. Therefore, we propose a novel line-based category transfer to replace the target category by another category, which has the highest probability among other categories in the same line. For instance, to hide a "person," if the highest probability in the same line called "background," the whole line data of "person" in the target channel will be removed and accumulated into the "background" category.

In our line-based target category transfer, we remove the target category line by line. For each line of the target category in its channel on the logits map, its value will be added to the same line in another channel, which has the maximum sum of the line value among all other channels except the target category channel. Define y_k as the number of current line, Z'can be obtained as follows:

$$Z'_{y_k,C_{embed}} = Z_{y_k,C_{embed}} + Z_{y_k,C_{target}} \tag{8}$$

where C_{embed} is the special channel of the y_k-th line to be transfer:

$$C_{embed} = \text{argmax} \sum_{i=0}^{W-1} Z_{i,y_k,C} \quad s.t. \quad C \neq C_{target} \tag{9}$$

W represents the number of pixels in each line of the image.

2.4. Generation of Adversarial Example

We let ξ to be the perturbation and I_{adv} to be the adversarial image. For both dot-based and line-based category transfer, let $\xi^{(0)} = 0$, $I^{(0)}_{adv} = I$, and ξ can be iteratively calculated by the gradient descent method described in [5,15,19].

$$\xi^{(n)} = \nabla_{I^{(n)}_{adv}} J\left(I^{(n)}_{adv}, Z'\right) \Big/ \left|\left|\nabla_{I^{(n)}_{adv}} J\left(I^{(n)}_{adv}, Z'\right)\right|\right|_2 \tag{10}$$

where J is the cost function [5].

To solve the final I_{adv}, the following formula is iteratively executed until no any pixel of the target category in I_{adv} is detected. α is the perturbation coefficient which determines the strength of distrubution.

$$I^{(n+1)}_{adv} = I^{(n)}_{adv} - \alpha \cdot \xi^{(n)} \tag{11}$$

3. EXPERIMENTAL RESULTS

3.1. Experiments on Pascal VOC2012 Dataset

In this section, various experiments show the effectiveness and efficiency of our nonspatial-search schemes. We adopt the famous U-net networks [20] as the attack model, which are trained by a 5-category subset of the Pascal VOC2012 dataset [21]. We use all images with people (including pedestrians, car-drivers, and motorcycle drivers), cars, motorcycles, and bicycles in Pascal VOC2012 dataset. By copying, cutting, horizontally flipping, and slightly rotating the selected images, our 5-category dataset is consisted of 5460 images. Besides the four categories mentioned above, the 5th category is "background," which represents all other objects. All images are resized to 256*256, and 1/4 of which are taken as validation set. After being fully trained, the model achieves 0.971 training accuracy and 0.893 validation accuracy. The "people" category is the target category, which we try to attack. A subset of 100 images with "people" category is selected from the validation dataset to accept attacking tests. All experiments in this paper are run on a workstation with NVIDIA RTX3000.

Firstly, consider the step of segmentation map modification. NN algorithm needs an average of 0.394 seconds to obtain a one-hot map, comparing to our dot-based and line-based algorithms, which only use 0.075 and 0.006 seconds to obtain a modified logits map, respectively. Generally speaking, our methods are able to run 5 times and 65 times faster than NN in such a step.

To generate adversarial examples, we set the perturbation coefficient α to be 1, 0.5, and 0.1, whose results are shown in Tables 1–3, respectively. According to these tables, we notice that a larger α leads to less iteration number but to more noise. Obviously, both of our dot-based and line-based schemes are faster and more effective than NN under different settings of α.

More specifically, in Table 3 with $\alpha = 0.1$, the results of our schemes are much better than NN. The main reason is the abandonment

Table 1 Comparison of different methods with $\alpha = 1$.

	Iterative number	Iterative time	Total time	MSE	L_∞ norm
NN (vanilla)	6.13	**0.543**	0.937	2.715	0.122
Dot-based (proposed)	6.55	0.613	0.688	**2.623**	**0.118**
Line-based (proposed)	**5.89**	0.556	**0.562**	2.703	0.121

The bold values indicate the best performance among the three methods.

Table 2 Comparison of different methods with $\alpha = 0.5$.

