Beyond Digital Domain: Fooling Deep learning Based Recognition System in Physical World

Kaichen Yang¹, Tzungyu Tsai², Honggang Yu¹, Tsung-Yi Ho², Yier Jin¹
¹ University of Florida, USA
² National Tsing Hua University, Hsinchu, Taiwan

bojanykc@ufl.edu, s107062519@m107.nthu.edu.tw, honggang.yu@ufl.edu, tyho@cs.nthu.edu.tw, yier.jin@ece.ufl.edu

Abstract

Adversarial examples that can fool deep neural network (DNN) models in computer vision present a growing threat. The current methods of launching adversarial attacks concentrate on attacking image classifiers by adding noise to digital inputs. The problem of attacking object detection models and adversarial attacks in physical world are rarely touched. Some prior works are proposed to launch physical adversarial attack against object detection models, but limited by certain aspects. In this paper, we propose a novel physical adversarial attack targeting object detection models. Instead of simply printing images, we manufacture real metal objects that could achieve the adversarial effect. In both indoor and outdoor experiments we show our physical adversarial objects can fool widely applied object detection models including SSD, YOLO and Faster R-CNN in various environments. We also test our attack in a variety of commercial platforms for object detection and demonstrate that our attack is still valid on these platforms. Consider the potential defense mechanisms our adversarial objects may encounter, we conduct a series of experiments to evaluate the effect of existing defense methods on our physical attack.

1 Introduction

Though deep learning is recognized as a promising way in processing a wide range of computer vision tasks, recent works (Szegedy et al. 2013; Goodfellow, Shlens, and Szegedy 2014; Papernot et al. 2016a; Moosavi-Dezfooli, Fawzi, and Frossard 2016; Carlini and Wagner 2017) have showed that deep learning models are vulnerable to deliberately crafted inputs known as adversarial examples. Researchers revealed the fact that adding small but intentionally selected perturbations to the original inputs can lead the target deep learning models to wrong decisions. Beyond the existence in image classification area (Szegedy et al. 2013), adversarial examples are also found in DNN models applied for other applications such as object detection (Xie et al. 2017), intrusion detection (Yang et al. 2018) and voice recognition (Yuan et al. 2018). Fast gradient sign method (FGSM) (Goodfellow, Shlens, and Szegedy 2014), Jacobian-based saliency map attack (JSMA) (Papernot et al. 2016a), DeepFool (Moosavi-Dezfooli, Fawzi, and Frossard 2016) and C&W attack (Carlini and Wagner 2017) are representative algorithms to generate adversarial examples.

Current studies of generating adversarial examples in computer vision field mainly focus on directly manipulating the pixels of the input images or videos in the digital domain. This type of adversarial attack assumes that the adversary has the ability to directly access and modify the input data of the target model (e.g., video streaming). However, in more realistic situations the DNN model is just one component of the entire system, and the input data of the DNN model is in charge by other components such as video surveillance system. In these cases the adversary may not access the digital inputs. Instead, a more realistic way for the adversary to generate adversarial examples is altering the objects physically outside the system.

Many challenges emerge when the adversary tries to generate physical adversarial examples against object detection models. First, current attacking algorithms (Goodfellow, Shlens, and Szegedy 2014; Papernot et al. 2016a; Moosavi-Dezfooli, Fawzi, and Frossard 2016; Carlini and Wagner 2017) are effective in fooling image classifier with single digital frame, but they may not deceive object detectors within the entire video frames where angles, distances and backgrounds keep on changing. In fact, it has already been shown that physical adversarial examples designed to fool an image classifier do not continuously fool a standard object detector (Lu et al. 2017), which suggests that constructing a physical adversarial example physically that can fool a detector under different environments might be hard. Second, the adversary can not directly add perturbations to anywhere in the streaming images, instead the adversary can only affect a small portion of the images by placing the adversarial object in the environment. When the adversarial objects are re-taken by the cameras, camera lens are also unable to capture full colors of the adversarial examples due to the various illumination conditions and the limitations of the lens itself.

