electronics

MDPI

Article

Fast and Lightweight Vision-Language Model for Adversarial Traffic Sign Detection

Furkan Mumcu and Yasin Yilmaz *

Department of Electrical Engineering, University of South Florida, Tampa, FL 33620, USA; furkan@usf.edu
* Correspondence: yasiny@usf.edu

Abstract: Several attacks have been proposed against autonomous vehicles and their subsystems that are powered by machine learning (ML). Road sign recognition models are especially heavily tested under various adversarial ML attack settings, and they have proven to be vulnerable. Despite the increasing research on adversarial ML attacks against road sign recognition models, there is little to no focus on defending against these attacks. In this paper, we propose the first defense method specifically designed for autonomous vehicles to detect adversarial ML attacks targeting road sign recognition models, which is called ViLAS (Vision-Language Model for Adversarial Traffic Sign Detection). The proposed defense method is based on a custom, fast, lightweight, and salable vision-language model (VLM) and is compatible with any existing traffic sign recognition system. Thanks to the orthogonal information coming from the class label text data through the language model, ViLAS leverages image context in addition to visual data for highly effective attack detection performance. In our extensive experiments, we show that our method consistently detects various attacks against different target models with high true positive rates while satisfying very low false positive rates. When tested against four state-of-the-art attacks targeting four popular action recognition models, our proposed detector achieves an average AUC of 0.94. This result achieves a 25.3% improvement over a state-of-the-art defense method proposed for generic image attack detection, which attains an average AUC of 0.75. We also show that our custom VLM is more suitable for an autonomous vehicle compared to the popular off-the-shelf VLM and CLIP in terms of speed (4.4 vs. 9.3 milliseconds), space complexity (0.36 vs. 1.6 GB), and performance (0.94 vs. 0.43 average AUC).

Keywords: adversarial machine learning; autonomous vehicle security; vision-language models; multimodal learning

Citation: Mumcu, F.; Yilmaz, Y. Fast and Lightweight Vision-Language Model for Adversarial Traffic Sign Detection. *Electronics* **2024**, *13*, 2172. https://doi.org/10.3390/electronics13112172

Academic Editor: Aryya Gangopadhyay

Received: 22 April 2024
Revised: 24 May 2024
Accepted: 30 May 2024
Published: 3 June 2024

1. Introduction

Adversarial machine learning (ML) attacks have been a very popular research topic since the introduction of the fast gradient sign method (FGSM) [1]. While the initial research on these attacks focused on general image recognition models, many recent works show that adversarial ML attacks can be a threat to autonomous vehicles.

Autonomous vehicles consist of various modules in which machine learning models are commonly used. These modules are often called advanced driver assistance systems (ADASs), and they serve different vehicle capabilities, such as traffic sign recognition, lane-keeping assistance, collision detection, and smart parking. Among these modules, traffic sign recognition models have repeatedly been a target of successful adversarial ML attacks. The increasing number of successful adversarial attacks against traffic sign recognition models raises real-world security concerns.

Adversarial ML attacks are achieved by creating input data with perturbations that are not obvious to the human eye but result in errors for the ML algorithms. It is suggested that many early adversarial ML attacks that target image recognition models apply perturbations to the whole input [1,2] or a specific part of the input in the form of pixels or

patches [3]. Natural aspects such as weather, lights, and shadows have also been considered while generating perturbations in attacks against traffic sign recognition models [4,5].

However, compared to the attacks, forms of defense mechanisms against adversarial ML attacks are scarce. Several defense mechanisms have suggested that robust training with perturbed images or changing a model's original architecture can help detect perturbed instances [6–8]. In addition, denoising methods were proposed to clean perturbed images. Ref. [9] provides a popular image denoiser developed by Microsoft, which uses randomized smoothing. In contrast to attacks, there is no defense method that is specially crafted for traffic sign recognition models. In this paper, as illustrated in Figure 1, we propose the first attack detection mechanism for traffic sign recognition models, which is

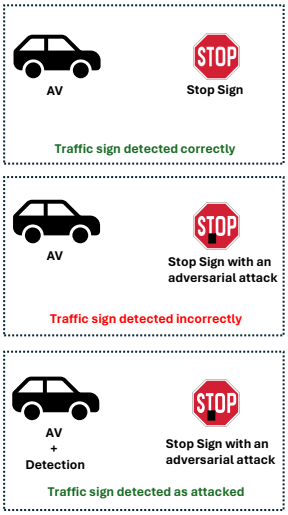

Figure 1. The proposed vision-language adversarial sign detection (ViLAS) method co-operates with the traffic sign recognition model to detect adversarial ML attacks.

Attacks against traffic signs can be achieved in various ways. In addition to possible perturbations suggested for images, physical adversarial patches and the angles of lights or shadows can also be used to achieve successful adversarial machine learning attacks. Therefore, we believe detecting adversarial attacks against traffic signs can be accomplished by observing the inputs and the model outputs with a separate modality subsystem.

In our approach, we use a vision-language model (VLM) as an observing subsystem. VLMs are gaining popularity with the introduction of CLIP [10] and have been used for several ML applications [11–15]. In our method, instead of using off-the-shelf CLIP, we train a smaller and faster VLM that is more suitable for timely decision-making in autonomous driving.

The multimodal processing of VLM for visual and language features provides additional information space for detecting attacks based on only visual features. By leveraging the capability of VLMs to make connections between texts and images, our method utilizes the context of images by computing the similarity scores between the image and the class labels. Then, it decides if the image is adversarial or clean based on the consistency/inconsistency between the predictions by the image recognition model and the similarity scores obtained by the VLM.

In this work, our contributions can be summarized as follows:

- We propose a custom VLM for universal attack detection, which can easily work with any traffic sign recognition model;
- To the best of our knowledge, this is the first defense mechanism against adversarial ML attacks that is specifically crafted for autonomous driving;
- We benchmark our method with extensive experiments and analyze its effectiveness compared to existing generic defense methods for image recognition.

