
APST

Asia-Pacific Journal of Science and Technology
<https://www.tci-thaijo.org/index.php/APST/index>

 Published by Research and Innovation Department,
 Khon Kaen University, Thailand

Copy-paste attacks: Targeted physical world attacks on self-driving

 Chuanxiang Bi¹ and Jian Qu^{1,*}
¹ Faculty of Engineering and Technology, Panyapiwat Institute of Management, Nonthaburi, Thailand

 *Corresponding author: jianqu@pim.ac.th

Received 30 January 2024

Revised 24 September 2024

 Accepted 13 January 2025

Abstract

Deep neural networks are susceptible to adversarial attacks, which ranging from unseen perturbations to tiny inconspicuous attacks that can cause deep neural networks (DNN) to output errors. Although many adversarial attack methods have been proposed, most methods cannot be easily applied in the physical (real) world due to their use of over-detailed images; such images could not be printed on a normal scale. In this paper, we propose a novel method of physical adversarial attack, the Copy-Paste Attack, by copying other image pattern elements to make stickers and pasting them on the attack target. This attack can be printed out and applied in the physical world. Moreover, this attack reduces the recognition accuracy of deep neural network by making the model misclassify the traffic signs as the attack pattern. We conducted our experiment with a model intelligent car in a physical world. We tested three well-known DNN models, on three different kinds of Datasets. The experimental results demonstrate that our proposed collaborative performance advertising solution (CPAs) greatly interferes with the recognition rate of traffic signs. Moreover, our CPAs outperform the existing method PR2. Furthermore, we tested one of our previous ResNet26-carbon border adjustment mechanism (CBAM) models, although it exhibits higher robustness against the CPA attack compared with other well-known CNN models, our ResNet26-CBAM also got misguided by CPAs with an accuracy of 60%. In addition, we trained the CNN models with the physical defense method of adversarial training; however, it had little effect on our CPA attacks.

Keywords: Deep neural networks, Copy-Paste Attack, Adversarial attacks, ResNet26-CBAM, Adversarial trained.

1. Introduction

Deep Neural Networks (DNNs) [1] have been widely used in fields like self-driving [2,3,4], mobile payment [5], speech processing [6,7,8], and so on. Recent research has shown that DNNs are vulnerable to adversarial attacks [9], which raises new security risks. Depending on the attack type, adversarial attacks in computer vision can now be divided into two categories: digital and physical attacks. While digital assaults such as fast gradient sign method (MI-FGSM) [10], C&W [7], and Deep fool [11] exhibit strong performance in DNN models, their effectiveness in physical world applications is limited due to the difficulty of detecting digital perturbations in the physical world [12,13]. In contrast, physical attacks are more threatening, especially in the self-driving domain, as self-driving cars rely on DNNs to recognize and respond to traffic signs. If an attacker is able to spoof DNNs through physical adversarial examples, self-driving cars may misinterpret traffic signals, leading to potential traffic accidents.

Since Kurakin *et al.* [14] initially showed that DNNs are vulnerable to physical attacks, research in this field has received much attention. ShapeShifter [15] demonstrates the feasibility of early adversarial physical attacks by executing formulas to generate adversarial stop signs with complex patterns. However, the ShapeShifter attack method is limited in practical application to real traffic signs. AdvCam [16] uses neural style migration to create naturally corrosive style stop signs that hide human attention and maintain large perturbations. This approach demonstrates that the stealth and effectiveness of physical attacks can be enhanced by introducing neural-style migration techniques. However, AdvCam may lose pixel points during the printing process, thus reducing its

effectiveness. Robust Physical Perturbations (RP2) takes a different approach, attacking by printing actual-sized traffic signs on paper and overlaying them onto existing signs. Although RP2 successfully spoofs the DNN classifier, its attack is mainly limited to STOP traffic signs.

In this paper, based on existing research, we propose a new approach that aims to extend the applicability of physical adversarial attacks on traffic signs and improve the feasibility of their application in the real world. By drawing on and improving existing methods, we hope to enhance the effectiveness of adversarial attacks further. The main contributions of this paper are:

- (1) We present a flexible adversarial attack methodology, the Copy-Paste Attacks (CPA), that can generate attacks ranging from simple traffic elements to complex mixes of multiple classes of traffic elements.
- (2) We utilize the ResNet26-CBAM model from our previous work [17], known for its superior interference resistance, to conduct a series of new experiments on our CPA.
- (3) Experiments with CPA in physical world scenarios show that the adversarial examples are highly deceptive and can successfully trick the network into incorrectly categorizing the examples into the attacking classes. Our experiments used a real intelligent model car to test against real traffic signs to validate the effectiveness of our attacks. Our experiments may get noise effects from light sources, and environmental shadow, thus, our experiments provide a more objective and rigorous factual basis compared to most experiments that employ only DNN classifiers in virtual simulation or using a webcam.

2. Materials and methods

In this section, we aim to verify the practicality and objectivity of the copy-paste attack in a self-driving scenario. Firstly, we explain the details of the intelligent model car, the simulation environment, and the 16 traffic signs used in the experiments. Secondly, we explain the design and construction of the copy-paste attack and the application of the adversarial training method. Finally, we explain the designs of each experiment, which is used to test the impact of the copy-paste attacks.

2.1 Materials

In this study, we conduct experiments using real model cars in the physical world rather than testing adversarial attacks using a webcam like most existing methods. Webcams are usually low-cost, and easy for adversarial attack experiments, however, it is difficult to show the details of how a real car behaves when it encounters an adversarial attack. Due to the expensive and safety issues of real self-driving cars, we built intelligent model cars that can be used for self-driving research, our car is shown in Figure 1. The intelligent model car imitates the real car, the difference is that there are multiple sensors in the real self-driving car, while there is only one camera in our intelligent model car. The experiments for the intelligent model car are conducted in a simulation environment. Which can be repeated under controlled conditions, the decision errors in the real environment can be better observed, that is crucial for verifying the experimental results and ensuring the scientific validity of the experiments.

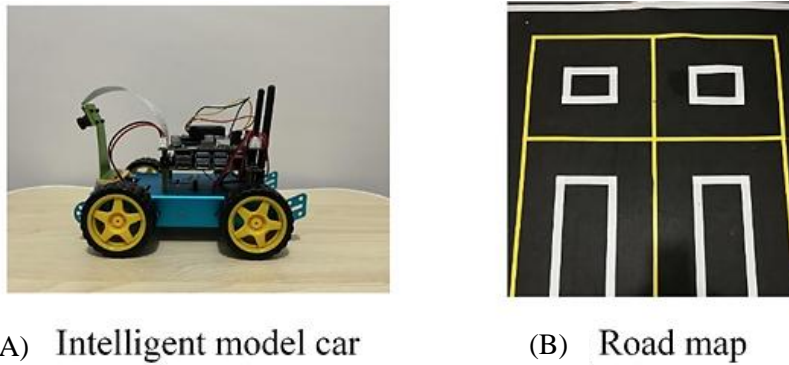


Figure 1 (A) Intelligent model car built in this paper, (B) Map of the road designed in this paper.

We designed simulated road maps and traffic signs in the laboratory to replicate the real-world driving environment. As shown in Figure 2, this paper uses four series of traffic signs, a total of 16 traffic signs, each category of traffic signs has a different shape and color. These 16 traffic signs are all made using the same base and material to reduce the likelihood of the model recognizing the traffic signs by looking at the differences in other aspects of the traffic signs. These 16 traffic signs were then used with equal frequency in subsequent experiments. With this design, this paper can conduct effective self-driving studies in more realistic environments and improve the robustness of the model for traffic signs.



Figure 2 The 16 traffic signs used in the experiment, (A) forward traffic signs, (B) right traffic signs, (C) left traffic signs, and (D) stop traffic signs. Each group is divided into four series of traffic signs, a, b, c, and d.