	Iterative number	Iterative time	Total time	MSE	L_∞ norm
NN (vanilla)	9.72	1.356	1.750	1.564	0.078
Dot-based (proposed)	**7.24**	1.108	1.183	**0.732**	**0.070**
Line-based (proposed)	7.61	**1.015**	**1.021**	0.778	0.096

The bold values indicate the best performance among the three methods.

Table 3 Comparison of different methods with $\alpha = 0.1$.

	Iterative number	Iterative time	Total time	MSE	L_∞ norm
NN (vanilla)	32.28	4.713	5.107	1.035	0.065
Dot-based (proposed)	**13.81**	**1.880**	**1.955**	**0.336**	0.043
Line-based (proposed)	14.24	2.112	2.118	0.364	**0.039**

The bold values indicate the best performance among the three methods.

of the one-hot map, which is a special reference where every pixel explicitly belongs to a specific category. Thus the value of every point at one-hot map is either 0 or 1, which is different from a normal prediction map (logits map). When we calculate the gradient and generate the perturbation, the acquired perturbation signal will fine-tune the pixels in both the target category region and the other region. Considering α=1, all three algorithm will converge with less iterations and the fine-tuning of other positions does not affect the detection results. When α is small, algorithms need more iterations to converge, while the increase of iterations makes some other category pixels be transferred into target category due to perturbation noise, which will further lead to more iterations. Under the same α, the more iterations, the more perturbation noise accumulated into the adversarial example, as well as more MSE and L_∞ norm.

In our methods, only the target category and the replaced category are changed. Therefore, perturbation noise only affects the target category and its replaced category, with no effects to other irrelevant categories. Even if α is smaller, the number of iterations will not increase sharply.

Tables 1–3 prove the advantages of our methods in both efficiency and effectiveness according to different numerical criterions. What about the visual effectiveness of our methods? In follows, we will provide such comparisons in many different images. Figure 5 illustrates the attacking results of an image with $\alpha = 1$. The first row shows the original image and its segmentation map of normal prediction. The other three rows show the individual perturbation map, adversarial example, and adversarial segmentation map of three attack algorithms, respectively. Among three algorithms, the line-based method shows the most promising result. After removing "people," the perturbation maps look similar and the adversarial examples are all imperceptible to human. However the visual qualities of the predicting segmentation maps (output maps) are different. Due to its line-based treatment approach, line-based method makes the segmentation result more connected than fragmented which leads to a more natural attack.

Figure 6 depicts two other instances, with $\alpha = 0.5$ and $\alpha = 0.1$, respectively. Although the adversarial output maps are similar, we notice that line-based method achieves the best results as well. Obviously, NN always leads to worst segmentation maps, because the error area of original segmentation (seen in red cycle) is enlarged by copying the neighborhood categories.

3.2. Experiments on Cityscapes Dataset

In this section, we use a different dataset Cityscapes [22] to verify the efficiency and effectiveness of our proposed methods. In Cityscapes dataset, the GT images for training are merged into five categories: people (including animals indeed), vehicles, lanes, buildings (including trees and facilities), and sky. In the preprocessing stage, we carry out image cut and image resize. Then we obtain a training set consists with 5948 images in a resolution of 256*256. After training, the model achieves 0.967 training accuracy and 0.881 verification accuracy.

We apply NN algorithm, the proposed dot-based and line-based algorithms to Cityscapes dataset. In addition to the person-hidden algorithm, this experiment increases the test of hiding vehicle and hiding building. Figures 7 and 8 show the experimental results.

From the output segmentation maps, When the hiding area is small (e.g., "person" area), all three algorithms perform well. But with large hiding areas, the NN algorithm is easy to cause abnormal expansion of some small category area. For example, when hiding "building," the "person" areas are magnified many times in NN methods (refer to the red boxes in Figures 7 and 8). In contrast, our two algorithms implement category hiding on category channels, which will not cause the unreasonable deformation of

