Adversarial Patch Attack on Traffic Sign Recognition of Automotive

CHUN-CHIA, HUANG
rXXXXXXXXcsie.ntu.edu.tw
National Taiwan University, Taiwan

TZU-MIN, YANG
rXXXXXXXX@csie.ntu.edu.tw
National Taiwan University, Taiwan

CHING, LO
rXXXXXXXX@csie.ntu.edu.tw
National Taiwan University, Taiwan

Abstract

Object classification is essential for autonomous vehicles (AVs), enabling them to make decisions like obstacle avoidance and speed regulation. This study investigates the vulnerability of AVs' traffic sign recognition (TSR) systems to adversarial attacks. We reproduce various patch attacks within a digital environment, evaluate them using our own Taiwan dataset, and review several defense methods. Our focus is on three types of stealthy, cost-effective attacks: retro-reflective patches, shadow-like patches, and light-reflection attacks. These attacks are simple to implement and difficult to detect. We assess their effectiveness under different conditions and test various TSR models. Our findings reveal that model capacity is critical to resistance, and a larger patch size does not necessarily correlate with increased attack efficiency.

This is a combination type project, including a research on multiple adversarial patch attacks and a brief survey of some defense methods. The main contributions of this project are:

- Study the effectiveness of three types of stealthy, cost-effective adversarial patch attacks against several TSR models
- Evaluate the proposed attacks on Taiwanese environment
- Survey some defense methods for adversarial patch attacks
- Open source the Taiwanese dataset we collected on GitHub

1 Introduction

The rapid development of autonomous vehicles (AVs) has the potential to revolutionize transportation, offering benefits such as enhanced safety, improved traffic efficiency, and reduced environmental impact. A critical component of AVs is their ability to accurately classify objects in their surroundings, which enables them to perform essential functions like obstacle avoidance and speed regulation. Among these capabilities, traffic sign recognition (TSR) plays a vital role, as it ensures that AVs can correctly interpret and respond to traffic signals, thereby adhering to road regulations and maintaining safety.

Despite significant advancements in TSR technology, these systems remain vulnerable to adversarial attacks. Adversarial attacks involve subtle modifications to input data that can deceive machine learning models, leading to incorrect classifications. Such attacks on TSR systems can cause AVs to misinterpret traffic signs, potentially resulting in dangerous driving decisions. The vulnerability of TSR systems to these attacks necessitates a thorough investigation to develop more robust and reliable AV technologies.

In this study, we focus on three specific types of adversarial attacks that are not only stealthy but also cost-effective and easy to implement: retro-reflective patches (Tsuruoka et al., 2023), shadow-like patches (Sato et al., 2023), and light-reflection attacks (Zhong et al., 2022). These methods exploit various aspects of TSR models, making them difficult to detect and mitigate. Our research aims to assess the potential severity of these attacks and explore the resilience of various TSR models under different conditions.

To understand the extent of the threat posed by these adversarial attacks, we conducted experiments using a dataset of Taiwanese traffic signs. Through our experiments, we aim to identify the factors that influence the robustness of TSR models and to propose potential defense mechanisms. By addressing these vulnerabilities, we hope to contribute to the development of more secure and reliable AV technologies.

To facilitate replication and further research, we have made our implementation publicly available at this github repository.

2 Related Work

2.1 Adversarial Retroreflective Patches Attack

Adversarial Reflective Patch (ARP) is a black-box attack introduced by Tsuruoka et al. (Tsuruoka et al., 2023). ARP employs reflective patches that are either transparent or blend with the background, making them nearly undetectable to humans, traffic sign detection, and classification systems during the day. Only at night, when the headlights of vehicles shine on them, these patches reveal their adversarial patterns, initiating the attack. The texture of these patches is specially designed to be retroreflective, meaning they reflect light back to its source. This unique property enables the ARP to create deceptive patterns that only become visible under nighttime illumination, effectively confusing detection and classification systems while remaining hidden during daylight hours.