2.2 Methods

As demonstrated in Figure 3, Shapshifter [15] is an algorithm that creates traffic sign surfaces that can mislead the model recognition. RP2 applies graffiti interference [18] on the traffic sign, and Advcam creates surface patterns resembling the surrounding environment using algorithms [13] to be pasted on the traffic sign. However, most of these physical attacks produce high classification accuracy when tested with our intelligent model car in the physical world. Mostly due to the detailed attack they used cannot be perfectly printed out in the physical world.



Figure 3 Examples of successful physical world attacks.

Figure 4 illustrates how we duplicate a portion of the incomplete pattern from certain traffic signs and apply it to other types of traffic indicators to create the CPAs. This type of attack may have a significant impact on the detection and decision-making of an AI agent, although such attacks usually do not cause a human driver to lose their visual judgment. We put an arrow indicator to the right on the stop sign, an arrow indicator to the left on the right-turn sign, and an "OP" style pattern on the left-turn sign. Through the CPAs, we can misguide the intelligent model car to make the desired reaction with a high probability according to the kind of attack pattern. As can be seen from Figure 5, the CPAs in this paper are classified into two styles: the simple style uses only one element from the other traffic signs, and the complex style has a stronger attack compared to the former, using multiple elements of the other traffic signs or multiple elements of traffic signs. The attack method in this paper is robust to a variety of physical conditions, modifying and adding the physical characteristics of the input samples to the deep neural network is adversarial in the physical world.

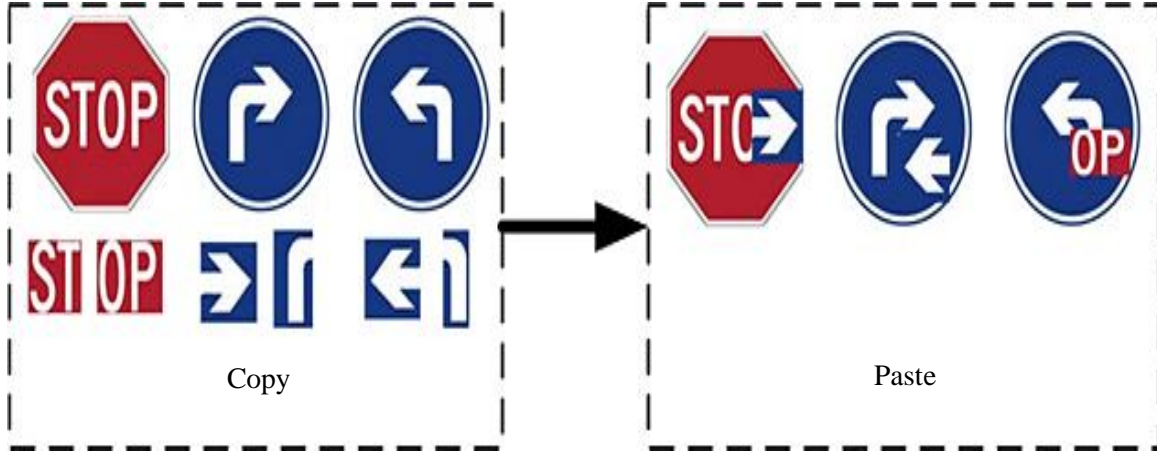


Figure 4 Pattern elements are copied from each category by the proposed CPA. Then paste them to create a camouflage counterexample.

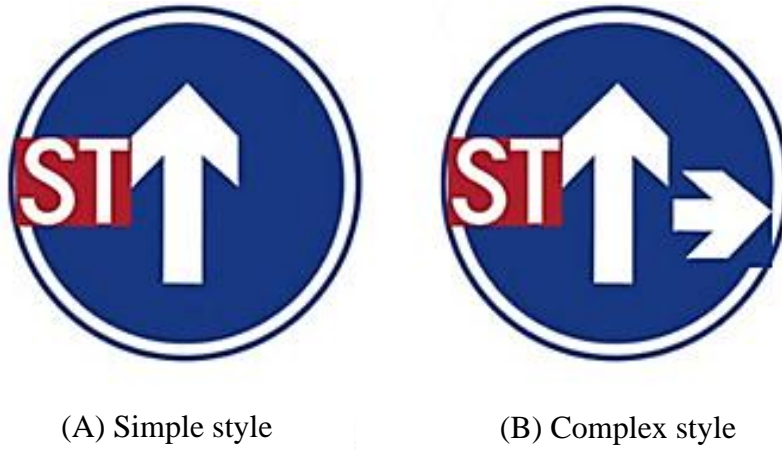


Figure 5 (A) simple style CPA, (B) complex style CPA.

This paper utilizes the previously developed ResNet26-CBAM model, known for its strong anti-interference performance [17]. As shown in Figure 6, ResNet employs a residual block design to combat vanishing and exploding gradients in deep networks. Each block consists of two convolutional layers, with the output of the second added to the first, allowing the network to learn residual mappings and simplify optimization.

To enhance the original ResNet-18 model, we increase its depth by adding extra convolutional and batch normalization layers, named conv3, bn3, conv4, and bn4, into each basic block. These additions improve feature extraction and accelerate training by normalizing feature maps. The details of the blocks are explained below.

Data first flows into the Channel Attention Model, where MaxPool and AvgPool operations generate two-channel feature maps. These maps are processed through a multi-layer perceptron (MLP) to calculate channel weights. The channel weights are then multiplied element-wise with the input feature map to produce the channel-weighted feature map.

This channel-weighted feature map is then passed into the Spatial Attention Model. In this model, spatial feature maps are computed through MaxPool and AvgPool operations, followed by a 7x7 convolution to generate the spatial attention map. The spatial attention map is multiplied element-wise with the channel-weighted feature map to produce the final optimized output.

These two modules operate sequentially: the Channel Attention Model is applied first, followed by the Spatial Attention Model, with both contributing to feature optimization and attention refinement.

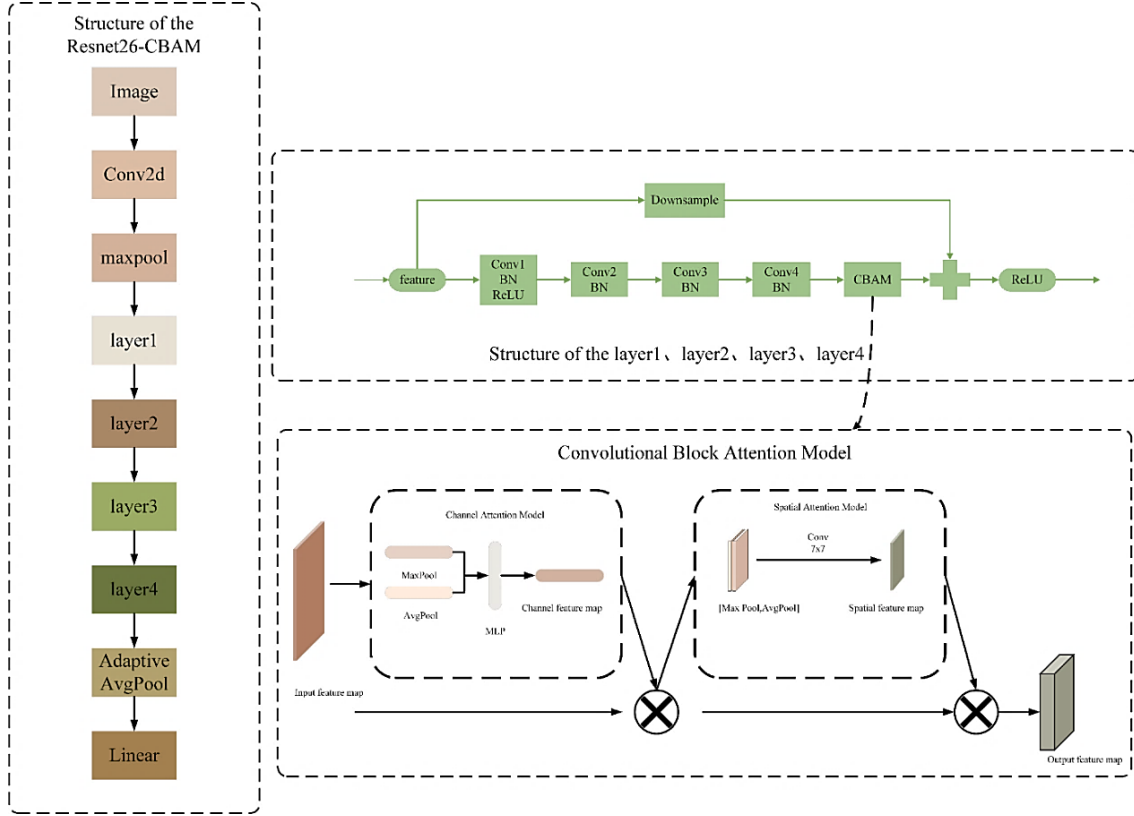


Figure 6 ResNet26-CBAM framework.