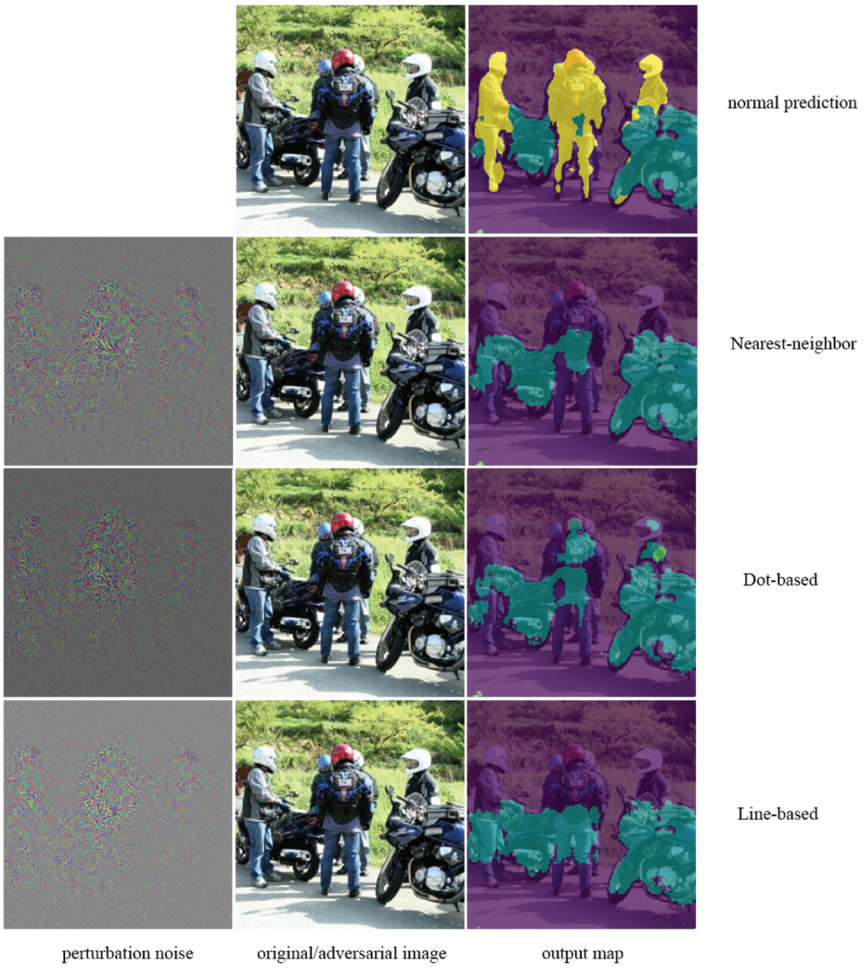

Figure 5 Category-hidden adversarial attacks on an image.

Figure 6 Comparisons of segmentation results.

small category area. However, the dot-based shows less coherence than the other two algorithms, and it is easy to appear holes or broken edges when hiding large category area. In comparison, the line-based algorithm considers the original category distribution in the horizontal direction, and its output map is the most natural and deceptive.

Table 4 shows the average total attack time with different hiding categories and Table 5 shows the average MSE of perturbation noise in adversarial samples. The perturbation coefficient α of all algorithms is 1.

4. CONCLUSIONS

In this paper, we propose two category-hidden methods by directly modifying the features in logits map. Without using spatial search, our two methods deal with dots or lines as a whole to significantly

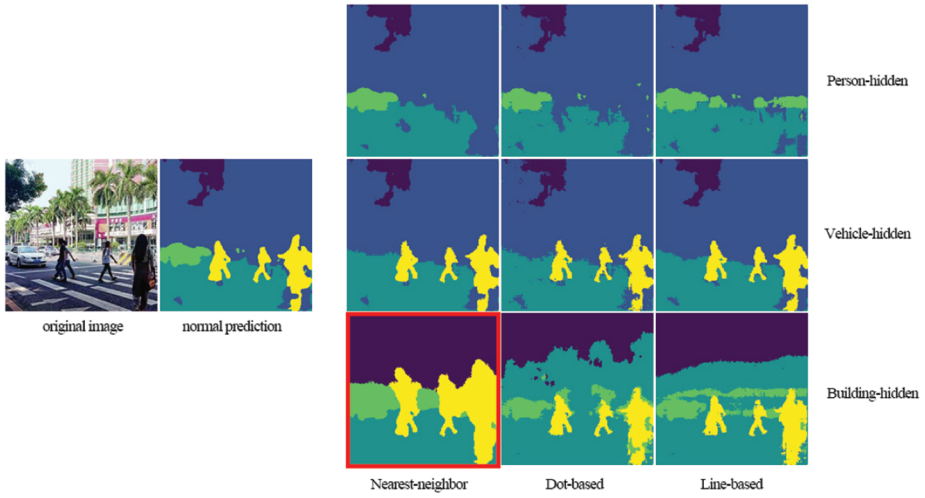

Figure 7 | Instance 1 with different category-hidden results.

