

Patch of Invisibility: Naturalistic Black-Box Adversarial Attacks on Object Detectors

Raz Lapid

Department of Computer Science

*Ben-Gurion University, Beer-Sheva 84105, Israel,
and DeepKeep, Tel-Aviv, Israel*

razla@post.bgu.ac.il

Moshe Sipper

Department of Computer Science

Ben-Gurion University, Beer-Sheva 84105, Israel

sipper@bgu.ac.il

Abstract

Adversarial attacks on deep-learning models have been receiving increased attention in recent years. Work in this area has mostly focused on gradient-based techniques, so-called “white-box” attacks, wherein the attacker has access to the targeted model’s internal parameters; such an assumption is usually unrealistic in the real world. Some attacks additionally use the entire pixel space to fool a given model, which is neither practical nor physical (i.e., real-world). On the contrary, we propose herein a gradient-free method that uses the learned image manifold of a pretrained generative adversarial network (GAN) to generate naturalistic physical adversarial patches for object detectors. We show that our proposed method works both digitally and physically.

1 Introduction

Deep Neural Networks (DNNs) are increasingly deployed in safety-critical applications, many involving some form of identifying humans. The risk of facing deceptive images—so called *adversarial* instances—grows with the use of DNN models. In order to trick the DNN into categorizing the input image differently than a human would, an adversarial sample is employed. As first demonstrated by Szegedy et al. (2013), neural networks are susceptible to such adversity.

The methods used to generate adversarial instances based on minimal input perturbations were enhanced by subsequent works (Carlini & Wagner, 2017; Moosavi-Dezfooli et al., 2016; Lapid et al., 2022). Instead of modifying existing images, Thys et al. (2019) generated unconstrained adversarial instances using generative adversarial networks (GANs). While many methods concentrate on digital, white-box attacks, where the attacker has access to the model, it is crucial to investigate and comprehend potential realistic *physical* attacks in more depth, i.e., attacks that occur in the physical world.

Adversarial attacks are a type of cybersecurity threat that aims to deceive deep learning (DL) systems by injecting carefully crafted inputs that are designed to fool them. These attacks exploit vulnerabilities in DL models and take advantage of their tendency to make mistakes when processing data. Adversarial attacks can be used for manipulation in a wide range of applications, including image recognition, natural language processing, autonomous vehicles, and medical diagnosis (Lapid et al., 2022; Tamam et al., 2022).

The implications of adversarial attacks can be far-reaching, as they can compromise the security and accuracy of systems that rely on DL. For instance, an adversarial attack on a vehicle-mounted, image-recognition system could cause it to misidentify a stop sign as a speed-limit sign (Eykholt et al., 2018), potentially

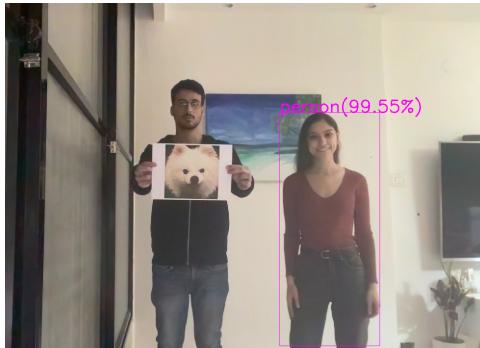


Figure 1: An adversarial patch evolved by our novel gradient-free algorithm, which conceals people from an object detector.

causing the vehicle to crash. As DL becomes increasingly ubiquitous, the need to mitigate adversarial attacks becomes more pressing. Therefore, research into adversarial attacks and defenses is a rapidly growing area, with researchers working on developing robust and secure models that are less susceptible to such attacks.

In this work we focus on fooling surveillance cameras (both indoor and outdoor), because of their ubiquity and susceptibility to attack, by creating adversarial patches (Figure 1). The next section presents previous work. Our method is described in Section 3, followed by experimental results in Section 4. We discuss our findings in Section 5, ending with concluding remarks in Section 6.

2 Previous work

This section offers some background and discusses relevant literature on adversarial attacks. We begin with digital adversarial attacks in classification tasks, followed by digital attacks on object detectors, ending with physical attacks.