Recently some works (Kurakin, Goodfellow, and Bengio 2018; Eykholt et al. 2018; Athalye et al. 2018; Sharif et al. 2016) tried to extend adversarial attacks to physical domain by generating adversarial examples that survive in various physical conditions. In the domain of object detection, some prior works (Song et al. 2018; Chen et al. 2018;
Lu, Sibai, and Fabry 2017) are proposed to create adversarial examples in the physical space. A perturbed stop sign is shown in (Lu, Sibai, and Fabry 2017) that cannot be detected by the Faster R-CNN object detector (Ren et al. 2015). However, the perturbation is very large and obvious, making the perturbed object hard to be defined as “stop sign” anymore. In (Song et al. 2018) they generated adversarial posters and patches against YOLO (Redmon et al. 2016) object detector, but their experiment is insufficient due to short testing videos and limited physical conditions. In (Chen et al. 2018) they successfully fooled Faster R-CNN object detector from different distances and angles, but their attack cannot fool other object detection models. All these prior works shared the following limitations: they only tested their algorithms in printed images and failed to manufacture practical and meaningful objects; their pattern of perturbations are also quite obvious and easily to be noticed by human, especially in their tests of traffic signs; they all lacked detailed experiments covering all representative object detection models and their adversarial printed images failed to transfer to other DNN models.

Considering that prior works only touched Faster R-CNN and YOLO series object detectors and physical adversarial attack against SSD object detection model (Liu et al. 2016) remains unexplored, in this paper we propose a novel physical attack against SSD object detection model and make the following contributions to address the limitations in prior works:

- We develop an effective algorithm to construct physical adversarial object (rather than printed photos) that can mislead target DNN models to certain incorrect decisions under various conditions including different distances, angles, light conditions and backgrounds.
- We successfully launch the physical adversarial attacks against DNN models applied for object detection. We start our attack at SSD model and then show our attack can transfer to other popular object detection models including Faster R-CNN and YOLO in commercial platforms with high success rates.
- We conduct a series of experiments to evaluate the effect of current defense mechanisms on our adversarial attacks.

The remainder of this paper is organized as follows: Section 2 reviews the related work. Section 3 describes our scheme of the physical adversarial attack. Section 4 presents the experimental results and analysis. Section 5 discusses the effect of some defense mechanisms. Section 6 concludes the paper.

## 2 Related Work

Since the discovery of adversarial examples by Szegedy (Szegedy et al. 2013) who observed that adding slightly but intentionally generated perturbations to legal inputs can mislead the deep learning models to incorrect decisions, many algorithms (Goodfellow, Shlens, and Szegedy 2014; Papernot et al. 2016a; Carlini and Wagner 2017; Moosavi-Dezfooli, Fawzi, and Frossard 2016) are proposed to launch more efficient and effective adversarial attacks. Despite the great progress those attacks have achieved, all these attack algorithms are only effective in the digital domain, i.e., they directly manipulate the pixels of the input images or videos under the assumption that the adversary can access and alter the inputs digitally in the DNN systems.

A more realistic threat model of the adversarial attack tries to launch attacks in the physical domain. Instead of altering the pixels of the digital inputs, physical adversarial attacks are launched by maliciously altering the images or environments outside the DNN systems. The first attempt of physical adversarial attack was proposed by Kurakin (Kurakin, Goodfellow, and Bengio 2018), who showed that some adversarial images stay effective as inputs to the classifiers after being printing out as photos and re-taken by cameras. Sharif (Sharif et al. 2016) proposed an adversarial attack against face recognition system by printing a pair of eyeglass frames that prevent individuals being recognized or impersonate another individual. They tried to maintain the effect of adversarial examples in physical world by integrating additional regularizers representing three factors: Robustness, Smoothness and Printability into the original loss function. Athalye (Athalye et al. 2018) extended the physical attack to 3D object by synthesizing examples that are adversarial over a chosen distribution of transformations called Expectation over Transformation (EOT). Eykholt (Eykholt et al. 2018) followed the way of Sharif (Sharif et al. 2016), and formulated more regularizers to represent the physical constraints on generating effective adversarial examples against image classifiers and object detectors.