2. Related Work

The robustness of deep neural networks and their vulnerability against adversarial ML attacks have been investigated for several years since the introduction of FGSM [1]. Many attacks for both white-box and black-box settings were proposed. In white-box settings, the attacker has access to the target model, including its parameters. Hence, very effective and strong methods were introduced, including C&W attacks [16] and PGD attacks [2]. For the black-box setting, in which the attacker does not have any prior information about the target model, different approaches have been proposed. One of the most common approaches is introduced in [17]. The idea is to use a substitute model for generating adversarial samples for an unknown target model, utilizing the transferability of adversarial samples to different CNNs. Another black-box attack approach is to estimate gradients to generate adversarial data without using a substitute model, e.g., ZOO [18], NES [19], and SPSA [20].

In addition to adversarial attacks developed for image recognition models, special attacks that leverage natural conditions, such as lights, shadows, and physical patches, have also been introduced for traffic sign recognition models [4,5,21–27]. Ref. [5] tries to apply adversarial patches to traffic signs where the patches have pictures of other traffic signs on them. Ref. [4] tries to investigate the effect of light on traffic signs and proves that, under certain lighting, traffic sign recognition models can be deceived.

However, defense against adversarial ML attacks has not been sufficiently investigated despite the increasing number and variety of attacks. Although early works suggest the usage of adversarial training [2,28], these defense approaches cannot be effective against the increasing number of adversarial attacks since they would require repeating the whole training process for each new attack. For this reason, several defense approaches were proposed, with the goal of inferring from the test images with no adversarial training. Ref. [29] aims to improve classification performance by adding an adversarial attack module and a data augmentation module to the model. Ref. [30] proposes a defense method in which the common information between clean and perturbed data is analyzed. Ref. [31] tries to remove perturbations with the help of adaptive compression and reconstruction. Ref. [32] implements random resizing to inputs to achieve robustness. Ref. [33] uses the compression of JPEG images to avoid any possible perturbations. In addition, some adversarial benchmark datasets, such as [34], were proposed to evaluate robustness against adversarial attacks. Ref. [9] tries to remove perturbation from input images by applying randomized smoothing.

Vision-language models (VLMs) are gaining popularity in recent years with the introduction of CLIP [10]. Many computer vision tasks benefit from VLMs, including object detection [11,12], video action recognition [13,35], and video anomaly detection [36]. In this work, we present the first use of a VLM for adversarial traffic sign detection.

3. Method

In this section, first, we discuss the threat model and then explain the details of our proposed method. After we present our architectural design and choices, we will conclude the section by demonstrating the implementation and real-world applicability of the system.

3.1. Threat Model

We assume an image classifier, $G(X)$, that is specially trained for recognizing traffic signs, and this is placed in the autonomous vehicle. After a traffic sign is captured by the vehicle, it is sent to the image classifier, G, as an input, $X \in \mathbb{R}^{H \times W \times C}$, which consists of $H \times W$ pixels and C channels. The image classifier outputs the predicted class probabilities $p_G = [p_1, p_2, p_3, ..., p_M]$, where M is the number of distinct class labels. By denoting the true label of the input with y, the true classification by the image recognition model is given by $G(X) = y$. However, an adversary can attack the system by generating an adversarial version, X^{adv}, of the input, which might be classified as $G(X^{adv}) = y'$, where $y \neq y'$. The

adversarial image, X^{adv}, can be either physically created or created by using malicious software in the autonomous vehicle.

3.2. Proposed Detection Workflow

There are many ways to generate successful image perturbations, such as adding noise to all inputs, adding only a small patch, and leveraging natural conditions, such as shadows. Due to the vastness of the attack space and the typical obliviousness of the image recognition models to the attack strategy, a defense mechanism should not use any bias regarding the attacks. Therefore, we propose a universal detection mechanism that does not rely on any assumption about the attack method or image recognition model and, hence, can work with any model to detect a broad range of adversarial attacks. An overview of the proposed detection method is depicted in Figure 2.

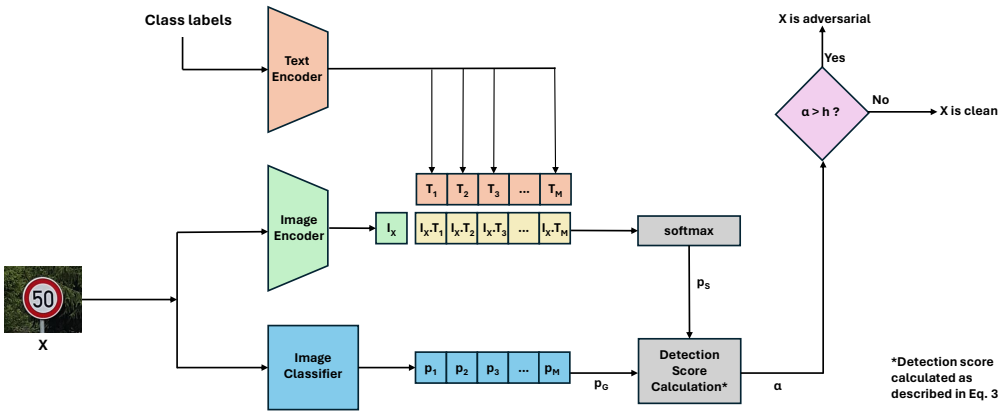

Figure 2. Overview of the proposed detection method, ViLAS. The predicted class, $G(X)$, by the image classifier is declared not valid if the attack detection statistic, α, is greater than the threshold, h.

