HIDING OBJECTS FROM DETECTORS: EXPLORING TRANSFERRABLE ADVERSARIAL PATTERNS

Anonymous authors
Paper under double-blind review

ABSTRACT

Adversaries in neural networks have drawn much attention since their first debut. While most existing methods aim at deceiving image classification models into misclassification or crafting attacks for specific object instances in the object setection tasks, we focus on creating universal adversaries to fool object detectors and hide objects from the detectors. The adversaries we examine are universal in three ways: (1) They are not specific for specific object instances; (2) They are image-independent; (3) They can further transfer to different unknown models. To achieve this, we propose two novel techniques to improve the transferability of the adversaries: *piling-up* and *monochromatization*. Both techniques prove to simplify the patterns of generated adversaries, and ultimately result in higher transferability.

1 INTRODUCTION

Despite the success of machine learning and deep learning models, recently it has been shown that these models are susceptible and sensitive to what is termed as *adversarial examples*, a.k.a. *adversaries* (Szegedy et al., 2013; Goodfellow et al., 2015). Adversaries are usually derived from ordinary data and retain the same semantic content, but can result in wrong predictions. Previous studies have shown that adversarial examples can be crafted efficiently and successfully in some conditions, which poses significant security threats (Kurakin et al., 2018). Formally speaking, given a model $y = F(x)$, input X and original or ground-truth output $Y = F(X)$, adversaries are modified versions of the original data, denoted as $X + \Delta X$ such that $F(X + \Delta X) \neq Y$. Generally, ΔX is constrained by its norm value (e.g. L_∞) or other metrics to preserve the original semantic meaning of input X.

Existing studies on adversarial examples focus on (1) designing effective and efficient methods to craft ΔX, e.g. L-BFGS (Szegedy et al., 2013), FGSM (Goodfellow et al., 2015), iterative methods (Kurakin et al., 2016); (2) defense methods including defensive distillation (Papernot et al., 2016b), random transformation (Xie et al., 2017), JPEG-compression (Dziugaite et al., 2016) and etc.; (3) how to improve the transferability of attacks crafted on one model to deceive another model, both for differently initialized and trained models, and models of different architecture (Liu et al., 2016b; Papernot et al., 2016a; Tramèr et al., 2017; Wu et al., 2018). Up till now, these efforts mainly focus on image classification models.

More recent work has studied the robustness of object detectors and tried to fool these models (Lu et al., 2017b; Chen et al., 2018; Eykholt et al., 2018; Li et al., 2018b;a; Rosenfeld et al., 2018). However, most of these works only attack specific object instances. Few proposed methods have attempted to attack multiple objects and images or verify the capacity to transfer to another model.

In this work, we aim to craft universal and transferable adversaries to fool object detectors and conceal objects. As far as we know, we are the first to carry out such large-scale attacks on object detectors. Our target is three-fold: (1) The adversary should work for different objects, regardless of their types, positions, sizes, and etc.. (2) The adversary is not limited to one image only, i.e. achieving image-independence. (3) The adversary should be able to attack detectors that they are not crafted on, i.e. achieving black-box attack.

Specifically, we craft an *adversarial mask* of the same size as input image, denoted as $\Delta X \in [0, 1]^{H_{image} \times W_{image} \times 3}$, and impose a norm-value constraint, $||\Delta X||_\infty \leq \epsilon$. Such an adversarial

mask is in fact similar to what the community has used to fool image classification models. However, optimizing over it is a non-trivial task. A full-sized mask would introduce a total amount of $0.5M$ parameters, putting our method on risk of overfitting. Further, using the concept of Effective Receptive Field (Luo et al., 2016), we found that gradients obtained through back propagation are sparse in spatial positions, making optimization difficult.

To achieve our objective, we propose to use the following techniques: (1) Optimizing ΔX over a set of images; (2) Using identical small patches that are piled-up to form the full-sized mask ΔX; (3) Crafting monochromatic masks instead of colorful ones as done in previous work. Our motivation is that piling-up identical small patches in a grid can incorporate translation invariance in a similar way to Convolutional Neural Networks (CNNs), which is also connected with the intuition that any part of the mask should perform equally to attack an object in any position. Constraining the adversarial mask to monochrome further forces the mask to learn coarse-grained patterns that may be universal.

In experiments, we compare with decent baseline methods and found that our methods can consistently surpasses them. While our adversarial mask can conceal as many as 80% objects from YOLO V3 (Redmon & Farhadi, 2018), on which it is crafted, it can also hide more than 40% objects from the eyes of Faster-RCNN (Ren et al., 2015), in a black-box setting. Further, we compare the patterns generated by different methods and carry out detailed analysis. We found that our techniques did help in crafting more coarse-grained patterns. These patterns have generic appearance, which we attribute as the key for good transferability.

In conclusion, we make the following contributions in this work: (1) We successfully craft universal adversarial mask that can fool object detectors that are independent in object-level, image-level and model-level. (2) We show that, with the proposed techniques, we can learn and generate masks that have generic and coarse-grained patterns. The pattern we generate is different from those in previous works by large, which may be the key for better transferability.

2 Related Work

Norm-Ball Attack Sabour et al. (2015) first demonstrates how deep learning models can be fooled by images, denoted as $X \in [0, 1]^{H \times W \times 3}$, that are mixed with imperceptible perturbations, denoted as $\Delta X \in R^{H \times W \times 3}$. Later, various methods for crafting such perturbations have been proposed (Szegedy et al., 2013; Goodfellow et al., 2015; Liu et al., 2016b; Kurakin et al., 2016; Brendel et al., 2018; Carlini & Wagner, 2017; Tramèr et al., 2017; Elsayed et al., 2018). A major common characteristic for these methods is that, the crafted perturbations satisfied the following constraint: $||\Delta X||_\infty \leq \epsilon$, where ϵ measures how much the images are perturbed. These efforts mainly focus on image classification models. Few shed light on object detectors. We also refer readers to these comprehensive surveys for more detailed introduction (Hazan et al., 2016; Gilmer et al., 2018; Kurakin et al., 2018).

Efforts on Transferability In real world application, the attackers usually have no knowledge about the target models, including their architecture, hyper-parameters, and learned parameters. Such situation is termed as *black-box* attack. Transferability between different models is thus a proxy for black-box methods, and several methods have been proposed. Ensemble attack (Tramèr et al., 2017) is based on the assumption that if an adversary can fool a set of N models, it is more likely to be able to generalize well and fool a $N + 1$-th one. Wu et al. (2018) analyze the cosine similarity between gradient obtained from different models and propose to smooth the loss landscape (Smilkov et al., 2017) to improve the generalization capacity among models. Specifically, they optimize over a set of data points sampled from the norm-ball of the target image. Another similar work (Brown et al., 2017) demonstrates how to generate image-independent adversaries for image classification. They optimize an adversarial patch that has not norm-value constraint but can only modify a small region of the target images. By optimizing over a set of images, the trained patch can transfer to new images successfully.

