Simplifying Adversarial Attacks Against Object Detectors: a Fundamental Approach

Thomas Cilloni[1], **Charles Fleming** [2] **Charles Walter** [1],

[1] Computer and Information Science Department, University of Mississippi, USA
[2] Cisco Research, USA
tcilloni@go.olemiss.edu, chflemin@cisco.com, cwwalter@olemiss.edu

Abstract

Current state-of-the-art object detection models are characterized by very large output manifolds due to the number of possible locations and sizes of objects in an image. This leads to their outputs being sparse. We propose a novel adversarial algorithm that leverages this output sparsity to propose more efficient adversarial attacks by limiting attacks to sensitive regions. We identify sensitive regions of an image as those that stimulate the greatest network activations and optimize adversarial perturbations to those regions only. Our *Focused Adversarial Attacks* (FA) algorithm is consistently more effective than other adversarial methods under the same perturbation constraints. We evaluate FA on the COCO 2017 and Pascal VOC 2007 detection datasets against a variety of SOTA object detector models and show that FA outperforms all popular and SOTA adversarial attacks against object detectors.

Introduction

Recent research in machine learning led to the discovery of *adversarial examples*, which are data points carefully tailored to induce errors in ML models. Many adversarial attacks likely take place every day without being detected or leaving traces. The introduction of Generative Adversarial Networks (GANs) (Goodfellow et al. 2014) gave rise to Deepfakes, synthetic videos, pictures, and voice recordings, which pose big threats such as fake news and crafted material. Detectors have been developed to distinguish real from synthetic images (Nguyen et al. 2019), but these have also been shown to be vulnerable to adversarial examples (Carlini and Farid 2020). Adversarial examples can also be used to promote user privacy, as shown in (Shan et al. 2020) and (Cilloni et al. 2022) for unauthorized facial recognition, and (Xiao et al. 2020) to prevent data inference from a model's parameters.

Most adversarial examples focus on fooling Object Detection models. Object Detection is the task of localizing and classifying objects in images. Most object detection models nowadays use deep CNNs and learn feature mappings that can be post-processed to produce detections. *Region Proposal Networks* (RPNs) first find regions in the image where objects may reside and then try to predict a label for each region, possibly rejecting any proposals. This approach allows the model to be flexible in predicting large, small, or numerous objects but requires significant computations. Examples of this architecture are RetinaNet (Lin et al. 2020), and Faster R-CNN (Ren et al. 2015). *Single Shot Detectors* (SSDs), on the other hand, lay out sparse and dense grids of anchor points over images. Then, for each point in each grid, a number of probability distributions over classes are computed. This feature map is processed with *non-max suppression* to discard overlapping and non-confident predictions. While faster, this method is less accurate than RPNs. An example of this architecture is SSD (Liu et al. 2016). SSDs and RPNs both output very large, sparsely activated feature maps that induce unnecessary computation in finding adversarial examples. By using only a subset of the outputs, we show how adversarial examples can be more effectively generated.

Fooling an Object Detection model with an adversarial example is defined as an adversarial attack. Adversarial attacks are algorithms that generate adversarial examples, crafted data points that are classified by neural networks differently than how they would be classified by a human. Adversarial attacks, given a real data point and an ML model, try to solve two problems at the same time: maximize the error of the model, and minimize the change introduced in the data point. The taxonomy of adversarial attacks identifies as *targeted* attacks those whose goal is to fool a model into predicting a certain target; otherwise, an attack is *untargeted*. *White-* and *black-box* methods refer to whether, in the threat model, attackers have internal access to the victim network or not. The most recent adversarial methods against object detectors are either gradient-based, such as DAG (Xie et al. 2017), or GAN-based, such as UEA (Wei et al. 2019) and MI (Liang, Wei, and Cao 2021). While gradient methods are faster, GAN-based ones are consistently more effective, and MI (Liang, Wei, and Cao 2021) is the current state-of-the-art.

Understanding adversarial attacks is the first step toward building defense mechanisms. In this paper, we propose a novel gradient-based adversarial algorithm that takes advantage of the semantics of the feature maps that models learn. This method is simple but fundamental as it exploits the very nature of object detectors as sparse networks. Isolating and backtracking only highly activated output neurons, we are able to determine which pixels in an image contribute

Copyright © 2023, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

most significantly to detections; we refer to their ensemble as *sensitive regions*. Constraining adversarial perturbations to sensitive regions only, cloaks are not only less perceptible but also more effective and equally fast or faster than comparable methods. We evaluate our strategy using four popular object detectors: RetinaNet, SSD300, Faster R-CNN, and DETR; and two datasets: COCO 2017 and Pascal VOC 2007. Results show that our algorithm is computationally more efficient and, at the same time, more effective than all other gradient or GAN-based attacks, including MI (Liang, Wei, and Cao 2021).