2.2 Stealthy and Effective Physical-world Adversarial Attack by Natural Phenomenon: Shadow

Zhong et al. studied the use of shadows as perturbations in adversarial attacks against TSR models, leveraging this common natural phenomenon for its stealthiness in real-world environments (Zhong et al., 2022). The proposed algorithms calculate the optimal location and color for triangular shadows, which are then synthesized onto images. The attack was evaluated in both digital and physical domains, achieving high success rates in both settings. However, this approach has two main limitations. First, it requires a strong, single light source; otherwise, it may struggle to generate significant shadows. Second, conducting explicit targeted attacks is difficult because the perturbation directions generated by shadows are relatively uniform, making precise targeted misclassifications challenging.

2.3 Infrared Laser Reflection Attack

The proposed attack in the paper (Sato et al., 2023) utilizes Infrared Laser Reflection (ILR) to create a stealthy and effective method for deceiving traffic sign recognition systems in autonomous vehicles. The ILR attack exploits the fact that while infrared (IR) light is invisible to the human eye, it is detectable by cameras that lack IR filters. By projecting an IR laser onto traffic signs, the attack creates a reflection that alters the sign's appearance to the camera, leading to misclassification. This makes the attack stealthy, as the changes are not visible to human drivers, thus making it difficult to detect and counteract. The attack is performed by strategically positioning an IR laser emitter to illuminate specific parts of a traffic sign. The experimental results indicate the attack is highly effectiveness and may post risk for real-world scenarios.

3 Methodology

3.1 Retro-reflective Patch Attack

3.1.1 Threat Model

We follow the same threat model as the prior research (Tsuruoka et al., 2023). This is a black-box attack, thus the attackers have no access to the internal architecture and parameters of the TSR models in the AVs, but they can obtain the output of the TSR models with arbitrary images and the information of the camera and headlights required for performing the attack. This requirement might be easily satisfied by analyzing an AV of the same kind. We assume that the attackers can apply a to the traffic signs, and the patch used is white and has a highly retro-reflective texture with zero to some level of transparency. Therefore, this attack can be very stealthy both at day and night due to the transparency and only be enabled by the headlights of the victim AVs.

3.1.2 Goal of Attack

The goal of the attackers is to affect the TSR models in the AVs, causing the AVs to classify the traffic sign as the wrong class. The steps to perform the attack can be described as follows: **Step 1.** Get access to an AV of the same kind. **Step 2.** Find the optimal combination of size, placement, and transparency of the patches. **Step 3.** Apply the patch to the traffic sign. **Step 4.** Wait for the victim AVs to pass by the patched traffic sign (Tsuruoka et al., 2023).

3.2 Shadow-Like Patch Attack

3.2.1 Threat Model

We use a similar threat model as the one described in the previous section. The main difference is that the patch used is partially transparent with black color and the texture is not reflective. This setup is similar to the related work (Zhong et al., 2022). The patch can be very stealthy during day since it looks like a shadow and can be hard to identify in a glimpse, and it can be even more stealthy at

night due to shadows being much more common at night.

3.2.2 Goal of Attack

The goal of the attackers and the steps to perform the attack are the same as described in the previous section (Zhong et al., 2022).

3.3 Light-Reflection Attack

3.3.1 Threat Model

We use a similar threat model as the one described in the previous section. The main difference is that instead of applying patches, the attackers direct a high-power light beam at the traffic sign to create a reflective area. This setup is similar to the related work (Sato et al., 2023). The attack can be completely invisible by using light at a frequency that is detectable by the AVs' cameras but invisible to the human eye.

3.3.2 Goal of Attack

The goal of the attackers is the same as described in the previous section. The steps to perform the attack are: **Step 1.** Get access to an AV of the same kind. **Step 2.** Find the optimal power of the light beam and the placement of the reflection on the traffic sign. **Step 3.** Direct the light beam to the traffic sign when the victim AVs is passing by (Sato et al., 2023).

3.4 Optimization of Size, Placement, and Transparency

To get the best chance of success while keeping the attack simple and easy to implement, we only discuss three properties of the patch and the light beam: size, placement, and transparency. The size of the patch or light beam is proportional to the size of the traffic sign, such as one-fifth the width and one-fifth the height of the traffic sign. For the results to be easily comparable, we choose the same setup in the inspired research (Tsuruoka et al., 2023), one-fifth the width and one-tenth the height, as the default when conducting the experiments. Regarding the placement, the area of the traffic sign is divided into multiple grids by the size of the patch or light beam, then we apply the patch or direct the light beam at each given grid one at a time, as shown in Figure 1.