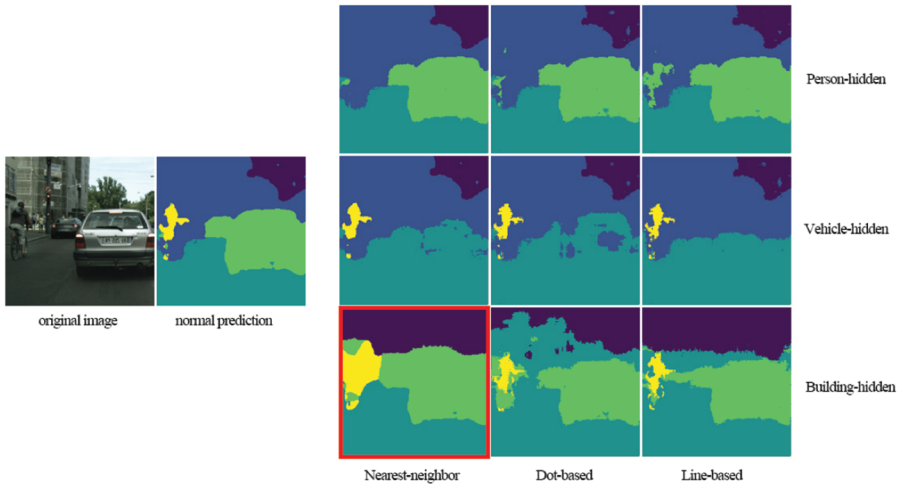

Figure 8 | Instance 2 with different category-hidden results.

Table 4 | Total attack time of different category-hidden.

	Person-hidden	Vehicle-hidden	Building-hidden
NN	0.683	4.716	12.284
Dot-based	0.704	**2.436**	6.151
Line-based	**0.497**	2.695	**5.229**

The bold values indicate the best performance among the three methods.

Table 5 | Perturbation of different category-hidden.

	Person-hidden	Vehicle-hidden	Building-hidden
NN	**1.912**	6.361	26.743
Dot-based	2.230	7.014	25.112
Line-based	2.283	**6.282**	**23.394**

The bold values indicate the best performance among the three methods.

accelerate the attacking speed. Experiments show that, with the same perturbation coefficient, our two methods are always better in effectiveness and efficiency than NN algorithm. Respectively, our dot-based method gets the fastest attacking speed in the cases of small perturbation coefficient. Our line-based method achieves the best results according to the adversarial segmentation maps, in different datasets and different experimental settings.

To the semantic segmentation system, category-hidden attack is one of the adversarial attacks which aim to challenging and improving existing deep learning algorithms. There are many other attacking forms, such as category-added (e.g., "pedestrian" appearing on the unmanned road), and category region tampering (e.g., straight lane being tampered to bend), and so on. Our future work may focus on these two aspects: (i) apply our category-channel-modified strategy to other adversarial attack algorithms. (ii) On the basis of this study, trying to expand the defense training to enhance the defense performance of the segmentation systems, so as to improve the robustness and security of their application.

DATA AVAILABILITY STATEMENT

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

CONFLICTS OF INTEREST

The authors declare that they have no competing interests.

AUTHORS' CONTRIBUTIONS

Yinghui Zhu: Conceptualization, validation, resources, writing—original draft preparation, supervision, project administration. **Yuzhen Jiang:** Conceptualization, software, validation, investigation, writing—original draft preparation, writing—review and editing. **Zhongxing Peng:** Validation, formal analysis, data curation, writing—review and editing. **Wei Huang:** Resources, validation, Visualization.

ACKNOWLEDGMENTS

The work was supported by the Teaching Quality and Teaching Reform Project of Guangdong University, China (191171-DXSSJJXJD-32); Research Project of Hanshan Normal University, China (XS201908, XN202034); Philosophy and Social Science "13th Five-Year plan" Project of Chaozhou, China (2019-A-05,2020-C-17).