In our detection mechanism, and in parallel with the image classification model, we used a custom VLM to detect adversarial traffic signs. Similarly to demonstrations of VLMs in existing works, our mechanism consists of a text encoder, E_T, and an image encoder, E_I. $L = [l_1, l_2, l_3, ..., l_M]$ is the label list that contains all possible labels that can be detected by the image classifier, G. Label list L is sent to the text encoder, E_T, in order to obtain label embedding vectors $E_T(L) = T = [T_1, T_2, T_3, ..., T_M]$, which contain the vector representations of each label.

In the next step, image X is sent to the image encoder, E_I, to obtain the image embedding vector, such that $E_I(X) = I_X$. Then, the cosine similarity, S, between the image embedding vector and label embedding vectors is taken by calculating the dot product:

$$S = I_X \cdot T = [I_X \cdot T_1, I_X \cdot T_2, \ldots, I_X \cdot T_M]. \tag{1}$$

By taking the softmax over S, we obtain the probability scores:

$$p_S = \text{softmax}(S) = [p_{S1}, p_{S2}, \ldots, p_{SM}]. \tag{2}$$

Similar to the score calculations in existing works [37,38], a final detection score, α, is calculated by obtaining the average of the forward and reverse KL divergences between the class probabilities, p_G, predicted by the image classifier and p_S predicted by the VLM:

$$\alpha = \frac{1}{2}[D_{KL}(p_S||p_G) + D_{KL}(p_G||p_S)]. \tag{3}$$

The detection score, α, is expected to be low for clean inputs and high for adversarial inputs. A threshold, h, is decided by calculating the detection scores for a set of clean

images, $\beta = [\alpha_1, \alpha_2, \ldots, \alpha_K]$, where K is the number of clean images and the scores in β are sorted in ascending order. Threshold h is selected as the θth percentile of the clean training scores:

$$h = \beta[\lfloor K\theta/100 \rfloor], \tag{4}$$

where $\lfloor \cdot \rfloor$ denotes the floor operator, and $\beta[i]$ denotes the ith element of β.

After obtaining the detection score for an input, decision d is made as follows:

$$d = \begin{cases} X \text{ is adversarial} & \text{if } \alpha > h \\ X \text{ is not adversarial} & \text{if } \alpha \leq h, \end{cases} \tag{5}$$

where α is computed, as per Equation (3).

3.3. Architectural Design

VLMs are very popular for solving various vision tasks since the introduction of CLIP [10]. While an off-the-shelf pretrained CLIP is used directly in many of the existing works, we designed and trained our custom VLM with ResNet50 [39] and DistillBERT [40]. This provided a smaller, faster, and more scalable VLM that is more suitable for use in autonomous vehicles. More importantly, in Section 5.2, we demonstrate that our custom VLM has better detection performance than a fine-tuned CLIP.

3.4. Implementation and Applicability

Our proposed method, ViLAS, is highly compatible with the existing image classifiers or object detectors that are used for traffic sign recognition in autonomous driving. ViLAS does not make any architectural changes to the existing models. Our method only needs the classification probabilities from the image recognition model. Algorithm 1 describes the overall workflow of our detection method. In addition, a Pytorch [41] implementation of our detection method is provided at https://github.com/furkanmumcu/ViLAS.

Algorithm 1 Vision-language model for adversarial traffic sign detection (ViLAS)

Input: input image X, labels vector $L = [l_1, l_2, ..., l_M]$ containing class labels as text, image classification model G, image encoder E_I, text encoder E_T, threshold h.
Output: detection result d.

image recognition model receives the input image, returns the classification probabilities
$p_G = G(X)$
E_I calculates the image embedding vector
$I_X = E_I(X)$
E_T calculates the text embedding vectors
$[T_1, T_2, ..., T_M] = E_T(L)$
calculate cosine similarity scores:
$S = [I_X \cdot T_1, I_X \cdot T_2, \ldots, I_X \cdot T_M]$
apply softmax to similarity scores to get the class probabilities by VLM
$p_S = \text{softmax}(S)$
Calculate detection score
$\alpha = \frac{1}{2}[D_{KL}(p_G||p_S) + D_{KL}(p_S||p_G)]$
if $\alpha > h$ **then**
    $d \leftarrow$ adversarial
else
    $d \leftarrow$ not adversarial
end if
return d

4. Experiments

In this section, we first provide the experimental settings and then present our results.

4.1. Experimental Settings

Throughout the experiments, we used the following settings:

Datasets: We used the GTSRB [42] dataset, which includes traffic signs photographed under various light conditions and distances, very similar to the images collected by the front cameras of autonomous vehicles. GTSRB contains 43 classes of traffic signs, split into 39,209 training images and 12,630 test images.

VLM training: In our VLM, we used ResNet50 as the image encoder and DistillBERT as the text encoder. First, we trained ResNet50 using the GTSRB dataset's training split. Then, the trained ResNET50 and DistillBERT were jointly trained. During the training process, GTSRB dataset class labels were sent to the text encoder, and the corresponding images were sent to the image encoder. The images were scaled to 224 $\times$ 224, and the probability vectors from the image encoder were used as the image-embedding vectors. The loss calculation and training were conducted as described in [10]. More details of our VLM structure and loss calculations can be found in our official implementation code.

Target models: Adversarial ML attack transferability between image classifiers has been studied in the literature, and it has been shown that similar architectures are more vulnerable to transferability. Therefore, we chose three convolutional neural network (CNN)-based image classifiers from different families, namely VGG16 [43], ResNet152 [39], and InceptionV3 [44], and one transformer-based image classifier, ViT [45].

Target models training: We trained VGG16, ResNet152, InceptionV3, and ViT with the training split of the GTSRB dataset. The networks were trained using the Adam optimizer [46] for 10 epochs, using a learning rate of 0.003. During the training, only the clean images were used, and no adversarial training method was applied. The detection accuracies are 97%, 99%, 98%, and 98%, respectively. The correctly classified images from the test split for each image classifier were used in our experiments.

