Phantom of the ADAS: Securing Advanced Driver-Assistance Systems from Split-Second Phantom Attacks

Ben Nassi[1], Yisroel Mirsky[1,2], Dudi Nassi[1], Raz Ben-Netanel[1], Oleg Drokin[3], Yuval Elovici[1]
[1] Ben-Gurion University of the Negev, [2] Georgia Tech, [3] Independent Researcher
{nassib, yisroel, razx, nassid, elovici}@post.bgu.ac.il, green@linuxhacker.ru

Abstract

In this paper, we investigate "split-second phantom attacks," a scientific gap that causes two commercial advanced driver-assistance systems (ADASs), Telsa Model X (HW 2.5 and HW 3) and Mobileye 630, to treat a depthless object that appears for a few milliseconds as a real obstacle/object. We discuss the challenge that split-second phantom attacks create for ADASs. We demonstrate how attackers can apply split-second phantom attacks remotely by embedding phantom road signs into an advertisement presented on a digital billboard which causes Tesla's autopilot to suddenly stop the car in the middle of a road and Mobileye 630 to issue false notifications. We also demonstrate how attackers can use a projector in order to cause Tesla's autopilot to apply the brakes in response to a phantom of a pedestrian that was projected on the road and Mobileye 630 to issue false notifications in response to a projected road sign. To counter this threat, we propose a countermeasure which can determine whether a detected object is a phantom or real using just the camera sensor. The countermeasure (*GhostBusters*) uses a "committee of experts" approach and combines the results obtained from four lightweight deep convolutional neural networks that assess the authenticity of an object based on the object's light, context, surface, and depth. We demonstrate our countermeasure's effectiveness (it obtains a TPR of 0.994 with an FPR of zero) and test its robustness to adversarial machine learning attacks.

Keywords

Advanced Driver-Assistance Systems; Security; Split-Second Phantom Attacks; Neural-Networks

ACM Reference Format:

Ben Nassi, Yisroel Mirsky, Dudi Nassi, Raz Ben-Netanel, Oleg Drokin, Yuval Elovici. 2020. Phantom of the ADAS: Securing Advanced Driver-Assistance Systems from Split-Second Phantom Attacks. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security (CCS'20), November 9–13, 2020, Virtual Event, USA*. ACM, New York, NY, USA, 16 pages. https://doi.org/10.1145/3372297.3423359

1 Introduction

After years of research and development, advanced driver assistance systems (ADASs) are now be used to support/replace manual driving with automated functionality [35, 53]. ADASs utilize artificial intelligence (AI) models which process data obtained from various sensors in real-time in order to steer the car, recognize traffic signs, trigger an alert when the car deviates from its lane, etc. Since these systems rely heavily on image input, their robustness against adversarial machine learning attacks has been tested in various studies in the last few years [10, 17, 31, 42, 54, 55, 66]. These studies have contributed to improved understanding of the limits of AI models and the need to increase their robustness against attacks that can endanger the lives of drivers, passengers, and pedestrians.

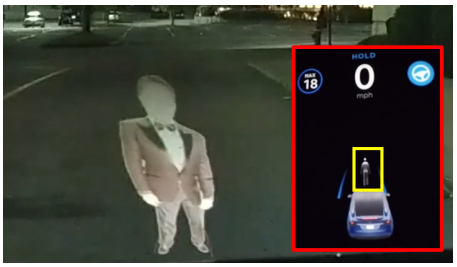

Figure 1: In practice, despite the fact that the Tesla Model X (HW 2.5) is equipped with radar and ultrasonic sensors, it recognizes a depthless person (phantom) projected on the road in front of it as a real obstacle due to its safety policy.

In this paper, we identify a limitation of AI models that can be exploited by attackers to perform a new type of attack called a "split-second phantom attack." This attack exploits a weakness of ADASs where (1) projected or digitally displayed imagery is perceived as a real object (i.e., phantom imagery), and (2) the imagery only has to appear briefly (for a few milliseconds) in order to be detected by the ADAS. Split-second phantom attacks fool AI models by presenting an image very briefly, for the minimum amount of time required to be captured by the camera, thus remaining imperceptible to the human eye. This attack raises great concern, because unskilled attackers can use split-second phantom attacks against ADASs with little fear of getting caught, because (1) there is no need to physically approach the attack scene (a drone can project the image or a digital billboard can display it), (2) there will be no identifying physical evidence left behind, (3) there will likely be few witnesses, since the attack is so brief, and (4) it is unlikely that the target of the attack will try to prevent the attack (by taking control of the vehicle), since he/she won't notice anything out of the ordinary.

Split-second phantom attacks are optical illusions that challenge computer vision algorithms. Since some commercial ADASs rely purely on imagery input obtained from video cameras (e.g., Mobileye 630) which does not have any depth, the authenticity of the detected objects cannot be verified at all, causing such ADASs to

CCS '20, November 9–13, 2020, Virtual Event, USA

ACM ISBN 978-1-4503-7089-9/20/11…$15.00
https://doi.org/10.1145/3372297.3423359

treat a phantom as a real object. One might argue that the authenticity of phantoms can be determined by commercial ADASs that use sensor fusion by cross-correlating the camera sensor with data obtained from depth sensors (e.g., radar, LiDAR, and ultrasonic sensors), however we show that the most advanced semi-autonomous vehicle (the Tesla Model X) resolves the disagreement between the camera and the integrated depth sensors (radar and ultrasonic sensors) regarding a phantom by treating the phantom as real, despite the fact that it does not have any depth (see Fig. 1). As a result, split-second phantom attacks are de facto a scientific gap.

To mitigate these attacks, we propose *GhostBusters*: a committee of machine learning models which validates objects detected by the on-board object detector. The *GhostBusters* can be deployed on existing ADASs without the need for additional sensors and does not require any changes to be made to existing road infrastructure. It consists of four lightweight deep CNNs which assess the realism and authenticity of an object by examining the object's reflected light, context, surface, and depth. A fifth model uses the four models' embeddings to identify phantom objects. This approach outperforms the baseline method and achieves an AUC of over 0.99 and a TPR of 0.994 with a threshold set to an FPR of zero. When applying the countermeasure to seven state-of-the-art road sign detectors, we were able to reduce the attack success rate from 99.7-81.2% without our module to 0.01% when using our module. Finally, we perform an ablation study and find that by separating the models, our solution is less reliant on specific features. This makes it more resilient than the baseline model and robust against potential adversarial attacks.

In summary, this paper makes the following contributions: (1) We expose a new type of attack against ADASs that does not leave any forensic evidence, is difficult to observe, and can be applied remotely, each of which limits the risk to the attacker. (2) We demonstrate the split-second phantom attacks on two popular commercial ADASs (Mobileye 630 and Tesla Model X HW 2.5/3) in real-life scenarios. (3) We propose a robust and efficient software-based countermeasure which can identify the attack using the camera sensor alone. The trained models, datasets, and source code are available online.[1]

The paper is structured as follows: First, we discuss why split-second phantom attacks represent a scientific gap (Section 3), discuss the significance of the threat model (Section 4), analyze the factors that influence on the attack (Section 5), and demonstrate these attacks on Mobileye 630 and the Tesla Model X HW 2.5/3 (Section 6). Then, we propose our countermeasure, evaluate it, and analyze its performance (Section 7). Finally, we discuss our findings and suggest future work (Section 8).

2 Background, Scope & Related Work

In this section, we provide the necessary background on advanced driver-assistance systems and review attacks against them.

Advanced driver-assistance systems are defined as *"vehicle-based intelligent safety systems which could improve road safety in terms of crash avoidance, crash severity mitigation, and protection and post-crash phases."* [11]. ADASs are integrated into cars and range from no automation (level 0) to fully automated systems (level 5) [15]. ADASs consist of sensors, actuators, and decision-making algorithms. Decision-making algorithms (e.g., collision avoidance systems) rely on AI models that base their actions (regarding, for example, notifications to drivers, alerts, steering, braking, etc.) on sensor input. In recent years, many studies have demonstrated how these AI models can be tricked using sensor attacks.

Visual spoofing attacks were demonstrated by [10, 17, 31, 42, 54, 55, 66]. Several studies have demonstrated adversarial machine learning attacks in the physical world against traffic sign recognition algorithms [10, 17, 54, 55, 66] by: (1) embedding 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 [54], and (2) adding a physical artifact (e.g., stickers, graffiti) that looks innocent to the human eye but can mislead traffic sign recognition algorithms [10, 17, 55, 66]. Two studies demonstrated adversarial machine learning attacks in the physical world against commercial ADASs by: (1) printing traffic signs that contain negligible changes that fool the road sign detection mechanism of HARMAN's ADAS [42], and (2) by placing stickers on the road that caused Telsa's autopilot to deviate to the lane of oncoming traffic [31].

Other attacks against sensors were demonstrated by [7, 49, 51, 57, 65]. One study [65] demonstrated spoofing and jamming attacks against Tesla's radar and ultrasonic sensors which caused the car to misperceive the distance to nearby obstacles. Another study [51] showed that GPS spoofing can cause Tesla's autopilot to navigate in the wrong direction. Black-box and white-box adversarial attacks on LiDAR were recently presented by [7, 49, 57].

3 The Scientific Gap

In this section, we define phantoms and split-second phantom attacks, and discuss why they are considered scientific gaps.