2.3 Evaluation methods

The experimental evaluation method in this paper utilizes a four-category confusion matrix. We used four evaluation metrics, Precision, Recall, F1 Score, and accuracy, which are defined as follows.

$$Precision = \frac{TP}{TP + FP} \quad (1)$$

$$Recall = \frac{TP}{TP + FN} \quad (2)$$

$$F1\ Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (3)$$

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (4)$$

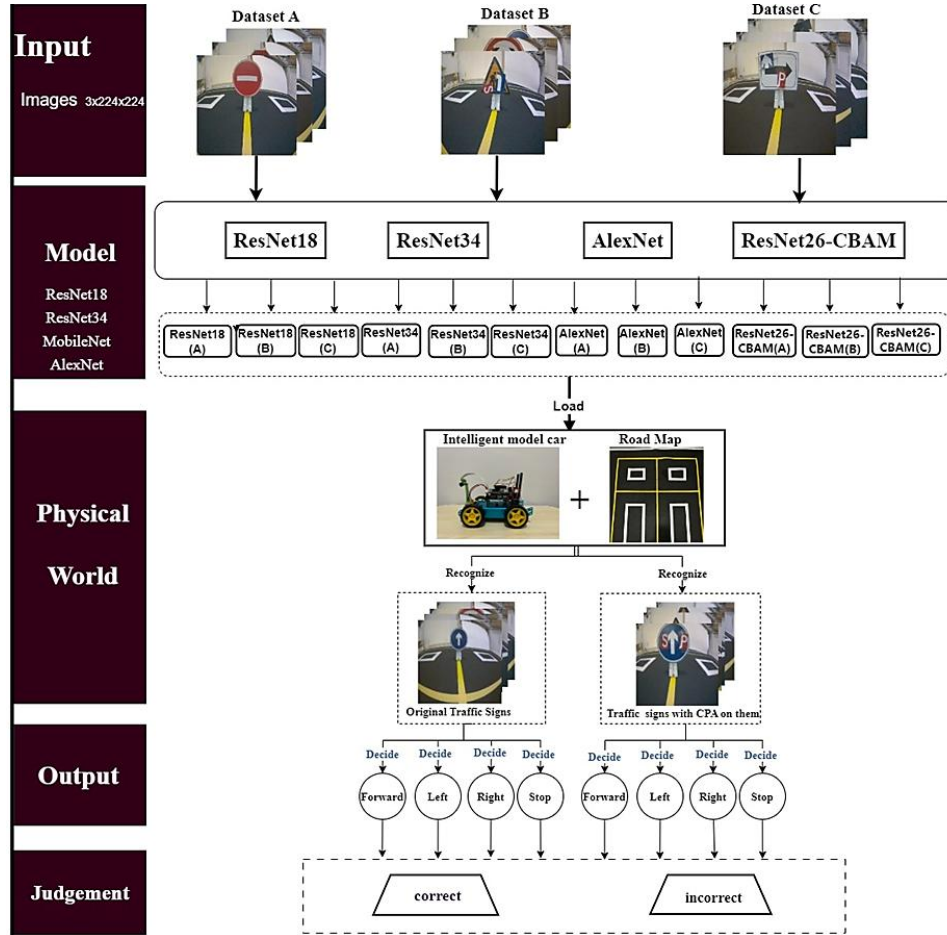
Here, we introduce four metrics, P (Positive), N (Negative), T (True), and F (False), for evaluating the environment perception ability of the intelligent model car. First, we make one of the classifications in the traffic sign as positive (P) and the rest as negative (N), T indicates that the intelligent model vehicle predicted correctly, while F indicates that the intelligent model vehicle predicted incorrectly. As shown in Table 1, for example, to calculate the correlation metrics of the forward traffic sign category, a 4×4 matrix is obtained with forward as the positive classification and the rest of the categories as the negative classification. TP (True Positive) indicates that the intelligent model car predicted correctly after testing the traffic signs belonging to positive classification, FP (False Positive) indicates that the intelligent model car predicted incorrectly after testing the traffic signs belonging to positive classification, TN (True Negative) indicates that the intelligent model car predicted correctly after testing the traffic signs belonging to negative classification, and FN (False Negative) indicates that the intelligent model car predicted incorrectly after testing the traffic signs belonging to the negative classification. If relevant metrics are to be calculated for other categories, the category to be calculated is used as a reference.

Table 1 Confusion matrix assessment table as an example for calculating forward-related metrics.

		Predicted			
		Forward	Left	Right	Stop
Actual	Forward	TP		FN	
	Left				
	Right	FP		TN	
	Stop				

2.4 Experimental Setup

To verify the practicality and objectivity of the attacks in this paper, we use an adversarial training [7] method in the physical defense method: interference injection [19], which involves injecting possible attacks in the training Dataset to reduce the model's sensitivity to such attacks. It is worth noting that most of the current research on adversarial attacks ignores the impact of adversarial defense and only proves if their attacks work on the original Dataset. As shown in Figure 7, this paper trains the same models with the Dataset without interference and the Dataset containing interference (interference injection) respectively and then loads them onto the intelligent model car. This paper tests the recognition ability of the normal traffic sign and the traffic sign labeled with CPA on a simulated route to compare multiple deep-learning models. In the subsequent experiments, we demonstrate that adversarial training, as an adversarial defense method, is less effective than the CPA method proposed in this paper.

**Figure 7** Experimental framework

2.4.1 Datasets

In terms of Datasets, this paper uses three kinds of Datasets that we have collected; they are Dataset A, Dataset B, and Dataset C. These three Datasets are four-category Datasets with four categories of "forward," "left," "right,"

and "stop," which are collected and put into the categories that they belong to. However, the difference is that Dataset A contains only normal traffic sign images, Dataset B has images of normal traffic signs and images of CPA-attacked traffic signs for adversarial training, and Dataset C contains only images of traffic signs attacked by CPAs.

Table 2 Dataset A, "without traffic sign," is used to train the intelligent model car to go straight, then 400 pictures of each traffic sign group are collected.

Class	Without traffic signs	(a)	(b)	(c)	(d)	Total pictures
Forward	200	400	400	400	400	1800
Left	0	400	400	400	400	1600
Right	0	400	400	400	400	1600
Stop	0	400	400	400	400	1600
Total pictures	200	1600	1600	1600	1600	6600

Before introducing Dataset B, we discuss which ratio is more appropriate to use for adversarial training images of physical defense, we reproduce the PR2 attack for experiments, and we set up the Dataset based on Dataset (A) with different ratios of the normal traffic signs and the attacked images being 2:1, 3:1, and 4:1, we then train these Datasets with resnet34, and carry out the 16 traffic signs affixed with the PR2 Tests, each traffic sign test 5 times, as shown in Table 3. When Dataset A's trained model faces the attack, the accuracy is 78%. While the model trained with an adversarial training Dataset is significantly improved, especially when using the 3:1 and 2:1 ratios of the trained model, the accuracy is 88% and 89%, respectively. So we used these two models to test the accuracy against Dataset A to check their ability for classification of normal traffic signs.