Related Work

One of the early works in adversarial attacks against object detectors is the Dense Adversary Generator (DAG) by Xie et al. (Xie et al. 2017). It proposes to apply gradient-based attacks previously used against classification models in the context of object detectors and semantic segmentation. Results show that the accuracy (mean Average Precision) of object detection can be reduced by DAG from $\approx 70\%$ to below 20%. Adversarial examples generated in white-box settings are also shown to transfer to the same detection models trained on different data. However, transferability to different models is low.

(Gu et al. 2021) propose *Gradient Shielding*, a gradient-based attack for image classifiers that targets sensitive regions of images. Such selection is made at the image level, either manually (Interactive Gradient Shielding, IGS) or automatically (Automatic Gradient Shielding, AGS). IGS can be visualized as a square-shaped adversarial perturbation applied to a square region of the image smaller than the image itself; in other words, IGS ignores image borders. AGS automates this process by finding sensitive regions beforehand and zeroing gradients for insensitive regions. Our work follows the same concept as gradient shielding, that is, applying perturbations only to sensitive regions, however, with some key differences: the loss function used in training the victim model is irrelevant to us; the method to find sensitive regions uses learned feature maps at the output level, instead of the loss function's gradient at the input level, which further optimizes the gradient calculation and leads to more strongly activated gradients; we apply our method to object detectors, which have much larger feature maps and complex gradients, instead of image classifiers.

(Wei et al. 2019) propose an adversarial algorithm that uses GANs and a multi-scale attention feature loss to produce adversarial examples that can fool object detectors and classifiers alike: Unified and Efficient Adversary (UEA). This loss function is an ensemble of a high-level class loss and a low-level feature loss and is used in training the GAN's generator. Their results show a drop in the accuracy of Faster R-CNN and SSD300 on the Pascal VOC 2007 dataset from 70% to 5% and from 68% to 20%, respectively.

A drawback of UEA is the time it takes to train the generator. Given a train set of a few thousand samples, the generator requires several days of training time. More Imperceptible attacks (MI) (Liang, Wei, and Cao 2021) overcome this problem by improving upon the generator used in UEA (Wei et al. 2019) with an early stopping condition and a noise reduction step. Instead of iterating the generator a fixed number of times, an object detector is used to determine when an example has become adversarial, and the generator is then stopped. This ensures that generated samples are adversarial and also minimally perturbed. The efficacy of attacks is at least as good as that of (Wei et al. 2019), but at a fraction of the time ($8.4s$ to $1.8s$).

The idea of focusing adversarial attacks toward specific regions in images has been studied in the past. (Xu et al. 2019) propose structured attacks (StrAttack), an adversarial algorithm to exploit semantic information in images to produce more targeted cloaks. The algorithm uses a small sliding window that scans an image in search of key identifying structures for objects. One of the motivations behind our work (i.e., exploiting image semantics to produce stronger attacks) is the same; however, the methodology and application are largely different. While StrAttack uses a brute-force approach to identify regions of interest, our algorithm uses a tailored optimization function that identifies sensitive regions and produces adversarial cloaks in a single operation. Additionally, while StrAttack is focused on classification problems, our interest is in object detectors, which present vastly sparser feature maps than classifiers.

Adversarial Algorithm

This section introduces the algorithm to generate focused adversarial examples. We first present some characteristics of object detectors that motivate the design of focused adversarial attacks. Then we introduce the algorithm itself, and finally provide some considerations on its implementation.

Inspecting Object Detectors

Object detection neural networks process images and produce a feature output that contains spacial and semantic information about any objects in the image. Unlike neural network classifiers, which output as many features as there are classes in the task they solve (typically with a softmax layer), object detectors' outputs are much larger.