We define a phantom as a depthless visual object used to deceive ADASs and cause these systems to perceive the object and consider it real. A phantom object can be created by a projector or be presented via a digital screen (e.g., billboard). The depthless object presented/projected is made from a picture of a 3D object (e.g., pedestrian, car, truck, motorcycle, traffic sign). The phantom is intended to trigger an undesired reaction from an ADAS. In the case of an automation level 0 ADAS, the reaction would be a driver notification about an event (e.g., maximal speed allowed) or even an alarm (e.g., collision avoidance). For an automation level 2 ADAS, the phantom could trigger an automatic dangerous reaction from the car (e.g., sudden braking). A split-second phantom attack is a phantom that appears for a few milliseconds and is treated as a real object/obstacle by an ADAS. There are two fundamental reasons why ADASs are susceptible to split-second phantom attacks:

1) **Perceptual weakness**: Phantoms challenge the perception of computer vision algorithms, because such algorithms do not base their decisions on personal experience, authenticity, or other factors that are taken into account by humans. Most algorithms are the output of a training phase performed using a dataset that consists of real objects. As a result, computer vision algorithms are not trained to identify phantoms and ignore them. Instead, these algorithms classify objects with high confidence if parts of the object (e.g., geometry, edges, outline) are similar to the training examples. Specifically, these algorithms do not take the following aspects into consideration: (i) Context - An object's location and local context within the frame are not taken into account. Fig. 2 presents an example where a phantom traffic sign projected on the

[1] https://github.com/ymirsky/GhostBusters

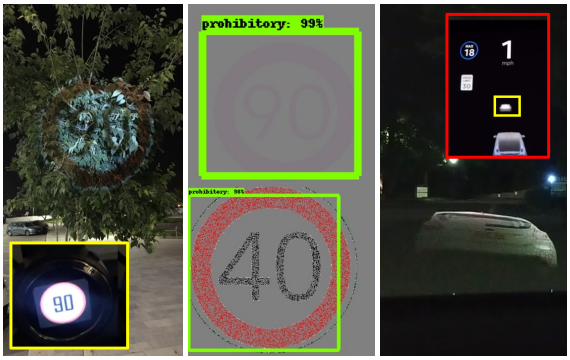

Figure 2: Left: a commercial ADAS (Mobileye 630) classifies a projection of a traffic sign on a tree. Middle: the Faster_rcnn_inception_v2 [3] classifies a traffic sign that was created by reducing the green component of the RGB value of the background color in Δ=3 from (128,128,128) to (128,125,128) and using only 25% of the pixels. Right: Tesla Model X recognizes a depthless phantom of a car as real.

tree is treated as a real traffic sign by a commercial ADAS (Mobileye 630). (ii) Color - An object's color is not taken into account. Fig. 2 shows how Faster_rcnn_inception_v2 [3] classifies a phantom traffic sign that was created by reducing the green component in Δ=3 from the gray background (128,128,128) to (128,125,128) and is made from gray colors. This is also true for other state-of-the-art traffic sign recognition algorithms (see Appendix 1 for more details). (iii) Texture - An object's texture is not taken into account. Fig. 2 shows how Faster_rcnn_inception_v2 [3] recognizes a traffic sign composed of 25% of the pixels (which were randomly drawn) from the original traffic sign as a real traffic sign. This is also true for other s.o.t.a traffic sign recognition algorithms (see Appendix 1 for more details). Fig. 2 shows how a projected traffic sign whose surface consists of the leaves of a tree is reported by Mobileye 630.

Therefore, although phantoms are perceived by humans as obvious fakes (defective, skewed, etc.), an ADAS will classify a phantom simply because its geometry matches its training examples.

2) **Disagreement between sensors**: Phantoms create AI models that rely on sensor fusion, an edge case where there is complete disagreement between an obstacle that was detected via a video camera and other depth sensors. In the face of phantoms, the Tesla trusts imagery input over radar input and considers depthless projected/presented objects as real obstacles. There are several reasons for this phenomenon which occurs despite the fact that sensor fusion is employed by the car: (1) Pedestrian detection via radar becomes unreliable from a range of 10-15 meters in real scenes due to reflections from other objects (humans have a weak radar cross-section) [19]. This fact requires the car to put more trust in video cameras for which pedestrian detection is considered highly reliable [19]. (2) Large areas of the vehicle are only covered by video cameras (see Fig. 13 in Appendix), so the car does not have any way of validating its findings with a depth sensor in these areas. (3) It is common practice in the automobile industry to program cars to ignore stationary objects identified by radar in order to focus on moving objects [56]. The implications of this fact are mentioned in the Tesla manual [60], other vendors' manuals [56], and in the NTSB report [47] where drivers are warned that the car's autopilot may not detect stationary obstacles. As a result, the car places greater confidence in other sensors to detect stationary obstacles (see how Tesla detects a stationary phantom car in Fig. 2).

Table 1: Mapping an attack to a desired result.

Desired Result	Triggered Reaction from the ADAS	Type of Phantom	Location of Appearance
Traffic collision	Sudden braking	Stop sign	Building, billboard
		No entry sign	
		Obstacle (e.g., car)	Road
Reckless/illegal driving behavior	Fast driving	Speed limit	Building, billboard
Traffic jam	Decreasing driving speed	Speed limit	
	Stopping	No entry sign	
Directing traffic to specific roads	Avoiding certain roads	No entry sign	

Disagreement regarding the existence of an obstacle among sensors must be resolved by an ADAS in real-time (in a few milliseconds), and an incorrect decision can endanger pedestrians, drivers, and passengers. Since car manufacturers want to prevent loss of life and accidents, a "safety first" approach is implemented for autonomous driving [27], and phantoms are treated as real objects by cars with integrated radar and ultrasonic sensors (see Fig. 2).

One may think that split-second phantom attacks can be dismissed by increasing the threshold for the period that a detected object must appear in the video stream before the object is treated as real by an ADAS. Considering the fact that ADASs must be able to react to a real obstacle that suddenly appears on the road, setting a minimal threshold that is too high (e.g., one second) may make it impossible for the ADAS to react in time. We do not consider split-second phantom attacks a bug, because they are not the result of poor code implementation. We consider phantoms a scientific gap, because they exploit a perceptual weakness that cannot be effectively addressed without decreasing the level of safety.

4 Threat Model

In this section, we present remote threat models for applying split-second phantom attacks and discuss their significance.

We consider an attacker as any malicious entity with the intention of creating chaos by performing a split-second phantom attack that will result in unintended car behavior. Table 1 maps the desired result (e.g., reckless driving behavior), triggered reaction (e.g., fast driving), and the phantom required (e.g., traffic sign).

In this threat model, the attacker simply presents the phantom for a few milliseconds, starting and stopping it whenever he/she wishes. The attacker can embed the phantom in a digital advertisement presented on a hacked digital billboard that faces the Internet [34, 61] or project the phantom (e.g., on a road, building, etc.) via a portable projector mounted to a drone. Since the phantom is presented for just a few milliseconds, it is very difficult for humans to detect. The attacker can amplify the desired effect by using a swarm of drones or a set of hacked digital billboards.

Split-second phantom attacks are a class of visual spoofing. Their significance with respect to related work [10, 17, 42, 54, 55, 66] is that they: (1) can be applied remotely by using a drone equipped with a portable projector or by embedding objects into digital advertisements presented on hacked digital billboards that face the Internet [34, 61] and are located near roads, (2) do not leave any

physical forensic evidence at the attack scene that can be used by investigators to trace the incident to the attackers, (3) do not require any complex preprocessing or professional sensor spoofing skills, and (4) allow attackers to manipulate ADAS obstacle detection systems (e.g., by using phantoms of cars, pedestrians, etc.).

5 Attack Analysis

In this section, we analyze the various factors that influence the success rate of split-second phantom attacks against a commercial ADAS: the distance and duration of the projected phantom.

The experiments presented in this section are performed against two commercial ADASs: (1) **Mobileye 630 PRO** is considered the most advanced external ADAS for automation level 0-1 cars. Mobileye sells its ADAS technology to 70% of car manufacturers [33]. In the rest of this section we refer to Mobileye 630 PRO as Mobileye. Mobileye relies solely on computer vision algorithms and consists of a video camera (that is installed on the front windshield) and a display which provides visual and audible alerts. Mobileye supports road sign recognition, car collision warning, and other features [39]. We installed Mobileye in a Renault Captur 2017 via their official local distributor. (2) **The Tesla Model X (HW 2.5/3)** is considered the most advanced commercial semi-autonomous car. In the rest of this section we refer to the Tesla Model X (HW 2.5/3) as Tesla. This model relies on eight surround video cameras, 12 ultrasonic sensors, and front-facing radar (see Fig. 13 in Appendix). These models support autopilot driving, cruise control, collision avoidance, and stop sign recognition. The most recent firmware was installed at the time the experiments were conducted.

5.1 Influence of Distance

There are two factors that determine whether the attack can be applied from a given distance: (1) The projection's intensity. If the projection's intensity is too weak, the phantom will not be captured by the video camera, because light deteriorates with distance. (2) The size of the phantom. If the projected object is too small, it will be ignored by the ADAS, because it is located far from the ADAS and is not considered an immediate obstacle or road sign that needs to be considered. We analyze how these factors influence the success of a split-second phantom attack against an ADAS with respect to the distance between the phantom and the ADAS.

5.1.1 Experimental Setup: We used a white screen to project the phantoms. We created 10 road signs with different opacity levels (10%, 20%,.., 100%). In addition, we created six different sized phantoms of a road sign with diameters smaller than our white screen (0.16, 0.25, 0.42, 0.68, 1.1, and 1.25 meters). We projected the 10 phantom road signs on the white screen via a Nebula Capsule projector, a portable projector with an intensity of 100 lumens and 854 x 480 resolution [2].

5.1.2 Experiment Protocol: We started by placing the ADAS (Mobileye and Tesla) one meter from the screen. In the first experiment, we projected the phantoms that we created (with varied brightness levels) on the screen. The 10 phantoms projected resulted in various projection intensities on the white screen, since they were based on various opacity levels. We kept projecting the phantoms (starting with the phantom with the strongest projection intensity) until the phantom was not identified by the ADAS. We measured the intensity of projection (in lux) on the white screen with a professional optical sensor and the distance between the ADAS and the white screen. In the second experiment, we projected the six different sized phantoms (starting with the largest one) until the phantom was not identified by the ADAS. Next we increased the distance between the screen and the ADAS by two meters and performed the experiments again.

The first dataset obtained at the end of these experiments mapped the minimal projection intensity (in lux) to the maximal distance at which the ADAS was able to detect the phantom (meters). We calculated the difference (Δ) between a measurement as it was captured on the screen (in lux) and the ambient light (in lux) as it was captured on the white screen. We consider this difference to be the additional projection intensity the attacker must use to project a phantom on the surface with respect to a given ambient light value. The second dataset mapped the minimal phantom size to the maximal distance at which the ADAS detects the phantom.

5.1.3 Results & Conclusions: Fig. 3 presents graphs that map the minimal projection intensity to the maximal range at which it can be detected by the ADAS. The results indicate that: (1) As the distance between the ADAS and the phantom increases, a stronger projector is required so the phantom will be detected by the ADAS. This is due to the fact that light deteriorates with distance. (2) It is easier to apply phantom attacks at night (in the dark) with weak projectors than during the day. This is due to the fact that the ambient light at night is zero lux, and during the day it is 1,000 - 2,000 lux. (3) Weaker intensities of projection (lux) are required to attack Tesla from a greater distance than that required to attack Mobileye. This can be explained by the fact that Mobileye relies on a single video camera while the area in front of the Tesla is covered by three HD video cameras (see Fig. 13 in Appendix).