To reduce the combinations of the properties, we assume the effect of transparency is independent of the placement. The optimal placement of a given size of patch or light beam is first obtained, then we apply different levels of transparency or power to the patch or light beam to examine the effectiveness of the attack.

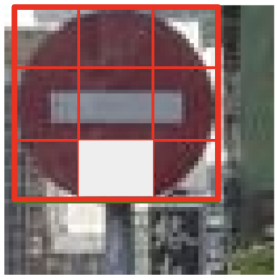

Figure 1: A patch or light beam is applied to one of the multiple grid in the image.

4 Evaluation

To evaluate the effectiveness of the attacks, we trained four different TSR models on a customized GTSRB dataset(Stallkamp et al., 2012), and then performed the attacks on the Taiwanese "No Entry" signs. The "No Entry" signs are chosen because they are the most easily accessible safety-critical traffic signs around the NTU campus.

4.1 Research Questions

Our evaluation addresses the following research questions:

- **RQ1.** How effective is the retro-reflective patch attack in daytime and nighttime conditions?
- **RQ2.** What is the optimal patch size in the retro-reflective patch attack?
- **RQ3.** What is the optimal patch placement in the retro-reflective patch attack?
- **RQ4.** How effective is the shadow-like patch attack in daytime and nighttime conditions?
- **RQ5.** What is the optimal patch size in the shadow-like patch attack?
- **RQ6.** What is the optimal patch placement in the shadow-like patch attack?
- **RQ7.** What is the optimal transparency of the patch in the shadow-like patch attack?
- **RQ8.** How effective is the light-reflection attack in daytime and nighttime conditions with different level of transparency of the reflection?
- **RQ9.** What is the optimal size of the reflection area in the light-reflection attack?

TSR Model	Original Accuracy
CNN	98.21%
DeepCNN	98.11%
ResNet	99.03%
VGG	97.71%

Table 1: The accuracy of TSR models after training or fine-tuning

Figure 2: The patch or light beam applied to the image for retro-reflective patch attack, shadow-like attack, and light-reflection attack (from left to right).

4.2 Dataset and TSR Model

We captured a total of 1,933 images of Taiwanese "No Entry" signs at two different locations. The images were captured in various settings: during day and night, at facing angles of 0, 30, and 60 degrees, as shown in Fig [facing angle], and from distances of 5m, 7.5m, and 10m. The camera was positioned at a height of approximately 160cm, which is similar to the height of the front camera on a Tesla Model Y.

To build the TSR models that can recognize Taiwanese "No Entry" signs, we created a customized GTSRB dataset (Stallkamp et al., 2012). In this dataset, all images of German 'No Entry' signs from the original GTSRB dataset have been replaced with the Taiwanese ones we captured. We then trained two CNN models from scratch on this customized dataset, named CNN and DeepCNN. DeepCNN has more convolutional layers than CNN. The detailed architecture of CNN and DeepCNN models are provided in Appendix A. We also fine-tuned a ResNet18 and a VGG16 , both provided by PyTorch Hub and pre-trained on the ImageNet-1K dataset (Deng et al., 2009), using our customized GTSRB dataset. The accuracy of them are listed in Table 1.

4.3 Experiment Design

4.3.1 Digital Experiment

In the digital experiment, 98 images of a Taiwanese "No Entry" sign captured at a distance of 5m, facing an angle of 0 degrees are chosen. 44 of them are captured in the morning, and the rest are at night. We labeled the area of the traffic sign in all the images manually, so the size and placement can be calculated precisely.

To simulate the retro-reflective patch attack, we assume the reflection is white with no transparency. For the shadow-like patch attack, we assume the patch is black with 0 to 80% transparency. To simulate the light-reflection attack, we assume the reflection of the light beam is white with 20 to 80% transparency to mimic different light intensities. We then apply these patches using image processing. The patch or light beam applied to the image for each attack are shown in Fig 2.

