Phantom of the ADAS: Phantom Attacks on Driver-Assistance Systems

Ben Nassi[1], Dudi Nassi[1], Raz Ben-Netanel[1], Yisroel Mirsky[1,2], Oleg Drokin[3], Yuval Elovici[1]

Video Demonstration - https://youtu.be/1cSw4fXYqWI
{nassib,nassid,razx,yisroel,elovici}@post.bgu.ac.il, green@linuxhacker.ru
[1] Ben-Gurion University of the Negev, [2] Georgia Tech,[3] Independent Tesla Researcher

Abstract

The absence of deployed vehicular communication systems, which prevents the advanced driving assistance systems (ADASs) and autopilots of semi/fully autonomous cars to validate their virtual perception regarding the physical environment surrounding the car with a third party, has been exploited in various attacks suggested by researchers. Since the application of these attacks comes with a cost (exposure of the attacker's identity), the delicate exposure vs. application balance has held, and attacks of this kind have not yet been encountered in the wild. In this paper, we investigate a new perceptual challenge that causes the ADASs and autopilots of semi/fully autonomous to consider depthless objects (phantoms) as real. We show how attackers can exploit this perceptual challenge to apply phantom attacks and change the abovementioned balance, without the need to physically approach the attack scene, by projecting a phantom via a drone equipped with a portable projector or by presenting a phantom on a hacked digital billboard that faces the Internet and is located near roads. We show that the car industry has not considered this type of attack by demonstrating the attack on today's most advanced ADAS and autopilot technologies: Mobileye 630 PRO and the Tesla Model X, HW 2.5; our experiments show that when presented with various phantoms, a car's ADAS or autopilot considers the phantoms as real objects, causing these systems to trigger the brakes, steer into the lane of oncoming traffic, and issue notifications about fake road signs. In order to mitigate this attack, we present a model that analyzes a detected object's context, surface, and reflected light, which is capable of detecting phantoms with 0.99 AUC. Finally, we explain why the deployment of vehicular communication systems might reduce attackers' opportunities to apply phantom attacks but won't eliminate them.

I. Introduction

After years of research and development, automobile technology is rapidly approaching the point at which human drivers can be replaced, as cars are now capable of supporting semi/fully autonomous driving [1, 2]. While the deployment of semi/fully autonomous cars has already begun in many countries around the world, the deployment of vehicular communication systems [3], a set of protocols intended for exchanging information between vehicles and roadside units, has been delayed [4]. The eventual deployment of such systems, which include V2V (vehicle-to-vehicle), V2I (vehicle-to-infrastructure), V2P (vehicle-to-pedestrian), and V2X (vehicle-to-everything) communication systems, is intended to supply semi/fully autonomous cars with information and validation regarding lanes, road signs, and obstacles.

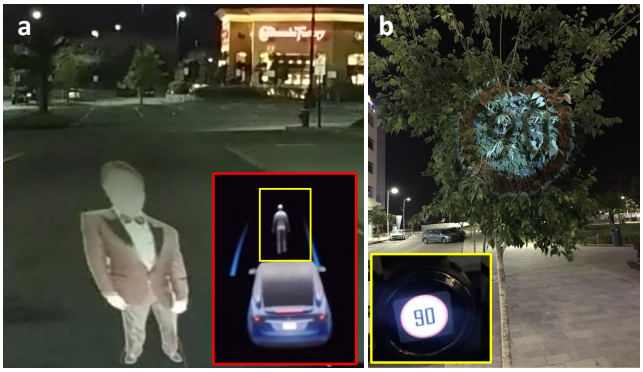

Fig. 1: Perceptual Challenge: Would you consider the projection of the person (a) and road sign (b) real? Telsa considers (a) a real person and Mobileye 630 PRO considers (b) a real road sign.

Given the delayed deployment of vehicular communication systems in most places around the world, autonomous driving largely relies on sensor fusion to replace human drivers. Passive and active sensors are used in order to create 360° 3D virtual perception of the physical environment surrounding the car. However, the lack of vehicular communication system deployment has created a validation gap which limits the ability of semi/fully autonomous cars to validate their virtual perception of obstacles and lane markings with a third party, requiring them to rely solely on their sensors and validate one sensor's measurements with another. Given that the exploitation of this gap threatens the security of semi/fully autonomous cars, we ask the following question: Why haven't attacks against semi/fully autonomous cars exploiting this validation gap been encountered in the wild?

Various attacks have already been demonstrated by researchers [5–14], causing cars to misclassify road signs [5–10], misperceive objects [11, 12], deviate to the lane of oncoming traffic [13], and navigate in the wrong direction [14]. These attacks can only be applied by skilled attackers (e.g., an expert

in radio spoofing or adversarial machine learning techniques) and require complicated/extensive preparation (e.g., a long preprocessing phase to find an evading instance that would be misclassified by a model). In addition, these methods necessitate that attackers approach the attack scene in order to set up the equipment needed to conduct the attack (e.g., laser/ultrasound/radio transmitter [11, 12, 14]) or add physical artifacts to the attack scene (e.g., stickers, patches, graffiti [5–10, 13]), risky acts that can expose the identity of the attacker. As long as the current exposure vs. application balance holds, in which attackers must "pay" for applying their attacks in the currency of identity exposure, the chance of encountering these attacks [5–14] in the wild remains low.

In this paper, we investigate a perceptual challenge, which causes the advanced driving assistance systems (ADASs) and autopilots of semi/fully autonomous cars to consider the depthless objects (phantoms) as real (demonstrated in Fig. 1). We show how attackers can exploit this perceptual challenge and the validation gap (i.e., the inability of semi/fully autonomous cars to verify their virtual perception with a third party) to apply phantom attacks against ADASs and autopilots of semi/fully autonomous cars without the need to physically approach the attack scene, by projecting a phantom via a drone equipped with a portable projector or by presenting a phantom on a hacked digital billboard that faces the Internet and is located near roads.

We start by discussing why phantoms are considered a perceptual challenge for machines (section III). We continue by analyzing phantom attack characteristics using Mobileye 630 PRO (section IV), which is currently the most popular external ADAS, and investigate how phantom attacks can be disguised such that human drivers in semi-autonomous cars ignore/fail to perceive them (in just 125 ms). We continue by demonstrating how attackers can apply phantom attacks against the Tesla Model X (HW 2.5), causing the car's autopilot to automatically and suddenly put on the brakes, by projecting a phantom of a person, and deviate toward the lane of oncoming traffic, by projecting a phantom of a lane (section V). In order to detect phantoms, we evaluate a convolutional neural network model that was trained purely on the output of a video camera. The model, which analyzes the context, surface, and reflected light of a detected object, identifies such attacks with high accuracy, achieving an AUC of 0.99 (section VI). We also present the response of both Mobileye and Tesla to our findings (section VII). At the end of the paper (section VIII), we discuss why the deployment of vehicular communication systems might limit the opportunities attackers have to apply phantom attacks but won't eliminate them.

The first contribution of this paper is related to the attack: We present a new type of attack which can be applied remotely by unskilled attackers and endanger pedestrians, drivers, and passengers, and changes the existing exposure vs. application balance. We demonstrate the application of this attack in two ways: via a drone equipped with a projector and as objects embedded in existing advertisements presented on digital billboards; further, we show that this perceptual challenge is currently not considered by the automobile industry. The second contribution is related to the proposed countermeasure: We present an approach for detecting phantoms with a model that considers context, surface, and reflected light. By using this approach, we can detect with 0.99 AUC.

II. Background, Scope & Related Work

In this section, we provide the necessary background about advanced driving assistance systems (ADASs) and autopilots, discuss autonomous car sensors and vehicular communication protocols, and review related work. The Society of Automotive Engineers defines six levels of driving automation, ranging from fully manual to fully automated systems [15]. Automation levels 0-2 rely on a human driver for monitoring the driving environment. Most traditional cars contain no automation and thus are considered Level 0; countries around the world promote/mandate the integration of an external ADAS (e.g., Mobileye 630) in such cars [16, 17] to enable them to receive notifications and alerts during driving about lane deviation, road signs, etc. Many new cars have Level 1 automation and contain an internal ADAS that supports some autonomous functionality triggered/handled by the car (e.g., collision avoidance system). Semi-autonomous driving starts at Level 2 automation. Level 2 car models are currently being sold by various companies [18] and support semi-autonomous driving that automatically steers by using an autopilot but requires a human driver for monitoring and intervention. In this study, we focus on Mobileye 630 PRO, which is the most popular commercial external ADAS, and on the Tesla Model X's (HW 2.5) autopilot, which is the most advanced autopilot currently deployed in Level 2 automation cars.

Cars rely on sensor fusion to support semi/fully autonomous driving and create virtual perception of the physical environment surrounding the car. They contain a GPS sensor and road mapping that contains information about driving regulations (e.g., minimal/maximal speed limit). Most semi/-full autonomous cars rely on two types of depth sensors (two of the following types: ultrasound, radar, and LiDAR) combined with a set of video cameras to achieve 360° 3D perception (a review about the use of each sensor can be found in [12]). Sensor fusion is used to improve single sensor-based virtual perception which is considered limited (e.g., lane detection can only be detected by the video camera and cannot be detected by other sensors), ambiguous (due to the low resolution of the information obtained), and not effective in adverse weather/light conditions. In this study, we focus on the video cameras that are integrated into autopilots and ADASs.

Vehicular communication protocols (e.g., V2I, V2P, V2V, V2X) are considered the X factor of a driverless future [19] (a review of vehicular communication protocols can be found in [3]). Their deployment is expected to improve cars' virtual perception regarding their surroundings by providing information about nearby (within a range of 300 meters) pedestrians, cars, road signs, lanes, etc. sent via short-range communication. They are expected to increase the level of semi/fully autonomous car safety, however these protocols are currently not in use for various reasons [3, 4], and it is not clear when these protocols will be more widely used around the world. In this study, we focus on the validation gap that

exists as a result of the delay in the deployment of vehicular communication systems.

Many methods that exploit the validation gap have been demonstrated in the last four years [5–13]. Physical attacks against computer vision algorithms for traffic sign recognition were suggested by various researchers [5–9]. *Sitawarin et al.* [6] showed that they could embed two traffic signs in one with a dedicated array of lenses that causes a different traffic sign to appear depending on the angle of view. *Eykholt et al.* [5], *Zhao et al.* [9], *Chen et al.* [8], and *Song et al.* [7] showed that adding a physical artifact (e.g., stickers, graffiti) that looks innocent to the human eye misleads traffic sign recognition algorithms. These methods [5–9] rely on white-box approaches to create an evading instance capable of being misclassified by computer vision algorithms, so the attacker must know the model of the targeted car.

Several attacks against commercial ADASs and autopilots have also been demonstrated in recent years [10–14]. An adversarial machine learning attack against a real ADAS was implemented by *Morgulis et al.* [10] against a car's traffic sign recognition system. Spoofing and jamming attacks against the radar and ultrasound of the Tesla Model S which caused the car to misperceive the distance to nearby obstacles were demonstrated by *Yan et al.* [12]. *Keen Labs* [13] recently demonstrated an attack that causes the autopilot of the Tesla Model S to deviate to the lane of oncoming traffic by placing stickers on the road. *Petit et al.* [11] showed that a laser directed at MobilEye C2-270 can destroy its optical sensor permanently. Other attacks against LiDAR sensors were also demonstrated by *Petit et al.* [11] and *Cao et al.* [20], however the success rate of these attacks in real setups against commercial cars is unknown. Another interesting attack against Tesla's navigation system was recently demonstrated by *Regulus* [14] and showed that GPS spoofing can cause Tesla's autopilot to navigate in the wrong direction.