Attack on Object Detector Methods to attack object detectors can be categorized into two classes: (1) stickers that are glued onto target objects to interfere with classification or onto backgrounds as counterfeit objects (Chen et al., 2018; Eykholt et al., 2018); (2) perturbation masks that are aligned to and trained for one specific object (Lu et al., 2017b) or one image only (Li et al., 2018a;b). In a

nutshell, these methods are specific to designated object instances, which means that to successfully fool detectors, one needs to craft adversaries and attacks the target objects one-by-one. Lu et al. (2017a) is the first to explore the possibility of transferability. However, the success rate it not very promising.

Recently, Rosenfeld et al. (2018) demonstrate the effects of feature inference, where randomly transplanted generic objects prove to have non-local adversarial effects, distorting detection results even far from the original transplantation position. More concretely, features attained from areas that do not belong to the object of interest have an impact on the detectors' behavior. This holds true both for pixels inside the region-of-interest (ROI) of the object and for those outside of it. The wide-range existence of such phenomenon is a proof that the obejct detectors are fragile and sensitive. Note that the probing approaches used in Rosenfeld et al. (2018) are not practical attack strategies, as the authors' method is a type of random search and the results are random. Such method may also be dependent on the architecture of target models, as we implemented this method on YOLO V3 but did not observe similar results. Besides, Rosenfeld et al. (2018) did not study how the objectiveness is influenced in this setting. Extending from Rosenfeld et al. (2018), we use a learned mask to probe how to hide an object by modifying its surroundings in a systematic way.

3 Background: the Object Detection Task

Object detection aims to localize the existence of objects of interest, and recognize the categories of them. There are mainly two branches, i.e. region-proposal based methods, including RCNN by Girshick et al. (2014), Fast-RCNN by Girshick (2015) and Faster-RCNN by Ren et al. (2015), and unified methods including SSD by Liu et al. (2016a) and YOLO by Redmon et al. (2016). In our research, we experiment with YOLOv3 as it runs the fastest and also performs at state-of-the-art level. We do experiment to see how well the adversaries crafted on YOLOv3, representative of unified methods, can transfer to Faster-RCNN, which is also a representative method for region-proposal based methods. We briefly introduce the core concept of YOLOv3 and Faster-RCNN.

YOLO V3 performs two functions: (1) spotting the existence of objects of interest, i.e. those in a pre-defined list; (2) classifying spotted objects into the correct categories. Input images are first fed into the *backbone* network, producing a sequence of $H \times W \times C$ feature maps. Each $1 \times 1 \times C$ vector represents the potential object at the corresponding position. Classifiers, which are 1×1 convolutional layers in practice, predict the existence of objects, its types and positions. Non-Maximal Suppression (NMS) is performed to deliver the final results. YOLO V3 has a set of 3 classifiers, each deployed in different layers and aimed at objects of different sizes. In total, there are $N_P = 10647$ such prediction points, also termed as *anchor*. In essence, YOLO V3 can be viewed as a *multi-head* image classifier.

Faster-RCNN incorporates a Region-Proposal Network (RPN) to make detection proposals, which are bounding boxes indicating the existence of objects. Sub-regions are cropped from a shared feature map to perform classification. However, the detection and localization of objects in Faster-RCNN is solely dependent on RPN, which works in the same way as YOLO V3.

4 Methodology

In this section, we introduce how to *obtain* adversarial masks, $\Delta X \in [-\epsilon, \epsilon]^{H \times W \times 3}$, and further introduce the two techniques we propose to generate adversarial masks that better transfer to other settings. Note that the attack is performed on YOLO V3, and therefore $H = W = 416$.

4.1 Baseline Method: Full-Mask Generation

The simplest way is to follow the tradition in adversaries for image classification, model the mask as a $416 \times 416 \times 3$ parameter, and optimize over some metric. We denote it as $\Delta X = m_{full} \in [-\epsilon, \epsilon]^{416 \times 416 \times 3}$. To conceal objects, we minimize the objectiveness score produced by the model. We set the minimization target as the average log-likelihood of the top-200 anchors of highest scores in YOLO V3. This is an adaptation of Online Hard Example Mining (OHEM) (Shrivastava et al., 2016) to balance the number of different categories. In our case, such Online Hard Positive Mining

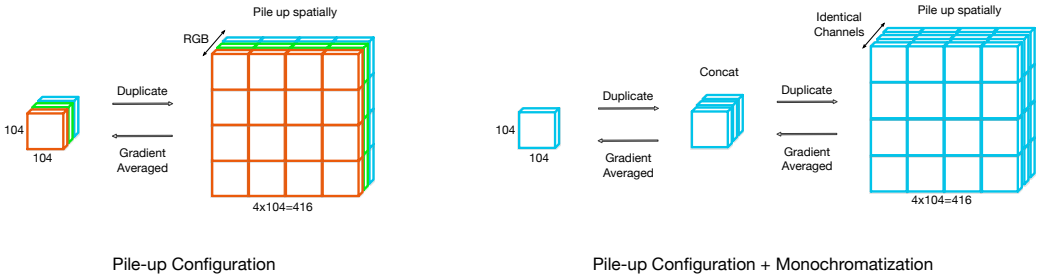

Figure 1: *Left*: pile-up configuration. *Right*: pile-up configuration + monochromatizaiton.

(OHPM) can avoid the overwhelming effects of the large number of negative anchors [1]. Optimization is done over a set of training images in the way of Brown et al. (2017). Data augmentation is used to improve the robustness of the trained mask. Specifically, we minimize the following target:

$$E_{x,A}[\frac{1}{N}\sum_{i=1}^{N} log\ p_{top-i}(Clip_{min=0}^{max=1}\{x + A(m_{full})\})]$$

where x's are images sampled from the training set, A is a randomly composed data augmentation scheme(rotation, translation, scaling), p_{top-i} is the probability value of the $i-$th highest scored anchor, $N = 200$, $Clip$ is a per-pixel clipping to ensure the attacked image is still in valid scope, and other symbols are as defined above. In practice, the norm-value constraint is done by applying an element-wise $tanh$ function to the parametrized mask and then multiply it with a designated distortion rate ϵ, which is proposed in (Carlini & Wagner, 2017). Training is continued until performance on a held-out test-set is not improved further.

As there are no other baselines, the basic setting of full-mask will in practice serve as a baseline for the two newly proposed techniques.

4.2 Technique 1: Pile-Up Configuration

The baseline setting of parameterization would result in a total number of $519K$ parameters. Although it allows for fine-grained patterns and thus stronger capacity, such exploitation of details may make it difficult to transfer to other models (Kurakin et al., 2016). Besides, an ideal adversarial mask should be translation-invariant, as it should be able to attack objects in any positions. To explicitly encode such intuitions, we propose a pile-up configuration to obtain adversarial masks.

As shown in Fig.1, we parametrize a much smaller mask, denoted as $m_{pile} \in R^{\lfloor 416r \rfloor \times \lfloor 416r \rfloor \times 3}$, where $r \in [0, 1]$ measures the size of the mask. To obtain a full-sized mask, we duplicate and pile up these small masks in a grid-aligned way. Specifically, we stack them into a $\lceil \frac{1}{r} \rceil \times \lceil \frac{1}{r} \rceil$ grid. We denote the aforementioned pile-up process as a function: $y = pile(x)$.