Table 3 Test results of four models (ResNet34(A), ResNet34(A (4:1)), ResNet34(A (3:1)), and ResNet34(A (2:1)) recognize traffic signs labeled with PR2.

	Class	ResNet34(A)	ResNet34(A (4:1))	ResNet34(A (3:1))	ResNet34(A (2:1))
Precision	Forward	0.84	0.68	0.76	0.79
	Left	0.64	0.94	0.94	0.95
	Right	0.9	0.81	0.86	0.86
	Stop	0.77	0.93	1	1
Recall	Forward	0.92	0.95	0.95	0.95
	Left	0.82	0.8	0.85	0.8
	Right	0.7	0.81	0.9	0.9
	Stop	0.7	0.65	0.8	0.8
F1	Forward	0.87	0.79	0.84	0.86
	Left	0.71	0.86	0.89	0.98
	Right	0.78	0.81	0.88	0.88
	Stop	0.73	0.76	0.89	0.89
Accuracy		78%	81%	88%	89%

As shown in Table 4, the accuracy of the 2:1 ratio model in recognizing the normal traffic sign is reduced by 6% compared to the 3:1 ratio model. So, combining the above experiments, we conclude that it is better to set the adversarial training Dataset with a 3:1 ratio and build The Dataset B.

Table 4 Test results of two models (ResNet34(A (3:1)), and ResNet34(A (2:1)) for recognizing normal traffic signs.

	Class	ResNet34(A (3:1))	ResNet34(A (2:1))
Precision	Forward	0.8	0.7
	Left	0.94	0.94
	Right	0.86	0.8
	Stop	0.94	0.88
Recall	Forward	1	0.95
	Left	0.8	0.75
	Right	0.9	0.8
	Stop	0.8	0.75

	Class	ResNet34(A (3:1))	ResNet34(A (2:1))
F1	Forward	0.88	0.81
	Left	0.86	0.83
	Right	0.88	0.8
	Stop	0.86	0.81
Accuracy		87%	81%

Table 5 Dataset B, "without traffic sign" is used to train the intelligent model car to drive straight, then 300 pictures of each traffic sign group are collected, and 100 pictures of each group are attacked with CPA.

Class	Without traffic signs	(a)	(b)	(c)	(d)	(a) +CPA	(b) +CPA	(c) +CPA	(d) +CPA	Total pictures
Forward	200	300	300	300	300	100	100	100	100	1800
Left	0	300	300	300	300	100	100	100	100	1600
Right	0	300	300	300	300	100	100	100	100	1600
Stop	0	300	300	300	300	100	100	100	100	1600
Total pictures	200	1200	1200	1200	1200	400	400	400	400	6600

Table 6 Dataset C, "without traffic sign" is used to train the intelligent model car to go straight, and then 400 images of each group are attacked with CPA.

Class	Without traffic signs	(a)+CPA	(b) +CPA	(c) +CPA	(d) +CPA	Total pictures
Forward	200	400	400	400	400	1800
Left	0	400	400	400	400	1600
Right	0	400	400	400	400	1600
Stop	0	400	400	400	400	1600
Total pictures	200	1600	1600	1600	1600	6600

Datasets A, B, and C allow our research to assess the effect of adding adversarial attacks on the performance of the model. Furthermore, it helps with the assessments of adversarial attacks both with and without the CPA interference injection.

2.4.2 Experiments

In order to verify the practicality and objectivity of our CPA proposed in self-driving scenarios, we designed six experiments. We utilize Dataset A, Dataset B, and Dataset C to train four CNN models (ResNet18, ResNet34, ResNet26-CBAM, and AlexNet), respectively. Then these models are tested on normal traffic signs and traffic signs affixed with CPA. The details of the experiments are shown in Table 7:

Table 7 Specifics of the six experiments.

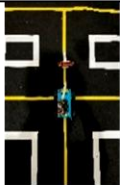
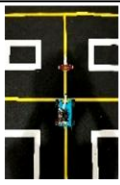








Dataset	Model	Test object	Goal
Dataset A	ResNet18(A), ResNet34(A), ResNet26-CBAM(A), AlexNet(A)	Normal traffic signs	Evaluating model performance on normal traffic signs
Dataset A	ResNet18(A), ResNet34(A), ResNet26-CBAM(A), AlexNet(A)	Traffic signs with CPA Attack	Evaluating model performance under CPA attacks
Dataset B	ResNet18(B), ResNet34(B), ResNet26-CBAM(B), AlexNet(B)	Normal traffic signs	Evaluating the Impact of adversarial training on model recognition of normal traffic signs
Dataset B	ResNet18(B), ResNet34(B), ResNet26-CBAM(B), AlexNet(B)	Traffic signs with CPA Attack	Evaluating the Impact of Adversarial Training on Model Recognition of CPA Attack Signs
Dataset C	ResNet18(C), ResNet34(C), ResNet26-CBAM(C), AlexNet(C)	Normal traffic signs	Evaluating the impact of recognizing normal signs by models trained on Dataset C
Dataset C	ResNet18(C), ResNet34(C), ResNet26-CBAM(C), AlexNet(C)	Traffic signs with CPA Attack	Evaluating the Impact of Models Trained on Dataset C to recognizing CPA attack signs

Firstly, this study trains four different CNN models using Dataset A, which contains only normal traffic signs. Next, we load the trained models onto the intelligent model car and perform Experiment 1 and Experiment 2.

Experiment 1 was conducted for 16 normal traffic signs. The experimental steps were to conduct 10 tests for each traffic sign, with each test being at a different angle and distance, as can be seen in Table 8, this was to test the recognition of the traffic sign by the intelligent model car from different angles and distances, the idea is to prevent a single location from being used for recognition. All the above four CNN models are tested sequentially to record the intelligent model car's response to each traffic sign to further analyze and evaluate the performance of each model.

Experiment 2 is called the random CPA attack, it is conducted on 16 traffic signs affixed with CPAs. The experimental steps are, firstly, we selected ten CPAs with different targeted attack patterns, as can be seen from Figure 8, our selection contains CPAs ranging from simple ones to complex ones. (1) "ST" for stop (a), (2) "OP" for stop (a), (3) elements for left (a), (4) Paste the elements of the right (b) traffic sign, (5) Paste the elements of the left (a) traffic sign, (6) Paste the elements of the stop (a) and right (a) traffic signs, (7) Paste the elements of the right (b) and left (a) traffic signs. (8) Paste the elements of right (b) and stop (a) traffic signs, (9) Paste the elements of left (a) and stop (a) traffic signs, (10) Paste the elements of right (a) and stop (a) traffic signs. We used 16 traffic signs affixed with the 10 attack patterns in different locations and conducted 10 tests (around 1600 random tests), we used random angles and distances for each test (Table 8, the same as in Experiment 1), and all of the four CNN models mentioned above are tested in turn to record the intelligent model car's response to the CPA attacked traffic signs.

Table 8 10 tests at different distances and angles. The same test method was used in the follow-up experiments.