Beyond adversarial examples against DNN models in image classification, many researchers tried to extend adversarial attacks to DNN models applied for object detection. Xie (Xie et al. 2017) first proposed a method to generate adversarial examples for object detection and semantic segmentation digitally. Physical adversarial attacks (Lu, Sibai, and Fabry 2017; Song et al. 2018; Chen et al. 2018) are proposed afterwards to launch attacks against object detection models by physically altering objects in the environment. Lu et al. (Lu, Sibai, and Fabry 2017) tried to attack YOLO object detector by printing adversarial images of road signs on the paper, but the success rate is not satisfactory. Song (Song et al. 2018) demonstrated that YOLOv2 object detection model can be fooled by printed images and stickers under some physical conditions. Their RP2 algorithm extended Eykholt’s work (Eykholt et al. 2018) that combined several regularizers representing physical factors. Chen (Chen et al. 2018) showed that Faster R-CNN object detector is also vulnerable to physical adversarial object. Their ShapeShifter algorithm applied Athalye’s EOT algorithm (Athalye et al. 2018) to simulate the potential transformation faced by adversarial objects in physical world. Current schemes of physical adversarial objects limited their experiments in printed images and failed to manufacture practical and meaningful objects. Their pattern of perturbations are also quite obvious and easily to be noticed by human. They also lack detailed experiments covering all representative object detection models and commercial platforms.

Several approaches are proposed to defend adversarial ex-
amples. Adversarial training proposed by Tramer (Tramèr et al. 2017) tries to make the model itself more robust during the training stage by augmenting the original training dataset with more pre-crafted adversarial examples. Defensive distillation proposed by Papernot (Papernot et al. 2016b) tries to re-train the model by smoothing the potential adversarial gradients which may be easily applied to craft the adversarial examples. Guo (Guo et al. 2017) tried to apply some pre-processing methods to defend potential adversarial inputs, such as compression and transformation.

3 Physical Attack

In this section we will present our algorithms of physically attacking SSD object detector. First we introduce some background knowledge about the target SSD model. We then describe the threat model. At last we present our methods of integrating iterative optimizations with outputs from SSD models and various physical constraints.

3.1 SSD Model

Object detectors locate and classify multiple objects in a given scene. Current popular deep neural network architecture for object detection can be roughly categorized into two classes: proposal based models like Faster R-CNN (Ren et al. 2015) and regression based models such as YOLO (Redmon et al. 2016), SSD (Liu et al. 2016). Proposal based models treat object detection as a two-stage problem consisting of region proposals followed by classifications for each of these regions. Faster R-CNN can achieve 83.8% mAP on VOC dataset, which is higher than YOLO and SSD. However, proposal based models such as Faster R-CNN suffer from low processing speed. In contrast, regression based models such as YOLO and SSD run a single Convolutional Neural Network (CNN) over the input image to jointly produce bounding boxes for object localization and confidence scores for classification. As the result, these networks can achieve similar accuracy as proposal based models while processing images much faster.

In this work, we focus on SSD model since it has the same level accuracy as Faster R-CNN while maintaining much higher processing speed. Further the adversarial attack against SSD is rarely discussed so far.

SSD only needs an input image and ground truth boxes for each object during training. Different from Faster R-CNN which applies sliding multibox to generate prediction bounding box, SSD and YOLO all evaluate a small set of default boxes of grid cells which in charge of prediction bounding boxes and class labelling. It enables SSD and YOLO the capability to process video in real time. To address the low accuracy the YOLO encountered, SSD adds six extra layers to evaluate default boxes of grid cells with different aspect ratios at each location in several feature maps with different scales. These six extra layers help SSD detect objects with multi-scale vision and achieve higher accuracy than YOLO while maintaining high processing speed.

3.2 Threat Model

Our work assumes an adversary who wishes to attack an object detection system by manipulating certain objects in the physical world. Specifically, our target model is an SSD model trained for vehicle license plate recognition and the adversary will try to launch the attack by designing and manufacturing certain types of license plates. We chose vehicle license plate instead of commonly tested traffic signs (Song et al. 2018; Chen et al. 2018) for the following reasons:

- Recognizing vehicle license plates in an accurate and real-time fashion is a realistic demand shared by automatic driving, highway management and monitoring at certain sites. Deep learning based detection systems are gradually applied in license plate recognition commercially. It is interesting to see the effect of adversarial attacks on such systems.

- The area of the license plate is much smaller than the area of the roadside sign, which makes it more difficult and challenging to add noise to the license plate for adversarial attacks.

- The adversarial attacks against road signs (e.g., stop sign) have been widely achieved before. However, these attacks are launched by printing the perturbed images of the road signs. Though optimized to limit the perturbation, these printed perturbed images are still obvious and easily to be detected and removed. In contrast, customized vehicle license plates widely exist in north America. It is practical to generate adversarial vehicle license plates by adding custom image in an unobtrusive way.

- The area of the license plate is much smaller than the area of the roadside sign, which makes it more difficult and challenging to add noise to the license plate for adversarial attacks.