4.3.2 Real-world Experiment (WIP)

In the real-world experiment, we planned to apply a patch or direct a light beam that has similar properties to the setup in the digital experiment to the same Taiwanese "No Entry" sign, take pictures of the patched sign, and input the pictures to the TSR models. The details of the real-world experiment design are still working in progress.

4.4 Results

4.4.1 Retro-reflective Patch Attack

- **RQ1: Effectiveness of Retro-reflective Patch Attack at Day & Night**: The results of daytime and nighttime are shown in Table 2 and Table 3. The largest accuracy drop among all placements, given the patch sizes of 1/10 and 1/5 in height and width of the traffic sign, is 92.59% on the CNN model under nighttime conditions. The overall results indicate that the attack is slightly more effective at night. Additionally, the larger, well-established models demonstrate greater robustness and are able to resist the attack.
- **RQ2: Optimal Size of Patch in Retro-reflective Patch Attack**: The results are shown in Table 4. The lowest accuracy among all possible placements with the given patch size is 1.85% for DeepCNN under nighttime conditions. According to the results, it is evident that larger patches result in more effective attacks. Meanwhile, ResNet demonstrates resistance to the attack.
- **RQ3: Optimal Placement of Patch in Retro-reflective Patch Attack**: We chose DeepCNN at night as the study subject because ResNet demonstrated great resistance, making it less

Model	Accuracy Before Attack	Accuracy After Attack	Accuracy Drop
CNN	100.00%	38.64%	61.36%
DeepCNN	100.00%	77.27%	22.73%
ResNet	100.00%	100.00%	0.00%
VGG	100.00%	100.00%	0.00%

Table 2: Accuracy of different TSR models before and after retro-reflective patch attack in daytime

Model	Accuracy Before Attack	Accuracy After Attack	Accuracy Drop
CNN	100.00%	7.41%	92.59%
DeepCNN	100.00%	79.63%	20.37%
ResNet	100.00%	100.00%	0.00%
VGG	100.00%	100.00%	0.00%

Table 3: Accuracy of different TSR models before and after retro-reflective patch attack in nighttime

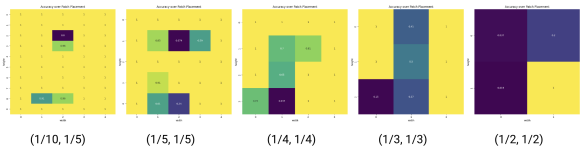

Figure 3: Accuracy of DeepCNN when a patch with given size was applied to each location.

effective for observing the impact of different placements. CNN performed too poorly, and VGG appeared to have overfitted on the Taiwanese dataset. Additionally, nighttime is the ideal period for performing the attack for better stealth. The result is shown in Figure 3.

The optimal placement of the patch of different sizes is not consistent, thus preventing us from drawing explicit conclusions for this research question.

In summary, the experiments reveal that the retro-reflective patch attack is more effective at night, and larger patches tend to increase the effectiveness of the attack. However, larger, well-established models like ResNet show greater robustness and resistance to the attack, indicating that the attack requires more elaborate adjustments to be effective in real-world scenarios.

4.4.2 Shadow-Like Patch Attack

- **RQ4: Effectiveness of Shadow-like Patch Attack at Day & Night**: The results of daytime and nighttime are shown in Table 5 and Table 6. The transparency of the patch was set to 20% in the experiments. The largest accuracy drop among all placements, given the patch sizes of 1/10 and 1/5 in height and width of the traffic sign, is 33.34% on the CNN model under nighttime conditions. This result contradicts our expectations. We suspect that this is because CNN models are particularly vulnerable to this type of attack. In contrast, ResNet showed no accuracy drop at night but did experience a 2.27% drop during the daytime. Larger, well-established models demonstrate greater robustness and are able to resist the attack more effectively.
- **RQ5: Optimal Size of Patch in Shadow-Like Patch Attack**: The results are shown in Table 7. The transparency of the patch was set to 20% in the experiments. The lowest accuracy among all possible placements with the given patch size is 74.07% for DeepCNN under nighttime conditions. The patch size of 1/3 in height and 1/3 in width of the traffic sign is the most effective for the attack across all the settings we studied.
- **RQ6: Optimal Placement of Patch in Shadow-Like Patch Attack**: We chose DeepCNN at day as the study subject due to the similar reasons described in RQ3. Dayttime is the ideal period for performing the attack since the shadow used in the attack might not be significant in nighttime (Zhong et al., 2022). The result is shown in Figure 4. The result of the experiment indicates that the center of the traffic sign is the optimal placement for this attack.
- **RQ7: Optimal Transparency of Patch in Shadow-Like Patch Attack**: Given the optimal placement of each patch size, in all settings, as the patch becomes less transparent,