In general, attacks can be divided into three categories: white box, black box, and gray box.

- **White-box** threat models assume that the attacker has complete knowledge of the system being attacked. This includes knowledge of the system’s architecture, implementation details, and even access to the source code. In a white-box threat model, the attacker can inspect the system’s internals and use this knowledge to launch targeted attacks.
- **Black-box** threat models assume that the attacker has no prior knowledge or access to the system being attacked—no knowledge of the system’s architecture or implementation details. This means that the attacker is only able to observe the system’s behavior from the outside, without any ability to inspect the internals of the system.
- **Gray-box** threat models are a mix of both black-box and white-box models, where the attacker has some knowledge of the system being attacked—but not complete knowledge.

Herein we focus on black-box threat models because we only assume access to the models’ output.

Convolutional Neural Networks (CNNs) have been of particular focus in the domain of adversarial attacks, due to their popularity and success in imaging tasks (Kurakin et al., 2018; Feng et al., 2021; Zolfi et al., 2021; Lapid et al., 2022; Sharif et al., 2019).

Digital adversarial attacks on classification models are a type of attack on learning models that includes adding a minimal perturbation to the input data, with the goal of causing the model to predict an incorrect class. There has been a plethora of studies over the past several years on creating and enhancing adversarial attacks on classification models.

The Fast Gradient Sign Method (FGSM) (Goodfellow et al., 2014), first presented in 2015, is one of the earliest and most well-known adversarial attacks. The FGSM is a straightforward and effective white-box

technique for generating adversarial attacks. It works by computing the gradient of the loss function with respect to the input data and perturbing the input data according to the gradient’s sign. FGSM has received much attention since its introduction, and has been proven quite successful.

Projected Gradient Descent (PGD) (Madry et al., 2017), a white-box attack introduced in 2017, executes several small steps along the gradient’s direction and projects the perturbed image back onto a sphere centered around the original image. PGD has been shown to outperform FGSM when applied to models with robust defenses against adversarial attacks.

The Carlini and Wagner (C&W) white-box attack (Carlini & Wagner, 2017), which was presented in 2016 as an optimization-based method for generating adversarial instances, is another well-known adversarial attack. In C&W, an optimization problem is solved to identify the smallest perturbation (in terms of $\|\cdot\|$) that leads to a misclassification. A wide variety of learning models—some with advanced defenses against adversarial attacks—have been demonstrated to be susceptible to C&W.

Overall, a significant amount of work has been done on generating and enhancing adversarial attacks on classification models (mostly in a white-box setting), and over the past several years many efficient methods have been developed. In order to protect the integrity and dependability of models deployed in an increasing number of applications it is crucial to continually seek out their weaknesses and create strong defenses against adversarial attacks.

Digital adversarial attacks on object-detection models, which are used in a variety of systems, from surveillance devices to autonomous cars, have become an increasing concern. These attacks involve intentionally malicious inputs to fool the model into generating bad predictions, and they can have severe effects, including inaccurate object identification or failing to detect.

One of the most commonly used object-detection models is You Only Look Once (YOLO) (Redmon et al., 2016), which is based on a single convolutional neural network (CNN) that simultaneously predicts the class and location of objects in an image. Several studies have shown that YOLO is vulnerable to adversarial attacks (Liu et al., 2018; Im Choi & Tian, 2022; Thys et al., 2019; Hu et al., 2021). For example, targeted perturbation attacks can be used to modify an input image in a way that causes YOLO to misidentify or fail to detect certain objects. An untargeted attack, on the other hand, seeks to create adversarial examples that cause general disruption to the model’s performance. Adversarial attacks can have a significant influence on object-detection models. Most prior research regarding such attacks on object-detection models focus on white-box, gradient-based attacks, which is, by and large, a non-real-world scenario.

Physical adversarial attacks pose an even greater threat than digital ones. Most digital adversarial attacks need access to the actual model in order to fool it. Further, many attacks use global perturbation over pixel space—e.g., changing the sky’s pixels—thus rendering it less realistic. Physical adversarial attacks can be engendered in a variety of ways, including covering an object with paint or some other material (Wang et al., 2021; Brown et al., 2017), applying stickers (Eykholt et al., 2018; Komkov & Petiushko, 2021) or camouflage (Duan et al., 2020; Wang et al., 2021), or changing the object’s shape or texture (Hu et al., 2022; Yang et al., 2020). Such changes are intended to alter how the object appears to the model while maintaining it recognizable to humans. Interestingly, the attack often looks identical to the naked eye—but *not* to the model (as we shall also show below).