We further assume that the adversary has white-box level access to the target SSD model. This means the adversary can access the outputs of the model and is aware of the internal information such as gradients. Instead of directly manipulating the pixels of the input images or videos, the adversary can only launch attacks by altering the physical objects (license plate in our case). Another constraint we put on adversary is limiting the pattern of perturbation added on the license plate. The attack will be meaningless if the license plate is so blurred that even human cannot recognize it as a license plate. It also violates the law if characters on the plates (i.e., numbers and texts) cannot be clearly recognized.

3.3 Attacking Algorithms

Our attacking algorithms followed the methods of generating adversarial examples proposed by Carlini and Wagner (Carlini and Wagner 2017). We integrate their C&W attack with the outputs of the SSD model along with additional constraints from colors and brightness. Given an digital template of the license plate $x$ as Figure 1(a) shows, we add perturbations $\delta$ on it to generate the digital adversarial license plate $x' = x + M \cdot \delta$, $M$ denotes the mask which defines the area where the perturbation can be added. We define three kinds of masks to prevent the important characters on plates from blurring while fit the popular styles of vehicle license plates:

- In mask 1 the perturbation is limited to the blank left part on the license plate, as Figure 1(b) shows, it is a common design of vehicle license plate.

- In mask 2 the perturbations is limited to the whole background of the license plate, as Figure 1(c) shows. It is also
M

cense plate shown in Figure 1(a),
where

\( k \)

We select top \( k \) predictions as a confidence score vector.

The bounding box is represented as four coordinates and the

class prediction is represented as a confidence score vector.

of SSD model are 1917 predictions (in our case) which con-
duction. Given an input image or video frame, the outputs

will act as inputs to the target SSD model for license plates


detection. For the distance, angle, and background, we simulate them

by placing adversarial license plates in a number of real pictures
which contain license plates with various angles, distances and background conditions as Figure 2 shows.

After the perturbation is added to the digital template of the ve-
hicle license plate, the adversarial license plate as an image
will be transformed and placed in a number of real back-
grounds, as Figure 2 shows. After

the transformation to our simulated images taken from the
camera. By experiencing the resizing and rotation process
in different real license plate pictures with various back-
grounds, the adversarial template of the license plates will
be robust under different physical conditions. To address
the problem of unstable illumination, we perform gray-scale
transformation to our simulated images taken by camera.

Particularly, by adopting gamma correction, we can simu-
late the images under different intensity illumination. The
overall process of placing and gamma correction constitute
the transformation function \( T \). The simulated images are de-
noted as

\[ x' = x + M \cdot \delta \]

\[ X_i = T(x', r_i) \tag{1} \]

where \( x \) denotes the digital template of the clean vehicle li-
cense plate shown in Figure 1(a), \( \delta \) denotes the perturbation,
\( M \) denotes the mask, and \( r_i \) denotes the \( i \)-th real background
image.

The set \( S \) of simulated images of adversarial license plates
will act as inputs to the target SSD model for license plates
detection. Given an input image or video frame, the outputs
of SSD model are 1917 predictions (in our case) which con-
sist bounding boxes and corresponding class label vectors.
The bounding box is represented as four coordinates and the

class prediction is represented as a confidence score vector.

We select top \( k \) predictions of the output as the target since

they are sufficient to represent the final predictions of the
SSD model. To achieve the attack, we apply the following

loss function:

\[ \ell_{CW} = \mathbb{E}_{X_i \in S} J(f(X_i), y^*) \tag{2} \]

where \( f \) denotes the extract function of SSD model which
extracts the logits before softmax or sigmoid layer from top
\( k \) predictions, and \( J \) denotes a C&W-like objective function
that measures the logits distance between the target class and
original class, \( J \) can be calculated as:

\[ J(f(X), y^*) = \sum_{j=1}^{k} (Z(X)_{y^*} - Z(X)_{t_j}) \tag{3} \]

where \( Z(X)_{y^*} \) denotes the logits of the original class and
\( Z(X)_{t_j} \) denotes the logits of the target class. By modify-
ing the loss function above, we can launch the hiding at-
tack against object detection model applied for license plate
recognition.