Distance	Angle	Example	Distance	Angle	Example
15cm	0°		20cm	0°	
20cm	15°		20cm	-15°	
30cm	0°		30cm	-15°	
30cm	15°		30cm	-30°	
30cm	30°		40cm	0°	

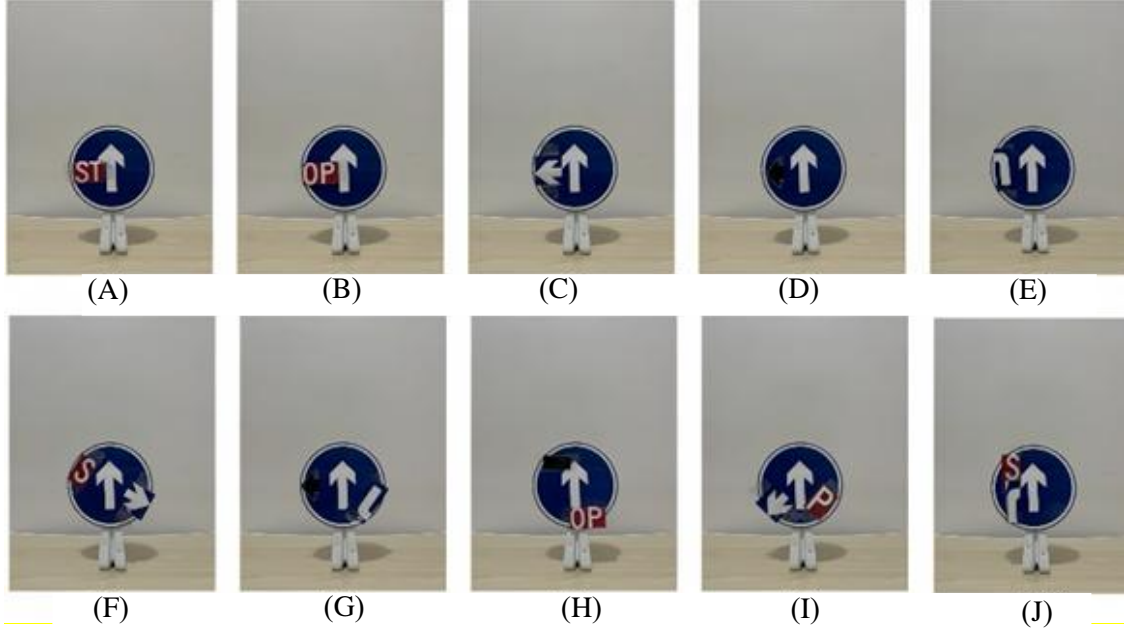


Figure 8 Take forward as an example and paste ten CPA attacks with different targeting positions. (A) (B) (C) (D) (E) are labelled with simple style CPAs and (F) (G) (H) (I) (J) are labelled with complex style CPAs.

Secondly, this study uses Dataset B to train the four CNN models. Next, we load the trained models onto the intelligent model car and conduct Experiment 3 and Experiment 4.

Experiment 3 has the same specific experimental steps as Experiment 1, and the data from Experiment 3 and Experiment 1 is compared to determine whether the adversarial training model with interference injection has weakened its ability to recognize normal targets.

Experiment 4 has the same steps as Experiment 2, including ten different CPA attacks. By comparing the data from Experiment 3 and Experiment 4, we try to find out which of the adversarial trained models performs better against traffic signs with CPAs, it is compared to the models without adversarial training.

Finally, this study uses Dataset C to train the four CNN models. Next, we load the trained models onto the intelligent model car and conduct Experiment 5 and Experiment 6.

The specific experimental steps of Experiment 5 are the same as Experiment 1, we evaluate the recognition performance of the four CNN models trained using Dataset C on the normal traffic signs.

The specific experimental steps of Experiment 6 are the same as Experiment 2, which includes ten different CPA attacks. We evaluate the recognition performance of four CNN models trained using Dataset C on CPA-attacked traffic signs.

As can be seen from Figure 9, which shows the heat-map, the intelligent model car mainly makes judgments by observing the patterns on the traffic signs and then displays the real-time predicted action probability distributions on our visualization interface (as in Figure 10), then makes the actions with the highest probability. We observe the distribution of the probability value of each classification, when the probability value is higher than 0.5, the intelligent model car judges it as belonging to the classification with a high probability value and successfully makes the corresponding action, and such a case is judged as a failure of the attack. Suppose it encounters a category in which the probability value does not exceed 0.5. In that case, the intelligent model car will make a wrong action or waver (due to there being two classifications with the same probability), in which case the attack is judged as successful. Ultimately, we will compare the accuracy of four CNN models trained on Dataset A in recognizing normal traffic signs and traffic signs labeled with CPA. This can provide proof of whether CPAs are effective in the physical world.

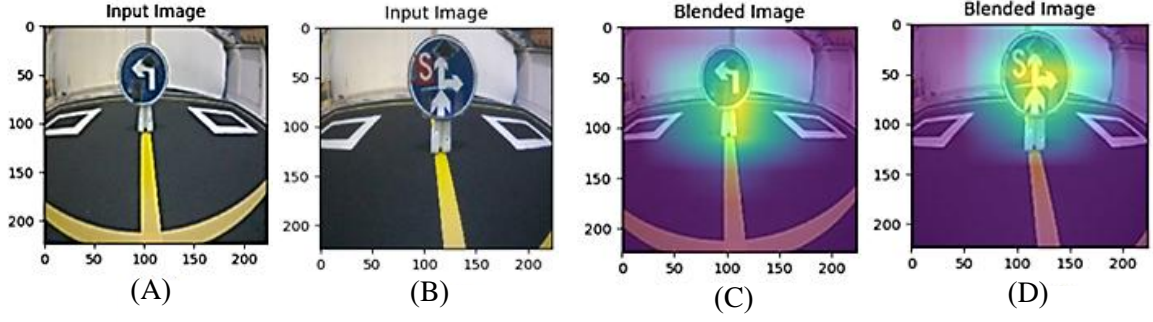


Figure 9 (A) (B) Images captured by the camera of the intelligent model car, (C) (D) Attention heatmap of the intelligent model car

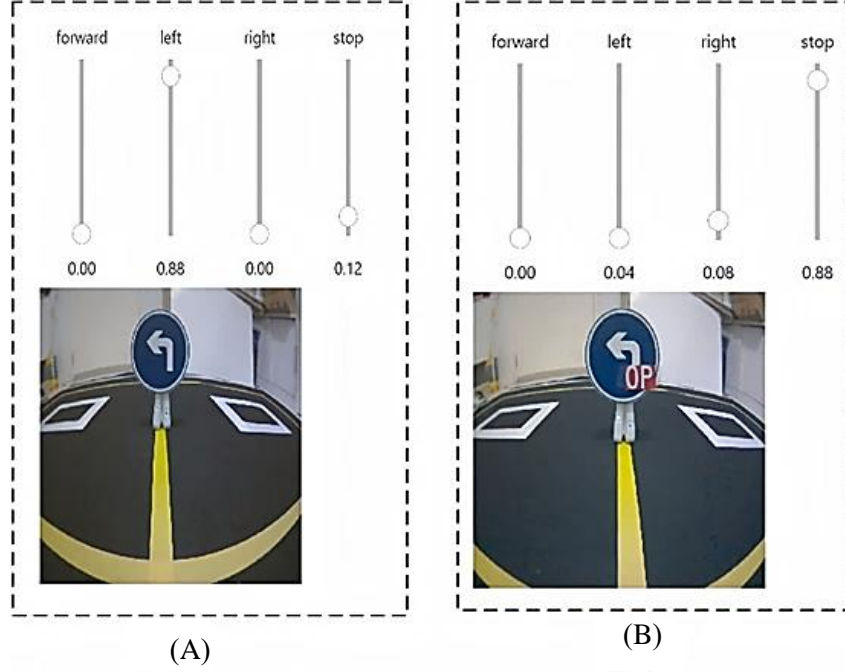


Figure 10 Probability distributions for each category displayed by the intelligent model car in the visualization interface. (A) is a normal left-turn traffic sign with a LEFT accuracy value of 0.88. (B) is a left-turn traffic sign with a CPA sticker with a LEFT accuracy value of 0.04.

3. Results

In this section, we will describe the results of six experiments in detail, focusing on comparing the recognition performance of different Datasets and different models against original and traffic signs labeled with CPA attacks. We will also discuss the effect of adversarial training on model robustness. With these results, we can verify the effectiveness of the copy-paste attack (CPA) proposed in this paper and the defense effect of adversarial training.