Adversarial ML attacks: We chose two traditional adversarial ML attacks, which were originally proposed for generic images, and two adversarial ML attacks, which were proposed specifically for traffic signs. We use FGSM [1] and PGD [2] as traditional adversarial ML attacks. Both of the methods take advantage of the model gradients and apply perturbation to all the pixels of an image. On the other hand, the natural light attack [4] simulates daylight or street lights on traffic signs, and the patch attack [5] inserts adversarial patches into traffic sign images. A clean test image from the GTSRB dataset

Figure 3. Samples of clean and attacked images.

During the experiments, all of the images that were correctly classified by the target models were used as the clean set. For each attack target model combination, we created an adversarial set by only taking the adversarial images misclassified by the target model. Then, the combination of the clean and adversarial sets was used for the score calculations, given by Equation (3).

Existing defense method: We used denoised smoothing [9] to compare our proposed detection method. Since denoised smoothing is not originally a detection method, we obtained a detection method from it by calculating the KL divergence between the image classification model's class probabilities for the original input image and the denoised input image.

Evaluation metric: In order to evaluate the attack detection performance of the defense methods, we report the commonly used area under the curve (AUC) metric from the receiver operating characteristic (ROC) curve, which shows a trade-off between the true positive rate (i.e., the ratio of successfully detected adversarial videos to all adversarial videos) and the false positive rate (i.e., the ratio of false alarms to all clean videos).

4.2. Results

In Table 1, we report the AUC scores for our method, ViLAS, and Denoise [9] against PGD [2], FGSM [1], Patch Attack [5], and Light Attack [4], targeting four image recognition models. Since it is known that different attacks might have different performances on different architectures [47], we selected both the CNN-based (VGG16, ResNet152, InceptionV3) and transformer-based (ViT) target models. We used the same parameter settings that were reported in the original source of the attacks.

Table 1. Attack detection results (AUC) for our method and Denoise, considering four adversarial ML attacks: (PGD [2], FGSM [1], Patch [5], and Light [4]) and four target models (VGG16 [43], ResNet152 [39], ViT [45], and InceptionV3 [44]). Best performance for each case is highlighted with bold.

		Denoise [9]	ViLAS
PGD [2]	VGG16 [43]	0.15	**0.99**
	ResNet152 [39]	0.25	**0.80**
	ViT [45]	0.81	**0.98**
	InceptionV3 [44]	0.21	**0.98**
FGSM [1]	VGG16 [43]	0.87	**0.99**
	ResNet152 [39]	0.87	**0.93**
	ViT [45]	0.94	**0.99**
	InceptionV3 [44]	0.91	**0.99**
Patch [5]	VGG16 [43]	0.88	**0.90**
	ResNet152 [39]	0.82	**0.84**
	ViT [45]	0.91	**0.94**
	InceptionV3 [44]	0.84	**0.85**
Light [4]	VGG16 [43]	0.77	**0.91**
	ResNet152 [39]	0.97	**0.99**
	ViT [45]	0.98	**0.99**
	InceptionV3 [44]	0.85	**0.98**
	Average	0.75	**0.94**

Denoise performs worst against PGD, with AUC scores of 0.15, 0.25, 0.81, and 0.21 for target models VGG16, ResNet152, ViT, and InceptionV3, respectiveley. Compared to the other target models, our method has lower performance against ResNet152, especially against PGD and Patch attacks, with AUC scores of 0.80 and 0.84. This is expected since ResNet152 is the most similar target model to the image encoder in our VLM (ResNet50), which results in attack transferability.

For each test combination for every attack on each target model, our method outperforms Denoise. While the average AUC score for Denoise is 0.75, ours is 0.94. The AUC of Denoise is as low as 0.15, whereas our lowest AUC score is 0.8, indicating that our method is robust against different types of adversarial ML attacks on different types of image classifiers.

5. Ablation Study

In this section, we first evaluate our method's performance against the transferability of adversarial ML attacks. Next, we compare our custom VLM to an existing pretrained VLM. Finally, we analyze the effect of the proposed KL divergence-based score calculation by comparing it with a different score calculation technique.

5.1. Transferability against ViLAS

Transferability is a common feature of adversarial ML attacks. It is known that there is high transferability between similar architectures and low transferability between diverse architectures. Table 2 shows our method's AUC values averaged over all adversarial attacks targeting each image recognition model. It can be observed that ResNet152 has the lowest AUC at 0.89 since it is the closest architecture to ResNet50, which is used as image encoder in our custom VLM.

Table 2. Average AUC performance of our defense method against all attacks. Larger values indicate a more defensible model.

VGG16 [43]	ResNet152 [39]	ViT [45]	InceptionV3 [44]
0.95	0.89	0.98	0.95

Because of the transferability of adversarial attacks, we recommend designing ViLAS using different architectures for image encoders and traffic sign recognition models. As an ablation study, in Table 3, we provide our detector's AUC score when we use the same architecture, namely ResNet50, as the traffic sign recognition model and the image encoder. In this identical setup, our detection method achieves 0.13, 0.81, 0.84, and 0.92 for the attacks, PGD, FGSM, Patch, and Light attacks, respectively. Except for the PGD attack, our method performs above 0.80, even when using the exact same architecture for the image encoder and traffic sign recognition model. This demonstrates the robustness provided by the text encoder in the proposed multimodal VLM structure.

Table 3. Our method's AUC scores against PGD, FGSM, Patch, and Light attacks when ResNet50 is used as the traffic sign recognition model and the image encoder.

PGD [2]	FGSM [1]	Patch [5]	Light [4]
0.13	0.81	0.84	0.92

5.2. Comparison to Pretrained VLMs

CLIP is a popular VLM that is used in several applications as a pretrained feature extractor. In order to compare this against the custom VLM of ViLAS, we fine-tuned CLIP (we used the official CLIP implementation with ViT-B/32 as the image encoder) using the GTSRB dataset, and we denote the fine-tuned version as CLIP*. In Table 4, we report the AUC scores for our detector, which are explained in Section 3.2, using pretrained CLIP and fine-tuned CLIP* instead of our custom VLM (Section 3.3) under the same experiments that were conducted in Section 4.