In the case of pile-up configuration, the adversarial mask is obtained by:

$$\Delta X = crop_{416\times416}\{A(pile(m_{pile}))\}$$

During training, the mask is applied to input images by addition followed by clipping: $x^* = Clip_{min=0}^{max=1}\{x + \Delta X\}$. Other training details are the same as Section 4.1. Gradients are averaged over the grid cells.

4.3 Technique 2: Monochromatization

The motivation for a monochromatic adversarial mask is two-fold. On the one hand, such adversaries require much fewer parameters. On the other hand, monochromatic patterns are less conspic-

[1] Statistically, at least 99.9% anchors are not activated given images from COCO dataset.

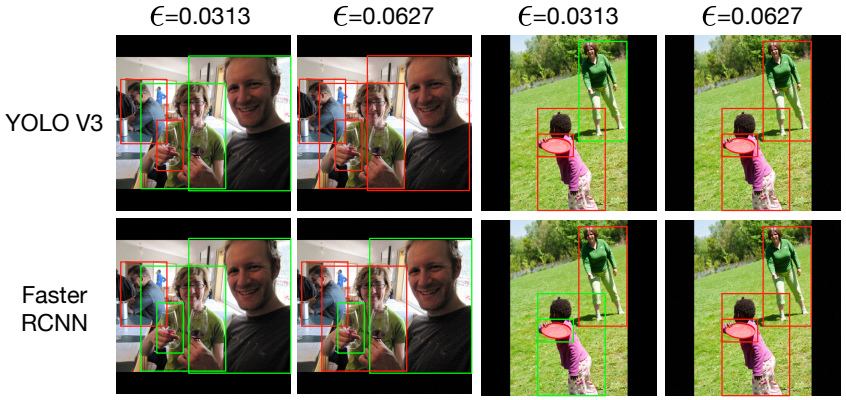

Figure 2: *Top*: detection results by YOLO V3; *Bottom*: by Faster-RCNN. The adversary is crafted solely on YOLO V3. *Green* bounding boxes represent objects that are detected both before and after attack; *Red* ones stand for those detected before attack, but concealed after attack.

uous and may better blend into objects, which may show potential in physical world application. Monochromatic mask can further be interpreted as changes of brightness.

We implement monochromatic adversarial masks by setting the values of the three color channels as the same. Training is the same as the one described above. Note that technique 1 and technique 2 can be combined together. In such case, there are only 10816 parameters, and only 2.1% of the baseline full-mask setting. Later we would show that such constraints results in simplified and even stylish adversarial patterns.

5 Experiments

We design experiments to answer the following questions: **(Q1)** How effective are our trained adversarial masks on YOLO V3 and Faster-RCNN respectively? How successful it the transfer to Faster-RCNN? **(Q2)** How do the techniques we employ help in generating effective attacks?

Empirically, we show that: (1) All three methods can achieve decent performance in the task of hiding objects from detectors. (2) The two proposed techniques improve transferability significantly. (3) Adversaries generated with the two proposed techniques demonstrate repetitive and coarse-grained patterns, which seems more robust than the finer ones. Samples for detection results are shown in Fig.2.

5.1 Setting

All experiments are based on off-the-shelf PyTorch implementations of YOLO V3[2] and Faster-RCNN[3]. Models are pretrained on COCO Detection dataset (Lin et al., 2014), with an mAP value of 33.0 for YOLO V3 and 37.0 for Faster-RCNN on test set. For the pile-up configuration, we set $r = 0.25$. We randomly sample images from the validation set of the COCO Detection dataset as training set and test set for our methods, 512 for each. The adversaries are constructed by applying mini-batch SGD with a batch size of 16, a learning rate of $1e + 2$, and momentum of $5e - 1$, until convergence.

As we aim at hiding objects from detectors, we propose to use the *average number of detected objects per image* as our main evaluation metric. Previous work on interfering with object detectors at large scale (Rosenfeld et al., 2018) uses more detailed evaluation method, taking into accounts cases like mis-classification of detected objects. We argue that our metric is suitable enough for our task, as it directly measures our main goal of making objects disappear. To make comparison easier, we use a derived metric in practice, where we compute the ratio of attacked image to clean image.

[2] https://github.com/eriklindernoren/PyTorch-YOLOv3
[3] https://github.com/jwyang/faster-rcnn.pytorch

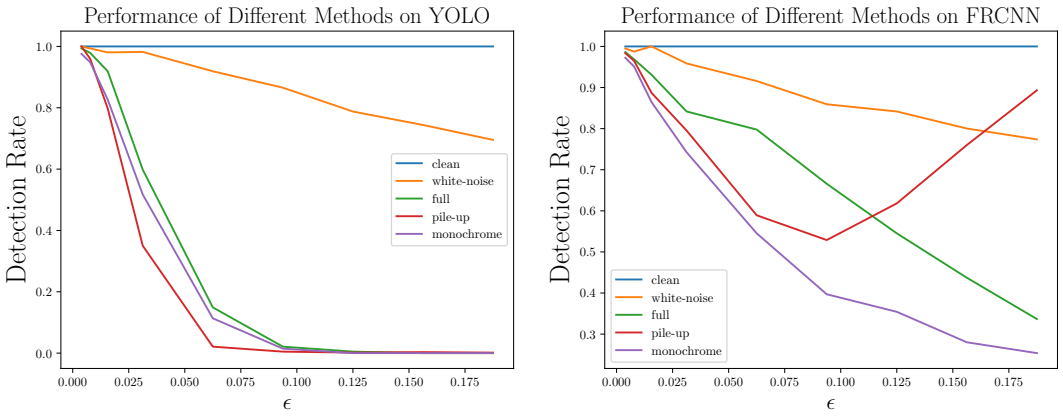

Figure 3: Performance on YOLO V3 and transfer performance on Faster-RCNN: $y-$axis:detection rate as introduced in Section 5.1; $x-$axis: distortion level ϵ. Different line represents different methods. *clean* stands for un-attacked setting, as control group. *full*, *pile-up*, *monochrome* stand for the three methods introduced in Section 4.

We term it as *Detection Rate*, measuring the proportion of objects that are still detected when being attacked. Here, the lower the measure, the better.

For a more comprehensive comparison, we train the adversaries with different values of ϵ, and plot a curve to characterize its dynamics. Samples for different levels of distortion are shown appendix.B. Overall performance evaluations are shown in Fig.3.

Further, we perform experiments under the setting of black box attack to truly evaluate the effectiveness of our methods: we transfer the adversaries trained on YOLO V3 to Faster-RCNN and compute the detection rate for each sample. Similarly, we evaluate over different levels of distortion. Results are shown in Fig.3 (*Right*).

5.2 Performance on YOLO V3: White Box Setting

Our first observation from Fig.3 (*Left*) is that, all the three methods' performance are decent under the imperceptible level of distortion ($\epsilon = \frac{8}{255}$) and are promising under the mild distortion ($\epsilon = \frac{16}{255}$). The pile-up setting can achieve nearly 100% success rate at concealing objects for $\epsilon = \frac{16}{255}$. For $\epsilon = \frac{8}{255}$, the monochronical mask can still conceal nearly half of the objects detected in clean images.