Fig. 3 presents graphs that map the minimal phantom size to the maximal range at which it can be detected by Mobileye and Tesla. As can be seen from the graphs, the results indicate that: (1) As the distance between the ADAS and the phantom increases, a larger phantom is required so the phantom will be detected by the ADAS as a nearby road sign. (2) The graphs obtained for Mobileye and Tesla show similar (almost identical) behavior. This can be explained by the fact that the size of the projected road sign (as it was captured by the video camera of the ADAS) is indicative of its location with respect to the ADAS and whether it should be considered by the ADAS. In Appendix 2, we explain how the phantom size and projection intensity required to attack an ADAS that is located beyond the ranges measured in our experiments can be calculated without additional experiments.

5.2 Influence of the Duration of the Phantom

Here we analyze how the duration of a split-second phantom attack influences the success rate of the attack. A phantom is the output of a digital visualization instrument (billboard, projector) which is sampled by a CMOS sensor of an ADAS video camera. We assume that attackers do not have any prior knowledge regarding: (1) the FPS rate of the video camera of the attacked ADAS, and (2) how the detection model works. We also assume that attackers cannot synchronize their digital visualization instrument to the video camera of the attacked ADAS (due to the nature of the attack).

5.2.1 Experimental Setup: We analyze the success rate of the attack as it is applied via a 24 FPS digital visualization instrument (the projector we used). In order to do so, we created 24 videos. Each of

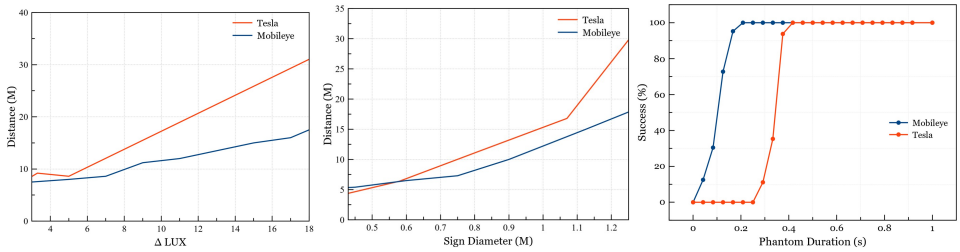

Figure 3: Left to right: Mapping distance between an ADAS to a required intensity of projection (left) and phantom size (middle). Attack's success rate as a function of its duration (right).

the 24 videos was used to evaluate the success rate of a split-second phantom attack for a specific duration of time. In order to simulate a real-world scenario where the phantom may appear in any frame, we shifted the time that the phantom appeared in each second in the video: The first video was created for a phantom that is projected for one frame (a duration of 41 milliseconds) and consists of 24 seconds, where in each second only one frame presents a phantom. In the first second of the first video, a phantom is presented in the first frame (the other 23 frames are empty). In the next second of the first video, a phantom is presented in the second frame (the other frames are empty). This pattern continues until the twenty-fourth second of the video, in which a phantom is presented in the twenty-fourth frame (the other frames are empty). The second video was created for a phantom that is projected for two consecutive frames (a duration of 82 milliseconds) and consists of 23 seconds, where in each second two consecutive frames present a phantom. In the first second of the second video, a phantom is presented in the first and second frames (the other 21 frames are empty). In the next second of the first video, a phantom is presented in the second and the third frames (the other 21 frames are empty). This pattern continues until the twenty-third second of the video, in which a phantom is presented in the twenty-second and twenty-third frames (the other 21 frames are empty). This pattern continues until the twenty-fourth video which was created for a phantom that is projected for 24 consecutive frames (a duration of one second) and is one second long. We created 24 videos of the split-second phantom attack of stop signs (for Tesla) and speed limit signs (for Mobileye). We placed a projector (Nebula 24 FPS [2]) in front of a white screen that was placed two meters in front of the ADAS.

5.2.2 Experiment Protocol: We tested Tesla and Mobileye with the videos that we created. Each video was presented 10 times. We report the success rate of the attack for each projection duration (the number of times that the phantom was detected by the ADAS divided by the total number of times the phantom appears).

5.2.3 Results & Conclusions: Fig. 3 presents the results of this set of experiments. An analysis of Fig. 3 reveals four interesting observations: (1) Phantom road signs with a duration that exceeds 416 milliseconds are detected by the tested ADAS with accuracy of 100%. This can be explained by the nature of the ADAS which is required to respond quickly to road signs. (2) The graphs present sigmoid behavior in the tested range: at short projection durations, the graphs show a 0% success rate which continues until a specific point at which the success rate quickly increases to 100% and is maintained at long projection durations. We concluded that the tested ADASs are configured to consider an object if it appears in a predefined threshold of t consecutive frames in the captured video stream, probably in order to decrease false positive detections. As a result, when a phantom appears in at least t consecutive frames in the captured video stream it is considered by the ADAS as a real object. When the phantom appears in less than t consecutive frames in the captured video stream, it is ignored by the ADAS. There is a probabilistic area at which the success rate of a split-second phantom attack is between 1-99% where a phantom projected for a given duration can appear in t consecutive frames or in t-*1* consecutive frames in the captured video stream depending on the differences in the FPS rate and synchronization of the video camera and digital visualization instrument. (3) The success rate for the same projection duration varies, depending on the ADAS employed (e.g., a phantom road sign that appears for 333 milliseconds is detected by Mobileye 100% of the time, but it is detected by Tesla just 35.3% of the time).

6 Evaluation

In this section, we demonstrate split-second phantom attacks on commercial ADASs while driving. The attack is validated by: (1) embedding a road sign into a digital advertisement presented on a hacked digital billboard, and (2) projecting a road sign via a portable projector mounted on a drone. The attacks are demonstrated against a Renault Captur (equipped with Mobileye 630) and a Tesla Model X (HW 2.5 and HW 3).

6.1 Phantom Attack via a Digital Billboard

We demonstrate the attack via a digital billboard by embedding a phantom into an existing advertisement and displaying it for a few milliseconds in a way that the imagery itself is hard to perceive. To accomplish this, we developed Algorithm 1 to embed phantoms into existing ads in a way that won't be recognized by the human eye. Based on the output of Algorithm 1, we embedded a phantom road sign into a McDonald's advertisement and applied the attack against Mobileye and Tesla.

6.1.1 Algorithm for Disguising Phantoms in Advertisements:

The procedure *main* in Algorithm 1 receives four arguments: *video* (a 3D matrix representing the advertisement), *nC* (the number of consecutive frames that need to be compromised), and the *height* and *width* (the size of a phantom to be embedded in the video). The algorithm returns a 3D array of scores, where the value in each cell of the array represents a "disguising" score in cases in which a phantom is embedded in the cell. First, the algorithm extracts key points for every frame based on the SURF algorithm [4]. Key points represent interest points in an image; they are used for many tasks in computer vision (e.g., object recognition, image registration) and are also used to represent interesting areas in a frame that a

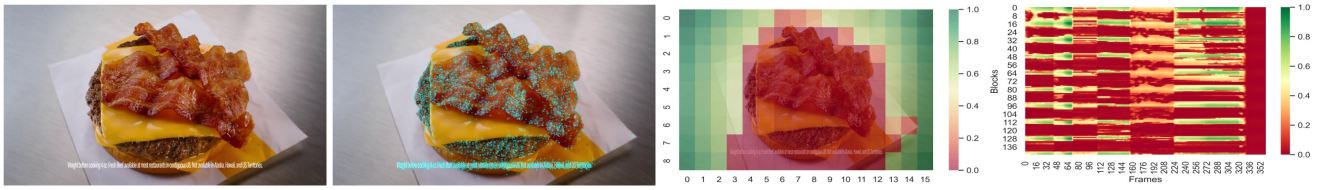

Figure 4: From left to right: Original frame; key points (in blue) extracted from the McDonald's advertisement; heatmap of a frame's local score; spectrogram of a block's global score with respect to the next three consecutive frames in the ad.

Algorithm 1 Rank-Blocks

```
1:  procedure MAIN(height, width, video, nC)
2:      grids[][][] = extract-key points(height, width, video)
3:      local [][][] = local-score(grids)
4:      global [][][] = global-score(local-scores,nC)
5:      return global
6:  procedure EXTRACT-KEY POINTS(height, width, video)
7:      for (i =0; i< len(video);i++) do
8:          k-p [][] = SURF(video[i])
9:          for (x =0; x< len(key points); x++) do
10:             for (y =0; y< len(key points[0]); y++) do
11:                 grids [i][x % width][y % height] += k-p[x][y]
12:     return grids
13: procedure LOCAL-SCORE(grids)
14:     for (f=0; f<len(grids); f++) do
15:         grid = grids[f]
16:         for (x1=0; x1<len(grid); x1++) do
17:             for (y1=0; y1<len(grid[0]); y1++) do
18:                 scores[f][x1][y1] = local-score-block(grid,x1,y1)
19:     return scores
20: procedure LOCAL-SCORE-BLOCK(grid,x1,y1)
21:     score = 0
22:     for (x2=0; x2<len(grid); x2++) do
23:         for (y2=0; y2<len(grid[0]); y2++) do
24:             score += (1 + sqrt((x1 − x2)^2 + (y1 − y2)^2))×grid[x2][y2]
25:     score/= (1 + grid[x1][y1])
26:     return score
27: procedure GLOBAL-SCORE(local-scores,nC)
28:     for (f=0; f<len(grids)-nC; f++) do
29:         for (x=0; x<len(grid); x++) do
30:             for (y=0; y<len(grid[0]); y++) do
31:                 for (i=f; i<f+nC; i++) do
32:                     global[f][x][y] += local[i][x][y]
33:     return global
```

viewer is likely to focus on. Fig. 4 presents a visual demonstration of the output of this stage on a frame from a random McDonald's advertisement that we picked. Then, the algorithm (1) divides the frame into a grid where each cell in the grid is the size of *height* x *width*, and (2) counts the number of key points within each cell.

Next, the algorithm computes a local score for every block in a frame that represents how distant a block is from the interesting parts of the frame (since attackers are more likely to embed phantoms into areas that viewers will not focus on). This is implemented according to the following approach: a block with a larger number of key points is likely to interest a viewer more than a block with a smaller number of key points or no key points. An optimal block in this respect is a block that is located far from all of the interesting blocks in a frame. In order to compute a score for how far the block b_i located at (x1,y1) is from the block b_j located at (x2,y2) with respect to the number of key points in block (x2,y2), the algorithm first extracts (1) the Euclidean distance between blocks b_i and b_j, and (2) $nKP(b_j)$, which is the number of key points in block b_j; then the algorithm calculates their product:

$$score(b_i, b_j) = \sqrt{(x1 - x2)^2 + (y1 - y2)^2} * nKP(b_j) \quad (1)$$
$$where\ b_i\ located\ at\ (x1,y1),\ b_j\ located\ at\ (x2,y2)$$

The result of this calculation is greater as the distance between blocks b_i and b_j increases or the number of key points in block b_j increases. The local score of a block b_i in a frame f is the sum of scores between block b_i to all other blocks in f, divided by the number of key points in b_i:

$$local - score_f(b_i) = \frac{\sum_{\forall b_j} score(b_i, b_j)}{1 + nKP(b_i)} \quad (2)$$

Fig. 4 illustrates this stage on a frame (after we normalized the scores to [0,1]) in the form of a heatmap.