Model	(1/10, 1/5)	(1/5, 1/5)	(1/4, 1/4)	(1/3, 1/3)	(1/2, 1/2)
DeepCNN-day	77.27%	18.18%	47.73%	52.27%	**11.36%**
DeepCNN-night	79.63%	7.41%	3.70%	14.81%	**1.85%**
ResNet-day	100.00%	100.00%	100.00%	**97.73%**	100.00%
ResNet-night	100.00%	100.00%	100.00%	100.00%	100.00%

Table 4: TSR Model accuracies under retro-reflective patch attack with different patch sizes

Model	Accuracy Before Attack	Accuracy After Attack	Accuracy Drop
CNN	100.00%	81.82%	18.18%
DeepCNN	100.00%	100.00%	0.00%
ResNet	100.00%	97.73%	2.27%
VGG	100.00%	100.00%	0.00%

Table 5: Accuracy of different TSR models before and after shadow-like patch attack in daytime

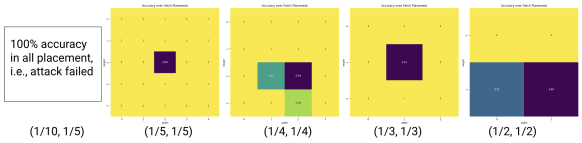

Figure 4: Enter Caption

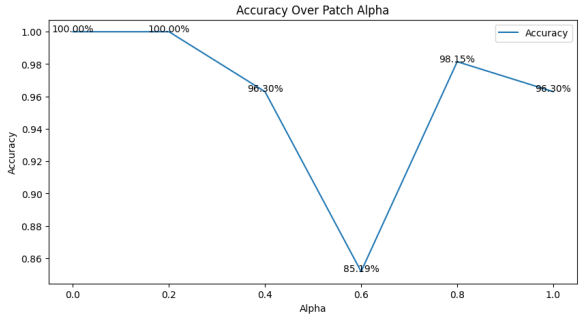

Figure 5: ResNet model accuracies over patch transparency after shadow patch attack in the night with (1/3, 1/3) patch size and optimal placement

the model accuracy decreases. Except in the ResNet-night setting with a patch size of 1/3 in both height and width of the traffic sign, the accuracy over patch transparency follows a U-shaped pattern, as shown in Figure 5. We don't have an explanation for this result.

In summary, the experiments reveal that the shadow-like patch attack is more effective at night. The optimal patch size and placement for the attack is 1/3 in height and 1/3 in width, positioned at the center of the traffic sign. As the patch becomes less transparent, model accuracy generally decreases, with the exception of a U-shaped pattern in one setting, for which we don't have a reasonable explanation. Overall, the attack was less effective than the retro-reflective patch attack.

4.4.3 Light-Reflection Attack

- **RQ8: Effectiveness of Light-Reflection Attack in Day & Night with Different Transparency of the Reflection**: Given the size of the reflection area as 1/5 in height and 1/10 in width of the traffic sign with different transparency of the light beam, the results are shown in Table 8. The worst accuracy among all placements is 86.36% for DeepCNN in daytime. The attack becomes more effective as the light intensity increases. Although the attack is shown to be more effective at night, it becomes less stealthy unless invisible light, such as Infrared Laser, is used as demonstrated in the paper by Sato et al. (Sato et al., 2023).
- **RQ9: Optimal Size of Reflection in Light-Reflection Attack**: The transparency of the light beam is set to 60% in this experiment. The results are shown in Table 9. The worst accuracy among all placements, given the reflection area size of 1/10 in height and 1/5 in width of the traffic sign, is 51.85% for DeepCNN at night. The experiment does not provide conclusive results regarding the optimal size for the attack.