**Hiding attack.** By setting \( y^* \) as class ‘plate’ and \( t \) as class
‘background’ the adversary can launch the hiding attack. In
the hiding attack the adversary tries to make specifically de-
digned license plate invisible in front of the object detection
models in the physical world by pushing the target model to
misclassify class ‘plate’ to class ‘background’. The distance
loss that measures the \( L_2 \) distance (\( p = 2 \)) between the clean
template and the perturbed template is expressed as

\[ \ell_{L_2} = c \cdot \| \delta \|_2^2 \tag{4} \]

In addition to the loss function \( \ell_{CW} \) that push the model to
output target class and \( \ell_{L_2} \) that push the \( L_2 \) distance, we
also need to consider the color constraints coming from the
plate manufacturing process. Considering the color space of
the printing machine will be limited, we set a constraint on
the total number of colors the perturbation could have, de-
noted as \( C_N \). Further considering the printing machine may
fail to accurately distinguish pixelated patterns, we followed
the smoothing method of total variation as (Song et al. 2018)
to encourage the pattern of perturbation being smooth and
continuous. The total variation of the perturbation pattern

\[ \ell_{TV} = \| \nabla \delta \|_2 \]

where \( \nabla \delta \) is the gradient of the perturbation. The color
constraint can be formulated as

\[ \ell_C = \frac{1}{N} \sum_{i=1}^{N} \left( I_{\text{pixels}}(x_i) - I_{\text{template}}(x_i) \right) \]

where \( I_{\text{pixels}}(x_i) \) and \( I_{\text{template}}(x_i) \) are the pixel values

of the perturbed image and the template image at position

\( x_i \), respectively. The total variation loss and color con-
straint loss are combined together to form the final objective
function:

\[ \ell_{final} = \frac{1}{\lambda} \left( \lambda_1 \ell_{TV} + \lambda_2 \ell_C \right) \]

where \( \lambda, \lambda_1, \lambda_2 \) are hyperparameters that control the
contribution of each loss term. By solving the optimization
problem with this objective function, we can generate adver-
sarial license plates that are robust against attacks from
object detection models.
can be formulated as:

\[
TV(M \cdot \delta) = \sum_{i,j} \left| (M \cdot \delta)^j_{i+1} - (M \cdot \delta)^j_i \right| + \left| (M \cdot \delta)^{j+1}_i - (M \cdot \delta)^j_i \right|.
\]  

(5)

So we formulate the object function for color constraints as follows:

\[
\ell_{\text{color}} = \alpha C_N + \beta TV(M \cdot \delta).
\]  

(6)

Given an SSD object detector model, our final robust spatially constrained perturbation is generated by iteratively optimizing the following object function:

\[
\arg \min_{\delta} \ell = \ell_{\text{CW}} + \ell_{\text{Lz}} + \ell_{\text{color}}.
\]  

(7)

Figure 3 depicts the process of optimizing the perturbation.

4 Experimentation

We evaluated our physical adversarial object using the state-of-the-art object detector SSD. Our SSD model is trained with InceptionV2 (Szegedy et al. 2016) convolutional network as the base network. The dataset we used to train the model contains 5012 images of license plates collected from the internet. For the hiding attack, the target SSD model is trained by this dataset labeled by only two possible classes ‘plate’ and ‘car’ since in this kind of attack we only concentrate on making the vehicle license plates invisible in front of the target SSD model and other classes are irrelevant. We first design the digital version of our adversarial vehicle license plate according to the physical attack algorithm described before, we then manufacture these plates. The size of the manufactured adversarial vehicle license plate is standard 6” x 12” and the material used to build the license plate is aluminum (0.40). We evaluate the effect of our manufactured adversarial vehicle license plates under various conditions including indoor environment and outdoor environment. In the outdoor environment the adversarial vehicle license plates were placed on vehicles in the parking station.

4.1 Experiment Setup

All experiments were carried out on a server with an Intel E5-2623 v4 2.60GHz CPU with 16GB RAM, Ubuntu 18.04, accelerated by NVIDIA CUDA Framework 10.0 and cuDNN 7.0 with two NVIDIA GeForce RTX 2080Ti GPUs. The videos for evaluation are taken by SONY DCR-SR40 Handy-cam and the built-in camera of iPhone 6s. We trained the SSD model using the Tensorflow Object Detection API (Huang et al. 2017).