Since an ADAS's cameras can work at a lower FPS rate (e.g., 12 FPS) than the FPS of the advertisement (e.g., 24 FPS), a phantom may need to appear on several consecutive frames (nC) which can be provided by the attackers as input to the algorithm. For example, Mobileye detects phantoms that are presented for 125 ms, so a phantom needs to appear on three frames of a 24 FPS video in order to be detected by Mobileye. In such cases (where $nC > 1$), the global score of a block that takes the notion of time into account needs to be computed. This is because the local score of a block can change between two consecutive frames, so the global score of block b_i in frame f with respect to the next nC consecutive frames is the sum of the block's $local - score_f(b_i)$ from frame f to $f + nC$:

$$global - score(b_i, frame, nC) = \sum_{f=frame}^{frame+nC} local - score_f(b_i) \quad (3)$$

The ideal block and frame for disguising a phantom is the block that received the highest score. Fig. 4 illustrates the global score of each block (with respect to the next three consecutive frames) for the entire advertisement.

6.1.2 Validation: We picked a random McDonald's advertisement from YouTube. We applied Algorithm 1 on the advertisement and located the optimal block and frame for hiding the phantom (the block with the maximal global score). Based on the analysis of the influence of the duration of a phantom's projection on the success rate of the attack (which is presented in Fig. 3), we embedded: (1) a speed limit sign into three consecutive frames (125 ms) of the advertisement in order to attack Mobileye 630, and (2) a stop sign into nine consecutive frames (500 ms) of the advertisement in order to attack the Tesla Model X. The compromised advertisements can

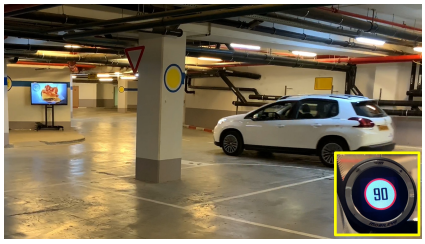

Figure 5: Mobileye issues an alert about a phantom road sign after the phantom speed limit was detected in the upper left corner of the digital billboard.

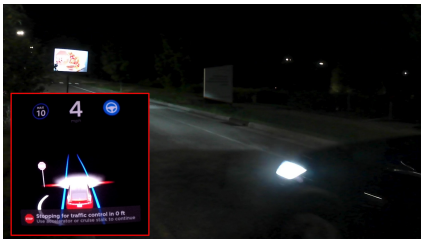

Figure 6: Tesla's autopilot triggers the car to stop suddenly after a phantom stop sign was detected in the upper left corner of the digital billboard.

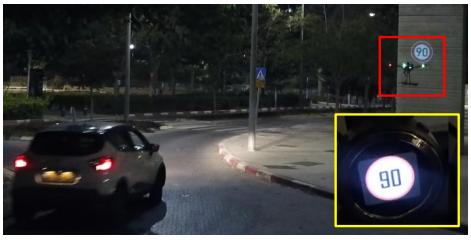

Figure 7: A phantom (boxed in red) is projected on a building for 125 ms from a drone and recognized by Mobileye (boxed in yellow) as real.

be seen here.[2] The size of the road sign in each advertisement was selected according to the size of the digital billboard that was used in the following experiments (according to the analysis in Fig 3). Below, we demonstrate how we used the advertisements to attack a car equipped with Mobileye 630 and the Tesla.

First we demonstrate the attack on a car equipped with Mobileye 630 via a digital advertisement presented on a digital billboard in an underground parking lot. The experiment was conducted in the underground parking lot of a building on our university campus after we received the proper approvals from the security department. The attacker used a 65 inch TV screen (which was used to simulate a digital billboard) that was placed on a sidewalk in the parking lot. We used this setup to demonstrate the attack, since we had no intention of hacking a real digital billboard in this study. The experiment was conducted as follows: While driving the car through the parking lot, the compromised Mcdonald's advertisement was played on the TV screen with a phantom of a 90 km/h speed limit sign presented for 125 ms. Fig. 5 presents a snapshot from the 125 ms that the phantom was presented on the left upper corner of the screen. The attack was successful, since Mobileye detected the phantom and notified the driver of the 90 km/h speed limit, although driving faster than 30 km/h in this parking lot is prohibited. The reader can view a video of this experiment online.[2]

We now demonstrate the attack on Tesla Model X (HW 3) via a digital advertisement presented on a digital billboard located near a road. The experiment was conducted on a road on our university campus after we received the proper approvals from the security department. The attacker used a 42 inch TV screen (which was used to simulate a digital billboard and was plugged into another car for electricity) that was placed in the middle of the road. We used this setup to demonstrate the attack, since we had no intention of hacking a real digital billboard in this study. The experiment was conducted as follows: We engaged Tesla's autopilot at the beginning of the road. The Tesla approached the middle of the road where the TV screen presented the compromised McDonald's advertisement (with the embedded 500 ms stop sign). Fig. 6 presents a snapshot from the 500 ms that the phantom was presented on the upper left corner on the screen. The attack was successful, since Tesla's autopilot identified the phantom stop sign and immediately triggered the car to stop in the middle of the road. The reader can view a video of this experiment online.[2]

[2] https://youtu.be/-E0t_s6bT_4

6.2 Phantom Attack via a Projector

First we demonstrated the attack on a car equipped with Mobileye 630. This experiment was conducted on a road with a 30 km/h speed limit, with the permission of the local authorities. The attacker used a drone (DJI Matrice 600) with a portable projector. The target was a car (Renault Captur) equipped with Mobileye. The experiment was conducted as follows: While driving the car through this urban environment, the drone projected a 90 km/h speed limit sign onto a building for 125 ms, within the Mobileye's field of view. Fig. 7 presents a snapshot of the phantom used in this experiment. The attack was successful, since Mobileye detected the phantom and notified the driver of the 90 km/h speed limit. The reader can view a video of this experiment online[3].

We then validated the phantom attack on a Tesla Model X (HW 2.5). As was shown in Fig. 1 and 2, this model recognizes phantoms of depthless objects as real obstacles despite the fact that it is equipped with depth sensors (see Fig. 13). In this experiment, we projected a phantom of a pedestrian onto a road which was closed after receiving permission to close the road from the local authorities. The phantom was not projected from a drone, since local flight regulations prohibit the use of drones near roads and highways. Furthermore, we projected a constant phantom instead of projecting it for a split second. This was done for ethical reasons, since the vehicle may swerve due to the sudden appearance of a pedestrian, and risk the lives of the driver and potential bystanders. We note that this experiment was performed to validate that phantom attacks are effective on moving vehicles; the effectiveness of split-second attacks on a Tesla was shown in sections 5 and 6.1.2. The experiment was performed by driving the Tesla towards the projected pedestrian with the cruise control set at 18 MPH (see Fig. 8). The result was that the Tesla automatically applied the brakes (slowing from 18 MPH to 14 MPH) to avoid a collision with the phantom pedestrian. In summary, the car detected the phantom as a real obstacle and reacted accordingly. The reader can view a video of this experiment online.[3]

The responsible disclosure of our findings to Tesla and Mobileye is presented in Appendix 3.

7 Detecting Phantoms

Even though autonomous systems also use RADAR and LIDAR to sense the vehicle's environment, we found that these systems rely on the camera sensor to avoid making potentially fatal mistakes

[3] https://youtu.be/1cSw4fXYqWI

Figure 8: The Tesla automatically driving at a speed of 18 MPH detects a phantom pedestrian that is projected on the road as a real obstacle (see a snapshot of its dashboard in the box on the left) and automatically triggers the brakes (see a later snapshot of its dashboard in the box on the right).

(e.g., failing to detect a pedestrian in the street). This makes sense since manufactures would rather have the system react (e.g., stop or swerve the vehicle) than run the risk of causing an accident. In these situations, we propose an add-on software module which can validate objects identified using the camera sensor.

As discussed in Section 3, 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, we propose that a phantom detection module should validate the legitimacy of the object given its context and authenticity. In general, we identify six 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 traffic sign which is not regulation size. The size and distance of an object can be determined via the camera sensors alone through stereoscopic imaging [43], though multiple cameras directed in the same direction are required.

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.

Focus. For projected phantoms, the detected object may be blurry in regions outside of the projector's focal range (see Fig. 17). For example, when projected at an angle to the surface or on a lumpy surface.

Context. If the placement of the object is impossible or simply abnormal, it is indicative of a phantom, e.g., a traffic 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 is likely a phantom. This can be determined passively through image analysis or actively by shining a light source onto the object (e.g., flash photography).

Depth. If the object has the wrong surface shape, or its placement in a 3D scene is abnormal, then the object may be a phantom. For example, a traffic sign projected on an uneven surface (like the tree in Fig. 2), or a 3D person or car projected on the surface of the road (figures 1 and 2). To extract an implicit depth perception from an existing camera on the vehicle, we suggest computing the optical flow between subsequent video frames (discussed more in detail later on).

These are the aspects were identified by visually comparing hundreds of phantoms in various environments to images of real objects. We note that this list may be incomplete and other aspects for identifying a phantom from an image may exist.

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

7.1 The *GhostBusters*

Overall, the proposed countermeasure works as follows. Whenever an object is detected using the vehicle's image recognition model, the detected object is cropped from the image and passed to our model. The model then predicts if the object is a phantom (has an abnormal setting is not realistic). If the object is determined to be a phantom, the system can then decide whether or not to trust the detected object. The model 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).

Let x^t be an RBG traffic sign image cropped from video frame t where the sign is centered and takes up approximately $\frac{1}{9}$-th of the image. To predict whether or not x^t is a phantom or real, we could build a convolutional neural network (CNN) classifier which receives x^t and predicts whether it is real or fake. However, this approach would make the CNN 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 traffic sign is an obvious way to visually distinguish between a real and projected sign. However, a CNN trained on the entire sign would primarily focus on this aspect alone and make errors when phantoms are 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 [26] in which there is an ensemble of models, each of which has a different perspective or capability of interpreting the training data. By separating these perspectives, we (1) become more resilient when one aspect fails to capture the evidence, and (2) we lower the false alarm rate by focusing the network on relevant features only.