In summary, the experiments reveal that the attack becomes more effective as the light intensity increases. The attack is particularly effective at night; however, it becomes less stealthy unless invisible light, such as Infrared Laser, is used. The optimal size of the attack cannot be determined from the experiments.

Model	Accuracy Before Attack	Accuracy After Attack	Accuracy Drop
CNN	100.00%	66.67%	33.34%
DeepCNN	100.00%	100.00%	0.00%
ResNet	100.00%	100.00%	0.00%
VGG	100.00%	100.00%	0.00%

Table 6: Accuracy of different TSR models before and after shadow-like patch attack in nighttime

Model	(1/10, 1/5)	(1/5, 1/5)	(1/4, 1/4)	(1/3, 1/3)	(1/2, 1/2)
DeepCNN-day	100.00%	100.00%	95.45%	97.73%	**90.91%**
ResNet-day	100.00%	97.73%	100.00%	**95.45%**	100.00%
DeepCNN-night	100.00%	100.00%	94.44%	**74.07%**	100.00%
ResNet-night	100.00%	100.00%	100.00%	**98.15%**	100.00%

Table 7: TSR Model accuracies under shadow-like patch attack with different patch sizes

5 Discussion

5.1 Attacking Model Is Not Equal to Attack AVs' Behavior

Even if we successfully attack the classifier model, it does not mean the entire AD system has been compromised.(Wang et al., 2023) In fact, if we do not consider attacking the entire system from the beginning and only target the model, it will hardly affect the whole autonomous driving system. The main issue is that the object tracking module can also affect the behavior of the AD system.

5.2 Defense Methods

5.2.1 Data pre-processing

Pre-processing the data before feeding it into the model can reduce the impact of perturbations.

- SHIELD–JPEG compression: This method uses JPEG compression to "compress away" perturbations from the input image. The classification model is then trained with these compressed images to ensure it learns to be resilient to the types of perturbations removed by compression (Das et al., 2018).

- DEFENSE-GAN: This approach involves using a generative adversarial network to randomly generate images and find one that closely resembles the input image. The generated image is then fed into the model, helping to mitigate the impact of adversarial attacks by filtering out adversarial perturbations (Pouya, 2018).

5.2.2 Model hardening

Making the model more robust by changing the model architecture or using adversarial training, which involves training the model with perturbed input data to improve its ability to defend against adversarial attacks.

Adversarial training with Projected Gradient Descent (PGD) is considered the state-of-the-art practical white-box defense. This method involves first generating adversarial examples using PGD, and then using these examples to train the model to enhance its robustness against such attacks (Madry et al., 2017). However, because PGD itself is computationally intensive and slow, adversarial training with PGD also tends to be slow. Despite this, PGD is recognized as the state-of-the-art practical method for adversarial attacks, known for its effectiveness despite not being theoretically proven.

6 Future Work

Our study has highlighted the importance of four key areas that warrant further investigation to enhance the robustness of TSR of AVs.

6.1 Impact of Dataset Size

The current research, including our own, has primarily utilized the GTSRB dataset, which contains only 40,000 images. We hypothesize that the size and diversity of the dataset may impact the robustness of models.

However, due to the lack of alternative datasets, we could not test this hypothesis. Future research should explore the effects of larger and more diverse datasets on model robustness, potentially by creating new datasets or augmenting existing ones.

Model	100%	80%	60%	40%	20%
DeepCNN-day	100.00%	100.00%	100.00%	100.00%	**86.36%**
DeepCNN-night	100.00%	100.00%	98.15%	96.30%	**94.44%**

Table 8: TSR Model accuracies under light-reflection attack with different transparency of the reflection

Model	(1/10, 1/5)	(1/5, 1/5)	(1/4, 1/4)	(1/3, 1/3)	(1/2, 1/2)
DeepCNN-day	100.00%	90.91%	100.00%	**81.82%**	90.01%
DeepCNN-night	98.15%	24.07%	55.56%	83.31%	**51.85%**

Table 9: TSR Model accuracies under light-reflection attack with different patch sizes