A few cyber-attacks against connected cars with 0-5 automation levels have been demonstrated [21–25]. However, we consider this type of attacks beyond of the scope of this paper, because they don't result from the validation gap. These attacks do not target sensors and are simply the result of poor implementation in terms of security.

III. Phantom Attacks & Threat Model

In this section, we define phantoms, discuss the perceptual challenge they create for machines, present remote threat models, and discuss the significance of phantom attacks. We define a phantom as a depthless object intended at causing ADASs and autopilot systems to perceive the object and consider it real. A phantom object can be projected by a projector or presented on a screen (e.g., billboard). The object can be an obstacle (e.g., person, car, truck, motorcycle), lane, or road sign. The goal of the attack is to trigger an undesired reaction from a target autopilot/ADAS. In the case of an ADAS, the reaction would be a driver notification about an event (e.g., lane changes) or even an alarm (e.g., collision avoidance). For autopilot systems, the phantom could trigger a dangerous reaction like sudden braking.

Fig. 2: An example showing how object classifiers are only concerned with matching geometry. In this case, Google Cloud's Vision API is used: https://cloud.google.com/vision/.

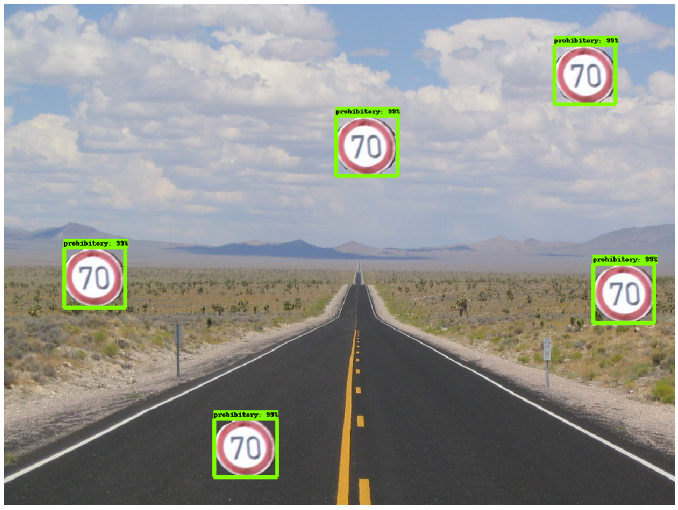

Fig. 3: An example demonstrating that object detectors aren't concerned about context. Here, the Faster R-CNN Inception ResNet model from [26] is used.

A. The Vulnerability

We consider phantom attacks as perceptual challenge for intelligence of machines. We do not consider phantom attacks bugs, since they don't exploit poor code implementation. There are two fundamental reasons why phantoms are considered a perceptual challenge for ADASs and autopilots. The first reason is because phantoms exploit the validation gap, i.e., the inability of semi/fully autonomous cars to verify their virtual perception with a third party. Instead, the semi/fully autonomous car must rely on its own sensor measurements. Therefore, when the camera detects an imminent collision or some other information critical for road safety, the system would rather trust that information alone, even if other sensors "disagree" in order to avoid accidents ("a better safe than sorry" approach).

The second reason is because the computer vision algorithms are trained to identify familiar geometry, without consideration for the object's context or how realistic they look. Most object detection algorithms are essentially feature matchers, meaning that they classify objects with high confidence if parts of the object (e.g., geometry, edges, textures) are similar to the training examples (see Fig. 2 for an example). Moreover, these algorithms don't care whether the scene makes sense or not; an object's location and local context within the frame are not taken into account. Fig. 1b presents an example where an

TABLE I: Phantom Projection Mapped to a Desired Result

Desired Result	Triggered Reaction	Type of Phantom	Place of Projection
Traffic collision	Deviation to pavement/lane of oncoming traffic	Lane	Road
	Trigger sudden brakes	Stop sign	Building, billboard
		No entry sign	
		Obstacles (cars, people, etc.)	Road
Reckless/illegal driving behavior	Triggering fast driving	Speed limit	Building, billboard
Traffic jam	Decreasing speed limitation	Speed limit	
	Stopping cars	No entry sign	
Directing traffic to chosen roads	Closing alternative roads	No entry sign	

ADAS positively identifies a road sign in an irregular location (on a tree), and Fig. 3 demonstrates this concept using a state-of-the-art road sign detector. Also, because an object's texture is not taken into account, object detectors still classify a phantom road sign as a real sign with high confidence although the phantom road sign is partially transparent and captures the surface behind it (see Fig. 1). Finally, these algorithms are trained with a ground truth that all objects are real and are not trained with the concept of fakes. Therefore, although projected images are perceived by a human as obvious fakes (florescent, transparent, defective, or skewed), object detection algorithms will report the object simply because the geometry matches their training examples (see Fig. 1b).

B. The Threat Model

We consider an attacker as any malicious entity with a medium sized budget (a few hundred dollars is enough to buy a drone and a portable projector) and the intention of creating chaos by performing a phantom attack that will result in unintended car behavior. The attacker's motivation for applying a phantom attack can be terrorism (e.g., a desire to kill a targeted passenger in a semi/full autonomous car or harm a nearby pedestrian by causing an accident), criminal intent (e.g., an interest in creating a traffic jam on a specific road by decreasing the allowed speed limit), or fraud (e.g., a person aims to sue Tesla and asks someone to attack his/her car). Table I maps a desired result (causing a traffic collision, triggering illegal driving behavior, routing cars to specific roads, and causing a traffic jam), a triggered reaction (triggering the car's brakes, deviating the car to the lane of oncoming traffic, reckless driving), and the phantom required (lane, road sign, obstacle). In this study, we demonstrate how attackers can cause a traffic collision and illegal driving behavior by applying phantom attacks against Mobileye 630 PRO and Tesla's Model X.

While many methods that exploit the validation gap have been demonstrated in the last four years [5–14], we consider their application as less desirable, because they can only be applied by skilled attackers with expertise in sensor spoofing techniques (e.g., adversarial machine leaning [5–10] or radio/ultrasound/LiDAR spoofing/jamming [11, 12, 14]).

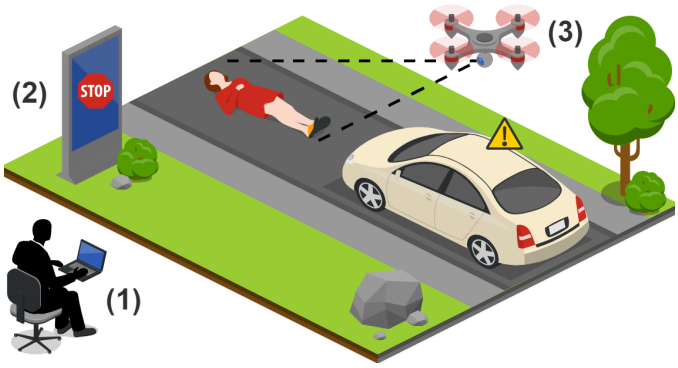

Fig. 4: The Threat Model: An attacker (1) either remotely hacks a digital billboard (2) or flies a drone equipped with a portable projector (3) to create a phantom image. The image is perceived as a real object by a car using an ADAS/autopilot, and the car reacts unexpectedly.

Some of the attacks [5–9] rely on white-box approaches that require full knowledge of the deployed models and a complex preprocessing stage (e.g., finding an evading instance that would be misclassified by a model). Moreover, the forensic evidence left by the attackers at the attack scene (e.g., stickers) can be easily removed by pedestrians and drivers or used by investigators to trace the incident to the attackers. Additionally, these attacks necessitate that the attackers approach the attack scene in order to manipulate an object using a physical artifact (e.g., stickers, graffiti) [5–10, 13] or to set up the required equipment [11, 12, 14], acts that can expose attackers' identities. The exposure vs. application balance which requires that attackers "pay" (with identity exposure) for the ability to perform these attacks is probably the main reason why these attacks have not been seen in the wild.

The phantom attack threat model is much lighter than previously proposed attacks [5–14]. Phantom attacks do not require a skilled attacker or white-box approach, and the equipment needed to apply them is cheap (a few hundred dollars). Any person with malicious intent can be an attacker. Since phantoms are the result of a digital process they can be applied and immediately disabled, so they do not leave any evidence at the attack scene. Finally, phantom attacks can be applied by projecting objects using a drone equipped with a portable projector or presenting objects on hacked digital billboards for advertisements that face the Internet [27, 28] and are located near roads, thereby eliminating the need to physically approach the attack scene, changing the exposure vs. application balance. The abovementioned reasons make phantom attacks very dangerous. The threat model is demonstrated in Fig. 4. In this study, we demonstrate the application of phantom attacks via a drone equipped with a projector and objects embedded in existing advertisements presented on digital billboards.

IV. Phantom Attacks on ADAS (Mobileye)

Commercial ADASs have been shown to decrease the volume of accidents in various studies [29] by notifying drivers about road signs, imminent collisions, lane deviations, etc. As a result, countries around the world promote/mandate the

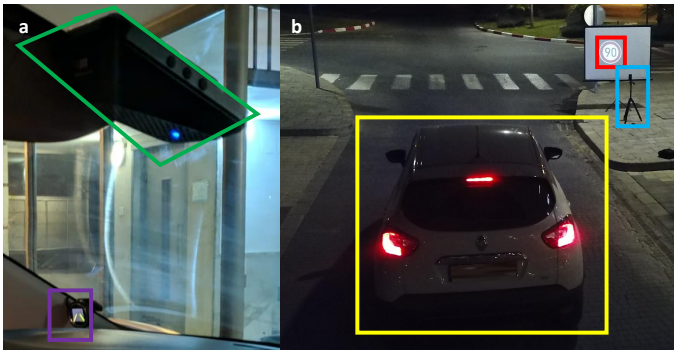

Fig. 5: (a) Mobileye 630 PRO consists of a video camera (boxed in green), which is installed on the windshield, and a display (boxed in purple). (b) Experimental setup: the phantom (boxed in red) projected from a portable projector placed on a tripod (boxed in blue), and the attacked vehicle equipped with Mobileye 630 (boxed in yellow).

use of ADASs in cars that were not manufactured with such systems [16, 17]. Phantom attacks against an ADAS can trigger reckless driving or traffic jams (by notifying drivers about abnormal speed limits), incorrect steering (by notifying drivers about lane deviations), and even sudden braking (by sounding an alarm about an imminent collision). Mobileye 630 PRO is considered the most advanced external ADAS for automation level 0-1 cars, so we decided to use Mobileye 630 PRO in this study. In the rest of this section we refer to Mobileye 630 PRO as Mobileye.