The second observation is that, while the full-mask attack has much more parameters than the other two methods, it achieves slightly lower success rate[4] at concealing objects. This may seem unreasonable at first glance as more parameters means stronger capacity. However, as we show in the next section, this may be due to the fact that the full-mask attack is much harder in training. We also notice that the colorful pile-up setting performs better that the other two methods by a significant margin. This can be explained by the fact that it has more parameters but not too many, therefore containing enough capacity and still being easy to train.

5.3 Transfer Performance on Faster-RCNN: Black Box Setting

From Fig.3 (*Right*), one basic observation is that, even with mild distortion ($\epsilon \leq \frac{16}{255}$), the best performing method can still conceal 40% of the objects. The success rate of the proposed methods perform better than the white noise baseline, demonstrating some inherent potential in transferring.

The most important observation is that, the two proposed techniques are significantly better than the full-mask baseline. At the same time, the monochromatic is better than the pile-up configuration. The comparison of the results supports our arguments that, pile-up and monochromatization are both effective technique in improving transferability, while monochromatization can further push

[4]The success rate gap is around 2%-5%, depending on the value of $epsilon$

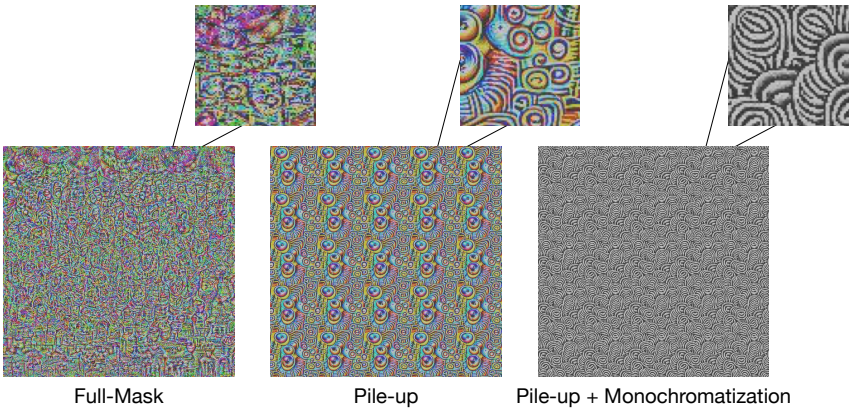

Figure 4: Visualization: magnified to $\epsilon = \frac{64}{255}$, shifted rightward 0.5, normalized back to $[0, 255]$. The upper row are zoom-in crops that illustrate the patterns learnt. **Zoom-in for better view.**

the envelope. Specifically, pile-up setting can reduce detection rate by a large margin over the full-mask baseline, ranging from 5% to 30%, depending on the ϵ value. Monochromatization can further reduce by 5%.

We also notice that the detection rate bounces back for pile-up setting when $\epsilon \geq \frac{24}{255}$. We manually check the attacked images and detection results, and found that the repetitive circle patterns in the trained pattern are sometimes mistaken as round objects, e.g. *apple* and *orange*, resulting in higher detection rate. We consider that this is one defect in the evaluation metric we use. We manually check all the attacked images again and found that this phenomenon only occurs in the colorful pile-up masks when $\epsilon \geq \frac{24}{255}$.

6 Analysis and Discussion

In this section, we visually evaluate how the two techniques play their role in improving transferability. Especially, we discuss about how pile-up helps in significantly improve transferability, as is shown in the experiments. Further, we study a strong method for comparison to provide further insight into adversaries for object detectors.

6.1 Visualization of Trained Adversaries

To observe what impact the two proposed techniques actually have, we visualize the trained adversaries in Fig.4 for full-mask, pile-up, and monochromitization respectively. We notice that, as we train with our techniques, the generated adversaries are much different from naively trained ones.

Specifically, when we use pile-up configuration, the adversaries are repetitive as expected by design, and more smooth. We zoom in to compare the pixel-level landscape, and found that while full-mask consists of tiny color lumps that are nearly as complex as white noise, the pile-up mask is much more smooth, containing less fine-grained details.

When we monochromatize the mask, the mask becomes highly repetitive and stylish. The zoom-in view shows that the patterns are even more coarse-grained than using pile-up solely. We attribute the success in improving transferability to such highly simplified patterns for adversarial attacks.

6.2 Pile-Up: Gridded Arrangement

In this section, we try to give a theoretical explanation as to why the less parametrized pile-up configuration can perform better even in the white-box setting. It would not be surprising for performing worse on Faster-RCNN, which can be attributed as overfitting, due to the large number of parameters. Here we introduce our hypothesis that *the full-mask adversary is harder to train*, and the reason may lie in the sparsity of gradients.

For crafting adversaries, most existing methods, including ours, are based on gradients propagated from the last layer, and thus the quality of the gradients are important. We found that for each image, the gradients propagated from the classifier layer only cover a small region of the adversarial mask. The large majority of the parameters in full-mask setting are not getting gradients at all[5]. Under such condition, using adaptive training methods e.g. Adam would result in erroneous estimation of amounts of recent updates; using momentum based methods would result in over-update; using vanilla SGD would only update a small fraction of the mask.

There is a related concept named Effective Receptive Field (ERF) (Luo et al., 2016), which essentially computes the gradient of a certain neuron over the input image. In fact, the objectiveness score is the activation value of that classifier neuron. Therefore, we can compute ERF for each anchor to analyze how the full-mask attack is updated. Some examples is shown in appendix.A. We notice that the gradients basically only cover the object region (not even the bounding box!). Significant variance exists for each pixel across different samples. As objects in images usually take up small fractions, the updates of such a large mask may thus be inefficient and difficult.

Different from full-mask setting, piled-up small and identical patches, in turn, can gather the gradients up, making the updates more efficient and accurate. We assume this is the main reason why pile-up configuration can beat full-mask setting by large.

6.3 Are There Any Other Stronger Baselines or Competitors?

As far as we know, there are no other methods that perform similar tasks to our target setting. Therefore, we use white noise of the same distortion level as the baseline for our three methods. We also contend that our main contribution rests in proposing and exploring two techniques to improve transferability, and that therefore, the full-mask method itself is a strong and appropriate baseline such that improvements over it provide appropriate experimental analysis.

However, we also provide experimentation below with a method adapted from adversaries for image classification approaches, to establish another benchmark for universal attack on object detectors. Following Adversarial Patch approach in Brown et al. (2017), we train a patch to conceal *neighboring* objects. One may argue that by placing the patch onto objects to conceal it can also serve as a baseline. While there is existing work on specific objects by applying stickers, they are specially designed for each object instance and still change the object a lot. One could argue that altering a scene with such large distortion does not make sense in the real world as it's too conspicuous or one could simply cover the object with a cloth. Therefore, we consider it interesting if we can design an object that can conceal *neighboring* objects *contactlessly*.

Basically we follow the original training method (Brown et al., 2017) . We found that the patch is indeed able to conceal other objects contactlessly. The training setting and more detailed results and are in appendix.C.