Physical adversarial attacks can have serious effects, e.g., when an autonomous car’s failed detection results in a crash. Physical attacks can also be intentionally used to get past security systems or enter restricted areas without authorization. Again, most prior research on physical attacks has been done using gradients, in a white-box setting.

Contrarily, we create **adversarial patches** in a **black-box manner**, i.e., without the use of gradients, leveraging the **learned image manifold of GANs** (Goodfellow et al., 2020).

Generative Adversarial Networks (GANs): The quality of generative models used for image generation has improved significantly with the advent of GANs (Goodfellow et al., 2020) and Diffusion Models (Ho et al., 2020). Herein, we focus on GANs, due to their relatively small latent-space dimension and their fast image generation. GANs utilize a learnable loss function, where a separate discriminator network is used to

distinguish between real and fake images, and the image generator network is trained to synthesize images that are indistinguishable from real ones. Despite their visually appealing results, GANs often face issues such as instability, vanishing gradients, and mode collapse. Researchers have suggested many different GANs to address these issues (Karras et al., 2019; Arjovsky et al., 2017; Mao et al., 2017). Herein we chose to use BigGAN2.

3 Method

Our objective is to generate *physically plausible* adversarial patches, which are performant and appear *realistic—without the use of gradients*. An adversarial patch is a specific type of attack, where an image is modified by adding a small, local pattern that engenders missclassification. The goal of such an attack is to intentionally mislead a model into making an incorrect prediction or decision.

By “physically plausible” we mean patches that not only work digitally, but also in the physical world, e.g., when printed—and used. The space of possible adversarial patches is huge, and with the aim of reducing it to afford a successful search process, we chose to use pretrained GAN generators.

Given a pretrained generator, we seek an input *latent vector*, corresponding to a generated image that leads the object detector to err. We leverage the latent space’s (relatively) small dimension, approximating the gradients using an Evolution Strategy algorithm (Wierstra et al., 2014), repeatedly updating the input latent vector by querying the target object detector until an appropriate adversarial patch is discovered.

Figure 2 depicts a general view of our approach. We search for an input latent vector that, given a pretrained generator, corresponds to a generated image that causes the object detector to mistakenly detect that image as a person.