REFERENCES

[1] C. Szegedy, W. Zaremba, I. Bruna, *et al.*, Intriguing properties of neural networks, in Proceedings of the International Conference on Learning Representations (ICLR), Banff, Canada, 2014, pp. 1–10.

[2] N. Carlini, D. Wagner, Towards evaluating the robustness of neural networks, in Proceedings of the 2017 IEEE Symposium on Security and Privacy (SP), San Jose, CA, USA, 2017, pp. 39–57.

[3] N. Papernot, P. McDanie, S. Jha, *et al.*, The limitations of deep learning in adversarial settings, in Proceedings of the 2016 IEEE European Symposium on Security and Privacy (EuroS&P), Saarbrucken, Germany, 2016, pp. 372–387.

[4] I.J. Goodfellow, J. Shlens, C. Szegedy, Explaining and harnessing adversarial examples, in Proceedings of the International Conference on Learning Representations (ICML), Lille, France, 2015.

[5] A. Kurakin, I.J. Goodfellow, S. Bengio, Adversarial machine learning at scale, in Proceedings of the International Conference on Learning Representations (ICLR), Toulon, France, 2017.

[6] J. Su, D.V. Vargas, S. Kouichi, One pixel attack for fooling deep neural networks, IEEE Trans. Evol. Comput. 23 (2019), 828–841.

[7] S.M. Moosavi-Dezfooli, A. Fawzi, 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 (CVPR), Las Vegas, NV, USA, 2016, pp. 2574–2582.

[8] A. Arnab, O. Miksik, P.H. Torr, On the robustness of semantic segmentation models to adversarial attacks, in Proceedings of International Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 2018, pp. 888–897.

[9] C. Xiao, R. Deng, B. Li, *et al.*, Characterizing adversarial examples based on spatial consistency information for semantic segmentation, in Proceedings of European Conference on Computer Vision (ECCV), Munich, Germany, 2018.

[10] G. Shen, C. Mao, J. Yang, *et al.*, AdvSPADE: realistic unrestricted attacks for semantic, 2019. https://arxiv.org/abs/1910.02354.

[11] L. Chen, W. Xu, Attacking Optical Character Recognition (OCR) systems with adversarial watermarks, 2020. https://arxiv.org/abs/2002.03095.

[12] M. Naseer, S. H. Khan, S. Rahman, *et al.*, Task-generalizable adversarial attack based on perceptual metric, 2018. https://arxiv.org/abs/1811.09020v3.

[13] A. Boloor, K. Garimella, X. He, *et al.*, *Attacking vision-based perception in end-to-end autonomous driving models,* J. Syst. Archit. 110 (2020), 101766.

[14] O. Poursaeed, I. Katsman, B. Gao, *et al.*, Generative adversarial perturbations, in Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 2018.

[15] J.H. Metzen, M.C. Kumar, T. Brox, *et al.*, Universal adversarial perturbations against semantic image segmentation, in Proceedings of IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 2017, pp. 2774–2783.

[16] V. Fischer, M.C. Kumar, J.H. Metzen, *et al.*, Adversarial examples for semantic image segmentation, in Proceedings of the International Conference on Learning Representations (ICLR), Toulon, France, 2017.

[17] C. Xie, J. Wang, Z. Zhang, Y. Zhou, *et al.*, Adversarial examples for semantic segmentation and object detection, in Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 2017, pp. 1369–1378.

[18] U. Osahor, N. Nasrabadi, Deep adversarial attack on target detection systems, in Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications, International Society for Optics and Photonics, Baltimore, MD, USA, 2019.

[19] T. Miyato, A.M. Dai, I.J. Goodfellow, Adversarial training methods for semi-supervised text classification, in Proceedings of the International Conference on Learning Representations (ICLR), San Juan, Puerto Rico, 2016.

[20] O. Ronneberger, P. Fischer, T. Brox, U-net: convolutional networks for biomedical image segmentation, in Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI), Munich, Germany, 2015, pp. 234–241.

[21] M. Everingham, S.M. Eslami, L. Van Gool, *et al.*, The Pascal visual object classes challenge: a retrospective, Int. J. Comput. Vis. 111 (2015), 98–136.

[22] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, *et al.*, The cityscapes dataset for semantic urban scene understanding, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016, pp. 3213–3223.