According to our evaluation method, we can calculate the accuracy, precision, recall, and F1 score of each group of experiments by using the experimental data. Accuracy provides the overall proportion of correct classification and is the most intuitive evaluation index, provides an overall performance, evaluation and reflects the average prediction ability of the model for all categories, so in this paper, we use accuracy as the main reference standard, and the other three metrics as the secondary reference standard.

As shown in Table 9, it could be concluded that the test results of Experiment 1 and Experiment 2, the four CNN models trained with Dataset A performed well in the normal traffic sign experiment, they yielded high accuracy, precision, recall, and F1 Score, which indicated that these models are able to accurately recognize traffic signs without adversarial attacks. The accuracy rate reached as high as 94%. However, in the CPA attack test, the performance of all models dropped significantly, with all metrics significantly lower than the normal traffic sign test. This indicated that our CPA attack has a strong interfering effect on the recognition ability of self-driving models. Only ResNet26-CBAM (A) achieved an accuracy higher than 50%, with a peak accuracy of 60%, while

other models fell below 50%. In particular, AlexNet (A) performed relatively poorly across all tests, especially under the CPA attack, where it recorded the lowest accuracy and all related metrics.

Table 9 Test results using four CNN models (ResNet18(A), ResNet34(A), AlexNet(A), and ResNet26-CBAM(A)) trained on Dataset A (contains only normal traffic signs) to recognize normal traffic signs, traffic signs with random CPA attack (experiment 1,2).

	Class	ResNet18 (A)	ResNet18 (A)+CPA	ResNet34 (A)	ResNet34 (A)+CPA	AlexNet (A)	AlexNet (A)+CPA	ResNet26- CBAM (A)	ResNet26- CBAM (A)+CPA
Precision	Forward	1	0.62	0.89	0.48	0.65	0.33	0.87	0.65
	Left	0.81	0.49	1	0.5	0.71	0.21	1	0.66
	Right	1	0.36	0.72	0.31	0.68	0.29	1	0.6
	Stop	0.88	0.32	0.94	0.26	0.77	0.27	0.9	0.54
Recall	Forward	1	0.69	1	0.72	0.9	0.39	1	0.75
	Left	0.95	0.22	0.75	0.25	0.62	0.24	0.9	0.6
	Right	1	0.34	0.95	0.29	0.6	0.12	0.9	0.45
	Stop	0.88	0.47	0.75	0.28	0.67	0.34	0.95	0.65
F1	Forward	1	0.65	0.94	0.58	0.75	0.36	0.93	0.9
	Left	0.87	0.3	0.86	0.33	0.66	0.22	0.95	0.62
	Right	1	0.35	0.82	0.3	0.64	0.17	0.95	0.51
	Stop	0.88	0.38	0.83	0.27	0.72	0.3	0.92	0.58
Accuracy		91%	43%	86%	39%	70%	27%	94%	60%
Ideal		1	0	1	0	1	0	1	0

Note: Ideal model performance with normal traffic signs: the higher the value the better,
Ideal model performance with attacked traffic signs⁷: the lower the value the better

We reproduced the graffiti attack in PR2 for comparison with our method. As seen in Table 10, the accuracy rate was much higher than our CPA method. This result suggested that the impact of the CPA attack is greater than that of the PR2 attack.

Table 10 Test results using four CNN models (ResNet18(A), ResNet34(A), AlexNet(A), and ResNet26-CBAM(A)) trained on Dataset A (contains only normal traffic signs) to recognize traffic signs labeled with RP2.

	Class	ResNet18 (A)+PR2	ResNet34 (A)+PR2	AlexNet (A)+PR2	ResNet26-CBAM (A)+PR2
Precision	Forward	0.86	0.84	0.61	0.86
	Left	0.93	0.64	0.66	0.89
	Right	0.67	0.9	0.67	0.89
	Stop	0.87	0.77	0.72	0.81
Recall	Forward	0.95	0.92	0.87	0.9
	Left	0.75	0.82	0.6	0.85
	Right	0.87	0.7	0.57	0.85
	Stop	0.7	0.7	0.6	0.85
F1	Forward	0.9	0.87	0.72	0.88
	Left	0.83	0.71	0.62	0.87
	Right	0.76	0.78	0.62	0.87
	Stop	0.78	0.73	0.65	0.83
Accuracy		81%	78%	66%	86%

Regarding the comparison between Experiment 3 and Experiment 4, Table 11 showed that all models performed well in the normal traffic sign test, we achieved an accuracy of 88% (ResNet18 (B)). We then used the interference-injected Dataset to train the models, and the results showed that the accuracy of these four models in recognizing traffic signs with CPAs is still low, reaching a maximum of 44%. This result suggests that our CPAs also highly aggressive to the physical adversarial trained models.

Table 11 Test results of four models (ResNet18(B), ResNet34(B), AlexNet(B), and ResNet26-CBAM(B)) trained with Dataset B (Contains normal traffic sign images and images of CPA attacks) to recognize normal traffic signs and traffic signs with random CPA attack (experiment 3,4).

	Class	ResNet18 (B)	ResNet18 (B)+CPA	ResNet34 (B)	ResNet34 (B)+CPA	AlexNet (B)	AlexNet (B)+CPA	ResNet26- CBAM (B)	ResNet26- CBAM (B)+CPA
Precision	Forward	1	0.57	0.95	0.55	0.63	0.42	0.86	0.64
	Left	0.74	0.6	0.67	0.31	0.72	0.25	0.9	0.71
	Right	1	0.34	0.94	0.61	0.67	0.12	0.94	0.69
	Stop	0.81	0.38	0.8	0.28	0.72	0.34	0.9	0.64
Recall	Forward	1	0.66	0.92	0.68	0.82	0.56	0.95	0.8
	Left	0.92	0.26	0.87	0.27	0.7	0.23	0.9	0.6
	Right	0.8	0.35	0.8	0.31	0.55	0.08	0.85	0.55
	Stop	0.77	0.52	0.7	0.39	0.65	0.38	0.9	0.7
F1	Forward	1	0.61	0.93	0.61	0.71	0.48	0.9	0.71
	Left	0.82	0.36	0.76	0.29	0.71	0.24	0.9	0.65
	Right	0.89	0.34	0.86	0.41	0.6	0.1	0.89	0.61
	Stop	0.79	0.44	0.75	0.33	0.68	0.36	0.9	0.67
Accuracy		88%	44%	83%	41%	68%	31%	90%	66%
Ideal		1	0	1	0	1	0	1	0

Note: Ideal model performance with normal traffic signs: the higher the value the better,
Ideal model performance with attacked traffic signs: the lower the value the better.

As can be seen in Figure 11, the comparison of the accuracy of Experiment 1 and Experiment 3, these four CNN models trained with Dataset A have higher accuracies than those trained with Dataset B for recognizing normal traffic signs, reaching a maximum of 95%. Regarding the comparison of the accuracy rates of Experiment 2 and Experiment 4, the difference was that these four CNN models trained with Dataset A have a lower accuracy rate than those trained with Dataset B for recognizing traffic signs with CPAs, reaching a maximum of 66% and a minimum of 27%.