We notice that the success rate depends on the distance to the objects. For objects that are close enough, the success rate can be as high as 50%. An interesting observation is that the trained patch contains circular patterns that are similar to those in the pile-up and monochromatization setting. Overall, it's a well-performing baseline, but it's essentially another type of attack. We have just included it here to provide a better understanding of the task of concealing objects.

7 Limitations

This paper provides effective methods to fool object detectors. We admit that one major limitation is: object detectors themselves are not robust enough yet. Current image classification can attain a top-1 accuracy higher than 80% and top-5 accuracy, which has surpassed the human level. Therefore, the wide-range existence of adversaries are intriguing: how are these intricate models fooled? On the contrary, the performance of object detectors are still far from human level. Though the experiments presented here show that our methods can beat baselines directly adapted from methods for attacking image classification models, the mechanism behind errors in object detectors still remains unknown.

[5]Gradients in these regions are smaller than the covered area by several orders of magnitude, thus making it difficult to train.

REFERENCES

Wieland Brendel, Jonas Rauber, and Matthias Bethge. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. *ICLR*, 2018.

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

Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)*, pp. 39–57. IEEE, 2017.

Shang-Tse Chen, Cory Cornelius, Jason Martin, and Duen Horng Chau. Shapeshifter: Robust physical adversarial attack on faster r-cnn object detector. ECML-PKDD, 2018.

Gintare Karolina Dziugaite, Zoubin Ghahramani, and Daniel M Roy. A study of the effect of jpg compression on adversarial images. *arXiv preprint arXiv:1608.00853*, 2016.

Gamaleldin F Elsayed, Ian Goodfellow, and Jascha Sohl-Dickstein. Adversarial reprogramming of neural networks. *arXiv preprint arXiv:1806.11146*, 2018.

Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Florian Tramer, Atul Prakash, Tadayoshi Kohno, and Dawn Song. Physical adversarial examples for object detectors. *arXiv preprint arXiv:1807.07769*, 2018.

Justin Gilmer, Ryan P Adams, Ian Goodfellow, David Andersen, and George E Dahl. Motivating the rules of the game for adversarial example research. *arXiv preprint arXiv:1807.06732*, 2018.

Ross Girshick. Fast r-cnn. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.

Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pp. 580–587, 2014.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *ICLR2015*, 2015.

Tamir Hazan, George Papandreou, and Daniel Tarlow. *Perturbations, Optimization, and Statistics*. 2016.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *ICLR*, 2015.

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

Alexey Kurakin, Ian Goodfellow, Samy Bengio, Yinpeng Dong, Fangzhou Liao, Ming Liang, Tianyu Pang, Jun Zhu, Xiaolin Hu, Cihang Xie, et al. Adversarial attacks and defences competition. *arXiv preprint arXiv:1804.00097*, 2018.

Yuezun Li, Xian Bian, and Siwei Lyu. Attacking object detectors via imperceptible patches on background. *arXiv preprint arXiv:1809.05966*, 2018a.

Yuezun Li, Daniel Tian, Xiao Bian, Siwei Lyu, et al. Robust adversarial perturbation on deep proposal-based models. *BMCV 2018*, 2018b.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pp. 740–755. Springer, 2014.

Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. SSD: Single shot multibox detector. In *In Proceedings of European Conference on Computer Vision (ECCV)*, pp. 21–37. Springer, 2016a.

Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. Delving into transferable adversarial examples and black-box attacks. *arXiv preprint arXiv:1611.02770*, 2016b.

Jiajun Lu, Hussein Sibai, and Evan Fabry. Adversarial examples that fool detectors. *arXiv preprint arXiv:1712.02494*, 2017a.

Jiajun Lu, Hussein Sibai, and Evan Fabry. Adversarial examples that fool detectors. *arXiv preprint arXiv:1712.02494*, 2017b.

Wenjie Luo, Yujia Li, Raquel Urtasun, and Richard Zemel. Understanding the effective receptive field in deep convolutional neural networks. In *Advances in neural information processing systems*, pp. 4898–4906, 2016.

Nicolas Papernot, Patrick McDaniel, and Ian Goodfellow. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. *arXiv preprint arXiv:1605.07277*, 2016a.

Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In *Security and Privacy (SP), 2016 IEEE Symposium on*, pp. 582–597. IEEE, 2016b.

Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 779–788, 2016.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pp. 91–99, 2015.

Amir Rosenfeld, Richard Zemel, and John K. Tsotsos. The elephant in the room, 2018.

Sara Sabour, Yanshuai Cao, Fartash Faghri, and David J Fleet. Adversarial manipulation of deep representations. *arXiv preprint arXiv:1511.05122*, 2015.

Abhinav Shrivastava, Abhinav Gupta, and Ross Girshick. Training region-based object detectors with online hard example mining. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 761–769, 2016.

Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*, 2017.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. Ensemble adversarial training: Attacks and defenses. *arXiv preprint arXiv:1705.07204*, 2017.

Lei Wu, Zhanxing Zhu, Cheng Tai, et al. Understanding and enhancing the transferability of adversarial examples. *arXiv preprint arXiv:1802.09707*, 2018.

Cihang Xie, Jianyu Wang, Zhishuai Zhang, Zhou Ren, and Alan Yuille. Mitigating adversarial effects through randomization. *arXiv preprint arXiv:1711.01991*, 2017.

APPENDIX A VISUALIZING GRADIENTS WITH EFFECTIVE RECEPTIVE FIELD

We randomly pick 100 samples to analyze how gradients are being propagated. Among the samples we analyzed, we randomly pick 4 as representatives and show them in Fig.5. For each image sample, we randomly select one object that's detected, compute gradients of its objectiveness scores, and visualize the gradients propagated to the adversarial mask. We manually check all the samples and found that only the object area can obtain gradients that are not negligible.

We need more detailed specification of being negligible. For object area, i.e. pixels belonging to the object we study, the average norm of gradient has a magnitude from $1e1$ to $1e2$. For area outside object, the magnitude drops to $1e-6$. Although some current optimization methods suggest that different parameters can have different level of gradients, e.g. Adam (Kingma & Ba, 2015), this is not the case for us. In our case, parameters are not receiving gradients of different magnitudes. On the contrary, parameters are receiving gradients of unstable magnitudes, while intuitively, there should be some level of symmetry or repetition in the pattern of the mask. We argue that such unstable gradient flow may make it hard to train, and finally result in lower performance despite larger potential capacity.

APPENDIX B IMAGES ATTACKED BY DIFFERENT VALUES OF ϵ

For better illustration of how much images are distorted, we randomly select some samples ans show them in Fig.6, Fig.7, Fig.8, and Fig.9 respectively. (Zoom-in for better view.)

APPENDIX C ADVERSARIAL PATCH THAT HIDES NEIGHBORING OBJECTS

C.1 TRAINING

We parametrize the artificial object as a tensor $p \in [0,1]^{h \times w \times 3}$ in round shape, and train the patch with gradient descent methods on a training set of images. Standard data augmentations are performed to improve robustness, including scaling and rotation. However, we need to adapt the training details for better suitability. Specifically, the artificial object is randomly placed around objects, but not overlapping with objects.