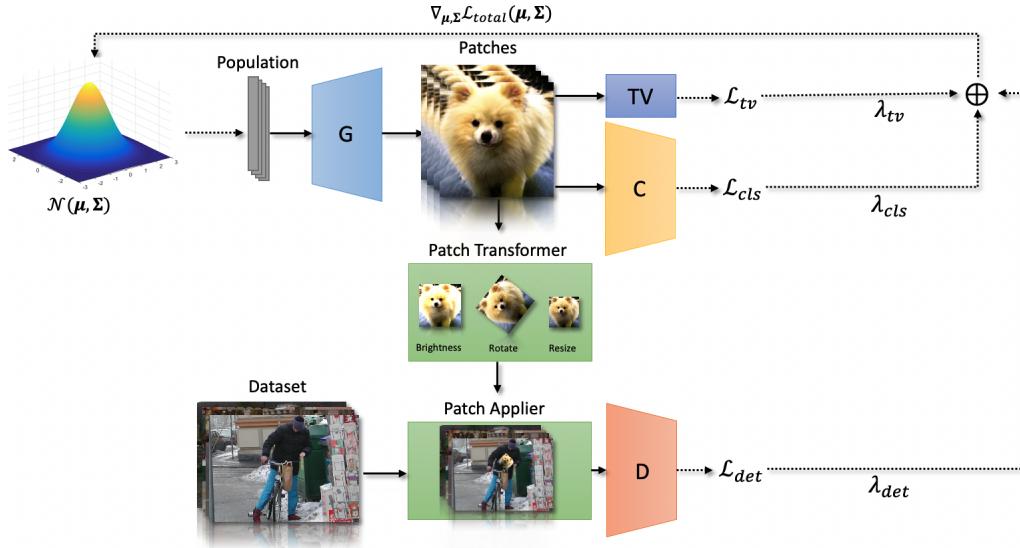


Figure 2: Naturalistic Black-Box Adversarial Attack: Overview of framework. The system creates patches for object detectors by using the learned image manifold of a pretrained GAN (G) on real-world images (as is often the case, we use the GAN’s generator, but do not need the discriminator). We use a pretrained classifier (C) to force the optimizer to find a patch that resembles a specific class, the TV component in order to make the images as smooth as possible, and the detector (D) for the actual detection loss. Efficient sampling of the GAN images via an iterative evolution strategy ultimately generates the final patch.

Algorithm 1 shows the pseudocode of the evolution strategy. In each iteration t , we sample n Gaussian noise values, $\epsilon_1, \epsilon_2, \dots, \epsilon_n$ (there is one latent vector per noise value). Then, we scale them using σ , add them to the latent vector z_t , and feed to the model F . The latent vector z_{t+1} is then updated using the weighted sum

of the loss values F_i for each $\epsilon_i, i \in \{1, 2, \dots, n\}$. This step can be done using any arbitrary optimizer—herein we used Adam (Kingma & Ba, 2014).

Algorithm 1: Evolution Strategy

Input: learning rate α , noise standard deviation σ , initial latent vector \mathbf{z}_0 , number of iterations T , population size n

for $t = 1, \dots, T$ **do**

- | Sample $\epsilon_1, \epsilon_2, \dots, \epsilon_n \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
- | Compute fitness $F_i = F(\mathbf{z}_t + \sigma\epsilon_i)$ for $i = 1, \dots, n$
- | Set $\mathbf{z}_{t+1} \leftarrow \alpha \frac{1}{n\sigma} \sum_{i=1}^n F_i \epsilon_i$

end

3.1 Generating adversarial patches

Previous research optimized adversarial patches in *pixel space*. We, on the other hand, focus on a GAN generator’s *latent space*. Our resultant adversarial patch will be closer to the manifold of natural pictures and hence appear more realistic (because GANs learn a latent space that roughly approximates the manifold of natural images). We employ a generator G that has been previously trained on ImageNet using a GAN framework, and we search the space of learned natural image manifold using an evolution strategy.

The evolution strategy algorithm begins by using an isotropic standard multivariate distribution, $\mathcal{N}(\boldsymbol{\mu} = \mathbf{0}, \boldsymbol{\Sigma} = \mathbf{I}_d)$, which parameterizes the evolved population with $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$. We begin with an initial latent vector \mathbf{z}_0 . We then randomly sample n noises $\epsilon_1, \epsilon_2, \dots, \epsilon_n \in \mathbb{R}^d$ using the standard multivariate distribution, scale them by σ and add them to the initial latent vector \mathbf{z}_0 resulting with $\mathbf{Z} \in \mathbb{R}^{d \times n}$ — which then are fed to the generator to create the population of patches $P = G(\mathbf{Z}) \in \mathbb{R}^{n \times 3 \times H \times W}$ — where n stands for the population size, 3 is the number of channels (RGB), H is the patch’s height, and W is the patch’s width.

Then, using gradient approximation through an evolution strategy, we repeatedly search the latent vector z that best achieves our objective function, which is:

$$\mathcal{L}_{total} = \mathcal{L}_{det} + \lambda_{tv}\mathcal{L}_{tv} + \lambda_{cls}\mathcal{L}_{cls}, \quad (1)$$

where:

- \mathcal{L}_{det} : detection loss of object detector of the specific class.
- \mathcal{L}_{tv} : total variation loss, to promote smoothness of generated patch.
- \mathcal{L}_{cls} : classification loss, to promote more-realistic patch generation.
- λ_{tv} and λ_{cls} are regularization weights.

We expound upon the these terms in below.

3.2 Adversarial gradient estimation

In order to generate patches that may deceive the target object detector, the generator uses adversarial gradient estimations as its primary navigation tool. We initially add the patch onto a given image. In order to compute an adversarial loss for the detection of bounding boxes (BBs), we feed the image to the object detector.