4.2 Experimental Results

Digital Template Starting with a clean digital template of the vehicle license plate shown in Figure 1, we generated the adversarial version by performing the optimization process defined in Equations (1)-(7) to find the suitable perturbation. The hyperparameters are chosen as follows: \(\alpha = \beta = 10^{-7}\), \(c = 0\). Since we believe that the attack is valid as long as the important characters on the plate can be clearly identified, we set \(c = 0\) to put no limit on the scale of the perturbation.

| SSD_Inception | indoor | outdoor |
|---------------|--------|---------|
| clean         | 478504 (94.8%) | 606625 (97.0%) |
| mask1         | 180612 (29.4%)  | 504631 (79.9%)  |
| mask2         | 59528 (11.2%)   | 178665 (25.8%)  |
| mask3         | 64540 (11.9%)   | 309733 (42.2%)  |

Table 1: Detection rate of the hiding attack on SSD model.

We define three kinds of masks where perturbations are allowed to be added in the previous section. We use 500 real images with vehicle license plates and replace the license part with our digital template to simulate the taken images containing adversarial license plate. After iterations of optimization the resulted digital version of the adversarial vehicle license plate with three different masks are shown in Figure 4.

Hiding Attack We test the hiding attack in both the indoor environment and the outdoor environment. In the indoor environment, we recorded videos of the manufactured adversarial license plates at a variety of distances (1m to 4m) and angles (0-60 from the plate’s tangent). The camera always pointed at the plate. The videos served as the inputs to our trained SSD_Inception model. For comparison we also tested clean license plates. We count the frames that the license plates are detected to evaluate the effect of the hiding attack. In the outdoor environment we place the adversarial vehicle license plate on a real car and recorded videos from a variety of distances (1m to 5m) and angles (0-60, from the plate’s tangent) using the build-in cameras of iPhone 6s. Some frames of the detection video are shown in Figure 5.

The results of the adversarial attack on SSD_Inception model are shown in Table 1. We tested three adversarial license plates with different masks of perturbations and one clean license plate. The table cells show the ratio: number of frames in which a license plate was detected / total number of frames, and a detection rate, which is the result of this ratio. We see that the SSD_Inception model we trained can detect clean license plate with high success rate (94.8% indoor and 97.0% outdoor). Our adversarial license plates can severely reduce the detection rate of the target SSD_Inception model in indoor and outdoor environment, especially for perturbation with mask 2 that can reduce the detection rate to 11.2% indoor and 26.8% outdoor.

From Table 1 we notice that the effect of our physical adversarial objects drops in the outdoor environment. One reason for this phenomenon may be the similarity between the scenes which license plate placing on the car in the outdoor environment with the images in the training set. The lower effect of the adversarial license plate with perturbation mask 1 might be due to the smaller space of the perturbation.

Evaluation of Transferability In the previous sections we show the adversarial license plates constructed following our methods can be hidden from the SSD_Inception model in white-box setting with high probability (reducing detection rate to 11.2% indoor and 26.8% outdoor.) However, in real-world scenarios the adversary may fail to access the internal information of the target object detection models.
To explore the transferability of our adversarial vehicle license plates, we fed our recorded videos to four extra models which are also applied for license plates recognition. The attack against these four extra models is in black-box model since the adversary has no access to the internal information of these target models.

The first model we tested is another SSD model with different base network, VGG. The dataset used to train this SSD_VGG model is the same as the one used to train the previous SSD_Inception model. The results of the attack against SSD_VGG model are shown in Table 2. We can see from these results that our adversarial license plates transfer with a relatively high probability in indoor settings where the environment conditions are stable. However, once outdoors, the effect for all adversarial license plates decreases significantly, but all of them retain moderate adversarial effect, especially for mask 2 based attacks.

The second model we tested is a Faster R-CNN model trained by the same dataset as the one used to train the previous two SSD models. The base network of the Faster R-CNN is the Inception network. The results we get on Faster R-CNN model are quite similar with the one on original SSD_Inception, as shown in Table 3. It demonstrates that the existence of transferability between SSD model and Faster R-CNN model. The higher performance of the same recorded video comparing to SSD_VGG may be due to the same base network of the original SSD_Inception model.