Specifically, for each image, we first randomly select one object, around which we place the artificial object. Then we rescale the artificial object to a proper size:

$$r = max(min(0.25, w_{object}, h_{object}), 0.1) \times U(0.9, 1.1)$$

where $U(a,b)$ is a uniform random variable used as scaling factor for data augmentation, $w_{object}, h_{object} \in [0,1]$ are the size of the selected object as proportion to input image size, r is the ratio of the proper size to the image size. The size ratio makes sure that the size of the artificial object is neither too big nor too small. Then we randomly rotate the object, ranging from $-\frac{1}{8}\pi$ to $\frac{1}{8}\pi$. The last step is to pick a point to place the artificial object. We enlarge the bounding box of the selected object in-place by 10%, and uniformly sample one point from the periphery. We place the artificial such that the center of it are located at the sampled point. We denote the transformation aforementioned as function A. For training, we minimize the expected log probability of objectiveness of all predictions in YOLOv3 over transformations and images in the training set:

$$\min : E_{x,A}(\frac{1}{N}\sum_{i=1}^{N} p_i(x + A(p)))$$

where p_i denote the probability of being an object for prediction point i in YOLOv3, and N denotes the number of prediction points in the model[6]. In practice, most prediction points (around 99.9%) in YOLOv3 are negative, which would obfuscate the positives that are in minority and be disastrous. We alternatively optimize over the top 12 positive prediction points.

C.2 EXPERIMENT

We explore the effect of different sizes and distances to target objects. We measure the size of the object as proportion of its diameter to the side length of the input image. The distance is computed as the absolute distance between the center of the trained adversarial object, and the center of the bounding box of the detected object, divided by the length of the bounding box's diagonal. For better vision, we take the logarithm of distance.

[6] $N = 10647$ in the case of YOLOv3

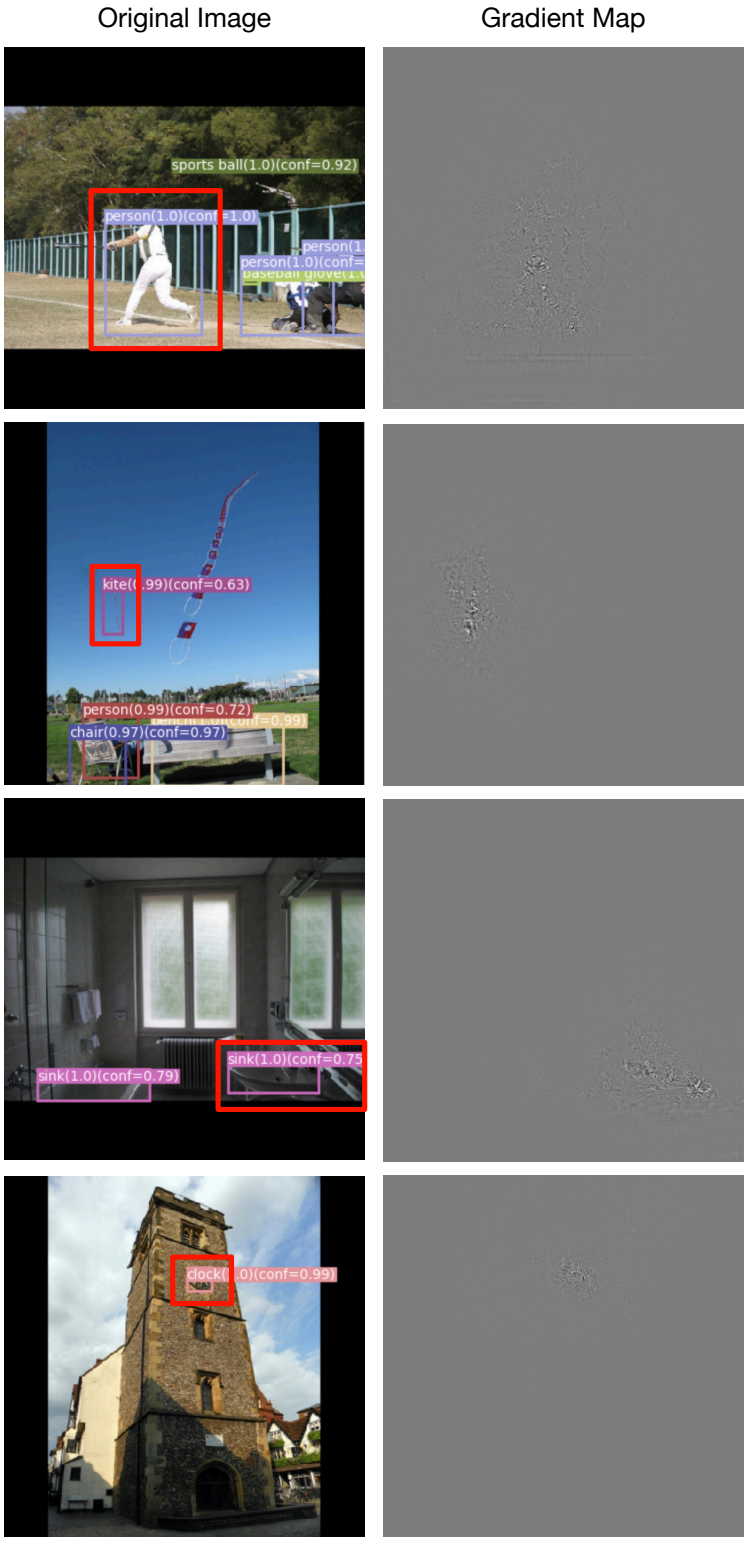

Figure 5: *Left*: detection results for original image; *Red BBox*: the target object from which we back-prop gradients; *Right*: the gradient flowing from the object bounded by red bounding box. Pixel values are normalized using the min-max rule.