Interestingly, fine-tuning does not help to considerably increase the detection performance of CLIP. While in some test cases, there is only a 0.01 increase from CLIP to CLIP*, in most of the cases, the detection accuracy stays the same. As a result, CLIP and CLIP* have an average AUC score of 0.431 and 0.438, respectively. This result (when compared to our 0.94 average AUC) shows the necessity of building a custom VLM for traffic sign recognition by training the image encoder on a relevant dataset. Fine-tuning the image encoder together with the text encoder is, apparently, not sufficient. In Figure 4, we provide the ROC curves for our method, Denoise, CLIP, and CLIP* under the experimental settings of light attack on VGG16, ResNet152, InceptionV3, and ViT.

Table 4. Attack detection results (AUC) for pretrained CLIP and fine-tuned CLIP*, considering four adversarial attacks: (PGD [2], FGSM [1], Patch [5], and Light [4]) and four target models (VGG16 [43], ResNet152 [39], ViT [45], and InceptionV3 [44]).

		CLIP [10]	**CLIP***
PGD [2]	VGG16 [43]	0.38	0.39
	ResNet152 [39]	0.94	0.95
	ViT [45]	0.68	0.69
	InceptionV3 [44]	0.90	0.90
FGSM [1]	VGG16 [43]	0.48	0.49
	ResNet152 [39]	0.33	0.34
	ViT [45]	0.41	0.42
	InceptionV3 [44]	0.32	0.32
Patch [5]	VGG16 [43]	0.23	0.24
	ResNet152 [39]	0.29	0.30
	ViT [45]	0.43	0.44
	InceptionV3 [44]	0.24	0.24
Light [4]	VGG16 [43]	0.29	0.29
	ResNet152 [39]	0.13	0.14
	ViT [45]	0.42	0.42
	InceptionV3 [44]	0.44	0.45
	Average	0.431	0.438

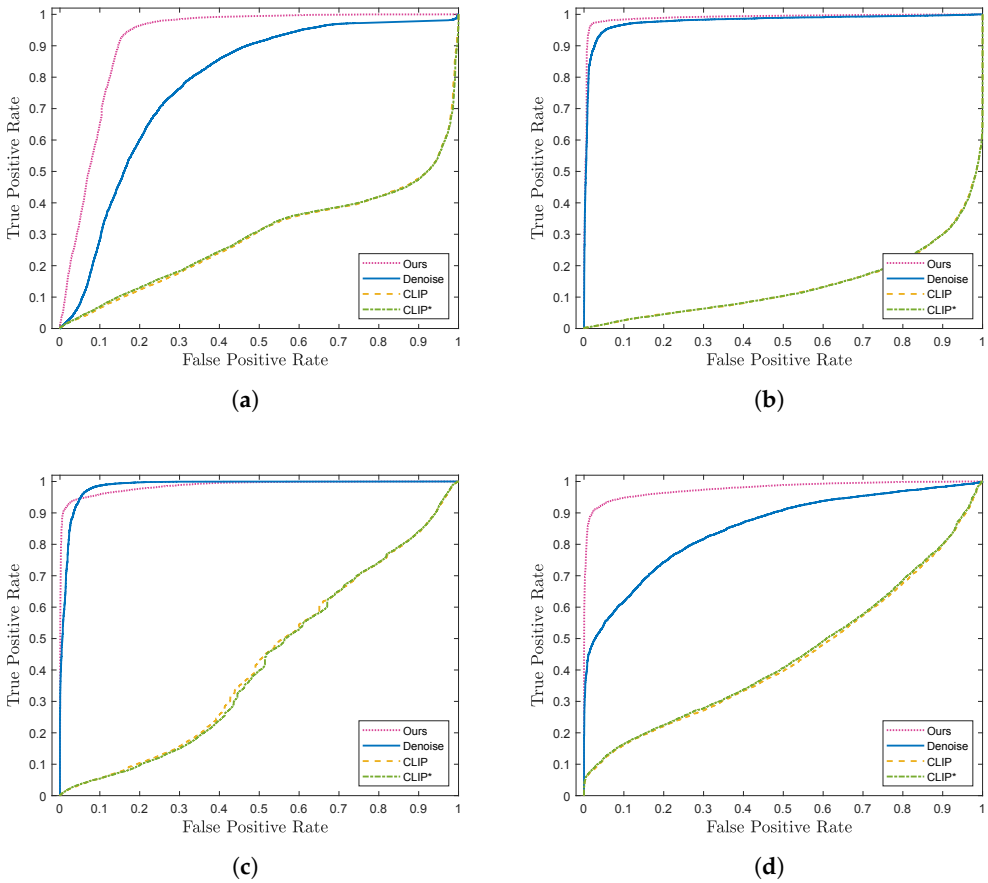

Figure 4. ROC curves for our method (ViLAS), Denoise, CLIP, and CLIP* against light attack, which targets VGG16, ResNet152, InceptionV3, and ViT. (**a**) VGG16; (**b**) ResNet152; (**c**) ViT; (**d**) InceptionV3.

In addition to performance gains, our VLM provides a more resource-efficient, scalable, and faster solution compared to CLIP. While CLIP holds 1.6 GB on the disk, our VLM

requires only 0.36 GB. More importantly, the image processing time of CLIP is more than twice that of our model's image processing time. CLIP calculates similarity scores for 12,630 images from the test split in GTSRB in 117.64 s using NVIDIA 4090 GPU (Santa Clara, CA, USA), whereas our model accomplishes the same task in 55.24 s with the same hardware configuration. Table 5 summarizes the differences between CLIP and our VLM in terms of size, image processing time (IPT) in milliseconds, and the average AUC scores in the experiments. We also present the IPT with an AMD Ryzen 9 7950X CPU.