The third model we tested is a YOLO based model. Different from the previous models which we trained ourselves with the same dataset, this YOLO based model is provided by a commercialized website which runs license plate recognition business through a prediction API
| Model          | Indoor          | Outdoor         |
|---------------|-----------------|-----------------|
| Clean         | 504/504 (100.0%) | 624/625 (99.8%) |
| Mask 1       | 320/612 (52.3%)  | 630/631 (99.8%) |
| Mask 2       | 5/528 (0.9%)    | 216/665 (32.5%) |
| Mask 3       | 167/540 (30.9%)  | 449/733 (61.3%) |

Table 2: Detection rate of the hiding attack on SSD_VGG model.

| Model          | Indoor          | Outdoor         |
|---------------|-----------------|-----------------|
| Clean         | 466/504 (93.5%)  | 572/625 (91.5%) |
| Mask 1       | 260/612 (42.5%)  | 144/631 (22.8%) |
| Mask 2       | 48/528 (9.1%)    | 6/665 (0.9%)    |
| Mask 3       | 114/540 (21.1%)  | 92/733 (12.6%)  |

Table 3: Detection rate of the hiding attack on Faster R-CNN model.

| Model          | Indoor          | Outdoor         |
|---------------|-----------------|-----------------|
| Clean         | 483/504 (95.8%)  | 600/625 (96.0%) |
| Mask 1       | 480/612 (78.4%)  | 519/631 (82.3%) |
| Mask 2       | 17/528 (0.2%)    | 0/665 (0.0%)    |
| Mask 3       | 394/540 (73.0%)  | 367/733 (50.1%) |

Table 4: Detection rate of the hiding attack on YOLO based model.

| Model          | Indoor          | Outdoor         |
|---------------|-----------------|-----------------|
| Clean         | 473/504 (93.8%)  | 625/625 (100.0%)|
| Mask 1       | 543/612 (88.7%)  | 560/631 (88.7%) |
| Mask 2       | 78/528 (14.8%)   | 2/665 (0.3%)    |
| Mask 3       | 419/540 (77.6%)  | 383/733 (52.3%) |

Table 5: Detection rate of the hiding attack on OpenALPR.

| Model          | Indoor          | Outdoor         |
|---------------|-----------------|-----------------|
| Clean         | 478/504 (94.8%)  | 562/625 (90.0%) |
| Mask 1       | 578/612 (94.4%)  | 603/631 (95.5%) |
| Mask 2       | 342/528 (64.7%)  | 276/665 (41.5%) |
| Mask 3       | 403/540 (74.6%)  | 598/733 (81.5%) |

Table 6: Detection rate of the model defended by adversarial training.

5 Evaluation of Defenses

We evaluate our physical attack under three defense mechanisms: adversarial training (Tramèr et al. 2017), distillation (Papernot et al. 2016b) and input transformation (Guo et al. 2017). For adversarial training, we add 20 adversarial images to the original training set and retrain a SSD model. For distillation, we smooth the training process by applying soft labels. For the input transformation, we resize to the input images to 200x200, 300x300, ..., 600x600 and then rotate them with degree 90, 180, 270 randomly. The detection rates of the defended model with the attack video are shown in Table 6, Table 7 and Table 8. From the tables we can see that the performance of our attack is not severely affected by the distillation and input transformation, illustrating the invalidity of these defense methods when facing robust physical adversarial objects. On the other hand, we can see that adversarial training achieves the best performance against our attack, it can be due to the possibility that the model memorizes the “noisy” numbers and texts. These experimental results indicate that adding intentionally noised samples to the training dataset may significantly reduce the risk of potential physical adversarial examples.

6 Conclusion

We show that the state-of-the-art SSD object detector is vulnerable to physically manufactured adversarial objects. Targeting on a specific deep learning based object detection model for license plate detection, we successfully lead the target model to neglect the existence of the perturbed li-
cense plates in different environments, revealing the threat of robust physical adversarial objects against current recognition systems based on deep learning. In the experiments we prove that our physical adversarial object is not only effective to the target model we trained ourselves, but also other black-box DNN models served for the similar task. Our attack scheme is even effective to those methods that do not rely on deep learning techniques. Our work reveals the urgency of practical defense mechanisms against physical adversarial objects. In the future, we will look into the counter measures that can efficiently detect physical adversarial attacks or smooth the effect of them.

References

Athalye, A.; Engstrom, L.; Ilyas, A.; and Kwok, K. 2018. Synthesizing robust adversarial examples. In Dy, J., and Krause, A., eds., Proceedings of the 35th International Conference on Machine Learning, volume 80 of Proceedings of Machine Learning Research, 284–293. Stockholmsmässan, Stockholm Sweden: PMLR.

Carlini, N., and Wagner, D. 2017. Towards evaluating the robustness of neural networks. In Proceedings of the Security and Privacy (SP) on 2017 IEEE Symposium. IEEE.