Adversarial detection loss. Detection can be arbitrarily produced by object detectors like YOLO (Redmon et al., 2016). We are interested in minimizing two terms for the patch detection i : its objectness probability D_{obj}^i , which specifies the model’s confidence regarding whether there is an object or not, and the class probability D_{cls}^i , which specifies the model’s confidence of a specific class. In this paper we focused on

generating patches that conceal humans. Thus, we want to minimize both the objectness D_{obj}^i and class probabilities D_{cls}^i for our generated patch with respect to the person class, i.e., minimizing the term:

$$\mathcal{L}_{det} = \frac{1}{n} \sum_{i=1}^n \sum_{j=1}^{det} D_{obj}^j(x_i) D_{cls}^j(x_i), \quad (2)$$

where n is the population size and det is the number of human detections by the model. By minimizing \mathcal{L}_{det} , we achieve a patch that minimizes the objectness and class probabilities of the target class.

Physical transformations. We have no influence over the adversarial patch’s viewpoint, position, or size, with respect to the images. In order to enhance the robustness of our adversarial patch, we overlay it onto a human and generate a variety of settings with distinct configurations. Furthermore, we subject our created adversarial patch P to various transformations, including rotation, brightness change, and resizing, to mimic the different visual appearances it may adopt in real-world situations. The transformations are applied by the Patch Transformer (Figure 2).

Smoothness. To promote smoothness in the generated image we apply the term \mathcal{L}_{tv} , which represents the total variation loss. The calculation of \mathcal{L}_{tv} from a patch P is done as follows:

$$\mathcal{L}_{tv}(P) = \sum_{i,j} \sqrt{(P_{i+1,j} - P_{i,j})^2 + (P_{i,j+1} - P_{i,j})^2}, \quad (3)$$

where subscripts i and j refer to the pixel coordinates of patch P . A constant value of $\lambda_{tv} = 0.1$ was employed in all experiments presented in this paper.

Class loss. Empirically, we noticed that the patches generated by the generator are very abstract and apparently far from the latent image manifold learned by the generator. Thus, we added another loss term in order to make the patch resemble a specific class, by adding a pretrained classifier, trained on ImageNet (Deng et al., 2009); this adds regularization that makes the image similar to a specific class.

Realism. To maintain realism we enforce a constraint on the norm of the latent vectors Z , which should not exceed a threshold τ . By adjusting this threshold we can balance realism versus attack performance. We use $\|\cdot\|_\infty$ to constrain z , resulting in:

$$z^t = \pi(z^{t-1} - \alpha \tilde{\nabla}_z \mathcal{L}_{total}), \quad (4)$$

where:

$$\pi(z) = \{z_i | z_i \leftarrow \min(\max(z_i, -\tau), \tau), \forall z_i \in z\}, \quad (5)$$

t is the timestep (epoch), α is the step size, and $\tilde{\nabla}_z \mathcal{L}_{total}$ is the gradient approximation of the total loss with respect to the latent vector z . We used $\tau = 20$ in all experiments.

4 Experimental results

We begin by delineating the implementation details of our experiments, followed by the qualitative and quantitative experiments themselves, focusing on the effectiveness of the proposed adversarial patch in two settings: 1) digital environments, specifically, the INRIA person dataset (Dalal & Triggs, 2005), where we target images with one person, to facilitate optimization; 2) physical environments, as demonstrated through recorded videos in various real-life scenes. We also perform several ablation studies, aimed at refining various hyperparameters, ultimately leading to the creation of a physically generated adversarial patch that appears more natural, while maintaining comparable attack performance. Additionally, subjective evaluations are conducted to assess the naturalness of the generated patches.