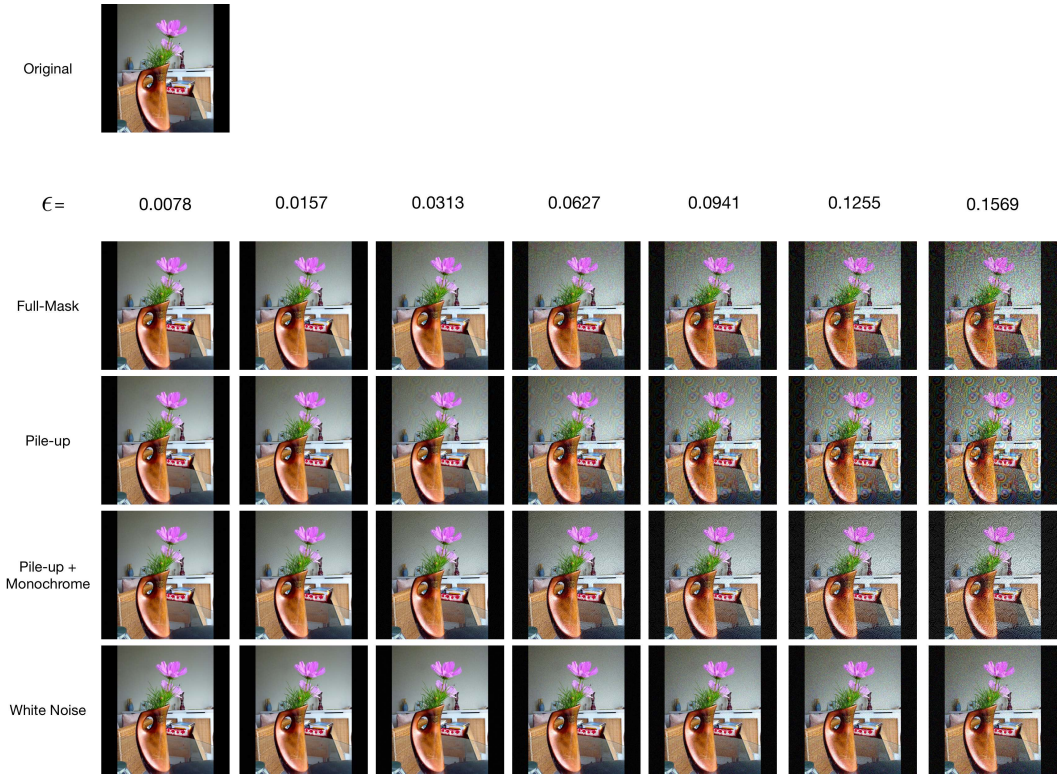

Figure 6: Sample 1: images attacked by different methods with different level of distortion.

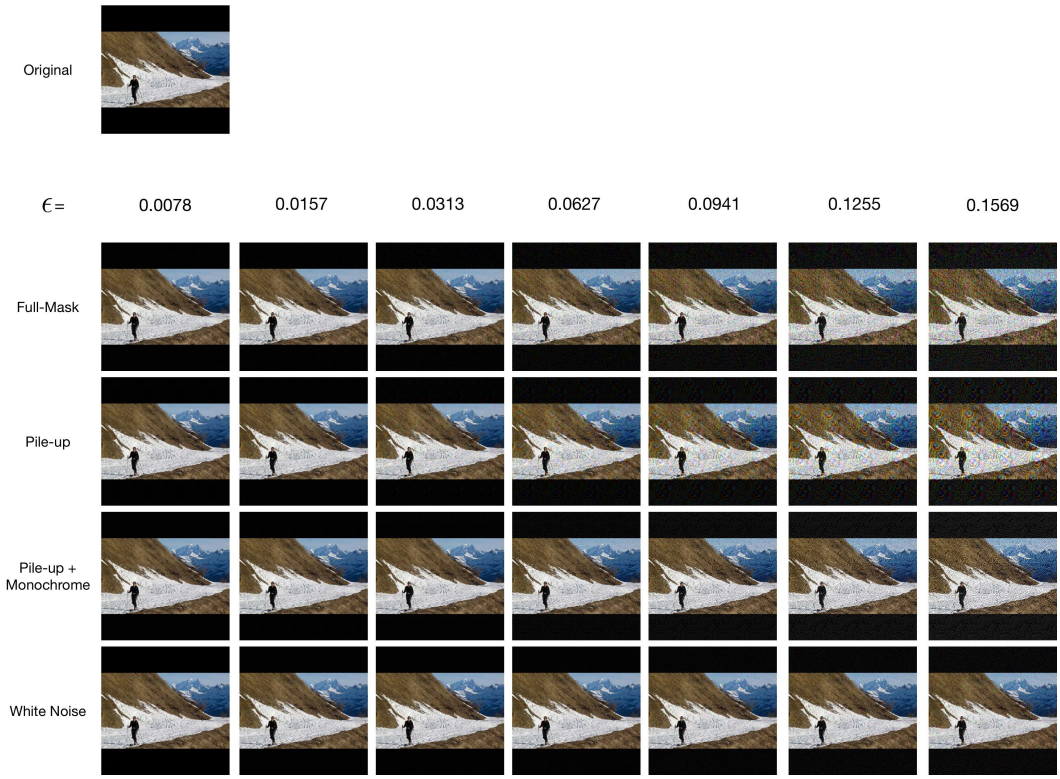

Figure 7: Sample 2: images attacked by different methods with different level of distortion.

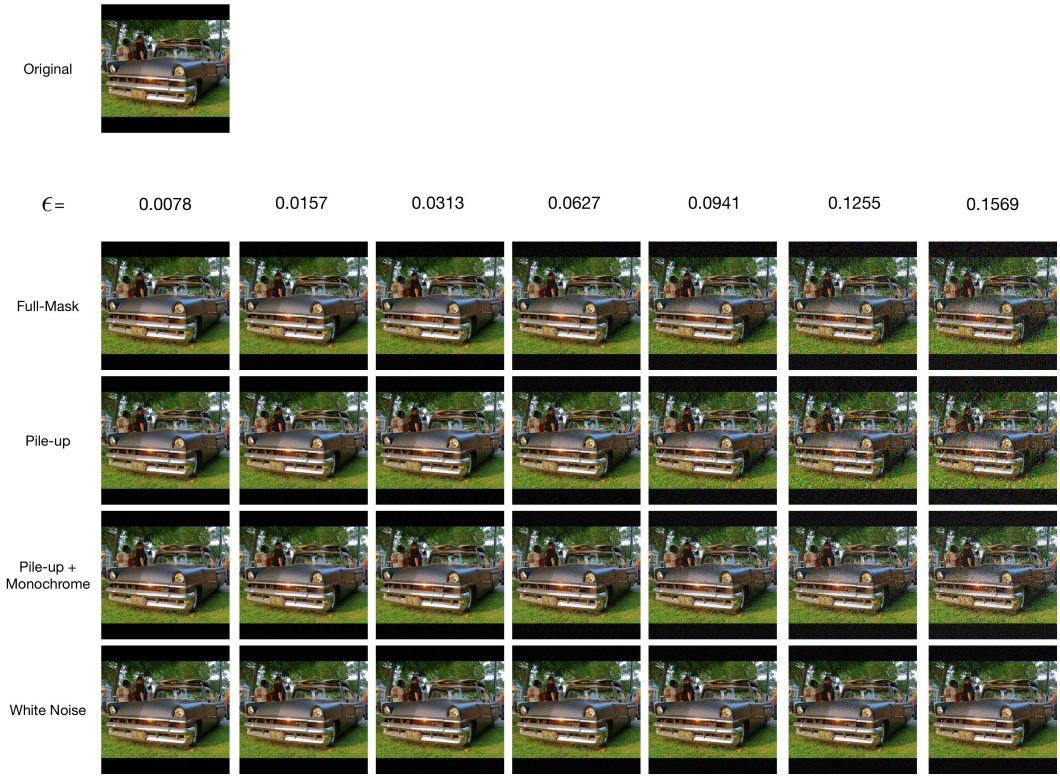

Figure 8: Sample 3: images attacked by different methods with different level of distortion.