Our committee, called the *GhostBusters*, consists of four deep CNN models, each focusing on a different aspect (see Fig. 9 for the

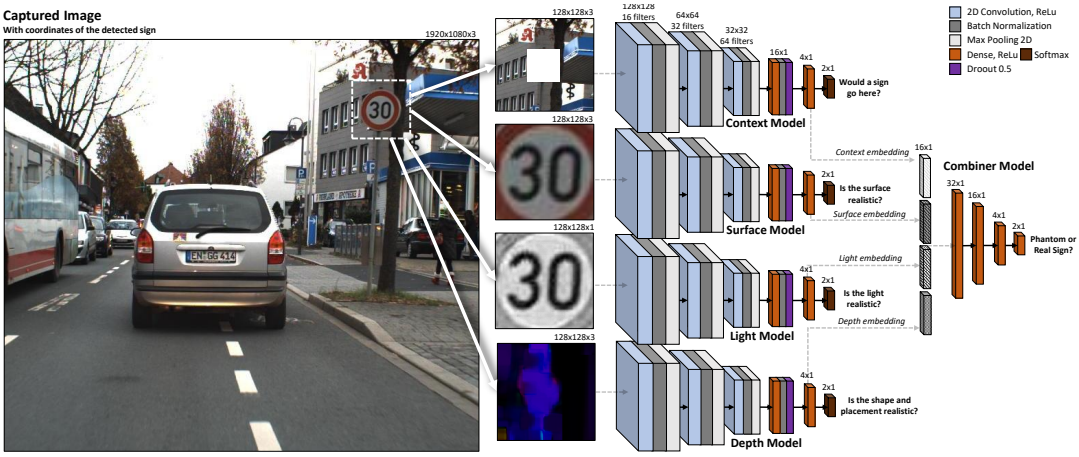

Figure 9: The architecture of proposed model: The *GhostBusters*. When a frame is captured, (1) the on-board object detector locates a road sign, (2) the road sign is cropped and passed to the Context, Surface, Light, and Depth models, and (3) the Combiner model interprets the models' embeddings and makes a final decision on the traffic sign (real or fake).

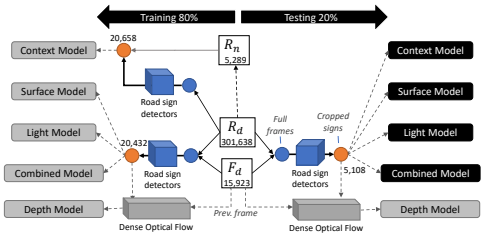

Figure 10: A diagram showing how the training and testing data was prepared from our data sources, and the number of instances.

architecture and the model's parameters). The models receive a cropped image of a traffic sign and then judge if the sign is authentic and contextually makes sense:

Context Model. This CNN receives the context: the area surrounding the traffic sign. This is obtained by taking x^t, re-scaling it to a 128×128 image, and then setting the center (a 45×45 box) to zero. 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. This is obtained by cropping x^t and re-scaling it to a 128×128 image. 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. This is created with the cropped and scaled x^t, by taking the maximum value from each pixel's RGB values. Formally

$$x[i,j] = \arg\max_k x^t[i,j,k] \tag{4}$$

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.

Depth Model. This CNN receives the apparent depth (distance) of the scenery in the image. To obtain this, we compute the optical flow between x^t and the same space from the last frame, denoted x^{t-1}. The optical flow is a 2D vector field where each vector captures the displacement of the pixels from x^{t-1} to x^t. In particular, with OpenCV, we utilize the Gunner Farneback algorithm [18] to obtain the 2D field v, and then convert it to an HSV image by computing each vector's angle and magnitude:

$$\begin{aligned} x[i,j,0] &= \sin^{-1}\left(v[i,j,1]/\sqrt{v[i,j,0]^2}\right) \times 180/2\pi \\ x[i,j,1] &= 255 \\ x[i,j,2] &= norm\_minmax\left(\sqrt{v[i,j,0]^2 + v[i,j,1]^2}\right) * 255 \end{aligned} \tag{5}$$

The HSV image is converted to an RGB image before passing it to the CNN. The significance of this approach is that we can obtain an implicit 3D view of the scenery while the vehicle is in motion. This enables the model to perceive the sign's placement and shape better using only one camera. Fig. 14 presents some examples of the model's input based on various sampled x^t.

To make a prediction on whether or not a sign is real or fake, we combine the knowledge of the four 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 on 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 fifth neural network is trained to classify the cropped sign as real or fake using the concatenated embeddings as its input. Overall, the entire neural network has 1, 145, 654 trainable parameters.

7.2 Experiment Setup

To validate the proposed countermeasure, we evaluate the (1) the system's detection performance, (2) the attack success rates

Table 4: Detection Rates Using s.o.t.a traffic sign Detectors

			Attack Success Rate		
			Countermeasure		
			With		Without
		Threshold:	*@0.5*	*@[FPR=0]*	
Sign Detector	[52]	faster_rcnn_inception_resnet_v2	0.098%	0.294%	87.16%
	[52]	faster_rcnn_resnet_101	0.098%	0.588%	96.08%
	[52]	faster_rcnn_resnet50	0.098%	0.588%	81.29%
	[28]	faster_rcnn_inception_v2	0.098%	0.588%	93.05%
	[13]	rfcn_resnet101	0.098%	0.588%	99.71%
	[25]	ssd_inception_v2	0.0%	0.294%	81.98%
	[24]	ssd_mobilenet_v1	0.098%	0.588%	83.45%

on different s.o.t.a traffic sign detectors, and (3) the models efficiency (speed/resources). We also analyse the importance of each individual model placed in a collective through an ablation study, and compare our models' performance to a baseline (a single CNN classifier trained on the cropped images).

Dataset. To generate training and testing data, we recorded videos using a dash cam with a driver seat perspective. The first set of videos contained three hours of driving around a city at night. We denote the frames extracted from these videos as R_d. The second set of videos were recorded while driving around an area where phantom traffic signs were projected. The frames from these videos are denoted as F_d. In the F_d dataset, we projected 40 different types of signs in a loop onto nine different surfaces.

We then used the highest performing traffic sign detector (faster rcnn inception resnet v2) described in [3] to detect and crop all of the traffic signs in R_d and F_d. Each cropped traffic sign x^t was collected and places in our dataset X. 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_d such that the center of the cropped images does not contain a sign.

A summary of how we created the dataset and its number of samples can be found in Fig. 10.

We note that the proposed models (experts) can work in practice when trained on data similar to ours. This is because the imagery captured while driving around different environments generalizes well to similar ones unseen during training [6, 20, 58, 62]. For example, NVIDIA recently demonstrated how a single camera can be used to accurately predict the size and depth of objects for autonomous driving, using similar training data [12]. Most companies have R_d available so the challenge is collecting F_d. Therefore, to expedite the collection of F_d, one can (1) mount a projector on top of a car, (2) drive around an urban area while displaying random phantoms, and (3) save all frames where the object detector identified the object (since it would potentially fool the ADAS). This approach would quickly generate a wide variety of successful phantom attacks on various surfaces and in different contexts and environments.

Training. The Context, Surface, Light, and Depth models were trained separately, and then the Combiner model was trained on their embeddings. Regarding the data, %80 of X was used to train the models, and the remaining %20 was used to evaluate them as a test set. To reduce bias, the test set contained videos and phantom projections on surfaces which were not in the training set. Finally, training was performed on an NVIDIA 2080 ti GPU for 25 epochs.

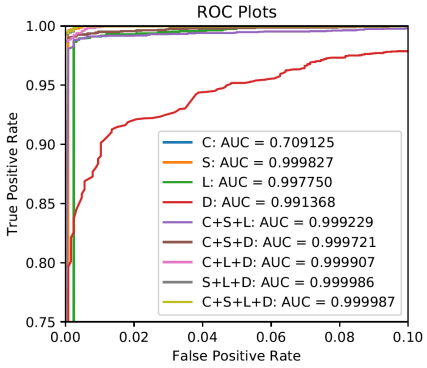

Figure 11: The receiver operating characteristic curve and AUC measure for different combinations of the models. Model Acronyms C: Context, S: Surface, L:Light, D: Depth.

7.3 Experiment Results

7.3.1 Model Performance: In Fig. 11 we present the receiver operating characteristic (ROC) plot and the area under the ROC (AUC) for combinations of the Context, Surface, Light, Depth models, where the combination of all four is the proposed model. 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: a lower threshold will decrease the FPR but often decrease the TPR as well. In our case, it is critical that the proposed module predict real signs as real every time because the vast majority of signs passed to our module will be real. Therefore, even a very small FPR would make the solution impractical. Therefore, in Table 2 we provide the TPR and FPR of the models when the threshold is set to 0.5 (the default for softmax) and for the threshold value at which the FPR is zero.

As can be seen in Table 2, the proposed model (C+S+L+D) out performs all other model's combinations and the baseline model (a CNN classifier trained on $x^t inX$). As noted earlier, the combined model outperforms the baseline model because it focuses on the relevant information and is less dependent on any one aspect/feature. We also note that the Depth model has a lower AUC than the others because it is only effective while the vehicle is in motion.

7.3.2 Ablation Study: Since the combination of all four models provides the best results, it is a clear indication that each aspect (context, surface, light, and depth) contributes a unique and important perspective on the difference between a real and phantom traffic sign. With the exception of a few cases, further evidence that each model has a unique contribution can be seen in Table 2 where all pairs of models are better than any single model, and all triplets are better than all pairs, etc. This is important since in order for the committee of experts approach to be effective there must be some disagreements between the models. Table 3 measures the number of disagreements each combination has on the test set. Interestingly, the full combination (C+S+L+D) has the highest number of disagreements and results in the highest TPR at FPR=0.