4.1 Ablation studies

The experiments were performed using the Adam (Kingma & Ba, 2014) optimizer, with a learning rate of 0.02 and $\beta_1 = 0.5, \beta_2 = 0.999$. If the changes in losses remain below $1e - 4$ for at least 50 epochs, we reduce the learning rate, following (Hu et al., 2021). Our generator consists of BigGAN2 (Brock et al., 2018), which has been pretrained on the ImageNet dataset, with a latent vector of size 120, and 128×128 output resolution. As the BigGAN generator is class-conditional, the generated patch can be guaranteed to belong to a specific class. We used two different object detectors—Tiny-YOLOv3 and Tiny-YOLOv4 (Redmon et al., 2016), trained on the COCO dataset (Lin et al., 2014)—with an input resolution of 416×416 . We chose these faster, memory-efficient models so as to be able to conduct far more experiments. For each of the detected bounding boxes, we placed an adversarial patch of size 25% of the bounding box. For both models the average precision rate on the original INRIA dataset using a single person was 100%. Further, the mAP of all experiments are shown in Table 4.1.

Table 1: Different evaluations of patches in terms of mAP(%) for the INRIA dataset using Eq 2. mAP is evaluated using best evolved patch per experiment.

Model	λ_{cls}	population	mAP (%)
Tiny-YOLOv3	0.1	110	23.2
		90	24.6
		70	31.5
		50	26.1
	0.2	110	30.0
		90	33.4
		70	28.9
		50	29.7
Tiny-YOLOv4	0.1	110	20.7
		90	16.9
		70	16.3
		50	31.4
	0.2	110	18.79
		90	32.7
		70	33.3
		50	31.0

4.2 Visual results

Figures 3 through 6 show evolved patches. Figures 7 and 8 show digital attacks. Figures 9 through 13 show physical attacks. Note that the figures show diverse situations and lighting conditions.



Figure 3: Patches evolved by our algorithm, on Tiny-YOLOv3, with $\lambda_{cls} = 0.1$, and (left to right) population sizes of 50, 70, 90, and 110, respectively.



Figure 4: Patches evolved by our algorithm, on Tiny-YOLOv3, with $\lambda_{cls} = 0.2$, and (left to right) population sizes of 50, 70, 90, and 110, respectively.



Figure 5: Patches evolved by our algorithm, on Tiny-YOLOv4, with $\lambda_{cls} = 0.1$, and (left to right) population sizes of 50, 70, 90, and 110, respectively.



Figure 6: Patches evolved by our algorithm, on Tiny-YOLOv4, with $\lambda_{cls} = 0.2$, and (left to right) population sizes of 50, 70, 90, and 110, respectively.



Figure 7: Digital examples of our proposed attack on Tiny-YOLOv3. The left image shows an example wherein the patch failed to “conceal” the person. In all other images the attack succeeded: no person was detected.

5 Discussion

5.1 Impact of loss function

We explored two different loss objectives. The first one focuses only on the objectness loss, i.e.,:

$$\mathcal{L}_{det} = \frac{1}{N} \sum_{i=1}^N \sum_{j=1}^{\#det} D_{obj}^j, \quad (6)$$

For Tiny-YOLOv3 (Figure 14), Equation 2 (which takes into account class probabilities as well) does better than Equation 6. For Tiny-YOLOv4 (Figure 15), the two are similar. This result is contrary to the result



Figure 8: Digital examples of our proposed attack on Tiny-YOLOv4. The left image shows an example wherein the patch failed to “conceal” the person. In all other images the attack succeeded: no person was detected.



Figure 9: Real-life images—taken by coauthor—of Tiny-YOLOv3 detection using no patch. Left: low-lighting condition, Right: high-lighting condition.



Figure 10: Real-life example of Tiny-YOLOv4 detection using no patch. Left: low-lighting condition, Right: high-lighting condition.

presented in Thys et al. (2019)—though we haven’t used any gradients. We think this might be due to the specific hyperparameters tested in our experiments—further analysis in the future will ascertain this point.

5.2 Impact of population size

We explored 4 different population sizes: 50, 70, 90, and 110. In order to estimate the gradient well enough, we need 2 queries per each coordinate of the latent vector (of size 120). Thus, we need $2d$ queries to the target model, resulting in 240 queries for one gradient estimation. We actually used far less queries, evidencing a strength of our approach. Further, we surmised that a larger population would yield better results, but, empirically, it seems that a population of size 70 yielded lower loss. We think this is because when we



Figure 11: Real-life example of evolved adversarial patch on Tiny-YOLOv3. Successful patch “conceals” the person. Low-lighting condition.