6.2 Impact of Model Capacity

Multiple experiments in this paper have shown that ResNet is much more robust than DeepCNN and CNN, with DeepCNN also proving to be more robust than CNN. Since both CNN models were trained on the same training dataset and evaluated on the same evaluation dataset, the only difference between the two models is their capacity. These results indicate that the capacity of a model can significantly impact its robustness. Some related works, such as those by Sato et al. (Sato et al., 2023) and Zhong et al. (Zhong et al., 2022), only use CNN models to evaluate their proposed attack methods. We argue that using only CNN models for evaluation is not sufficient to demonstrate the real-world impact of these attacks. Therefore, we advise future research to include more deep, well-established image recognition models, like ResNet and ViT, to better demonstrate their impact on real-world scenarios.

6.3 Building a Realistic Dataset using 3D Ray Tracing Environments

To create a realistic dataset, we used Blender to construct a 3D ray tracing environment. In Blender, we modeled a "No Entry" traffic sign and incorporated background lighting using an HDRI light obtained online. We have control over the camera's angle, distance, and environmental lighting. The rendering outcome is depicted in Figure 6.

After completing the modeling, we need to decide how to apply the patch. However, we have not yet determined the placement of the patch. Additionally, we have not found a suitable texture for the patch that can effectively represent the reflective effect. For future work, we may optimize the patch placement and find the appropriate texture for the patch to create a realistic dataset that fits our experiment.

Figure 6: A "No Entry" traffic sign rendered using Blender's 3D modeling and rendering capabilities.

6.4 Real-World Experimentation

While our study has been conducted in a controlled digital environment, real-world experiments are crucial for validating the findings. We are in the process of designing experiments to test patch adversarial attacks in real-world settings.

The planned approach involves applying tape or using light beams to simulate attacks on the same Taiwanese "No Entry" sign used in the evaluation, capturing images of the attacked sign, then testing the robustness of TSR models against these real-world adversarial examples. This will provide insights into the practical challenges and effectiveness of such attacks outside of a simulated environment.

By addressing these areas, future research can significantly advance the field of adversarial attack mitigation in traffic signal recognition, ultimately contributing to the development of more secure and reliable AVs technologies.

7 Conclusion

In our study, we reproduced and developed three patch adversarial attacks on various models within a digital environment and evaluated several attack parameters, such as patch size, transparency, and

placement.

Our findings indicate that a larger patch size does not necessarily result in a more effective attack. Interestingly, attacks tend to be more effective at night. We also observed that larger, well-established models demonstrate greater robustness and can resist these attacks more effectively.

The capacity of a model significantly impacts its robustness; if a model's capacity is insufficient, it cannot learn the correct, complex classification boundaries. This is evidenced by our experimental results comparing CNN and DeepCNN models. DeepCNN, which has more layers than a standard CNN, proved to be more robust in our tests.

These conclusions underscore the importance of various factors in enhancing resistance to adversarial attacks in TSR for AVs. Model capacity is crucial, but other elements such as model architecture, and training methodologies also play significant roles. This holistic approach can lead to the development of more secure and resilient AI systems in the autonomous driving domain.

Author Contributions

All authors have made significant contributions to this project. The specific contributions are as follows:

- **CHUN-CHIA, Huang**: Surveying attack methods, implementing the TSR models, implementing and evaluating the attacks, preparing slides, and writing the report.
- **TZU-MIN, YANG**: Collection and preprocessing of Taiwanese real-world dataset, surveying defense methods, preparing presentations and slides, and writing the report.
- **CHING, LO**: Collectoin of Taiwanese real-world datasetn, surveying TSR models, preparing slides and presentations, building traffic signs in 3D ray tracing environments using Blender and writing the report.

8 Acknowledgement

This research is inspired by the paper (Tsuruoka et al., 2023). It was planned to first partially reproduce the results of the paper (Tsuruoka et al., 2023) in a Taiwanese setting, then develop and expand the research further to better control the scope as the research progresses. Consequently, this research bears some similarities in design to the paper (Tsuruoka et al., 2023). It is crucial to understand the development of this research before criticizing it as plagiarism, as we do not intend to, nor do we, plagiarize the research.

Additionally, this report does not involve any double assignment or include any completed work from before this semester.