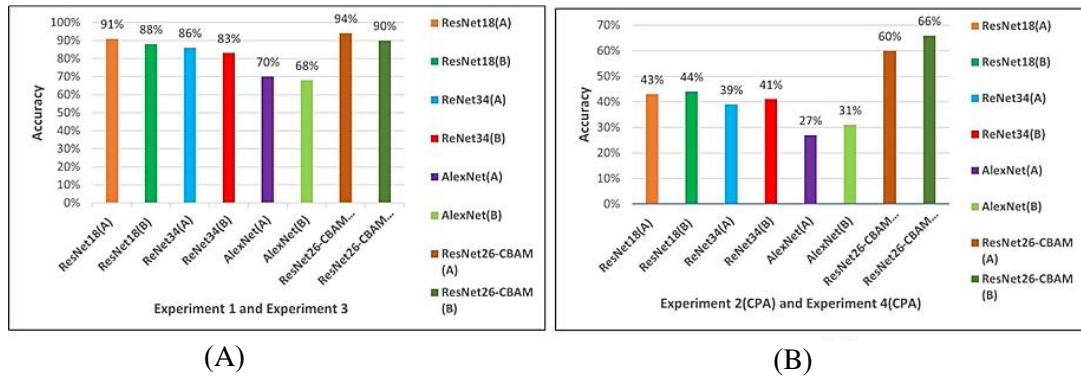


Figure 11 (A) Comparison of accuracy between Experiment 1 and Experiment 3, (B) Comparison of accuracy between Experiment 2 and Experiment 4.

Table 12 Test results of four models (ResNet18 (C), ResNet34 (C), AlexNet (C), and ResNet26-CBAM (C)) trained with dataset B to recognize normal traffic signs and traffic signs labeled with CPA.

	Class	ResNet18 (C)	ResNet18 (C)+CPA	ResNet34 (C)	ResNet34 (C)+CPA	AlexNet (C)	AlexNet (C)+CPA	ResNet26- CBAM (C)	ResNet26- CBAM (C)+CPA
Precision	Forward	0.58	0.58	0.61	0.55	0.42	0.58	0.64	0.72
	Left	0.62	0.62	0.75	0.62	0.33	0.66	0.69	0.77
	Right	0.5	0.42	0.71	0.56	0.25	0.57	0.62	0.66
	Stop	0.63	0.38	0.58	0.57	0.36	0.41	0.56	0.63
Recall	Forward	0.7	0.7	0.8	0.75	0.5	0.7	0.8	0.8
	Left	0.5	0.5	0.6	0.5	0.3	0.6	0.55	0.7
	Right	0.4	0.3	0.5	0.45	0.2	0.4	0.5	0.6
	Stop	0.7	0.5	0.7	0.6	0.4	0.5	0.65	0.7
F1	Forward	0.63	0.63	0.69	0.63	0.45	0.63	0.71	0.75
	Left	0.55	0.55	0.66	0.55	0.31	0.62	0.61	0.73
	Right	0.44	0.35	0.58	0.5	0.22	0.47	0.55	0.62
	Stop	0.66	0.43	0.63	0.58	0.37	0.45	0.6	0.66
Accuracy		55%	65%	50%	57%	35%	55%	63%	70%

As can be seen from Table 12, the four models trained using dataset C have low recognition accuracy on normal traffic signs, indicating that the models trained on the full attack image dataset perform poorly in recognizing normal traffic signs. This may be since the models mainly learned the adversarial features and ignored the normal features. The recognition accuracy of the four models trained using dataset C on CPA attack traffic signs is also not high, indicating that the models trained on the full attack image dataset still perform poorly in recognizing CPA attack traffic signs.

4. Discussions

In this paper, the attacks targeted the intelligent model car to misclassify the attack pattern according to the kind of CPA. As could be seen from Figure 12, the intelligent model car could recognize the original right-turn sign as "Right Turn" with a classification value of 0.93, but when it was attacked with "ST" (which belongs to the element of stop sign), the classification value of recognized it as "Stop" becomes 0.67. Similarly, a left-turn sign with a stop element was mistakenly recognized as "stop" by the intelligent model car. The Stop sign, labeled with the element of turn, was mistaken as "left" or "right" by the intelligent model car. Therefore, the CPA attacked in this paper can intentionally make the intelligent model car get confused with the attacked traffic signs.

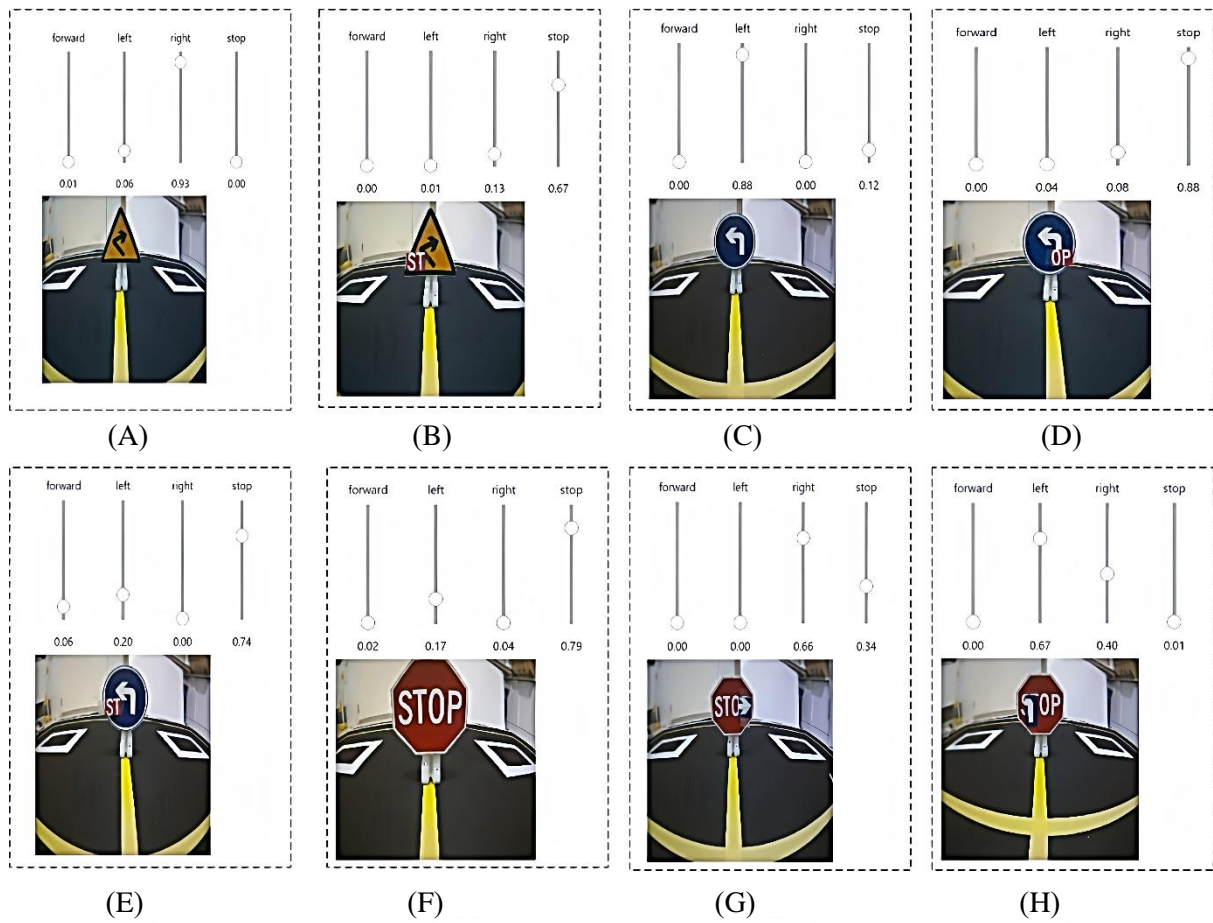


Figure 12 (A), (B), (C) Intelligent model car identifies original traffic sign, (D), (E), (F), (G), (H) Intelligent model car identifies CPA-stickered traffic sign.

It is worth noting that simple CPAs could not attack most simple-styled traffic signs, the source image would change the validity of the attack, as shown in Figure 13, forward (2) paste "ST," would still get recognized as "forward," while left (4) paste "ST," would get recognized as "stop," forward (3) paste complex style CPAs, would get recognized as "left," and left (4) pastes "ST," which is recognized as "stop." Therefore, some simple-styled traffic signs could not be attacked successfully with simple CPAs, but complex CPAs (complex stickers with multiple CPAs overlaid on top of each other) could successfully attack them.