Table 5. Comparison of CLIP and our custom VLM in terms of their size on the disk, image processing time (IPT) in milliseconds, and the average AUC scores in the experiments. The first number for IPT is the time when using NVIDIA 4090 GPU, and the second number is the time when using AMD Ryzen 9 7950X CPU.

	Size (GB)	**IPT (ms)**	**AUC**
CLIP [10]	1.6	9.3, 333.6	0.43
Our VLM	0.36	4.4, 18.5	0.94

5.3. Score Calculation

In Equation (3), we use KL divergence to compute the detection score. In addition to KL divergence, in this ablation study, we considered another way to calculate the detection score.

In this version, instead of using probability vectors, we used the probability of the predicted class directly and take the difference between the traffic sign recognition model's probability for the predicted class and the probability of VLM for the predicted class, such that

$$\alpha_1 = p_G[y] - p_S[y], \tag{6}$$

where y is the predicted class by the traffic sign recognition model.

For this new score calculation, we repeat the experiments in Section 4.2 and report the results in Table 6. While Denoise's average AUC drops from 0.75 to 0.69 compared to the original experiments using KL divergence-based score calculation, our average AUC score drops by only 0.01, from 0.94 to 0.93. This result indicates that our method performs well with different score calculations.

Table 6. Attack detection results (AUC) for our method and Denoise using a different attack score calculation. The drop in Denoise's performance is more significant than our model when compared to Table 1. Best performance in each case is highlighted with bold.

		Denoise [9]	**ViLAS**
PGD [2]	VGG16 [43]	0.51	**0.99**
	ResNet152 [39]	0.49	**0.94**
	ViT [45]	0.39	**0.97**
	InceptionV3 [44]	0.68	**0.99**
FGSM [1]	VGG16 [43]	0.66	**0.99**
	ResNet152 [39]	0.72	**0.90**
	ViT [45]	0.75	**0.98**
	InceptionV3 [44]	0.83	**0.99**
Patch [5]	VGG16 [43]	0.70	**0.90**
	ResNet152 [39]	0.74	**0.82**
	ViT [45]	0.85	**0.94**
	InceptionV3 [44]	0.82	**0.85**
Light [4]	VGG16 [43]	0.60	**0.88**
	ResNet152 [39]	0.81	**0.95**
	ViT [45]	0.89	**0.95**
	InceptionV3 [44]	0.75	**0.96**
	Average	0.69	**0.93**

6. Discussions and Future Work

Our model's superior performance across various attacks and target models is due to two factors: extra information coming from the class labels regarding the image context, which is typically not utilized in attack design, and the visual clues coming from the image encoder in VLM, which is significantly differentiated from the targeted image recognition model. By using an image encoder in VLM in addition to the one in the image recognition model, we introduce an intentional redundancy for attack detection. The utility of having two image models for attack detection stems from the fact that the architecture of the image encoder in VLM should intentionally be chosen to be as distinct as possible from the image recognition model, as summarized in Table 2. This choice is due to the lack of attack transferability between significantly different architectures. The image encoder in ViLAS is further differentiated from the image recognition model by jointly training it with the text encoder. That is why it is still effective in detecting attacks when both image models in ViLAS and the target model are ResNet (Table 2).

Thanks to its orthogonal features not being utilized by classical attacks, ViLAS can also be used as a robust image recognition model. The fact that it is able to identify the true class in the presence of attacks is the reason that enables superior attack detection performance. It can be used as a standalone image classifier, as well as a backup decision-maker that can override the original image classifier when needed (e.g., when an attack is detected or, for some reason, the image classifier fails).

While we demonstrated that ViLAS does not burden an autonomous vehicle with impractical load, as it only needs 360 MB of space and takes milliseconds to process an image (note that the detection score computation time is insignificant compared to the inference time of VLM), the computational footprint of ViLAS can be further reduced by restricting its use to only a subset of classes that are vulnerable to attacks.

In our implementation of a custom VLM, we used ResNet50 [39] as the image encoder and DistillBert [40] as the text encoder. Our aim when choosing this specific configuration was to use small models for compactness and speed, which is suitable for the local computing resources and response time requirements in an autonomous vehicle. Since our detection framework is not restricted to any specific image/text encoder, the performance when using other encoder models can also be evaluated under similar setups.

Since many existing traffic sign datasets contain images taken by standard optical cameras [42], we evaluated our method by using a CNN-based image encoder with the assumption of traffic signs recorded by standard optical cameras. However, it is possible that some autonomous vehicles use different types of cameras (e.g., infrared) or sensors (e.g., lidar and radar). Similarly, the experiments in this work are limited to the operating conditions demonstrated in the GTSRB dataset. The proposed main idea of using a custom VLM-based detector is potentially compatible with any input visual data type, but further experiments are needed to study different data modalities.

An important limitation of ViLAS is that it can be targeted by query-based black-box attacks, which iteratively design adversarial images in a data-driven fashion by sending queries to the target model and computing the gradient of its decision with respect to input modifications to optimize the attack. Since the final decision of autonomous vehicle hosting ViLAS will be either suppressed or corrected for adversarial queries, a data-driven black-box attack can learn specific noise masks to fool ViLAS. This can be an interesting future study. We should also note here that such data-driven black-box attacks are known to require large numbers of queries (on the order of millions) to design successful noise masks, which can be a practical limitation in real-world scenarios.

7. Conclusions

The increasing number of successful attacks against traffic sign recognition models are raising real-world security concerns. In addition to classical adversarial ML attacks that target image classifiers, it has been shown in the literature that natural aspects, such as weather, lights, or shadows, can be used to attack traffic sign recognition models. In

contrast to the growing numbers of threatening adversarial ML attacks against traffic sign recognition models, defending against these attacks was not studied sufficiently. In this paper, we proposed the first attack detection mechanism for traffic sign recognition, ViLAS (vision-language model for adversarial traffic sign detection).