First, we show how attackers can identify and analyze the various factors that influence the success rate of phantom attacks against a real ADAS/autopilot, and we use Mobileye to demonstrate this process. Then, we show how attackers can disguise phantom attacks so they won't be recognized by a human driver using black-box techniques. Finally, we demonstrate how attackers can leverage their findings and apply phantom attacks in just 125 ms via: 1) a projector mounted to a drone, and 2) an advertisement presented on a hacked digital billboard.

Given the lack of V2I, V2V, and V2P protocol implementation, Mobileye relies solely on computer vision algorithms and consists of two main components (see Fig. 5a): a video camera and a display which provides visual and audible alerts about the surroundings, as needed. Mobileye is also connected to the car's CAN bus and obtains other information (e.g., speed, the use of turn signals). Mobileye supports the following features: lane deviation warning, pedestrian collision warning, car collision warning, and road sign recognition. The accuracy of Mobileye's road sign recognition feature is stable, even in extreme ambient light or weather conditions, and is considered very reliable. Thus, we decided to focus our efforts on trying to fool this feature, with the aim of challenging Mobileye's most robust functionality (the functionality of some of their other features, like pedestrian collision warning, does not work in the dark/night [30]).

A. Analysis

In this subsection, we show how attackers can identify the various factors that influence the success rate of phantom attacks against a real ADAS/autopilot. We show how attackers can determine: 1) the diameter of the phantom road sign required to cover a given attack range, 2) the projection intensity required to cover a given attack range given the ambient light. Throughout the subsection we refer to a projected road sign as a phantom. Fig. 5b presents an illustration of the experimental setup used in the experiments described in this subsection. We used the Nebula Capsule projector, a portable projector with an intensity of 100 lumens and 854 x 480 resolution, which we bought on Amazon for $300 [31]. The portable projector was placed on a tripod located about 2.5 meters from a white screen (2.2 x 1.25 meters), and the phantom was projected onto the center of the screen. Mobileye is programmed to work only when the car is driving, so to test whether the phantom was captured by Mobileye, we drove the car (a Renault Captur 2017 equipped with Mobileye) in a straight line at a speed of approximately 25-50 km/h and observed its display.

Fig. 6: Examples of road signs with different opacity levels.

Experimental Setup: We started by demonstrating how attackers can calculate the diameter of the projected phantom road sign required to attack a driving car located a desired distance from the phantom. We tested six different sized phantoms of a road sign (20 km/h speed limit) with diameters smaller than our white screen (0.16, 0.25, 0.42, 0.68, 1.1, and 1.25 meters). We report the minimal and maximal distances for which the phantom road sign was detected as real by Mobileye.

Results: Fig. 7 presents the results from this set of experiments. The black points on the graph indicate the minimal and maximal distances for each phantom size. The gray area on the graph shows the detection range for the entire sample set. The red points indicate the midpoint between the maximal and minimal distance. First, we report that road signs with a diameter of less than 0.16 meters were not detected by Mobileye at all. Beyond the minimal and maximal distances, Mobileye ignores the phantoms and does not consider them at all. This is probably due to an internal mechanism that calculates the distance from a detected road sign based on the size of the road sign in pixels. Mobileye only presents a road sign to the driver if the sign is located within the specific distance range (1-5 meters) of the car [32]. If Mobileye detects a road sign which is very small, it interprets this as being far from the car; if the road sign is viewed by Mobileye as very large, Mobileye considers it too late to notify the driver about the sign. Mobileye only notifies the driver about a sign when the size of the detected road sign is within the desired size range (in terms of pixels). This is the reason why the red points on the graph maintain a linear behavior between the distance and the diameter. Our white screen is limited by its size (a height of 1.25 meters), so the maximal distance we were able to validate is 14.8 meters when using a phantom road sign with a diameter of 1.2 meters. However, distances beyond 14.8

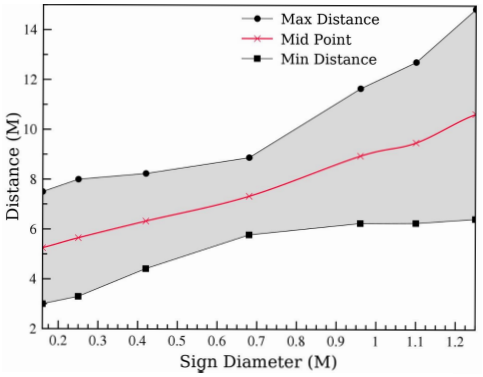

Fig. 7: Required diameter of phantom as a function of distance.

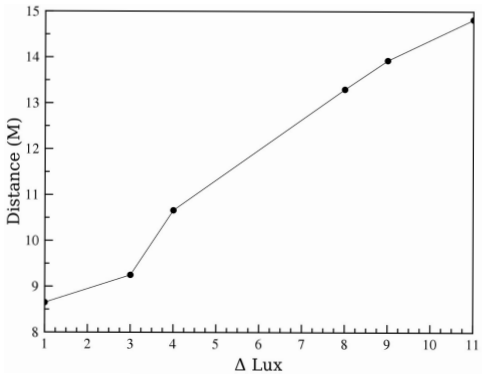

Fig. 8: Required intensity of projection (delta from ambient light) as a function of distance.

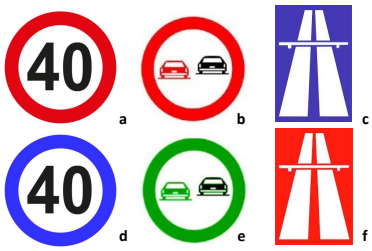

Fig. 9: Real road signs (a-c) and fake road signs with different outline color (d), different color of the sign's inner content and outline (e), different background color (f).

meters can be assessed by calculating the equation of the red linear curve by applying linear regression to the results. The function calculated is presented in Equation 1:

$$\textbf{Diameter (Range)} = 0.206 \times Range - 0.891 \quad (1)$$

Equation 1 results in the following: correlation coefficient (r) = 0.995, residual sum of squares (rss) = 0.008, and coefficient of determination (R^2) = 0.991. This equation can be used by attackers to calculate the phantom diameter required as a function of the distance between the phantom and the car they want to attack for a range $\geq$ 14.8 meters.

Experimental Setup: We continue by demonstrating how attackers can calculate the intensity of projection required to attack a driving car located at a desired distance from the phantom. Since light deteriorates with distance, a weak projection may not be captured by Mobileye's video camera beyond a given distance. In order to investigate this effect, we tested ten phantoms (a 20 km/h speed limit sign) with different opacity levels (10%, 20%,.., 100%). These phantoms created various projection intensities, as can be seen in Figure 6. For every projected phantom, we measured the intensity of projection (in LUX) on the white screen with a professional optical sensor, and the maximal distance from which Mobileye could detect this phantom. We also measured the ambient light (in LUX) on the white screen when no projection was applied. We calculated the difference between a measurement as it was captured on the white screen (in LUX) and the ambient light (in LUX) as it was captured on the white screen. We consider this difference the intensity the attacker must use to project a phantom on the surface with a given ambient light.

Results: Fig. 8 presents the results of this set of experiments. This graph indicates that 1) it is easier to apply phantom attacks at night (in the dark) with weak projectors, and 2) stronger projectors are needed to apply phantom attacks during the day. The graph shows a polynomial behavior in the distances evaluated. The required projection intensity for ranges that are beyond 14.8 meters can be calculated using Lagrange interpolation. The result is presented in Equation 2:

$$\begin{aligned}\Delta\ \textbf{Lux (Range=r)} = 0.01 \times r^5 - 0.90 \times r^4 + \\ 21.78 \times r^3 - 258.86 \times r^2 + 1525.72 \times r - 3566.76\end{aligned} \quad (2)$$

This equation can be used by attackers to calculate the projection intensity required as a function of the distance from the car they want to attack for distances $\geq$ 14.8 meters.

B. Disguising the Phantoms to Avoid Detection by Drivers

In this subsection, we demonstrate how attackers can disguise the phantoms so that they 1) aren't detected by a driver while he/she is driving the car, and 2) are misclassified by Mobileye.

Experimental Setup: First, we assess whether Mobileye is sensitive to the color of the sign. The motivation behind this set of experiments is that ambient light conditions can change the perception of the colors and hues of the captured road signs; we assumed that Mobileye contains an internal mechanism that compensates for this fact. We chose three road signs (presented in Fig. 9a-c) and verified that Mobileye detects their phantoms (projected in their real colors) as real road signs. Next, we projected a phantom of the same traffic sign outlined in a different color (presented in Fig. 9d), a phantom of a road sign with a different color of both its inner content and outline (Fig. 9e), and a phantom sign with a different background color (Fig. 9f).

Results: We found that Mobileye is not sensitive to color, since all of the phantoms presented in Fig. 9d-f were classified by Mobileye as real road signs. Based on this, we concluded that Mobileye either obtains the pictures in grayscale (digitally/physically) or its road sign recognition system ignores the detected road sign's color.

Experimental Setup: In this experiment, we aimed to determine the minimal projection time required to ensure that Mobileye detects the phantom. The projector we used works at the rate of 25 FPS. We created 25 videos that present a black background for 10 seconds. In each of the videos, we embedded a road sign (30 km/h speed limit) in a few consecutive frames (1,2,3...,25). Then, we projected the videos with the embedded road signs.

Results: We discovered that Mobileye is capable of detecting phantoms that are projected for 125 ms. We were unable to fool Mobileye with shorter projection times, likely due to an internal mechanism that validates a detected traffic sign against a consecutive number of frames that exceeds 125 ms or due to the low FPS rate of its video camera.

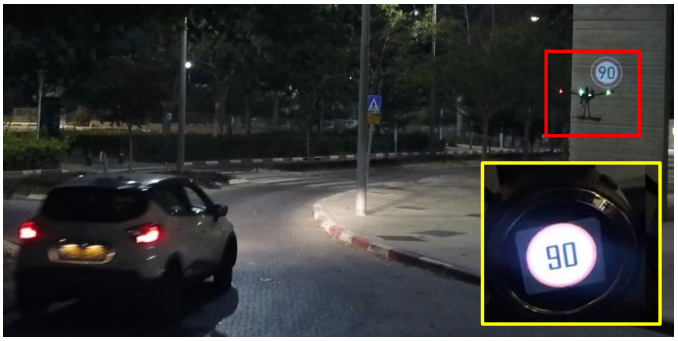

Fig. 10: A phantom (boxed in red) is projected on a building for 125 seconds from a drone; the phantom is captured by the passing Renault Captur, and Mobileye 630 PRO (boxed in yellow) identifies the phantom as real.

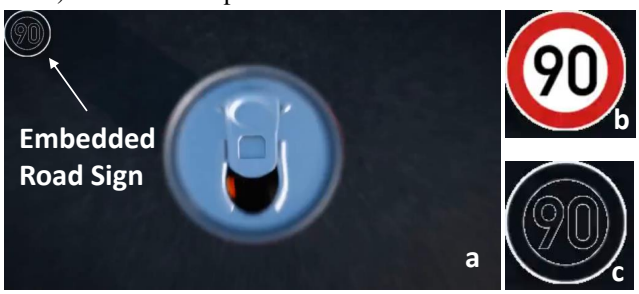

Fig. 11: Embedded road sign in a Coca-Cola advertisement (a): full, (b) outline, and (c) embedding.