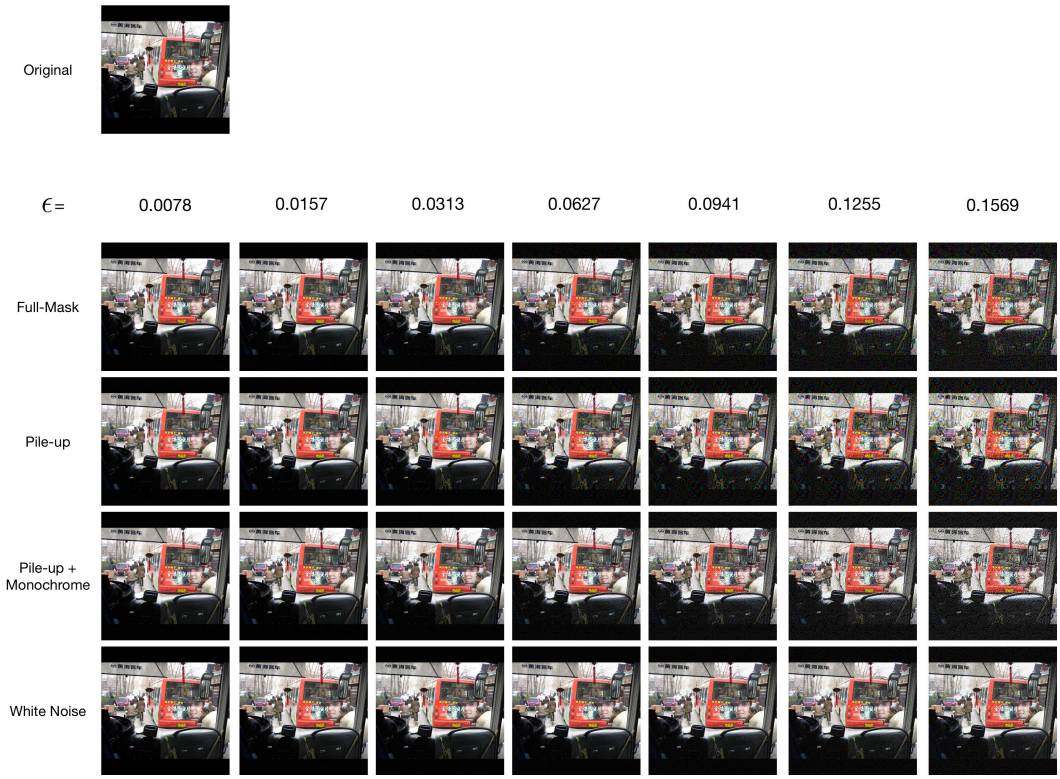

Figure 9: Sample 4: images attacked by different methods with different level of distortion.

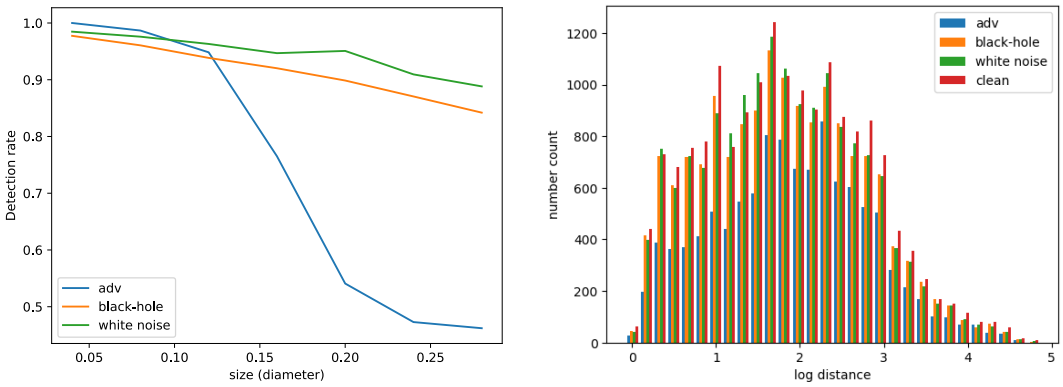

Figure 10: Overall performance of adversarial patch of different sizes and distances from target objects. We use two baselines: black-hole that replaces all pixels in a sub-region with 0; white-noise that replaces all pixels in a sub-region with random Gaussian noise.

We evaluate the performance with the using held-out test set. Quantitative results are shown in Fig.10. One important conclusion is that, although baseline methods similarly change the image significantly by replacing a sub-region completely, they can barely affect the detection results. We also include some samples from the test set in Fig.11. The trained object can indeed conceal other objects in a contactless way.

When we look can the effect of size, we notice that the trained object, of a reasonable size (0.28), can conceal more than 50% of the existence of other objects when simply placed around them. We did not consider patches of size larger than 0.30, as we consider it impractical under the real world setting.

Distance also plays an important role. As we randomly place the object for training, where the actual size ranges from 0.10 to 0.25, we notice that for neighboring objects, more than half the objects can be concealed. As the distance to the target objects grows, success rate drops.

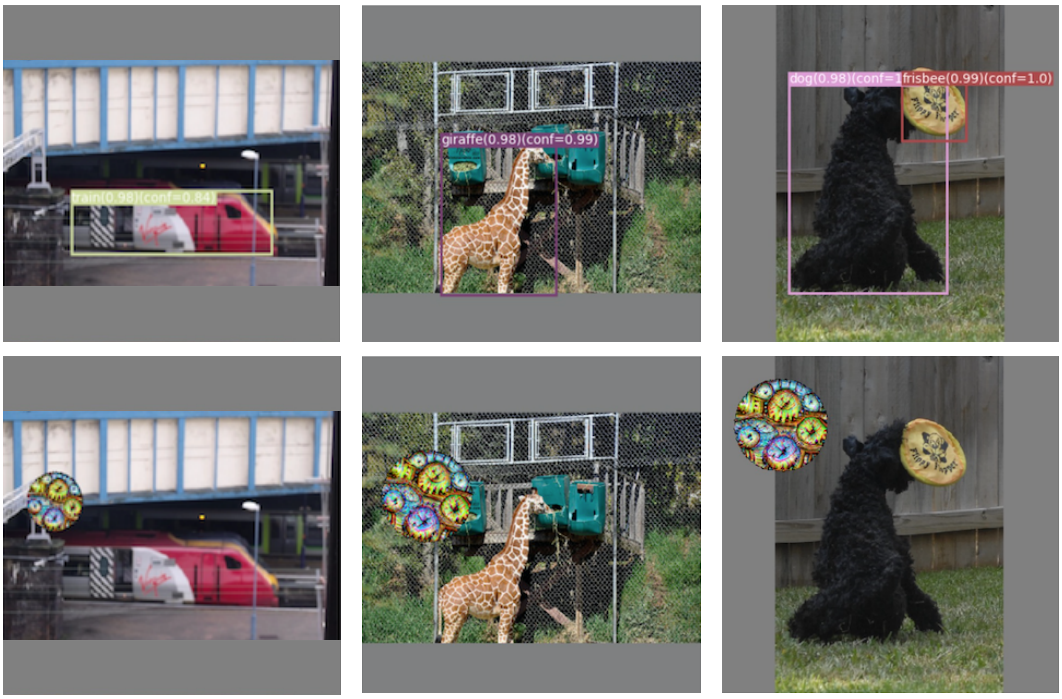

Figure 11: Demonstration of successful attack. *Top*: original image and its detection result; *Bottom*: attacked image and its detection result.