ViLAS leverages the context of an image by computing the similarity scores between the image and the class labels in a custom vision-language model (VLM), thereby deciding if the image is adversarial or clean. By training the image encoder and VLM on a relevant dataset, we showed that ViLAS provides protection against adversarial ML attacks that target popular image classifiers, thanks to the orthogonal information coming from the class label text data. In our extensive experiments, we demonstrated that our detection method outperforms a state-of-the-art defense method by a wide margin (0.94 vs. 0.75 average AUC). Our ablation studies also proved the necessity of a custom VLM design instead of using an off-the-shelf VLM. Our custom VLM requires a minimal overhead when compared to the existing traffic sign recognition models (only 360 MB of disk space and an ITS of as low as 4.4 ms), which can facilitate practical implementation in existing vehicular technology. Furthermore, our detection mechanism is compatible with any existing traffic sign recognition system since it can be applied without any modification to the existing sign recognition models. Although being much smaller in size and faster than the widely used VLM and CLIP, our custom VLM significantly outperforms the same detector based on CLIP being fine-tuned on the same dataset.

Author Contributions: Conceptualization, F.M. and Y.Y.; Methodology, F.M. and Y.Y.; Software, F.M.; Validation, F.M.; Formal analysis, F.M. and Y.Y.; Investigation, F.M. and Y.Y.; Resources, Y.Y.; Data curation, F.M.; Writing—original draft, F.M.; Writing—review & editing, Y.Y.; Visualization, F.M.; Supervision, Y.Y.; Project administration, Y.Y.; Funding acquisition, Y.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by U.S. National Science Foundation (NSF) grant number 2040572.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Goodfellow, I.J.; Shlens, J.; Szegedy, C. Explaining and harnessing adversarial examples. *arXiv* **2014**, arXiv:1412.6572.
2. Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; Vladu, A. Towards deep learning models resistant to adversarial attacks. *arXiv* **2017**, arXiv:1706.06083.
3. Pomponi, J.; Scardapane, S.; Uncini, A. Pixle: A fast and effective black-box attack based on rearranging pixels. In Proceedings of the 2022 International Joint Conference on Neural Networks (IJCNN), Padua, Italy, 18–23 July 2022; pp. 1–7.
4. Hsiao, T.F.; Huang, B.L.; Ni, Z.X.; Lin, Y.T.; Shuai, H.H.; Li, Y.H.; Cheng, W.H. Natural Light Can Also Be Dangerous: Traffic Sign Misinterpretation under Adversarial Natural Light Attacks. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Waikoloa, HI, USA, 3–8 January 2024; pp. 3915–3924.
5. Ye, B.; Yin, H.; Yan, J.; Ge, W. Patch-based attack on traffic sign recognition. In Proceedings of the 2021 IEEE International Intelligent Transportation Systems Conference (ITSC), Indianapolis, IN, USA, 19–22 September 2021; pp. 164–171.
6. Ross, A.; Doshi-Velez, F. Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients. In Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018; Volume 32.
7. Garg, S.; Sharan, V.; Zhang, B.; Valiant, G. A spectral view of adversarially robust features. *Adv. Neural Inf. Process. Syst.* **2018**, *31*, 10138–10148.
8. Kolter, J.Z.; Wong, E. Provable defenses against adversarial examples via the convex outer adversarial polytope. *arXiv* **2017**, arXiv:1711.00851.
9. Salman, H.; Sun, M.; Yang, G.; Kapoor, A.; Kolter, J.Z. Denoised smoothing: A provable defense for pretrained classifiers. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 21945–21957.
10. Radford, A.; Kim, J.W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. Learning transferable visual models from natural language supervision. In Proceedings of the International Conference on Machine Learning, PMLR, Virtual Event, 18–24 July 2021; pp. 8748–8763.
11. Zhang, H.; Zhang, P.; Hu, X.; Chen, Y.C.; Li, L.; Dai, X.; Wang, L.; Yuan, L.; Hwang, J.N.; Gao, J. Glipv2: Unifying localization and vision-language understanding. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 36067–36080.