C. Evaluation (Split Second Attacks)

We now show how attackers can leverage this knowledge to apply a phantom attack in a split second attack (125 ms) disguised as 1) a drone delivery, and 2) an advertisement presented on a digital billboard; in this case, the attacker's objective is to cause a driver that follows Mobileye notifications and adjusts his/her driving accordingly to drive recklessly.

Applying a Phantom Attack Using a Drone: This experiment was conducted on the premises of our university after we received the proper approvals from the security department. We mounted a portable projector on a drone (DJI Matrice 600) carrying a delivery box, so it would look like a drone delivery. In this experiment, our car (a Renault Captur equipped with Mobileye) was driven in an urban environment as the attacker operated the drone; the attacker positioned the drone in front of a building so the phantom speed limit sign (90 km/h) could be projected onto the wall so as to be in Mobileye's field of view. The attacker then waited for the car to arrive and projected the incorrect 90 km/h speed limit sign for 125 ms. A snapshot from the attack can be seen in Fig. 10, and the recorded video of the attack was uploaded.[1] Mobileye detected the phantom and notified the driver that the speed limit on this road is 90 km/h, although driving faster than 30 km/h on this road is not permitted. Attackers can also mount lightweight projectors onto much smaller drones; we were able to apply the same attack using an AAXA P2-A LED projector (weighs just 8.8 ounces) mounted on a DJI Mavic.

Applying a Phantom Attack via a Digital Billboard: Attackers can present phantoms via a desired digital billboard that is located near roads by hacking a billboard that faces the Internet (as was shown in [27, 28]) or by renting the services of a hacked billboard on the darknet. Attackers can disguise the phantom in an existing advertisement to make the attack more difficult to detect by drivers, pedestrians, and passengers. There are two methods of embedding phantoms within the content of an existing advertisement, as presented in Fig. 11: 1) a split second attack with full embedding in which a phantom is added to a video of an advertisement as is for 125 ms, and 2) a split second attack with outline embedding in which a phantom's outline is added to a video of an advertisement for 125 ms. Embedding a phantom within a video of an advertisement is a technique that attackers can easily apply using simple video editors, in order to disguise the attack as a regular advertisement presented on a digital billboard. We demonstrate these techniques using a random Coca-Cola ad. We added the content of a road sign (a speed limit of 90 km/h) to three consecutive frames in a Coke ad using the two methods mentioned above (snapshots from the compromised frames of the ads are presented in Fig. 11), and the ad was uploaded. [2] With the experimental setup seen in Fig. 5b, we projected the two advertisements on the white screen to simulate a scenario of a phantom attack applied via a hacked digital billboard. The road sign was detected by Mobileye in both cases, and the driver was notified that the speed limit was 90 km/h although driving faster than 50 km/h is not permitted on the road.

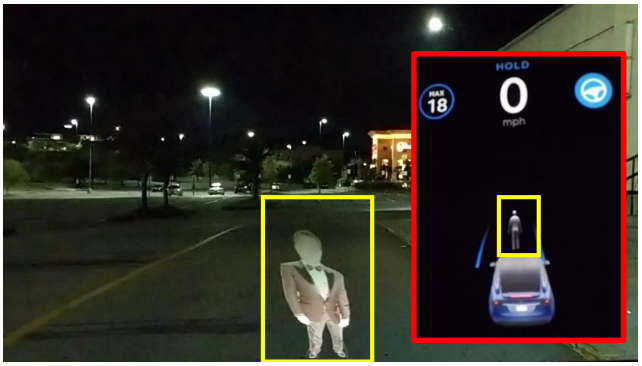

Fig. 12: Tesla's autopilot identifies the phantom as a person and does not start to drive. The red box contains a picture of the car's dashboard.

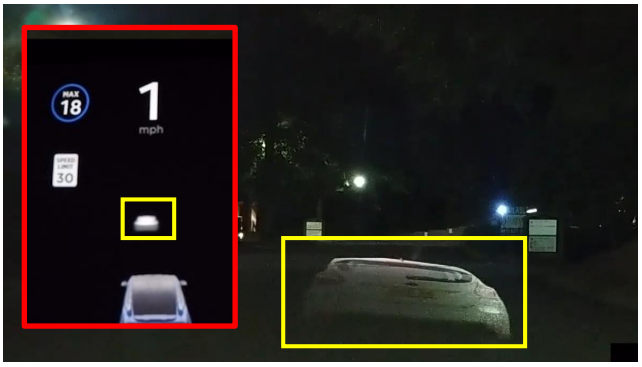

Fig. 13: The Tesla identifies the phantom as a car. The red box contains a picture of the car's dashboard.

[1] https://youtu.be/sMsaPMaHWfA

[2] https://youtu.be/sMsaPMaHWfA?t=31

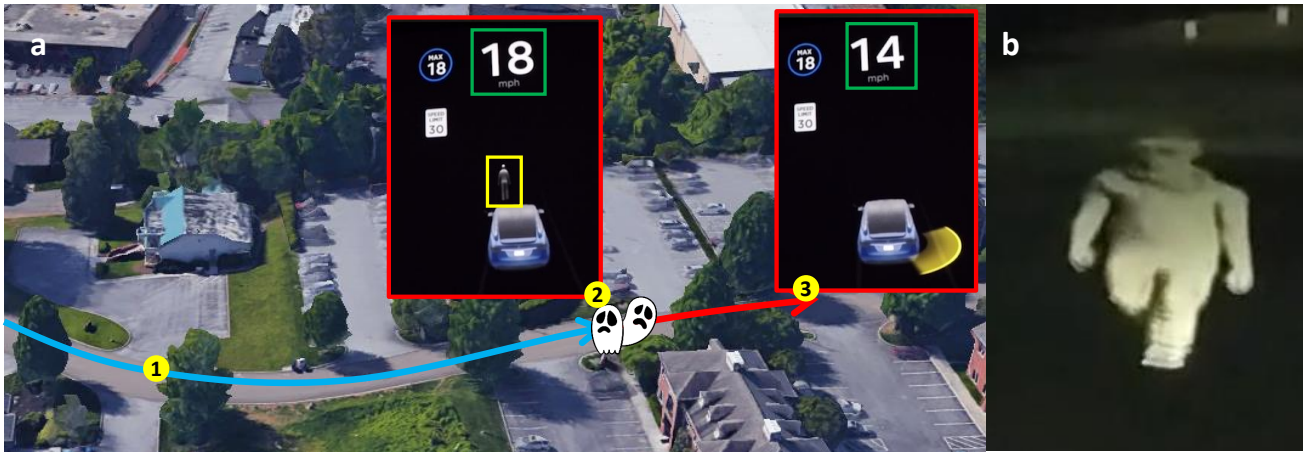

Fig. 14: Fooling the obstacle detection system: (a) A Tesla operating in cruise control (at location 1) approaches a phantom (at location 2). As a result, the car's collision avoidance system automatically triggers the brakes which reduces the car's speed from 18 MPH to 14 MPH while traveling to location 3. Snapshots of the car's dashboard (at locations 2 and 3) are presented in the red boxes. (b) The projected phantom as it was captured from a camera placed inside the car.

V. Phantom Attacks on Semi-Autonomous Cars (Tesla)

Autopilots have been deployed in semi-autonomous cars since the last quarter of 2015, and many car manufacturers have recently started to include them in level 2 automation cars [18]. Phantom attacks against semi-autonomous cars can trigger an unintended reaction from the autopilot that will result in a collision. Tesla's autopilot is considered statistically safer than a human driver [33], so we decided to test its robustness to phantom attacks in this study. All of the experiments described in this section were conducted with the Tesla Model X HW 2.5 which was manufactured in November 2017. The most recent firmware (2019.31.1) was installed at the time the experiments were conducted (September 2019). This model supports cruise control and autopilot functionalities. It also provides an anti-collision system to prevent the car from accidents with pedestrians, cars, etc.

First, we show that no validation is performed when an obstacle has been visually detected, likely due to a safety policy. Then, we show how attackers can exploit this fact and cause Tesla's autopilot to automatically and suddenly put on the brakes (by projecting a phantom of a person) and deviate from its path and cross the lane of oncoming traffic (by projecting a phantom of a lane). The set of experiments presented in this section was not performed in the same country that the experiments against Mobileye were performed. Flight regulations in the country that the experiments against Tesla were conducted prohibit the use of drones near roads and highways, so all of the attacks discussed in this section were applied via a portable projector (LG - CineBeam PH550 720p DLP projector) mounted on a tripod, although they could be implemented from a drone as was done in the experiments described in the previous section.

A. Fooling the Obstacle Detection System

In the absence of V2V and V2P protocols, Tesla's obstacle detection system obtains information about its surroundings from eight surround video cameras, twelve ultrasonic sensors, and front-facing radar [34]. Any obstacle (e.g., person, car, motorcycle, truck) detected by this system is presented to the driver on the dashboard. In this subsection, we evaluate the robustness of this system to phantom attacks.

Experimental Setup: We started by testing the system's robustness to a phantom of a picture of a person. Since the projector was placed on the sidewalk on the side of the road, we applied a morphing process to the picture, so it would look straight at the Tesla's front video camera (this process is described in the Appendix) and projected the morphed phantom on the road about one meter in front of the car. We then engaged the Tesla's autopilot.

Results: As can be seen from the results presented in Fig. 12, the Tesla's autopilot did not start to drive, since the phantom was detected as a real person (a picture of the car's dashboard appears in the red box, with the "person" detected boxed in yellow). We were only a bit surprised by this result, because the radar cross section of humans is dramatically lower than that of a car due to differences in their size, material, and orientation. This fact makes Tesla's front-facing radar measurements ambiguous and unreliable for the task of sensing people. In addition, ultrasound measurements are known to be effective for just short ranges (~ 5-8 meters) [12], so the obstacle detection system cannot rely on ultrasound measurements to sense people. These two facts can explain why the Tesla did not validate the existence of the phantom person detected by the front-facing camera with the front-facing radar and the set of ultrasound sensors, and thus considers it a real obstacle.

Experimental Setup: Next, we aimed at testing the obstacle detection system's response to a projected phantom of a car. We took a picture of a car and morphed it so it would look straight at the car's front video camera and projected the phantom car on the road about one meter in front of the Tesla.