In Fig. 15, 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

Table 2: The TPR and FPR of the Countermeasure Models at Different Thresholds. C: Context, S: Surface, L: Light, D: Depth

		C	S	L	D	C+S	C+L	C+D	S+L	S+D	L+D	C+S+L	C+S+D	C+L+D	S+L+D	Proposed C+S+L+D	Baseline Cropped Image
Threshold	**TPR**	0.730	0.997	0.990	0.992	0.998	0.990	0.990	0.990	0.995	0.996	0.989	0.995	0.992	0.999	0.998	1.000
@ 0.5	**FPR**	0.281	0.002	0.006	0.214	0.002	0.004	0.165	0.004	0.009	0.002	0.004	0.016	0.002	0.002	0.002	1.000
Threshold	**TPR**	0.049	0.796	0.017	0.150	0.986	0.221	0.725	0	0.990	0.985	0.667	0.987	0.987	0.993	**0.994**	**0.906**
@ [FPR=0]	**FPR**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Table 3: The Disagreement Between the Models

	Disagreement on…	C, S	C, L	C, D	S, L	S, D	L, D	C, S, L	C, S, D	C, L, D	S, L, D	C, S, L, D
Threshold	Phantoms	86.12%	88.03%	75.65%	3.65%	17.92%	15.70%	88.90%	89.85%	89.69%	18.64%	**90.25%**
@ 0.5	Real Signs	10.80%	11.74%	11.50%	0.99%	1.22%	2.21%	11.76%	11.76%	12.72%	2.21%	**12.72%**
Threshold	Phantoms	55.75%	0%	3.97%	55.75%	54.16%	3.97%	55.75%	56.94%	3.97%	56.94%	**56.94%**
@ [FPR=0]	Real Signs	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	**0%**

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. 15). In other cases, a model misclassifies 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. 15). Regardless, in all cases the other models provided a strong vote of confidence contrary to the erroneous opinion, which led to the correct prediction.

However, this approach is not perfect. Fig. 16 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.

7.3.3 Countermeasure Performance: The proposed module filters out untrusted (phantom) traffic signs detected by the on-board object detector. Since there are many different implementations of traffic 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 seven s.o.t.a traffic sign detectors [3]. We measured the attack success rate on a detector as the percent of phantom signs in F_d identified as a sign. In Table 4 we present the attack success rates on each detector before and after applying our countermeasure, and the impact of the different thresholds. The results show that the detectors are highly susceptible to phantom attacks and that our countermeasure provides effective mitigation, even at the threshold where there are zero false positives on the test set.

Therefore our countermeasure is reliable enough for daily usage (it is expected to make very few false positives) and will detect a phantom most 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. We also suggest that additional models which consider size and angle be considered as well.

7.3.4 Adversarial Machine Learning Attacks: It is well known that deep learning models, such as those used in our countermeasure, are susceptible to adversarial machine learning attacks. A concern with our system is that an attacker will craft a phantom projection that will both fool the traffic sign detector and cause our model to predict 'real' with high confidence. This attack would be relatively easy to achieve when using a single CNN as a phantom detector. However, our committee of experts is strong against such attacks. This is because the attacker only has control over the pixels which he projects but not the physical attributes of the projected image. Furthermore, the attacker is limited since he must project on a relatively smooth surface, making it easier for the Context and Depth models to identify the attack, regardless of the content. Therefore, even if the attacker fools the Surface Model CNN, the combined model will still detect the object as fake and mitigate the attack.

To evaluate this concept, we used the framework in [46] to attack each expert with eight different white box adversarial machine learning evasion attacks [5, 8, 9, 21, 29, 32, 40, 48]. For each expert and attack, we measured the expert's accuracy alone and the committee's accuracy when that expert is targeted.

In Fig. 12 we present the results along with visual examples. The results demonstrate that the committee is much stronger than any one expert. Since it is especially challenging to physically craft an adversarial sample for the light and depth experts, the collective model remains robust to targeted attacks. For results on attacking different combinations of experts, please see Table 6 in the appendix.

One exception is the IFGSM method which negatively impacts the committee's decision (except when targeting the Surface model). However, there exists many works which specifically mitigate the IFGSM attacks, for example [14, 16, 30, 37, 38, 41, 63]. Regardless, it is important to note that adversarial attacks are a cat and mouse game and our solution is not impervious to future adversarial attacks. However, these results indicate that the committee approach raises the difficulty bar for attackers, since the attacker (1) must craft complex physical features (e.g., light and depth), (2) is limited in the location they can perform the attack (context), and (3) must generate adversarial samples that capture multiple physical aspects (experts) at once.

7.4 Generalization to other Phantom Objects

In our evaluation, we showed how the *GhostBusters* work in the context of phantom road signs. However, the same approach generalizes to other phantom objects.

Light Consider a phantom pedestrian. Compared to a real pedestrian, a phantom pedestrian emits more light than usual, especially when considering the reluctance of different materials in direct light (glasses, belt, clothes, etc.) Moreover, if the phantom is on the road in front of the car, the car's headlights saturate the projection and erase many of the details. This is also observed in the left of Fig. 3 where a more intense phantom was required at 3m than 4.5m since the headlights were covering the phantom.

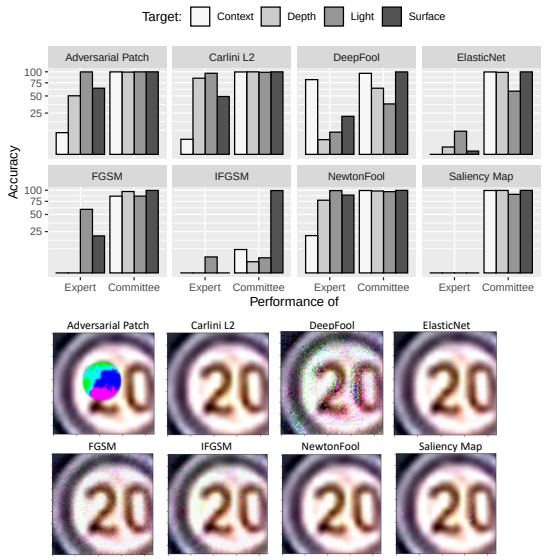

Figure 12: Results from the adversarial experiment. Top: The accuracy of each expert under adversarial attacks, both alone and within the committee. Bottom: A phantom road sign perturbed to be detected as 'real' by the Surface Model.

Surface When projected on the road (to trigger a collision) the image will receive abnormal textures and color patterns due to the surface. For example, sewers, cracks, street markings, and the asphalt all distort the projected image (see Fig. 17). The same can be said when road signs and traffic lights are projected on walls and other vertical surfaces.

Depth All 3D objects projected onto a flat surface will not be detected as 3D by the depth expert. This is because there is no foreground/background so the entire surface will have the same shift between frames. The exception is when 2D objects, such as road signs, are projected. In this case the strength of the committee's different perspectives are leveraged to make the decision. On the contrary, when other experts fail, the depth expert can assist in mitigating false positives as seen in Fig. 15.

Context The context of road sign is very clear, but it may be difficult to capture the context for other objects. In general, objects displayed in digital billboards will be within the frame of the display, and phantoms projected on irregular surfaces (e.g., a pedestrian floating on the back of a truck) will also be detected. However, a road projected pedestrian may legitimately look like someone crossing a street. Regardless, the committee should still work well with and without this expert, as demonstrated in Table 2. Furthermore, one can also add new expert, not evaluated in this paper, to strengthen the diversity. For example, a size expert based on [12] can be used to identify contradictory sizes (a road projection can stretch several meters as seen in Fig. 17).

8 Conclusions, Discussion & Future Work

The findings of this research are not intended to minimize the findings of previous studies that presented adversarial machine learning attacks against computer vision algorithms. Split-second phantom attacks represent an additional class of visual spoofing that should be considered. Our research is not aimed at casting doubt on the car industry's many years of experience, discounting the use of sensor fusion for cyber-physical systems, or taking any side in the ongoing debate regarding the use of LiDAR or radar for semi/fully autonomous cars [1]. The fact is that sensor fusion based on radar is the safest approach for semi/fully autonomous vehicles, and recent reports from Tesla show that its autopilot is safer than a human driver [59]. The purpose of this research is to demonstrate that like previous attacks on autonomous vehicles [65] have shown, sensor fusion still has edge cases that need to be taken into account.

One might get the impression that we claim that Teslas are programmed to always trust a video camera over radar. We can only hypothesize how Teslas are programmed, because Telsa's engineers did not provide any details. However, we believe that the reason that Tesla followed the video camera over the radar is the result of another policy. A previous study [65] demonstrated that the Tesla can be tricked into considering obstacles by applying spoofing and jamming attacks to an integrated sensor (radar or ultrasonic sensors). In both cases, the car follows a single sensor despite the fact that there was no visual validation. Having that in mind, we believe that the Tesla is programmed to consider classifications that were made from just a single sensor if they have crossed a predefined level of confidence. This policy can explain why the Tesla considers radio and ultrasonic phantoms [65], as well as visual phantoms, even without any additional validation.

Another question that arises from this research is: Can split-second phantom attacks be applied to cars that rely on a different set of sensors than those used by Tesla? Visual phantoms represent an edge case of disagreement between depth sensors and video cameras. We approached two self-driving car manufacturers (level 4) and asked them to allow us to evaluate the behavior of their cars in response to split-second second phantom attacks, however neither of them responded positively. However, we find the question of whether the attacks can be applied to other semi/fully autonomous cars as irrelevant, because attackers can create the same sort of challenge for self-driving cars that rely on LiDAR and video cameras. In our opinion, phantoms are the result of a policy regarding how the car should behave when there is a case of disagreement between sensors. This question is not addressed specifically by any of the recent standards, leaving car manufacturers the space to implement a policy based on their experience. We hope that this research will result in guidelines/standards regarding such edge cases.

As future work, we suggest exploring whether split-second phantom attacks can be applied remotely via digital bus ads. We also suggest examining whether phantom attacks can be applied using infrared projection, exploiting the fact that a narrow spectrum of frequencies, the near infrared, is also captured by some CMOS sensors (this fact was exploited to mislead facial recognition algorithms [64, 67], to establish an optical covert channel [22, 45], and to break the confidentiality of FPV channel of commercial drones [44]). It would also be interesting to evaluate a specific kind of phantom where a complete sign or part of a sign is projected onto another. We also suggest testing the robustness of other commercial ADASs to phantom attacks and investigating whether phantoms of other 3D objects can trick the ADAS (e.g., pets, traffic cones, traffic lights).

References

[1] Simon Alvarez. [n.d.]. Tesla's approach for Full Self-Driving gets validated by Cornell researchers, LiDAR pioneer. https://www.teslarati.com/tesla-elon-musk-full-self-driving-lidar-waymo-cornell-study/.

[2] Anker. 2019. Nebula Capsule. https://www.amazon.com/Projector-Anker-Portable-High-Contrast-Playtime/dp/B076Q3GBJK.

[3] Alvaro Arcos-Garcia, Juan A. Alvarez-Garcia, and Luis M. Soria-Morillo. 2018. Evaluation of deep neural networks for traffic sign detection systems. *Neurocomputing* 316 (2018), 332 – 344. https://doi.org/10.1016/j.neucom.2018.08.009

[4] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. 2006. Surf: Speeded up robust features. In *European conference on computer vision*. Springer, 404–417.

[5] Tom B Brown, Dandelion Mané, Aurko Roy, Martín Abadi, and Justin Gilmer. 2017. Adversarial patch. *arXiv preprint arXiv:1712.09665* (2017).