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

Eykholt, K.;Evtimov, I.; Fernandes, E.; Li, B.; Rahmati, A.; Xiao, C.; Prakash, A.; Kohno, T.; and Song, D. 2018. Robust physical-world attacks on deep learning visual classification. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2014. Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572.

Guo, C.; Rana, M.; Cisse, M.; and van der Maaten, L. 2017. Countering adversarial images using input transformations. arXiv preprint arXiv:1711.00117.

Huang, J.; Rathod, V.; Sun, C.; Zhu, M.; Korattikara, A.; Fathi, A.; Fischer, I.; Wojna, Z.; Song, Y.; Guadarrama, S.; 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.

Kurakin, A.; Goodfellow, I. J.; and Bengio, S. 2018. Adversarial examples in the physical world. In Artificial Intelligence Safety and Security. Chapman and Hall/CRC. 99–112.

Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; and Berg, A. C. 2016. Ssd: Single shot multibox detector. In European conference on computer vision, 21–37. Springer.

Lu, J.; Sibai, H.; and Fabry, E. 2017. Standard detectors aren’t (currently) fooled by physical adversarial stop signs. arXiv preprint arXiv:1710.03337.

Moosavi-Dezfooli, S.-M.; Fawzi, A.; and Frossard, P. 2016. Deepfool: a simple and accurate method to fool deep neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. IEEE.

Papernot, N.; McDaniel, P.; Jha, S.; Fredrikson, M.; Celik, Z. B.; and Swami, A. 2016a. The limitations of deep learning in adversarial settings. In Proceedings of the Security and Privacy (EuroS&P) on 2016 IEEE European Symposium. IEEE.

Papernot, N.; McDaniel, P.; Wu, X.; Jha, S.; and Swami, A. 2016b. Distillation as a defense to adversarial perturbations against deep neural networks. In Security and Privacy (SP), 2016 IEEE Symposium on, 582–597. IEEE.

Redmon, J.; Divvala, S.; Girshick, R.; and Farhadi, A. 2016. You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition, 779–788.

Ren, S.; He, K.; Girshick, R.; and Sun, J. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In Advances in neural information processing systems, 91–99.

Sharif, M.; Bhagavatula, S.; Bauer, L.; and Reiter, M. K. 2016. Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition. In 23rd ACM Conference on Computer and Communications Security, CCS 2016, 1528–1540. Association for Computing Machinery.

Song, D.; Eykholt, K.; Evtimov, I.; Fernandes, E.; Li, B.; Rahmati, A.; Tramer, F.; Prakash, A.; and Kohno, T. 2018. Physical adversarial examples for object detectors. In 12th {USENIX} Workshop on Offensive Technologies ({WOOT} 18).

Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; and Fergus, R. 2013. Intriguing properties of neural networks. arXiv preprint arXiv:1312.6199.

Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; and Wojna, Z. 2016. Rethinking the inception architecture for computer vision. In Proceedings of the IEEE conference on computer vision and pattern recognition, 2818–2826.

Tramèr, F.; Kurakin, A.; Papernot, N.; Goodfellow, I.; Boneh, D.; and McDaniel, P. 2017. Ensemble adversarial training: Attacks and defenses. arXiv preprint arXiv:1705.07204.

Xie, C.; Wang, J.; Zhang, Z.; Zhou, Y.; Xie, L.; and Yuille, A. 2017. Adversarial examples for semantic segmentation and object detection. In Proceedings of the IEEE International Conference on Computer Vision, 1369–1378.

Yang, K.; Liu, J.; Zhang, C.; and Fang, Y. 2018. Adversarial examples against the deep learning based network intrusion detection systems. In MILCOM 2018-2018 IEEE Military Communications Conference (MILCOM), 559–564. IEEE.

Yuan, X.; Chen, Y.; Zhao, Y.; Long, Y.; Liu, X.; Chen, K.; Zhang, S.; Huang, H.; Wang, X.; and Gunter, C. A. 2018. Commander-song: A systematic approach for practical adversarial voice recognition. In 27th {USENIX} Security Symposium ({USENIX} Security 18), 49–64.

| Input transformation | indoor | outdoor |
|----------------------|--------|---------|
| clean                | 395/504 (78.3%) | 498/625 (79.6%) |
| mask1                | 236/612 (38.6%) | 69/631 (10.9%) |
| mask2                | 71/528 (13.4%) | 46/665 (6.9%) |
| mask3                | 180/540 (33.31%) | 115/733 (15.6%) |

Table 8: Detection rate of the model defended by input transformation.