Results: We were surprised to see that the depthless phantom car projected on the road was detected as a real car, as can be seen in Fig. 13. This is a very interesting result, because the phantom car was projected about one meter in front of the Tesla to the area in the driving environment which is covered

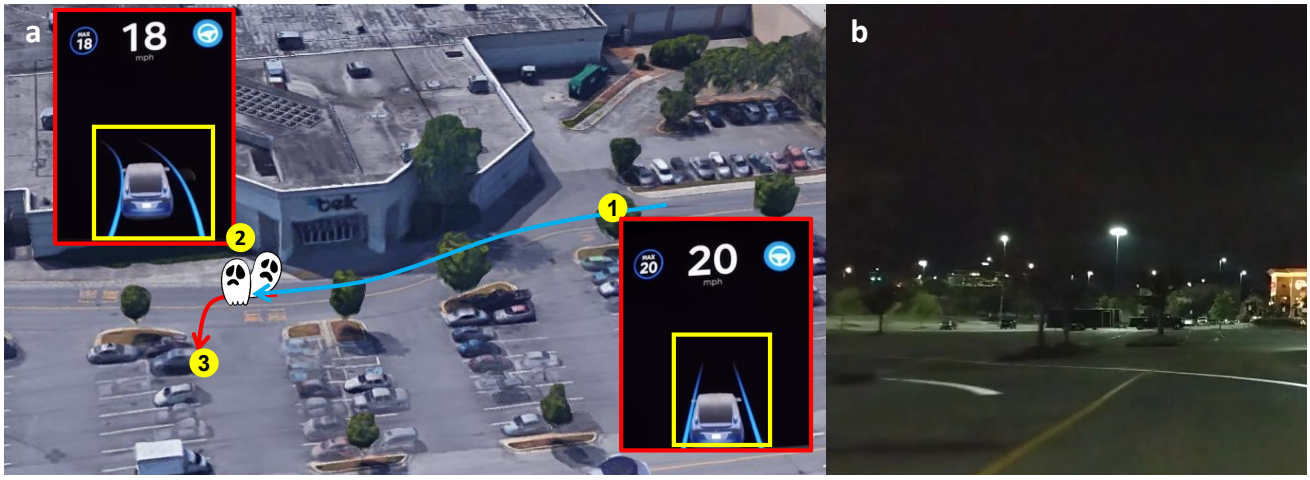

Fig. 15: Fooling the lane detection system: (a) A Tesla with its autopilot engaged (at location 1) approaches a phantom lane projected on the road (at location 2). As a result, Tesla's lane detection system causes the car to turn to the left, following the phantom white lane and crossing the real solid yellow lane, so that the car is driving across the lane of oncoming traffic to location 3 (the result is marked with a red arrow). Pictures of the car's dashboard at locations 1 and 2 are presented in the red boxes. (b) The projected phantom lanes as captured from a camera placed inside the car.

by the car's front-facing radar and ultrasound. Considering the fact that a car's radar cross section is very reliable, since cars are made of metal, the existence of visually identified cars can be validated with the front-facing radar. Based on this experiment, we concluded that Tesla's obstacle detection system does not cross-validate the existence of a visually detected obstacle with another sensor. When we contacted Tesla's engineers they did not share the reasons for our findings with us, but we assume that a "better safe than sorry" policy is implemented, i.e., if an obstacle is detected by one of Tesla's sensors with high confidence, Teslas are designed to consider it as real and stop rather than risking an accident.

Experimental Setup: With the observation noted above in mind, we show how attackers can exploit the "better safe than sorry" policy and cause Tesla's collision avoidance system to trigger sudden braking, by applying a phantom attack of a person. We drove the car to a deserted location to conduct this experiment. Fig. 14a presents the attack stages. At the beginning of the experiment we drove the car at a speed of 18 MPH (which is the slowest speed at which the cruise control can be engaged) and engaged the cruise control at location 1 in Fig. 14a. The cruise control system drove the car at a fixed speed of 18 MPH from location 1 to location 2. At location 2 a phantom of a person was projected in the middle of the road (as can be seen in Fig. 14b).

Results: A few meters before location 2 where the phantom was projected, the Tesla's obstacle detection system identified a person, as can be seen in Fig. 14a which presents a picture of the dashboard, as it appeared when the car reached location 2. Again, there was no validation with another sensor to detect fake objects, and the collision avoidance system caused the car to brake suddenly (at location 2), decreasing the car's speed from 18 MPH to 14 MPH by the time the car reached location 3. The experiment was recorded and uploaded.[3] While we performed this experiment carefully, implementing the attack when the car was driving at the lowest speed possible with cruise control (18 MPH), attackers can target this attack at semi/fully autonomous cars driving on highways at speeds of 45-70 MPH, endangering the passengers in the attacked car as well as those in other nearby cars.

B. *Fooling the Lane Detection System*

Tesla's lane detection system is used by its autopilot to steer the car safely. It is also used to notify the driver about lane deviations in cases in which the car is manually driven. This system shows the driver the detected lane on the dashboard. In the absence of deployed V2I protocols, Tesla's lane detection system is based purely on a video camera. In this subsection, we test the robustness of Tesla's lane detection system to a phantom attack.

Experimental Setup: We demonstrate how attackers can cause Tesla's autopilot to deviate from its path and cross the lane of oncoming traffic by projecting phantom lanes. We created a phantom consisting of two lane markings which gradually turn to the left, using a picture that consists of two white lanes on a black background. We drove the car on a road with a single lane in each direction. The two lanes were separated by a solid yellow line, as can be seen in Fig. 15a. We engaged the autopilot functionality (at location 1), and the car was steered by the autopilot on the road towards location 2, traveling toward the phantom that was projected at location 2 (the driving route is indicated by the blue arrow in Fig. 15a). The two red boxes are pictures of the car's dashboard taken at each of the locations. A video demonstrating this experiment was recorded and uploaded.[4] A picture of the road taken from the driver's seat showing the white phantom lanes that cross the real solid yellow is presented in Fig. 4b.

Results: As can be seen from the red box at location 2 in Fig. 15a, Tesla's lane detection system detected the phantom lanes turning toward the left as the real lanes. The autopilot turned

[3] https://youtu.be/sMsaPMaHWfA?t=43

[4] https://youtu.be/sMsaPMaHWfA?t=77

the car toward the left, following the phantom white lanes and crossing the real yellow solid lane (the path is marked with the red arrow in the figure) and driving across the lane of oncoming traffic until we put on the brakes and stopped the car at location 3 in Fig. 15a. Tesla's lane detection system was unable to differentiate between the real yellow lane and the white phantom lanes although they were different colors.

In the Appendix we demonstrate another application of phantom attacks against Tesla's stop sign recognition system. We show how a Tesla considered a phantom stop sign that was projected on a road that does not contain a stop sign. Since Tesla's stop sign recognition system is experimental and is not considered a deployed functionality, we chose to exclude this demonstration from the paper.

VI. Detecting Phantoms

Phantom attacks work well because autonomous systems consider the camera sensor alone in order to avoid making a potentially fatal mistake (e.g., failing to detect a pedestrian in the street). Since it makes sense to rely on just the camera sensor in these situations, we propose that an add-on software module be used to validate objects identified using the camera sensor.

As discussed in section III-A, ADASs and autonomous systems often ignore a detected object's context and authenticity (i.e., how realistic it looks). This is because the computer vision model is only concerned with matching geometry and has no concept of what fake objects (phantoms) look like. Therefore, the module should validate the legitimacy of the object given its context and authenticity. In general there are five aspects which can be analyzed to detect a phantom image:

Size. If the size of the detected object is larger or smaller than it should be, the detected object should be disregarded, e.g., a road sign which is not regulation size. The size and distance of an object can be determined via the camera sensors alone through stereoscopic imaging [35].

Angle. If the angle of the object does not match its placement in the frame, it is indicative of a phantom. The skew of a 2D object facing a camera changes depending on which side of the frame it is situated. A phantom may be projected at an angle onto a surface, or the surface may not be directly facing the camera. As a result, the captured object may be skewed in an anomalous way.

Context. If the placement of the object is impossible or simply abnormal, it is indicative of a phantom, e.g., a road sign that does not have a post or a pedestrian 'floating' over the ground.

Surface. If the surface of the object is distorted or lumpy, or has features which do not match the typical features of the detected object, then it is likely a phantom, e.g., when a phantom is projected onto a brick wall or an uneven surface.

Lighting. If the object is too bright given its location (e.g., in the shade) or time of day, then it can be assumed to be a phantom. This can be determined passively through image analysis or actively by shining a light source onto the object (e.g., flash photography).

In the following subsections, we present one possible implementation this countermeasure module which considers the last three aspects. We focus on detecting projected phantom road signs, because we can evaluate our approach in conjunction with eight state-of-the-art road sign detectors. We also note that road sign location databases do not mitigate road sign phantom attacks. This is because temporary road signs are very common. For example, caution, speed, and stop signs in construction zones, and stop signs on school buses. Finally, although we focus on road signs, the same approach can be applied to other types of phantom objects (pedestrians, cars, etc.).

A. The Detection Module

Overall, our module works as follows. First, the module receives a cropped image of a road sign from the on-board object detector. The module uses a model to predict whether or not the object's setting makes sense and whether or not the object is realistic and reports the decision back to the system. The module can be used on every detected object or only on those which the controller deems urgent (e.g., to avoid an imminent collision with a person).

To predict whether or not an object is a phantom or real, we could build a simple convolutional neural network (CNN) classifier which receives a cropped image of a road sign and then predicts whether it is real or fake, however this approach would make the neural network reliant on specific features and thus would not generalize to phantoms projected on different surfaces or made using different types of projectors. For example, the light intensity of a road sign is an obvious way to visually distinguish between a real and projected sign. As a result, a neural network trained on the entire sign would primarily focus on this aspect alone and make errors with phantoms projected on different surfaces or made using different projectors (not used in the training set).

To avoid this bias, we utilize the committee of experts approach used in machine learning [36] in which there is an ensemble of models, each of which has a different perspective or capability of interpreting the training data. Our committee consists of three deep CNN models, each focusing on a different aspect (see Fig. 16 for the model parameters). The models receive a cropped image of a road sign. The models then judge if the sign is authentic and contextually makes sense:

Context Model. This CNN receives the context: the area surrounding the road sign with the road sign itself erased. Given a context, the model is trained to predict whether a sign is appropriate or not. The goal of this model is to determine whether the placement of a sign makes sense in a given location.

Surface Model. This CNN receives the sign's surface: the cropped sign alone in full color. Given a surface, the model is trained to predict whether or not the sign's surface is realistic. For example, a sign with tree leaves or brick patterns inside is not realistic, but a smooth one is.

Light Model. This CNN receives the light intensity of the sign. The light level of a pixel is the maximum value

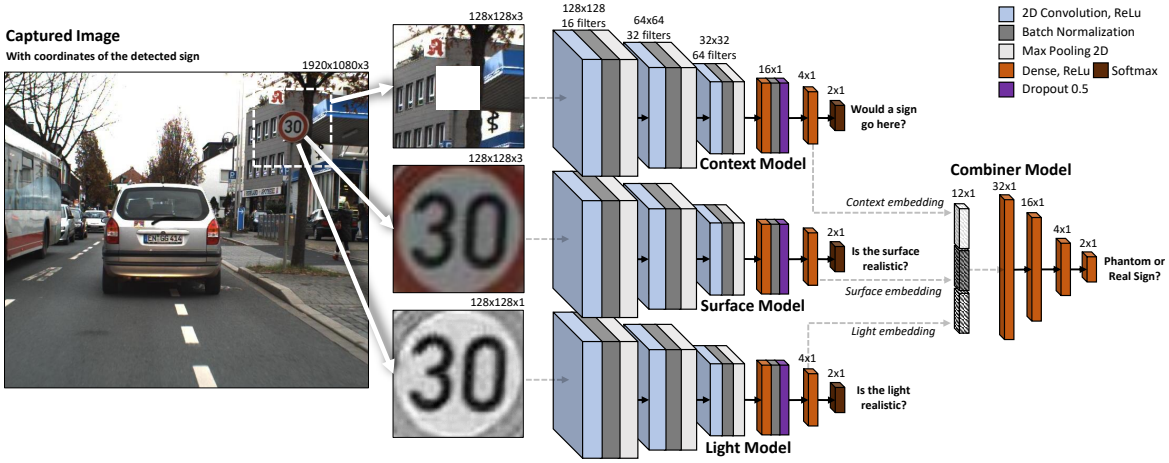

Fig. 16: The proposed phantom image detection module. When a frame is captured, (1) the on-board object detector locates a road sign, (2) the road sign is cropped and passed to the Context, Surface, and Light models, and (3) the Combiner model interprets the models' embeddings and makes a final decision on the road sign (real or fake).