[6] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. 2020. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 11621–11631.

[7] Yulong Cao, Chaowei Xiao, Benjamin Cyr, Yimeng Zhou, Won Park, Sara Rampazzi, Qi Alfred Chen, Kevin Fu, and Z Morley Mao. 2019. Adversarial sensor attack on lidar-based perception in autonomous driving. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*. 2267–2281.

[8] Nicholas Carlini and David Wagner. 2017. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*. IEEE, 39–57.

[9] Pin-Yu Chen, Yash Sharma, Huan Zhang, Jinfeng Yi, and Cho-Jui Hsieh. 2017. Ead: elastic-net attacks to deep neural networks via adversarial examples. *arXiv preprint arXiv:1709.04114* (2017).

[10] Shang-Tse Chen, Cory Cornelius, Jason Martin, and Duen Horng Polo Chau. 2018. Shapeshifter: Robust physical adversarial attack on faster r-cnn object detector. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 52–68.

[11] European Comission. 2016. Advanced driver assistance systems. https://ec.europa.eu/transport/road_safety/sites/roadsafety/files/ersosynthesis2016-adas15_en.pdf.

[12] NEDA CVIJETIC. 2019. DRIVE Labs: Detecting the Distance – NVIDIA Blog. https://blogs.nvidia.com/blog/2019/06/19/drive-labs-distance-to-object-detection/. (Accessed on 08/24/2020).

[13] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. 2016. R-fcn: Object detection via region-based fully convolutional networks. In *Advances in neural information processing systems*. 379–387.

[14] Nilaksh Das, Madhuri Shanbhogue, Shang-Tse Chen, Fred Hohman, Siwei Li, Li Chen, Michael E Kounavis, and Duen Horng Chau. 2018. Shield: Fast, practical defense and vaccination for deep learning using jpeg compression. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 196–204.

[15] Automated Driving. 2014. Levels of driving automation are defined in new SAE international standard J3016: 2014. *SAE International: Warrendale, PA, USA* (2014).

[16] Gamaleldin Elsayed, Dilip Krishnan, Hossein Mobahi, Kevin Regan, and Samy Bengio. 2018. Large margin deep networks for classification. In *Advances in neural information processing systems*. 842–852.

[17] Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, and Dawn Song. 2017. Robust physical-world attacks on deep learning models. *arXiv preprint arXiv:1707.08945* (2017).

[18] Gunnar Farnebäck. 2003. Two-frame motion estimation based on polynomial expansion. In *Scandinavian conference on Image analysis*. Springer, 363–370.

[19] David Geronimo, Antonio M Lopez, Angel D Sappa, and Thorsten Graf. 2009. Survey of pedestrian detection for advanced driver assistance systems. *IEEE transactions on pattern analysis and machine intelligence* 32, 7 (2009), 1239–1258.

[20] Jakob Geyer, Yohannes Kassahun, Mentar Mahmudi, Xavier Ricou, Rupesh Durgesh, Andrew S Chung, Lorenz Hauswald, Viet Hoang Pham, Maximilian Mühlegg, Sebastian Dorn, et al. 2020. A2d2: Audi autonomous driving dataset. *arXiv preprint arXiv:2004.06320* (2020).

[21] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572* (2014).

[22] Mordechai Guri and Dima Bykhovsky. 2019. air-jumper: Covert air-gap exfiltration/infiltration via security cameras & infrared (ir). *Computers & Security* 82 (2019), 15–29.

[23] K. He, X. Zhang, S. Ren, and J. Sun. 2016. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 770–778.

[24] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. 2017. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861* (2017).

[25] Jonathan Huang, Vivek Rathod, Chen Sun, Menglong Zhu, Anoop Korattikara, Alireza Fathi, Ian Fischer, Zbigniew Wojna, Yang Song, Sergio Guadarrama, et al. 2017. Speed/accuracy trade-offs for modern convolutional object detectors. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 7310–7311.

[26] Jenq-Neng Hwang and Yu Hen Hu. 2001. *Handbook of neural network signal processing*. CRC press.

[27] Car Industry. 2019. Safety First For Automated Driving. https://www.daimler.com/documents/innovation/other/safety-first-for-automated-driving.pdf".

[28] Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167* (2015).

[29] Uyeong Jang, Xi Wu, and Somesh Jha. 2017. Objective metrics and gradient descent algorithms for adversarial examples in machine learning. In *Proceedings of the 33rd Annual Computer Security Applications Conference*. 262–277.

[30] Jiahuan Ji, Baojiang Zhong, and Kai-Kuang Ma. 2019. Multi-Scale Defense of Adversarial Images. In *2019 IEEE International Conference on Image Processing (ICIP)*. IEEE, 4070–4074.

[31] keen labs. 2019. 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/.

[32] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. 2016. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533* (2016).

[33] Timothy Lee. [n.d.]. Intel's Mobileye has a plan to dominate self-driving—and it might work. https://arstechnica.com/cars/2020/01/intels-mobileye-has-a-plan-to-dominate-self-driving-and-it-might-work/.

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

[35] Timothy Lee. 2019. 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/.

[36] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. 2016. Ssd: Single shot multibox detector. In *European conference on computer vision*. Springer, 21–37.

[37] Bo Luo, Min Li, Yu Li, and Qiang Xu. 2018. On Configurable Defense against Adversarial Example Attacks. *arXiv preprint arXiv:1812.02737* (2018).

[38] Shiqing Ma and Yingqi Liu. 2019. Nic: Detecting adversarial samples with neural network invariant checking. In *Proceedings of the 26th Network and Distributed System Security Symposium (NDSS 2019)*.

[39] Mobileye. [n.d.]. Mobileye 6-Series - User Manual. http://www.c2sec.com.sg/Files/UserManualMobileye6.pdf.

[40] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. 2016. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2574–2582.

[41] Seyed-Mohsen Moosavi-Dezfooli, Ashish Shrivastava, and Oncel Tuzel. 2018. Divide, denoise, and defend against adversarial attacks. *arXiv preprint arXiv:1802.06806* (2018).

[42] Nir Morgulis, Alexander Kreines, Shachar Mendelowitz, and Yuval Weisglass. 2019. Fooling a Real Car with Adversarial Traffic Signs. *arXiv preprint arXiv:1907.00374* (2019).

[43] Y. M. Mustafah, R. Noor, H. Hasbi, and A. W. Azma. 2012. Stereo vision images processing for real-time object distance and size measurements. In *2012 International Conference on Computer and Communication Engineering (ICCCE)*. 659–663. https://doi.org/10.1109/ICCCE.2012.6271270

[44] Ben Nassi, Raz Ben-Netanel, Adi Shamir, and Yuval Elovici. 2019. Drones' Cryptanalysis-Smashing Cryptography with a Flicker. In *2019 IEEE Symposium on Security and Privacy (SP)*. IEEE, 1397–1414.

[45] Ben Nassi, Adi Shamir, and Yuval Elovici. 2018. Xerox day vulnerability. *IEEE Transactions on Information Forensics and Security* 14, 2 (2018), 415–430.

[46] Maria-Irina Nicolae, Mathieu Sinn, Minh Ngoc Tran, Beat Buesser, Ambrish Rawat, Martin Wistuba, Valentina Zantedeschi, Nathalie Baracaldo, Bryant Chen, Heiko Ludwig, Ian Molloy, and Ben Edwards. 2018. Adversarial Robustness Toolbox v1.2.0. *CoRR* 1807.01069 (2018). https://arxiv.org/pdf/1807.01069

[47] NTSB. 2020. Collision Between a Sport Utility Vehicle Operating With Partial Driving Automation and a Crash Attenuator Mountain View, California. https://www.ntsb.gov/investigations/AccidentReports/Reports/HAR2001.pdf.

[48] Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z Berkay Celik, and Ananthram Swami. 2016. The limitations of deep learning in adversarial settings. In *2016 IEEE European symposium on security and privacy (EuroS&P)*. IEEE, 372–387.

[49] Jonathan Petit, Bas Stottelaar, Michael Feiri, and Frank Kargl. 2015. Remote attacks on automated vehicles sensors: Experiments on camera and lidar. *Black Hat Europe* 11 (2015), 2015.

[50] J. Redmon and A. Farhadi. 2017. YOLO9000: Better, Faster, Stronger. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 6517–6525.

[51] Regulus. [n.d.]. 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/.

[52] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*. 91–99.
[53] Anjali Berdia Simona Shemer. 2019. Self-Driving Spin: Riding In An Autonomous Vehicle Around Tel Aviv. https://nocamels.com/2019/06/autonomous-vehicle-yandex-tech/.
[54] Chawin Sitawarin, Arjun Nitin Bhagoji, Arsalan Mosenia, Mung Chiang, and Prateek Mittal. 2018. Darts: Deceiving autonomous cars with toxic signs. *arXiv preprint arXiv:1802.06430* (2018).
[55] Dawn Song, Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Florian Tramer, Atul Prakash, and Tadayoshi Kohno. 2018. Physical adversarial examples for object detectors. *USENIX Workshop on Offensive Technologies (WOOT 18)* (2018).
[56] Jack Stewart. 2018. Why Tesla's Autopilot Can't See a Stopped Firetruck. https://www.wired.com/story/tesla-autopilot-why-crash-radar/.
[57] Jiachen Sun, Yulong Cao, Qi Alfred Chen, and Z. Morley Mao. 2020. Towards Robust LiDAR-based Perception in Autonomous Driving: General Black-box Adversarial Sensor Attack and Countermeasures. In *29th USENIX Security Symposium (USENIX Security 20)*. USENIX Association, 877–894. https://www.usenix.org/conference/usenixsecurity20/presentation/sun
[58] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. 2020. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2446–2454.
[59] Tesla. [n.d.]. Tesla Vehicle Safety Report. https://www.tesla.com/VehicleSafetyReport.
[60] Tesla. 2020. Model S - Owner's Manual. https://www.tesla.com/sites/default/files/model_s_owners_manual_north_america_en_us.pdf.
[61] Security Tutorials. [n.d.]. Hacking Digital Billboards. https://securitytutorials.co.uk/hacking-digital-billboards/.
[62] Peng Wang, Xinyu Huang, Xinjing Cheng, Dingfu Zhou, Qichuan Geng, and Ruigang Yang. 2019. The apolloscape open dataset for autonomous driving and its application. *IEEE transactions on pattern analysis and machine intelligence* (2019).
[63] Shangxi Wu, Jitao Sang, Kaiyuan Xu, Jiaming Zhang, Yanfeng Sun, Liping Jing, and Jian Yu. 2018. Attention, Please! Adversarial Defense via Attention Rectification and Preservation. *arXiv preprint arXiv:1811.09831* (2018).
[64] Takayuki Yamada, Seiichi Gohshi, and Isao Echizen. 2013. Privacy visor: Method for preventing face image detection by using differences in human and device sensitivity. In *IFIP International Conference on Communications and Multimedia Security*. Springer, 152–161.
[65] Chen Yan, Wenyuan Xu, and Jianhao Liu. 2016. Can you trust autonomous vehicles: Contactless attacks against sensors of self-driving vehicle. *DEF CON* 24 (2016).
[66] Yue Zhao, Hong Zhu, Ruigang Liang, Qintao Shen, Shengzhi Zhang, and Kai Chen. 2019. 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 (CCS '19)*. ACM, New York, NY, USA, 1989–2004. https://doi.org/10.1145/3319535.3354259
[67] Zhe Zhou, Di Tang, Xiaofeng Wang, Weili Han, Xiangyu Liu, and Kehuan Zhang. 2018. Invisible mask: Practical attacks on face recognition with infrared. *arXiv preprint arXiv:1803.04683* (2018).