12. Liu, X.; Zhou, J.; Kong, T.; Lin, X.; Ji, R. Exploring target representations for masked autoencoders. *arXiv* **2022**, arXiv:2209.03917.
13. Wu, W.; Wang, X.; Luo, H.; Wang, J.; Yang, Y.; Ouyang, W. Bidirectional cross-modal knowledge exploration for video recognition with pre-trained vision-language models. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 6620–6630.
14. Wu, W.; Sun, Z.; Ouyang, W. Transferring textual knowledge for visual recognition. *arXiv* **2022**, arXiv:2207.01297.
15. Lin, Z.; Geng, S.; Zhang, R.; Gao, P.; de Melo, G.; Wang, X.; Dai, J.; Qiao, Y.; Li, H. Frozen clip models are efficient video learners. In Proceedings of the European Conference on Computer Vision, Tel Aviv, Israel, 23–27 October 2022; Springer: Berlin/Heidelberg, Germany, 2022; pp. 388–404.
16. Carlini, N.; Wagner, D. Towards evaluating the robustness of neural networks. In Proceedings of the 2017 IEEE Symposium on Security and Privacy (sp), San Jose, CA, USA, 22–26 May 2017; pp. 39–57.
17. Papernot, N.; McDaniel, P.; Goodfellow, I.; Jha, S.; Celik, Z.B.; Swami, A. Practical black-box attacks against machine learning. In Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security, Abu Dhabi, UAE, 2–6 April 2017; pp. 506–519.
18. Chen, P.Y.; Zhang, H.; Sharma, Y.; Yi, J.; Hsieh, C.J. Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security, Dallas, TX, USA, 3 November 2017; pp. 15–26.
19. Ilyas, A.; Engstrom, L.; Athalye, A.; Lin, J. Black-box adversarial attacks with limited queries and information. In Proceedings of the International Conference on Machine Learning, PMLR, Vienna, Austria, 25–26 July 2018; pp. 2137–2146.
20. Uesato, J.; O'donoghue, B.; Kohli, P.; Oord, A. Adversarial risk and the dangers of evaluating against weak attacks. In Proceedings of the International Conference on Machine Learning, PMLR, Vienna, Austria, 25–26 July 2018; pp. 5025–5034.
21. Gnanasambandam, A.; Sherman, A.M.; Chan, S.H. Optical adversarial attack. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 92–101.
22. Duan, R.; Mao, X.; Qin, A.K.; Chen, Y.; Ye, S.; He, Y.; Yang, Y. Adversarial laser beam: Effective physical-world attack to dnns in a blink. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 16062–16071.
23. Woitschek, F.; Schneider, G. Physical adversarial attacks on deep neural networks for traffic sign recognition: A feasibility study. In Proceedings of the 2021 IEEE Intelligent Vehicles Symposium (IV), Nagoya, Japan, 11–17 July 2021; pp. 481–487.
24. Yan, J.; Yin, H.; Ye, B.; Ge, W.; Zhang, H.; Rigoll, G. An Adversarial Attack on Salient Regions of Traffic Sign. *Automot. Innov.* **2023**, *6*, 190–203. [CrossRef]
25. Sitawarin, C.; Bhagoji, A.N.; Mosenia, A.; Mittal, P.; Chiang, M. Rogue signs: Deceiving traffic sign recognition with malicious ads and logos. *arXiv* **2018**, arXiv:1801.02780.
26. Li, Y.; Xu, X.; Xiao, J.; Li, S.; Shen, H.T. Adaptive square attack: Fooling autonomous cars with adversarial traffic signs. *IEEE Internet Things J.* **2020**, *8*, 6337–6347. [CrossRef]
27. Morgulis, N.; Kreines, A.; Mendelowitz, S.; Weisglass, Y. Fooling a real car with adversarial traffic signs. *arXiv* **2019**, arXiv:1907.00374.
28. Tramèr, F.; Kurakin, A.; Papernot, N.; Goodfellow, I.; Boneh, D.; McDaniel, P. Ensemble adversarial training: Attacks and defenses. *arXiv* **2017**, arXiv:1705.07204.
29. Yang, S.; Li, J.; Zhang, T.; Zhao, J.; Shen, F. AdvMask: A sparse adversarial attack-based data augmentation method for image classification. *Pattern Recognit.* **2023**, *144*, 109847. [CrossRef]
30. Yu, X.; Smedemark-Margulies, N.; Aeron, S.; Koike-Akino, T.; Moulin, P.; Brand, M.; Parsons, K.; Wang, Y. Improving adversarial robustness by learning shared information. *Pattern Recognit.* **2023**, *134*, 109054. [CrossRef]
31. Niu, Z.H.; Yang, Y.B. Defense Against Adversarial Attacks with Efficient Frequency-Adaptive Compression and Reconstruction. *Pattern Recognit.* **2023**, *138*, 109382. [CrossRef]
32. Xie, C.; Wang, J.; Zhang, Z.; Ren, Z.; Yuille, A. Mitigating adversarial effects through randomization. *arXiv* **2017**, arXiv:1711.01991.
33. Dziugaite, G.K.; Ghahramani, Z.; Roy, D.M. A study of the effect of jpg compression on adversarial images. *arXiv* **2016**, arXiv:1608.00853.
34. Pintor, M.; Angioni, D.; Sotgiu, A.; Demetrio, L.; Demontis, A.; Biggio, B.; Roli, F. ImageNet-Patch: A dataset for benchmarking machine learning robustness against adversarial patches. *Pattern Recognit.* **2023**, *134*, 109064. [CrossRef]
35. Ni, B.; Peng, H.; Chen, M.; Zhang, S.; Meng, G.; Fu, J.; Xiang, S.; Ling, H. Expanding language-image pretrained models for general video recognition. In Proceedings of the European Conference on Computer Vision, Tel Aviv, Israel, 23–27 October 2022; Springer: Berlin/Heidelberg, Germany, 2022; pp. 1–18.
36. Zanella, L.; Liberatori, B.; Menapace, W.; Poiesi, F.; Wang, Y.; Ricci, E. Delving into CLIP latent space for Video Anomaly Recognition. *arXiv* **2023**, arXiv:2310.02835.
37. Xiao, C.; Deng, R.; Li, B.; Lee, T.; Edwards, B.; Yi, J.; Song, D.; Liu, M.; Molloy, I. Advit: Adversarial frames identifier based on temporal consistency in videos. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 3968–3977.
38. Mumcu, F.; Yilmaz, Y. Multimodal Attack Detection for Action Recognition Models. *arXiv* **2024**, arXiv:2404.10790.
39. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.

40. Sanh, V.; Debut, L.; Chaumond, J.; Wolf, T. DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter. *arXiv* **2019**, arXiv:1910.01108.
41. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. Pytorch: An imperative style, high-performance deep learning library. *Adv. Neural Inf. Process. Syst.* **2019**, *32*, 8026–8037.
42. Stallkamp, J.; Schlipsing, M.; Salmen, J.; Igel, C. Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. *Neural Netw.* **2012**, *32*, 323–332. [CrossRef] [PubMed]
43. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
44. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1–9.
45. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv* **2020**, arXiv:2010.11929.
46. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
47. Mumcu, F.; Yilmaz, Y. Sequential architecture-agnostic black-box attack design and analysis. *Pattern Recognit.* **2023**, *147*, 110066. [CrossRef]