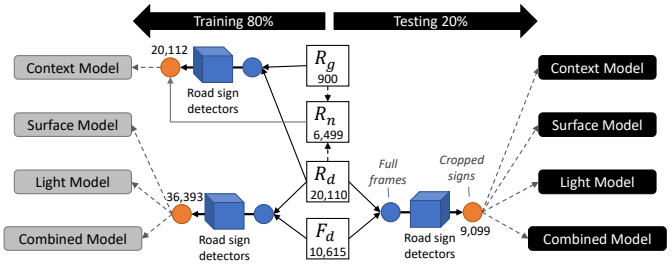

Fig. 17: A diagram showing how the training and testing data was prepared from our data sources, and the number of instances.

of the pixel's RGB values (the 'V' in the HSV image format). The goal of this model is to detect whether a sign's lighting is irregular. This can be used to differentiate real signs from phantom signs, because the paint on signs reflects light differently than the way light is emitted from projected signs.

To make a prediction on whether or not a sign is real or fake, we combine the knowledge of the three models into a final prediction. As an image is passed through each of the models, we capture the activation of the fifth layer's neurons. This vector provides a latent representation (embedding) of the model's reasoning as to why it thinks the given instance should be predicted as a certain class. We then concatenate the embeddings to form a summary of the given image. Finally, a fourth neural network is trained to classify the image as real or fake using the concatenated embeddings. The entire neural network has $860,216$ trainable parameters.

B. Experimental Setup

To evaluate the proposed detector, we combined three datasets containing driver seat perspective images (see Fig. 17 for a summary). The first is the GTSRB German traffic sign dataset [37] denoted as (R_g). The second is a dataset we recorded from a dash cam while driving at night for a three hour period in a city, which is denoted as (R_d). The third is another dash cam dataset we recorded while driving an area where phantom road signs were projected, denoted as (F_d). In the F_d dataset, we projected 40 different types of signs in a loop onto nine different surfaces while driving by. We then used eight state-of-the-art road sign detectors (described in [26]) to detect and crop all of the road signs in R_g, R_d, and F_d. The cropped road signs were then passed as input to the models.

To train the context model, we needed examples which do not contain signs (denoted as R_n) to teach the model the improper placement of signs. For this dataset we cropped random areas from R_g and R_d such that the center of the cropped images does not contain a sign.

The Context, Surface, and Light models were trained separately, and then the Combiner model was trained on their embeddings. Regarding the data, $\%80$ was used to train the models, and the remaining $\%20$ was used to evaluate them. To reduce bias, the evaluation samples taken from F_d contained phantom projections on surfaces which were not in the training set. Training was performed on an NVIDIA Titan X (Pascal) GPU for 100 epochs.

C. Experimental Results

1) Model Performance: In Fig. 19 we present the receiver operating characteristic (ROC) plot and the area under the ROC for of the Context, Surface, Light, and Combiner models. The ROC shows the true positive rate (TPR) and false positive rate (FPR) for every possible prediction threshold, and the AUC provides an overall performance measure of a classifier (AUC=1 : perfect predictions, AUC=0.5 : random guessing).

There is a trade-off when setting a threshold. This is because a lower threshold will decrease the FPR but often decrease the TPR as well. In our case, it is critical that our module predicts real signs as real every time. This is because the vast majority of signs passed to our module will be real. Therefore, even a very small FPR would make the solution impractical. For this reason, in Table II we provide the TPR and FPR of the models

Fig. 18: Examples of disagreements between the models for real and fake road signs which led to correct predictions.

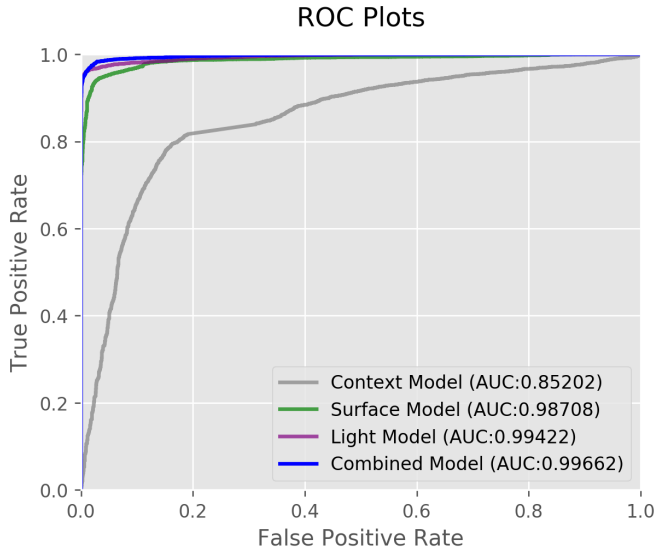

Fig. 19: The receiver operating characteristic curve and AUC measure for each model. A larger AUC is better.

when the threshold is set to 0.5 (the default for softmax) and for the threshold value at which the FPR is zero.

2) The Committee at Work: In Table II, we note that the Combiner model performs better than any of the individual models alone. In Table II we also show that there is no combination of models that performs as well as the combination consisting of all three models. This means that each aspect (context, surface, and light) contributes a unique and important perspective on the difference between a real and phantom road sign.

This is important since in order for the committee of experts approach to be effective there must be some disagreements between the models. In Fig. 18, we provide some visual examples of the disagreements which resulted in a correct prediction by the Combined model. In some cases, a model simply misclassifies although the evidence is clear. For example, sometimes the Context model does not realize that the sign is on the back of a truck (bottom right corner of Fig. 18). In other cases, a model misclassifies simply because its perspective does not contain the required evidence. For example, sometimes the Context model finds it abnormal for a sign to be floating on a horizontal structure (top left corner of Fig. 18). Regardless, in all cases the other models (experts) provided a strong vote of confidence contrary to the erroneous opinion, and this ultimately led to the correct prediction.

However, the committee of experts approach is not perfect. Fig. 20 provides an example of a case in which the Combiner model failed. Here the sign is real, but only the Context model identified it as such. However, due to motion blur, the other models strongly disagreed.

3) Module Performance: Our module filters out untrusted (phantom) road signs detected by the on-board object detector. Since there are many different implementations of road sign detectors, one detector may be fooled by a specific phantom while another would not. Therefore, to determine how effective our module is within a system, we evaluated phantom attacks on eight state-of-the-art road sign detectors [26]. We measured

TABLE II: The TPR and FPR of the Countermeasure Models at Different Thresholds. **C**: Context, **S**: Surface, **L**: Light

		C	S	L	C+S	C+L	S+L	C+S+L
Threshold @ 0.5	**TPR**	0.778	0.960	0.972	0.969	0.971	0.973	**0.976**
	FPR	0.151	0.072	0.039	0.071	0.035	0.021	**0.022**
Threshold @ [FPR=0]	**TPR**	0.006	0.724	0.870	0.726	0.884	0.896	**0.902**
	FPR	0	0	0	0	0	0	**0**

TPR: % of phantoms detected, **FPR**: % of road signs misclassified as phantoms

TABLE III: The Disagreement Between the Models

	Disagreement on…	C vs S	C vs L	S vs L	C vs S vs L
Threshold @ 0.5	Phantoms	0.3%	0.4%	0.5%	0.6%
	Real Signs	29.7%	32.0%	6.2%	33.9%
Threshold @ [FPR=0]	Phantoms	21.0%	95.8%	74.8%	95.8%
	Real Signs	0%	0%	0%	0%

TABLE IV: The Detection Rates Using s.o.t.a Road Sign Detectors

		Detection Rate *given real signs* (baseline)	*given fake signs* (attack)	
			Countermeasure?	
			No	***Yes***
Road Sign Detector	faster_rcnn_inception_resnet_v2	70.4%	92.57%	0.36%
	faster_rcnn_resnet_101	70.7%	87.62%	0.67%
	faster_rcnn_resnet50	70.2%	97.53%	0.56%
	faster_rcnn_inception_v2	70.4%	99.02%	1.19%
	rfcn_resnet101	71.2%	93.00%	0.49%
	ssd_inception_v2	62.8%	59.06%	0.96%
	ssd_mobilenet_v1	58.5%	40.70%	1.41%
	yolo_v2	69.4%	79.22%	5.70%

Fig. 20: An example of a false positive, where the Combiner model failed due to a disagreement.

the attack success rate on a detector as the percent of phantom signs identified in F_d. In Table IV we present the attack success rates on each detector before and after applying our countermeasure.[5] We also provide each of the detector's accuracy with real road signs (R_g) as a baseline. The results show that the detectors are highly susceptible to phantom attacks and that our countermeasure provides effective mitigation.

In summary, with all models combined as a committee and the FPR tuned to zero, the TPR is 0.9. This means that our countermeasure is reliable enough for daily usage (is not expected make false alarms) and will detect a phantom 90% of the time. However, our training set only contained several hours of video footage. For this solution to be deployed, it is recommended that the models be trained on much larger datasets with phantoms projected from other devices as well. We also suggest that additional models which consider size and angle be considered.

VII. Responsible Disclosure

This research shows that the absence of deployed vehicular communication systems limits the ability of semi/fully autonomous cars to validate virtual perception and that the car industry doesn't take phantom attacks into consideration. We have nothing against Tesla or Mobileye, and the reason that their products were used in our experiments is because their products are the best and most popular products available on the market.

We shared our findings with Mobileye's bug bounty via email and received the following response: *"There was no exploit, no vulnerability, no flaw, and nothing of interest: the road sign recognition system saw an image of a street sign, and this is good enough, so Mobileye 630 PRO should accept it and move on."* We agree with Mobileye regarding their claim that there wasn't an exploit or vulnerability. We do not consider phantom attacks bugs, since they don't exploit poor code implementation. Instead, phantom attacks pose a perceptual challenge to ADASs and autopilots which are unable to validate their findings with a third party due to the lack of deployed vehicular communication systems. However, we disagree with Mobileye's claims that there is "nothing of interest and no flaw," because Mobileye 630 PRO considered a phantom as a legitimate street sign. Considering the fact that Mobileye's technology is currently integrated in semi-autonomous cars (e.g., the Tesla with HW 1) which will eventually be programmed to stop when a stop sign is recognized, the inability of Mobileye's technology to distinguish between a phantom and a real stop sign may be exploited by attackers to target semi-autonomous cars driving on highways at speeds of 45-70 MPH in order to trigger sudden braking using a phantom stop sign.