Figure 12: Real-life example of evolved adversarial patch on Tiny-YOLOv3. Successful patch “conceals” the person. High-lighting condition.



Figure 13: Real-life example of evolved adversarial patch on Tiny-YOLOv4. Successful patch “conceals” the person. Low-lighting condition.

estimate a gradient using a relatively small population we increase the algorithm’s exploration, resulting in a lower loss value.

6 Concluding remarks

We presented a novel algorithm that generates naturalistic, physical adversarial patches for object detectors. The patches can be printed and used in the real world. Our approach involved optimizing a latent vector to minimize probabilities associated with the appearance of a person in the detector’s output, without using any internal information of the model—a realistic scenario that does not rely on the use of gradients.

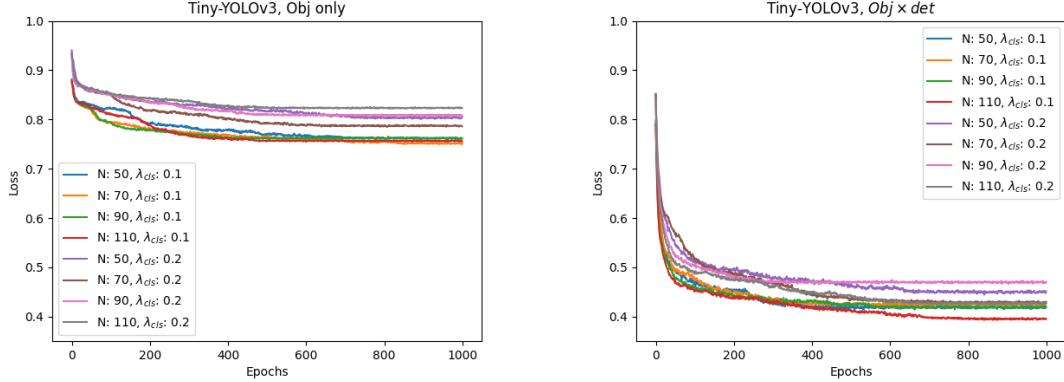


Figure 14: Total loss as function of number of epochs, on Tiny-YOLOv3. Left: Optimization process using Equation 6. Right: Optimization process using Equation 2

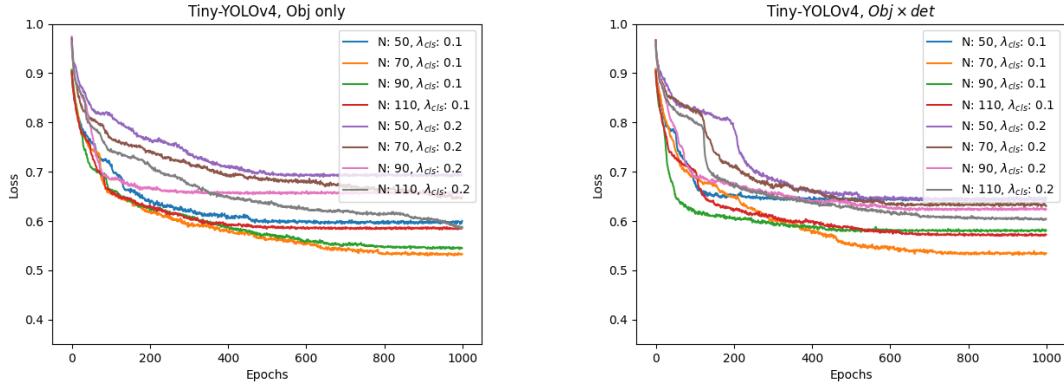


Figure 15: Total loss as function of number of epochs, on Tiny-YOLOv4. Left: Optimization process using Equation 6. Right: Optimization process using Equation 2

We compared different deep models and concluded that it is possible to generate patches that fool object detectors. The real-world tests of the printed patches demonstrated their efficacy in “concealing” persons, evidencing a basic threat to security systems. In future work we wish to expand our algorithm to other types of attacks, such as false-positive attacks (which cause the appearance of a person where none exists), evasion attacks, and more.