Appendix 1 - Extended Discussion on the Scientific Gap

As was discussed in Section 3, object detectors do not take the following aspects into consideration:

Color - An object's color is not taken into account by many state-of-the-art (s.o.t.a.) object detectors. In practice, these object detectors base their decisions on the difference between the color of the object and the background. An object detector classifies a phantom traffic sign that was created by reducing the green component in Δ=3 in the gray background from (128,128,128) to (128,125,128). We prove that this is a fact for s.o.t.a. traffic sign recognition algorithms [3]. Table 5 presents the minimal RGB difference required to change a given background color so that s.o.t.a. algorithms will consider it real. Four out of seven of the algorithms examined can be fooled into detecting a traffic sign by reducing the green component in the gray background in Δ=6 from (128,128,128) to (128,124,128).

(iii) Texture - An object's texture is not taken into account. Object detectors classify a phantom traffic sign as a real sign even if the phantom traffic sign is partially filled or even transparent. We prove that this is a fact for s.o.t.a. traffic sign recognition algorithms [3]. Table 5 presents the minimal percentage of pixels from the original speed limit sign required to cause s.o.t.a. algorithms to consider the sparse speed limit sign as real. Four out of seven of the algorithms examined detected phantoms that were composed of less than half of the pixels from the original sign.

Therefore, although phantoms are perceived by humans as obvious fakes (defective, skewed, etc.), object detection algorithms will classify a phantom simply because its geometry matches their training examples.

Table 5: The minimal background RGB gradient required to change a gray background in order to fool existing s.o.t.a traffic sign detectors [3]. The minimal percentage of pixels of a real traffic sign required to fool the algorithms.

		RGB Components				Density
		∇ RGB	∇ R	∇ G	∇ B	Pixels(%)
Traffic Sign Detector	faster_rcnn_inception_resnet_v2 [28, 52]	12	36	12	18	80
	faster_rcnn_resnet_101 [23, 52]	6	18	6	6	45
	faster_rcnn_resnet50 [23, 52]	4	21	3	6	25
	faster_rcnn_inception_v2 [28, 52]	4	18	6	12	30
	rfcn_resnet101 [13, 23]	8	24	3	6	30
	ssd_mobilenet_v1 [24, 36]	22	66	18	36	75
	yolo_v2 [50]	10	33	21	30	55

Appendix 2 - Using the Collected Data to Calculate Phantom Projection Intensity and Size for Greater Distances

Based on the data presented in Fig. 3, we can calculate the phantom projection intensity and size required to attack an ADAS that is located at distances greater than those used in the experiment.

The required projection intensity/size of the projection to attack an ADAS that is located in greater ranges that were measured in the experiment in Section 5 can be calculated by applying linear regression to the results. For example, the following equation was created from applying a linear regression to the dataset that maps projection intensity to maximal distance:

$$\textbf{Range } (\Delta \textbf{ Lux=l}) = 0.66 \times l + 7.71 \tag{6}$$

Equation 6 results in the following: the correlation coefficient (r) = 0.98, the residual sum of squares (rss) = 0.75, and the coefficient of determination (R^2) = 0.97. The same calculation can also be applied to calculate the required phantom size to greater ranges than the ranges measured in the experiment.

Appendix 3 - Responsible Disclosure

We shared our findings with Mobileye's bug bounty via email. On 28/6/19 Mobileye responded: "There was no exploit here, no vulnerability, no flaw, and nothing of interest. The system saw an image of a street sign-good enough, accept it and move on."

We also shared our findings with Tesla's bug bounty via email. The course of events is summarized as follows:

On 1/5/19, we sent Tesla a demonstration of a phantom speed limit sign captured by Mobileye 630. Tesla replied: "This is an interesting concept. Rest assured that Tesla is doing ongoing research to

Table 6: The accuracy of the experts alone (left) and as a committee (right) when targeted with various adversarial machine learning attacks. Note that the adversarial samples were generated directly on the input images, yet it is difficult to physically create an adversarial samples for depth, light, and in some cases context –via a phantom attack.

		Accuracy of Expert				Accuracy of Committee													
	Target:	C	S	L	D	C	S	L	D	C+S	C+L	C+D	S+L	S+D	L+D	C+S+L	C+S+D	C+L+D	S+L+D	C+S+L+D
[5]	**Adversarial Patch**	6.9	64.1	99.6	50.3	99.7	99.8	99.7	99.2	99.6	98.7	95.6	99.8	98.7	96.0	99.7	99.7	99.7	99.8	99.7
[8]	**Carlini L2**	3.3	49.1	96.4	85.0	99.5	99.8	98.7	99.8	99.4	98.7	99.8	99.0	99.8	99.4	99.5	99.5	99.5	99.8	99.5
[40]	**DeepFool**	81.8	21.2	7.2	3.2	96.2	99.6	37.3	64.1	95.6	36.1	34.5	33.9	58.4	6.7	96.2	96.2	96.2	99.6	96.2
[9]	**ElasticNet**	0.0	0.2	7.9	0.8	99.5	99.7	58.5	98.7	99.6	56.5	96.4	57.9	98.7	11.1	99.5	99.5	99.5	99.7	99.5
[21]	**FGSM**	0.0	20.1	59.2	0.0	86.3	99.7	86.4	97.1	96.2	30.1	18.9	83.5	96.7	10.9	86.3	86.3	86.3	99.7	86.3
[32]	**IFGSM**	0.0	0.0	3.7	0.0	8.0	99.4	3.3	1.8	65.0	70.7	14.6	2.4	0.1	0.0	8.0	8.0	8.0	99.4	8.0
[29]	**NewtonFool**	20.4	88.7	99.3	77.4	99.4	99.8	96.7	98.3	99.4	93.0	97.6	96.3	98.3	91.7	99.4	99.4	99.4	99.8	99.4
[48]	**Saliency Map**	0.0	0.0	0.0	0.0	99.5	99.8	90.4	99.6	99.6	88.2	99.6	93.4	99.6	78.3	99.5	99.5	99.5	99.8	99.5

ensure the operational safety of our vehicles. We will be happy to receive and evaluate any further report from you and your team on practical attacks." On 20/9/19 we sent Tesla an email that contained video demonstrations of phantom attacks against their cars. On 22/9/19, we sent another email to Tesla asking them whether they believe their car to be immune against phantom attacks. At the time of writing, Tesla has not responded to our question. At

Appendix 4 - Additional Evaluations for the Countermeasure

Speed Performance

The proposed module must run in real-time and share the vehicle's computing resources. Therefore, We have performed a speed benchmark to measure it's performance. To perform the benchmark, we used an NVIDIA 2080 ti GPU and processed 100 batches of 500 signs.

We found that the module was took an average of 1.2 ms to process each sign, with a deviation of 242 ns. This translates to approximately 838 signs per second. We also note that the model is relatively small and only utilized 0.04% of the GPU per sign, maxing out at about 35% utilization at 838 signs per second.

Appendix 5 - Additional Figures

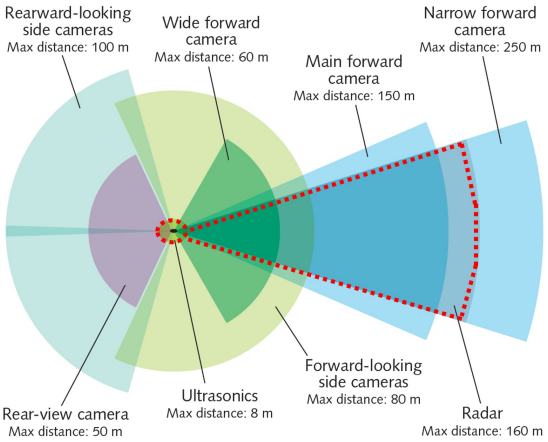

Figure 13: The areas that are covered by each of Tesla sensors. The dashed red lines show areas that are covered by depth sensors. Most of the areas around the car are covered only by video cameras.

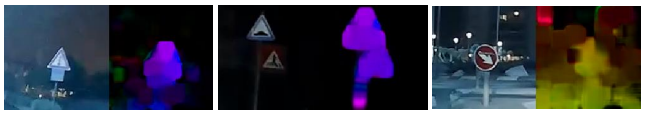

Figure 14: Examples of how dense optical flow between frames captures the depth of the environment. Left: the current frame, right: the our extracted RGB optical flow image for the Depth Model.

Figure 15: A few examples of disagreements between experts which led to correct predictions.

Figure 16: An example of a false positive, where the Combiner model failed due to a disagreement.

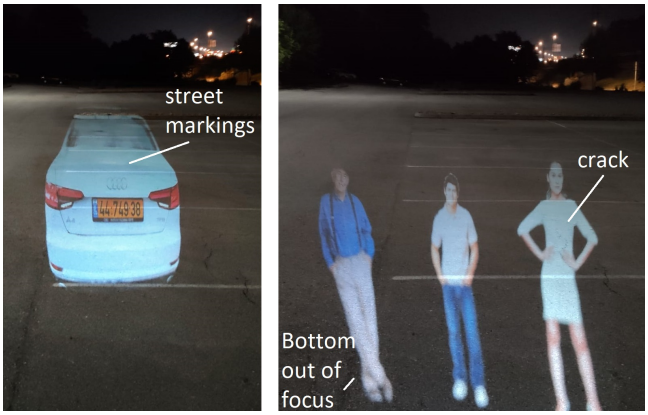

Figure 17: Examples of artifacts of road phantoms detected by the surface expert. Note that the projection is at a slight angle so although the images aren't perceived as skewed by the victim, the bottom of the phantom will still be out of focus.