We also shared our findings with Tesla's bug bounty via email. Tesla decided to dismiss all of our findings due to the fact that the experiments that are presented in the Appendix, were performed after enabling the experimental stop sign recognition system, claiming: *"We cannot provide any comment on the sort of behavior you would experience after doing manual modifications to the internal configuration - or any other characteristic, or physical part for that matter - of your vehicle"*. Tesla engineers removed the experimental code from the firmware about two weeks after we contacted them about this matter. While we did indeed enable the stop sign recognition feature in the experiments presented in the Appendix, we did not influence the behavior that led the car to steer into the lane of oncoming traffic or suddenly put on the brakes after detecting a phantom.

VIII. Discussion

One might argue that the deployment of vehicular communication systems will prevent attackers from applying phantom attacks in the wild, however this is unlikely to be the case. We don't believe that full deployment of vehicular communication systems that support V2V, V2I, V2P, and V2X protocols will cause the manufacturers of semi/full autonomous cars to abandoned the "better safe than sorry" policy, because they cannot rely on the assumption that if no validation was obtained for a detected visual object, then the object must be a phantom. There are other reasons why there might not be validation. V2P communication relies on the fact that pedestrians are carrying devices (e.g., smartphones) and requires that they carry such devices with them. If a pedestrian's device is turned off (e.g., drained battery) or the pedestrian isn't carrying a device (e.g., forgot it at home), validation based on V2P communication isn't possible. Since car manufacturers cannot rely on the assumption that they will be able to validate the presence of pedestrians with V2P protocols, they must implement a "better safe than sorry approach" policy. This is also the reason why car manufacturers cannot completely rely on V2V validation in the case of a visually detected car - not all cars contain a fully functioning V2V device. The complete deployment of V2I systems around the world might limit the attackers' ability to project a phantom lane or road sign, but the full deployment of such systems might not be practical, since doing so is very expensive, and currently most places around the world don't utilize V2I systems at all.

An interesting observation made during this study is that the perceptual challenge that phantoms create is, in some cases, an intelligence discriminator between people and machines. Distinguishing between a projected object and a real object is something that in some cases can be solved by examining the context. This fact can be used to perform a Turing test [38] for machine vs. human perception with an interesting application in areas such as CAPTCHA, i.e., detecting Internet sessions launched by bots.

[5]Here the Combiner model's threshold is set to the value where the FPR=0.

REFERENCES

[1] T. Lee, "Waymo tells riders to get ready for fully driverless rides," https://arstechnica.com/cars/2019/10/waymo-starts-offering-driverless-rides-to-ordinary-riders-in-phoenix/, 2019.

[2] A. B. Simona Shemer, "Self-driving spin: Riding in an autonomous vehicle around tel aviv," https://nocamels.com/2019/06/autonomous-vehicle-yandex-tech/, 2019.

[3] W. Chen, *Vehicular communications and networks: Architectures, protocols, operation and deployment*. Elsevier, 2015.

[4] E. C. Eze, S. Zhang, and E. Liu, "Vehicular ad hoc networks (vanets): Current state, challenges, potentials and way forward," in *2014 20th International Conference on Automation and Computing*. IEEE, 2014, pp. 176–181.

[5] K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, C. Xiao, A. Prakash, T. Kohno, and D. Song, "Robust physical-world attacks on deep learning models," *arXiv preprint arXiv:1707.08945*, 2017.

[6] C. Sitawarin, A. N. Bhagoji, A. Mosenia, M. Chiang, and P. Mittal, "Darts: Deceiving autonomous cars with toxic signs," *arXiv preprint arXiv:1802.06430*, 2018.

[7] D. Song, K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, F. Tramer, A. Prakash, and T. Kohno, "Physical adversarial examples for object detectors," *USENIX Workshop on Offensive Technologies (WOOT 18)*, 2018.

[8] S.-T. Chen, C. Cornelius, J. Martin, and D. H. P. Chau, "Shapeshifter: Robust physical adversarial attack on faster r-cnn object detector," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2018, pp. 52–68.

[9] Y. Zhao, H. Zhu, R. Liang, Q. Shen, S. Zhang, and K. Chen, "Seeing isn't believing: Towards more robust adversarial attack against real world object detectors," in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '19. New York, NY, USA: ACM, 2019, pp. 1989–2004. [Online]. Available: http://doi.acm.org/10.1145/3319535.3354259

[10] N. Morgulis, A. Kreines, S. Mendelowitz, and Y. Weisglass, "Fooling a real car with adversarial traffic signs," *arXiv preprint arXiv:1907.00374*, 2019.

[11] J. Petit, B. Stottelaar, M. Feiri, and F. Kargl, "Remote attacks on automated vehicles sensors: Experiments on camera and lidar," *Black Hat Europe*, vol. 11, p. 2015, 2015.

[12] C. Yan, W. Xu, and J. Liu, "Can you trust autonomous vehicles: Contactless attacks against sensors of self-driving vehicle," *DEF CON*, vol. 24, 2016.

[13] keen labs, "Tencent keen security lab: Experimental security research of tesla autopilot," https://keenlab.tencent.com/en/2019/03/29/Tencent-Keen-Security-Lab-Experimental-Security-Research-of-Tesla-Autopilot/, 2019.

[14] Regulus, "Tesla model 3 spoofed off the highway - regulus navigation system hack causes car to turn on its own," https://www.regulus.com/blog/tesla-model-3-spoofed-off-the-highway-regulus-researches-hack-navigation-system-causing-car-to-steer-off-road/.

[15] A. Driving, "Levels of driving automation are defined in new sae international standard j3016: 2014," *SAE International: Warrendale, PA, USA*, 2014.

[16] D. Dutta, "Advanced driver aids might be mandatory by 2022 says nitin gadkari: Will make indian roads safer!" https://www.financialexpress.com/auto/car-news/advanced-driver-aids-might-be-mandatory-by-2022-says-nitin-gadkari-will-make-indian-roads-safer/1312218/, 2018.

[17] "Advanced driver assistance systems," https://ec.europa.eu/transport/road_safety/sites/roadsafety/files/ersosynthesis2016-adas15_en.pdf, 2016.

[18] D. DeMuro, "7 best semi-autonomous systems available right now," https://www.autotrader.com/best-cars/7-best-semi-autonomous-systems-available-right-now-271865, 2018.

[19] C. Forrest, "The x-factor in our driverless future: V2v and v2i," https://www.zdnet.com/article/the-x-factor-in-our-driverless-future-v2v-and-v2i/, 2018.

[20] Y. Cao, C. Xiao, B. Cyr, Y. Zhou, W. Park, S. Rampazzi, Q. A. Chen, K. Fu, and Z. M. Mao, "Adversarial sensor attack on lidar-based perception in autonomous driving," *arXiv preprint arXiv:1907.06826*, 2019.

[21] C. Miller and C. Valasek, "Remote exploitation of an unaltered passenger vehicle," *Black Hat USA*, vol. 2015, p. 91, 2015.

[22] S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, S. Savage, K. Koscher, A. Czeskis, F. Roesner, T. Kohno *et al.*, "Comprehensive experimental analyses of automotive attack surfaces." in *USENIX Security Symposium*, vol. 4. San Francisco, 2011, pp. 447–462.

[23] K. Koscher, A. Czeskis, F. Roesner, S. Patel, T. Kohno, S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham *et al.*, "Experimental security analysis of a modern automobile," in *2010 IEEE Symposium on Security and Privacy*. IEEE, 2010, pp. 447–462.

[24] keen labs, "Car hacking research: Remote attack tesla motors," https://keenlab.tencent.com/en/2016/09/19/Keen-Security-Lab-of-Tencent-Car-Hacking-Research-Remote-Attack-to-Tesla-Cars/, 2016.

[25] L. Wouters, E. Marin, T. Ashur, B. Gierlichs, and B. Preneel, "Fast, furious and insecure: Passive keyless entry and start systems in modern supercars," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, pp. 66–85, 2019.

[26] A. Arcos-Garcia, J. A. Alvarez-Garcia, and L. M. Soria-Morillo, "Evaluation of deep neural networks for traffic sign detection systems," *Neurocomputing*, vol. 316, pp. 332 – 344, 2018.

[27] "Men hack electronic billboard, play porn on it," https://arstechnica.com/tech-policy/2019/10/men-hack-electronic-billboard-play-porn-on-it/, 2019.

[28] S. Tutorials, "Hacking digital billboards," https://securitytutorials.co.uk/hacking-digital-billboards/.

[29] M. Kyriakidis, C. van de Weijer, B. van Arem, and R. Happee, "The deployment of advanced driver assistance systems in europe," *Available at SSRN 2559034*, 2015.
[30] Mobileye, "Mobileye 6-series - user manual," http://www.c2sec.com.sg/Files/UserManualMobileye6.pdf.
[31] "Nebula capsule," https://www.amazon.com/Projector-Anker-Portable-High-Contrast-Playtime/dp/B076Q3GBJK.
[32] Scantips, "Calculate distance or size of an object in a photo image," https://www.scantips.com/lights/subjectdistance.html.
[33] Tesla, "Tesla vehicle safety report," https://www.tesla.com/VehicleSafetyReport.
[34] "Tesla autopilot," https://www.tesla.com/autopilot?utm=.
[35] Y. M. Mustafah, R. Noor, H. Hasbi, and A. W. Azma, "Stereo vision images processing for real-time object distance and size measurements," in *2012 International Conference on Computer and Communication Engineering (ICCCE)*, July 2012, pp. 659–663.
[36] J.-N. Hwang and Y. H. Hu, *Handbook of neural network signal processing*. CRC press, 2001.
[37] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, "The German Traffic Sign Recognition Benchmark: A multi-class classification competition," in *IEEE International Joint Conference on Neural Networks*, 2011, pp. 1453–1460.
[38] C. Machinery, "Computing machinery and intelligence-am turing," *Mind*, vol. 59, no. 236, p. 433, 1950.
[39] Wikipedia, "Tesla autopilot," https://en.wikipedia.org/wiki/Tesla_Autopilot.
[40] T. M. Club, "Hw2.5 capabilities," https://teslamotorsclub.com/tmc/threads/hw2-5-capabilities.95278/page-76#post-2386639.
[41] "What is the government's commitment to gps accuracy?" https://www.gps.gov/systems/gps/performance/accuracy/.

IX. Appendix

A. Morphing a Picture for a Projection

Figure 21 presents three locations. Location 1 is the location of the front facing camera of the targeted car. Location 2 shows where the attack will be implemented from (this can be a sidewalk, a drone, etc.). Location 3 indicates where the attacker wishes to project the phantom. When projecting a picture from location 2 toward location 3 (at a non-90 degree horizontal/vertical angle), the picture loses its form and looks distorted when it is captured by the front facing camera of the targeted car (positioned at location 1). In order to project a picture that will look straight at the car's forward facing camera, we performed the following steps:

- Downloading a Picture - We downloaded a picture of an object that the car's obstacle detection system can identify. Currently, Tesla signals the driver about pedestrians, cars, trucks, motorcyclists, etc. The original picture used is presented in Figure 22a.