Acknowledgement

This research was partially supported by the Israeli Innovation Authority through the Trust.AI consortium.

References

- Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International Conference on Machine Learning*, pp. 214–223. PMLR, 2017.
- Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 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.
- Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pp. 886–893. Ieee, 2005.
- Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255. Ieee, 2009.
- Ranjie Duan, Xingjun Ma, Yisen Wang, James Bailey, A Kai Qin, and Yun Yang. Adversarial camouflage: Hiding physical-world attacks with natural styles. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1000–1008, 2020.
- Kevin Eykholt, Ivan Evtimov, Earlene Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, and Dawn Song. Robust physical-world attacks on deep learning visual classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1625–1634, 2018.
- Weiwei Feng, Baoyuan Wu, Tianzhu Zhang, Yong Zhang, and Yongdong Zhang. Meta-attack: Class-agnostic and model-agnostic physical adversarial attack. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7787–7796, 2021.
- Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.
- Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.
- Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.
- Yu-Chih-Tuan Hu, Bo-Han Kung, Daniel Stanley Tan, Jun-Cheng Chen, Kai-Lung Hua, and Wen-Huang Cheng. Naturalistic physical adversarial patch for object detectors. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7848–7857, 2021.
- Zhanhao Hu, Siyuan Huang, Xiaopei Zhu, Fuchun Sun, Bo Zhang, and Xiaolin Hu. Adversarial texture for fooling person detectors in the physical world. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13307–13316, 2022.
- Jung Im Choi and Qing Tian. Adversarial attack and defense of yolo detectors in autonomous driving scenarios. In *2022 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1011–1017. IEEE, 2022.
- Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4401–4410, 2019.
- Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- Stepan Komkov and Aleksandr Petushko. Advhat: Real-world adversarial attack on arcface face id system. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pp. 819–826. IEEE, 2021.
- Alexey Kurakin, Ian J Goodfellow, and Samy Bengio. Adversarial examples in the physical world. In *Artificial Intelligence Safety and Security*, pp. 99–112. Chapman and Hall/CRC, 2018.
- Raz Lapid, Zvika Haramaty, and Moshe Sipper. An evolutionary, gradient-free, query-efficient, black-box algorithm for generating adversarial instances in deep convolutional neural networks. *Algorithms*, 15(11):407, 2022.

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 *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part V 13*, pp. 740–755. Springer, 2014.

Xin Liu, Huanrui Yang, Ziwei Liu, Linghao Song, Hai Li, and Yiran Chen. Dpatch: An adversarial patch attack on object detectors. *arXiv preprint arXiv:1806.02299*, 2018.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2794–2802, 2017.

Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2574–2582, 2016.

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*, pp. 779–788, 2016.

Mahmood Sharif, Sruti Bhagavatula, Lujo Bauer, and Michael K Reiter. A general framework for adversarial examples with objectives. *ACM Transactions on Privacy and Security (TOPS)*, 22(3):1–30, 2019.

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.

Snir Vitrack Tamam, Raz Lapid, and Moshe Sipper. Foiling explanations in deep neural networks. *arXiv preprint arXiv:2211.14860*, 2022.

Simen Thys, Wiebe Van Ranst, and Toon Goedemé. Fooling automated surveillance cameras: Adversarial patches to attack person detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 0–0, 2019.

Jiakai Wang, Aishan Liu, Zixin Yin, Shunchang Liu, Shiyu Tang, and Xianglong Liu. Dual attention suppression attack: Generate adversarial camouflage in physical world. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8565–8574, 2021.

Daan Wierstra, Tom Schaul, Tobias Glasmachers, Yi Sun, Jan Peters, and Jürgen Schmidhuber. Natural evolution strategies. *Journal of Machine Learning Research*, 15(1):949–980, 2014.

Chenglin Yang, Adam Kortylewski, Cihang Xie, Yinzhi Cao, and Alan Yuille. Patchattack: A black-box texture-based attack with reinforcement learning. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVI*, pp. 681–698. Springer, 2020.

Alon Zolfi, Shai Avidan, Yuval Elovici, and Asaf Shabtai. Adversarial mask: Real-world adversarial attack against face recognition models. *arXiv preprint arXiv:2111.10759*, 2021.