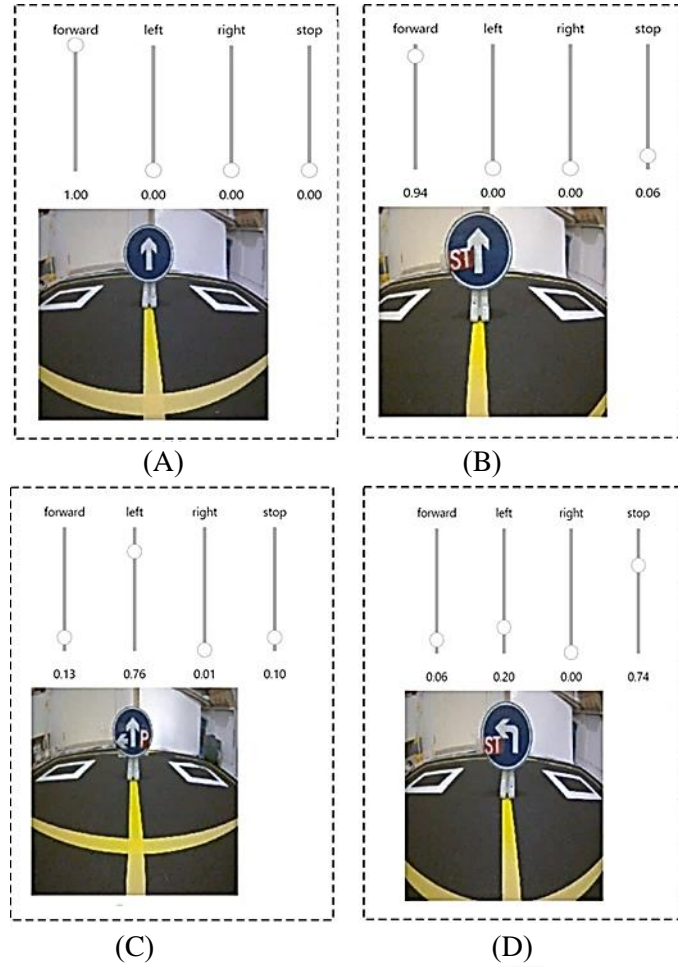


Figure 13 (A) add forward (a) without interference, (B) add forward (a) for simple style CPAs, (C) add forward (a) for complex style CPAs, and (D) add left (a) for the same simple style CPAs as (2).

5. Conclusions

This paper proposed and demonstrated a real and effective adversarial attack: the copy-paste attack (CPA). CNN models were deployed on an intelligent model car for experiments in physical environments. We tested three well-known CNN models, on 3 different kinds of Datasets. The experimental results demonstrate that our proposed CPAs greatly interfere with the recognition rate of traffic signs. CPA can misguide the intelligent model car into wrong behaviors. We trained the CNN models with the physical defense method of adversarial training; however, it had little effect on CPA attacks. Moreover, we compared our CPAs with the existing method PR2, Resnet18 can achieve a high accuracy of 81% on PR2, while only achieving an accuracy of 43% on our CPA attacks. Furthermore, we tested one of our previous ResNet26-CBAM model, which has been modified to pay more attention to CPA, although it exhibits higher robustness against the CPA attack compared with other well-known CNN models, our ResNet26-CBAM also got misguided by CPAs with an accuracy of 60%. The research can be used in the future to further optimize adversarial training methods, explore other types of physical adversarial attacks, and possibly validate these attacks and defense methods on real autonomous vehicles.

6. Acknowledgments

Conceptualization, CX. B. and J. Q.; methodology, CX. B. and J. Q.; software, CX. B. and J. Q.; validation, CX. B. and J. Q.; formal analysis, CX. B. and J. Q.; investigation, CX. B. and J. Q.; data curation, CX. B. And J. Q.; writing—original draft preparation, CX. B. and J. Q.; writing—review and editing, CX. B. and J.Q.; visualization, CX. B. and J. Q.; supervision, J. Q.; All authors have read and agreed to the published version of the manuscript. J. Q. is the corresponding author.

7. References

- [1] Aldahdooh A, Hamidouche W, Fezza SA, Déforges O. Adversarial example detection for DNN models: A review and experimental comparison. *Artif Intell Rev.* 2022;55(6):4403–62.
- [2] Ni J, Chen Y, Chen Y, Zhu J, Ali D, Cao W. A survey on theories and applications for self-driving cars based on deep learning methods. *Appl Sci.* 2020;10(8):2749.
- [3] Li YW, Qu J. Intelligent road tracking and real-time acceleration-deceleration for autonomous driving using modified convolutional neural networks. *Curr Appl Sci Technol.* 2022 ID 10-55003.
- [4] Ding S, Qu J. Research on multi-tasking smart car based on autonomous driving systems. *SN Comput Sci.* 2023;4(3):292.
- [5] Du M. Mobile payment recognition technology based on face detection algorithm. *Concurr Comput Pract Exp.* 2018;30(22): e4655.
- [6] Hinton G, Deng L, Yu D, Dahl GE, Mohamed AR, Jaitly N, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Process Mag.* 2012;29(6):82–97.
- [7] Carlini N, Wagner D. Towards evaluating the robustness of neural networks. In: *Proc IEEE Symp Secur Privacy.* 2017. p. 39–57.
- [8] Kos J, Fischer I, Song D. Adversarial examples for generative models. In: *Proc IEEE Secur Privacy Workshops.* 2018. p. 36–42.
- [9] Finlayson SG, Bowers JD, Ito J, Zittrain JL, Beam AL, Kohane IS. Adversarial attacks on medical machine learning. *Science.* 2019;363(6433):1287–1289.
- [10] Dong Y, Liao F, Pang T, Su H, Zhu J, Hu X, et al. Boosting adversarial attacks with momentum. In: *Proc IEEE Conf Comput Vis Pattern Recognit (CVPR).* 2018. p. 9185–9193.
- [11] Moosavi-Dezfooli SM, Fawzi A, Frossard P. DeepFool: A simple and accurate method to fool deep neural networks. In: *Proc IEEE Conf Comput Vis Pattern Recognit (CVPR).* 2016. p. 2574–2582.
- [12] Eykholt K, Evtimov I, Fernandes E, Li B, Rahmati A, Xiao C, et al. Robust physical-world attacks on deep learning visual classification. In: *Proc IEEE Conf Comput Vis Pattern Recognit (CVPR).* 2018. p. 1625–1634.
- [13] Duan R, Ma X, Wang Y, Bailey J, Qin AK, Yang Y. Adversarial camouflage: Hiding physical-world attacks with natural styles. In: *Proc IEEE/CVF Conf Comput Vis Pattern Recognit (CVPR).* 2020. p. 1000–1008.
- [14] Kurakin A, Goodfellow IJ, Bengio S. Adversarial examples in the physical world. In: *Artif Intell Saf Secur.* 2018. p. 99–112.
- [15] Chen ST, Cornelius C, Martin J, Chau DH. Shapeshifter: Robust physical adversarial attack on Faster R-CNN object detector. In: *Proc Eur Conf Comput Vis (ECCV) Workshops.* 2018. p. 10–14.
- [16] Akhtar N, Mian A. Threat of adversarial attacks on deep learning in computer vision: A survey. *IEEE Access.* 2018; 6:14410–14430.
- [17] Bi C, Shi S, Qu J. Enhancing autonomous driving: A novel approach of mixed attack and physical defense strategies. *ASEAN J Sci Technol Rep.* 2025;28(1): e254093.
- [18] Evtimov I, Eykholt K, Fernandes E, Kohno T, Li B, Prakash A, et al. Robust physical-world attacks on machine learning models. *arXiv.* 2017; arXiv:1707.08945.
- [19] Akhtar N, Mian A, Kardan N, Shah M. Advances in adversarial attacks and defenses in computer vision: A survey. *IEEE Access.* 2021; 9:155161–155196.