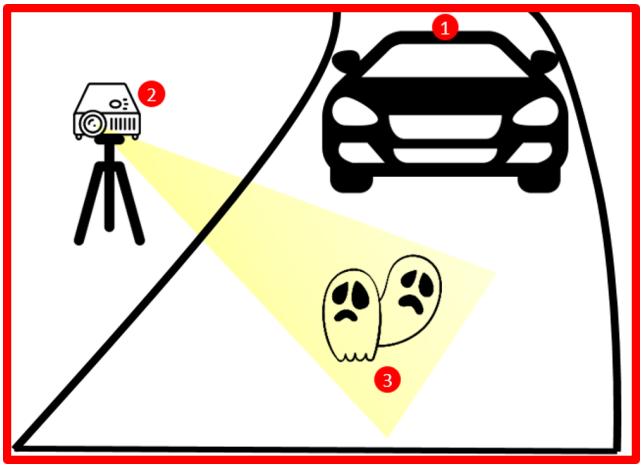

Fig. 21: Morphing Process Attacked Scene: Location 1 is where the front facing camera of the targeted car is located. Location 2 is where the attacker will perform the attack. Location 3 shows where the attacker wants to project the phantom.

- Brightening the Picture - We brightened the picture in order to emphasize its projection on the road. This is an optional step. The brightened picture is presented in Figure 22b.
- Projecting the Picture from the Front Facing Camera - We placed the portable projector at location 1, near the car's front facing camera, and projected it on the road.
- Taking a picture of the Projected Object from Location 2 - We took a picture of the projected object from location 2, which is the place that we would like the attacker to apply the attack. Figure 22c shows how Figure 22b was captured on the road from a smartphone's camera located at location 2.
- Morphing the Original Object Using GIMP - We morphed Figure 22b using GIMP according to Figure 22c and created a new picture. The result is presented in Figure 22d.
- Projecting the Morphed Picture from Location 2 - Finally, we projected the picture presented in Figure 22d from location 2 to location 3. The result as it was captured from a camera that was placed in the driver's seat is presented in Figure 22e.

B. Fooling Tesla's Road Sign Recognition System

In this subsection, we evaluate the robustness of Tesla's stop sign recognition system to phantom attacks. In the absence of V2I protocols, Tesla HW 2.5 uses a geolocation mechanism to obtain the information needed as the car is driving; this mechanism uses an internal database (without the use of the video camera) which is queried with location and orientation data in order to obtain the necessary information regarding traffic laws on a given road. In order to obtain the location and orientation data required to query the database, the car calculates its location via a GPS sensor over time and infers the driving orientation on a road. This new functionality is used

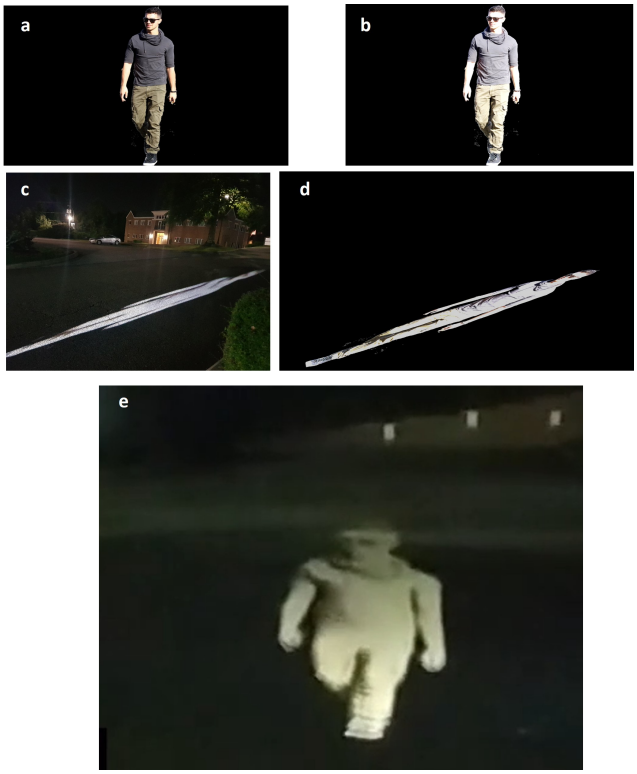

Fig. 22: Morphing Process Outputs: (a) the original picture of the obstacle, (b) the brightened picture, (c) the object projected from the car's front facing camera as it was captured from a camera placed where we want to apply the attack, (d) the morphed picture, and (e) the object as it was captured from the driver's seat.

by Tesla to obtain speed limits on roads. This mechanism, which is only available in Tesla's new models (HW 2, 2.5 and 3), replaced Tesla's old autopilot (HW 1) which relied on Mobileye technology and was based on visual detection [39]. The new mechanism was designed to decrease false detection rates as a result of unintended edge cases (e.g., speed limits were detected from the back of trucks, parallel streets, parking lot entrances [40]) and attacks [5–10].

Tesla recently deployed firmware which supports stop sign and traffic light recognition. This functionality was recently integrated into Tesla's HW 2.0,2.5 and 3.0, and it is considered experimental and is disabled by default. It can be enabled by changing a system variable in Tesla's computer, and we did just that. A stop sign/traffic light recognition system requires a high level of physical recognition accuracy which cannot be obtained via standard GPS devices due to their known average error under open skies (up to 7.8 m, with 95% probability [41]). Given this limitation, Tesla cannot rely purely on the geolocation mechanism as in the case of speed limit sign recognition. Another reason is because in cases in which there is no line on the road indicating where to stop, the location of the stop sign/traffic light itself is considered the point at which to stop. In order to compensate for this fact, Tesla uses an additional means and considers a stop sign/traffic light (and presents a signal to the driver to this effect) only if the stop sign/traffic light has also been recognized by the car's object detection mechanism via the front video camera. If one of the following conditions does not hold, the car will not consider a stop sign/traffic light: 1) the stop sign/traffic light was not detected by the front video camera, 2) the car is not located within a geolocated area with an intersection that known to contain a stop sign/traffic light, or 3) the orientation of the car is not facing the stop sign/traffic light.

With that in mind, we show how an attacker can fool Tesla's stop sign recognition system, so that it considers a phantom stop sign projected on a road that does not contain a stop sign when the car is in fact located 50 meters from a nearby intersection that contains a traffic light. Fig. 23a shows a road (marked with a yellow arrow) that ends at a intersection that contains a traffic light with stop line. When the car approached the intersection, the traffic light recognition system informed us about the traffic light visually detected. We then looked for a nearby road with the same orientation that didn't contain a stop sign. We decided to conduct our experiments on the road marked with a blue arrow in Fig. 23a; we first validated that a stop sign is not detected by the Tesla on this road by driving down the road. The Tesla did not recognize a stop sign, and as a result, no indication appeared on the dashboard. The selection of this road allowed us to orient the car such that it was traveling in the direction of a nearby intersection that contains traffic light but on a different road.

Experimental Setup: We started by trying to determine the radius of the geolocation mechanism by finding maximal distance from the intersection that Tesla's stop sign recognition system considers a phantom as a real stop sign. We drove the car on the road marked with the blue arrow and projected a phantom of a stop sign on a white board at various distances (50, 60, and 70 meters) from the original stop sign.

Results: The Tesla identified the projected stop signs as real on places located at a distance of 50 meters or less from the intersection. Phantoms projected at distances of 60 and 70 meters from the intersection were not considered by Tesla's stop sign recognition system as real. Interestingly, although the global average user range error of GPS measurements is $\leq$ 7.8 meters, with 95% probability [41], the radius of the geolocation mechanism is greater than that by six times.

Experimental Setup: Next, we decided to test whether Tesla's stop sign recognition system considers colorless projection of phantom stop signs as real. The motivation behind this set of experiments is the same as in the Mobileye experiments described earlier: ambient light conditions can change the way colors and hues are perceived by the system of a captured stop sign, so we assumed that Tesla cars contain an internal mechanism that compensates for this fact. In order to conduct this experiment, we projected two phantoms of colorless stop signs (presented in Figs. 23c and d) on a wall located 50 meters from the real stop sign (marked with a phantom in Fig. 23a). We drove the car on the road marked with the blue arrow in Fig. 23a.

Results: As in the Mobileye case, we found that Tesla's stop sign recognition system does not take the color of the stop sign into account. It detected all of the projected stop signs as real stop signs, regardless of the presence of color

and issued notifications about them.

Experimental Setup: Next, we aimed to test which features are more important to Tesla's stop sign recognition system. Our analysis of the two phantoms in Figs. 23c and d, shows that they consist of two components: the hexagon shape and the word "STOP." Using the same experimental setup as the previous experiment, we projected two more phantoms, one consisting of only the word "STOP" (see Fig. 23e) and another consisting of an empty hexagon on the wall.

Results: While we expected that Tesla's stop sign recognition system would consider the hexagon shape and ignore the word "STOP," the results of this experiment were surprising. The word "STOP" was recognized as a stop sign, while the empty hexagon was ignored. This experiment confirms that the most dominant feature recognized by Tesla's stop sign recognition system is the word "STOP."

Experimental Setup: Next, we decided to evaluate whether a stop sign projection can be disguised so it won't be seen by a human driver (in the case of a semi-autonomous car). We created phantom videos that present regular stop signs for 250ms, 125ms, 82ms, and 41ms. With the same experimental setup described above, we projected each video while we were driving the car on the road marked with the blue arrow.

Results: We found that the minimal time period required for Tesla's stop sign recognition system to identify a phantom is 125 ms. We were unable to fool Tesla's stop detection system with projection periods shorter than 125 ms.

As mentioned earlier, the Tesla stop sign recognition system is currently experimental and we are confident that when it is officially deployed it won't misclassify phantom the word "STOP" as real stop sign. However, attackers might still be able to fool a robust stop sign recognition system by applying a phantom projection attack using the original stop sign (a red hexagon with the word "STOP") because: 1) The Tesla must be able to detect stop signs visually in cases in which a stop line does not exist or in cases of temporal stop signs (e.g., stop sign extended from a school bus driver's window), so Tesla cars will need to rely on a video camera for detecting a stop sign, leaving the option for attackers to project phantom stop signs. In addition, while Tesla's engineers did not reveal the reason why they decided to use a radius of 50 meters for their geolocation mechanism, we believe that the reason for this decision is the following: While the GPS measurement's average error is is $\leq$ 7.8 meters with 95% probability [41], there are various cases (e.g., tunnels) in which the error of the obtained GPS measurements can be greater than the average error ($\geq$ 7.8). Limiting the geolocation area to 7.8 meters will probably result in many false negatives, i.e., a detected stop sign/traffic light will not be considered by the system as a real due to incorrect GPS measurements. Again, the absence of V2I protocols can be exploited by attackers to cause greater harm. While Tesla's current stop sign recognition mechanism does not cause the car to stop, full autonomous cars must have the functionality that stops the car at a detected stop sign. Given that the geolocation radius will probably be beyond 7.8 meters, attackers can target autonomous cars driving at speeds of 45-70 MPH on a highway by projecting phantom stop signs in specific locations (e.g., near intersections that contain stop signs and located at a distance which is less than the geolocation's radius), causing autonomous cars to stop in the middle of a highway.

Fig. 23: Fooling the stop sign recognition system: Each of the four phantoms (b-e), projected for just 125 ms, were recognized by Tesla's stop sign recognition system. The phantoms were projected on a white wall located 50 meters from a nearby intersection that contained a real stop sign.