References

N. Das, M. Shanbhogue, S. T. Chen, F. Hohman, S. Li, L. Chen, and D. H. Chau. 2018. Shield: Fast, practical defense and vaccination for deep learning using jpeg compression. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 196–204.

J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, and L. Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255. IEEE.

A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. 2017. Towards deep learning models resistant to adversarial attacks. https://arxiv.org/abs/1706.06083. ArXiv preprint arXiv:1706.06083.

S. Pouya. 2018. Defense-gan: protecting classifiers against adversarial attacks using generative models. https://arxiv.org/abs/1805.06605. Retrieved from https://arXiv: 1805.06605.

T. Sato, S. H. Bhupathiraju, M. Clifford, T. Sugawara, Q. A. Chen, and S. Rampazzi. 2023. Wip: Infrared laser reflection attack against traffic sign recognition systems. In *ISOC Symposium on Vehicle Security and Privacy (VehicleSec)*.

J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel. 2012. Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. *Neural Networks*, 32:323–332.

G. Tsuruoka, T. Sato, Q. A. Chen, K. Nomoto, R. Kobayashi, Y. Tanaka, and T. Mori. 2023. Poster: Adversarial retroreflective patches: A novel stealthy attack on traffic sign recognition at night. In *Proceedings of the Conference*. Poster presentation.

N. Wang, Y. Luo, T. Sato, K. Xu, and Q. A. Chen. 2023. Does physical adversarial example really matter to autonomous driving? towards system-level effect of adversarial object evasion attack. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4412–4423.

Y. Zhong, X. Liu, D. Zhai, J. Jiang, and X. Ji. 2022. Shadows can be dangerous: Stealthy and effective physical-world adversarial attack by natural phenomenon. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15345–15354.

A Model Architecture of CNN and DeepCNN

```
class CNN(nn.Module):
    def __init__(self, num_classes=43):
        super(CNN, self).__init__()
        self.conv1 = nn.Conv2d(3, 32,
            kernel_size=3, stride=1,
            padding=1)
        self.conv2 = nn.Conv2d(32, 64,
            kernel_size=3, stride=1,
            padding=1)
        self.pool = nn.MaxPool2d(
            kernel_size=2, stride=2,
            padding=0)
        self.fc1 = nn.Linear(64 * 16 *
            16, 512)
        self.fc2 = nn.Linear(512, 128)
        self.fc3 = nn.Linear(128,
            num_classes)

    def forward(self, x):
        x = self.pool(torch.relu(self.
            conv1(x)))
        x = self.pool(torch.relu(self.
            conv2(x)))
        x = x.view(-1, 64 * 16 * 16)
        x = torch.relu(self.fc1(x))
        x = torch.relu(self.fc2(x))
        x = self.fc3(x)
        return x

class DeepCNN(nn.Module):
    def __init__(self, num_classes=43):
        super(DeepCNN, self).__init__()
        self.conv1 = nn.Conv2d(3, 32,
            kernel_size=3, stride=1,
            padding=1)
        self.conv2 = nn.Conv2d(32, 64,
            kernel_size=3, stride=1,
            padding=1)
        self.conv3 = nn.Conv2d(64, 128,
            kernel_size=3, stride=1,
            padding=1)
        self.conv4 = nn.Conv2d(128, 256,
             kernel_size=3, stride=1,
            padding=1)
        self.pool = nn.MaxPool2d(
            kernel_size=2, stride=2,
            padding=0)
        self.fc1 = nn.Linear(256 * 4 *
            4, 512)
        self.fc2 = nn.Linear(512, 128)
        self.fc3 = nn.Linear(128,
            num_classes)

    def forward(self, x):
        x = self.pool(torch.relu(self.
            conv1(x)))
        x = self.pool(torch.relu(self.
            conv2(x)))
        x = self.pool(torch.relu(self.
            conv3(x)))
        x = self.pool(torch.relu(self.
            conv4(x)))
        x = x.view(-1, 256 * 4 * 4)
        x = torch.relu(self.fc1(x))
        x = torch.relu(self.fc2(x))
        x = self.fc3(x)
        return x
```

Listing 1: Model Architecture of CNN and DeepCNN