As images may contain multiple objects, and the objects may have different scales and be of different types, the feature map of object detectors must contain all such information. Determining the class of an object is done in the same manner a classifier makes predictions: generating a probability distribution over classes, which is a one-dimensional vector. Objects have drastically different shapes, and this is partially handled by having either multiple candidate bounding boxes or dynamic bounding boxes. Minor improvements to the fit of bounding boxes are also typically controlled with a scale and offset adjustment. The spatial information of detections is represented by a grid of candidate locations in an image, and to support small and large-sized objects, multiple grids are employed. Two of the most popular object detectors, SSD(Liu et al. 2016) and RetinaNet(Lin et al. 2020), have around 8K and 100K candidate boxes, respectively. Each candidate box has a related probability distribution over target classes, which we can assume to be COCO's 80 classes. The total size of the output of these networks, therefore, becomes enormous: 640K in SSD and 8M in RetinaNet.

The first consideration of traditional adversarial machine learning methods is to be made with regard to the intuition behind the effect of perturbations on predictions. In classification models, the gradient of the loss function, computed with respect to the input and for a single output feature, intuitively points perturbations in the direction to optimally disrupt that feature or class. Similarly, considering a distribution over target classes, perturbations can either confuse a model by spreading the distribution or fool it by targeting a feature other than the target. In object detectors, however, it is unclear what gradient-based attacks actually do. Given the high dimensionality of the detection features, targeted attacks can incur in racing conditions where gradients with respect to input pixels cancel each other out.

A second consideration is the characteristics of object detectors' feature mappings. The large outputs of detectors are very sparsely activated. Significant detections usually occur only in extremely small subsets of the feature maps, and they become even fewer when they are filtered with a confidence threshold. We explored this behavior by feeding an example image to RetinaNet and studying its output mapping. 99.98% of the 8M outputs had activations at or below 0.05, showing how sparse the features are. Fig.1a shows the distribution of the 0.02% most activated outputs: only a very small subset of this already minuscule set would actually contribute to candidate box selection algorithms at later stages (which typically filter out activations below 0.5).

These insights are some of the motivations that led to our design of Focused adversarial attacks. We show that adversarial examples can be generated more efficiently and effectively by filtering out non-contributing parts of the feature maps of object detectors.

Problem Statement

Focused adversarial attacks are a form of gradient-based attacks and therefore require white-box access to the victim model. We assume that the attacker has access to the structure and the parameters of the model to attack and can replicate the data pre-processing pipeline used in training. The loss function used in training does not need to be known. Finally, it is assumed that the model outputs the feature map of the activation of classes at various locations in the image.

In object detection tasks, images $\mathbf{x} \in \mathcal{X}$ are sampled from an unknown distribution $\mathcal{X} \subset [0,1]^{H \times W \times 3}$ and have corresponding one-hot encoded labels $\mathbf{y} \in \mathcal{Y} \subset [0,1]^{A \times C}$ where $\mathbf{y}_i j = 1$ indicates that there is an instance of the object j at the anchor location i (typically if $j = 0$ there is no object or it is just background). An object detection model f with parameters θ maps images onto a feature space $\mathcal{Y}$ as $\hat{\mathbf{y}} = f(\mathbf{x};\theta) \in \mathcal{Y}$, such that $\arg\max_j \hat{\mathbf{y}}_{ij}$ is the predicted class for location i, and its confidence is $\hat{\mathbf{y}}_{ij}$. The position i in a vector $\hat{\mathbf{y}}$ tells the location of detection, and its coordinates in an image can be calculated taking into consideration the number of grids in the image, the size of each grid, and the number of anchors. Focused adversarial attacks, however, being model-agnostic, are not concerned with the location of detections, and therefore, the details of their implementation can be safely ignored.

We look at adversarial attacks as optimization problems. Images are perturbed with a minimal mask $\delta \in [0,1]^{H \times W \times 3}$ with the goal of removing all confident predictions from a model's feature map. If all $\hat{\mathbf{y}}_{ij}$ in a model's output are low enough (the exact value depends on model implementation and choices in interpreting results), then the model will not detect any object in an image. We indicate this upper bound on ignored detections as c and define the optimization problem to find perturbations δ as follows:

$$\begin{aligned} \text{find} \quad & \arg\min_{\delta} \|\mathbf{x} + \delta\|_\infty \\ \text{s.t.} \quad & \hat{\mathbf{y}}_{ij} \leq c, \forall\, \hat{\mathbf{y}}_{ij} \in \hat{\mathbf{y}} = f(\mathbf{x} + \delta; \theta) \end{aligned} \tag{1}$$

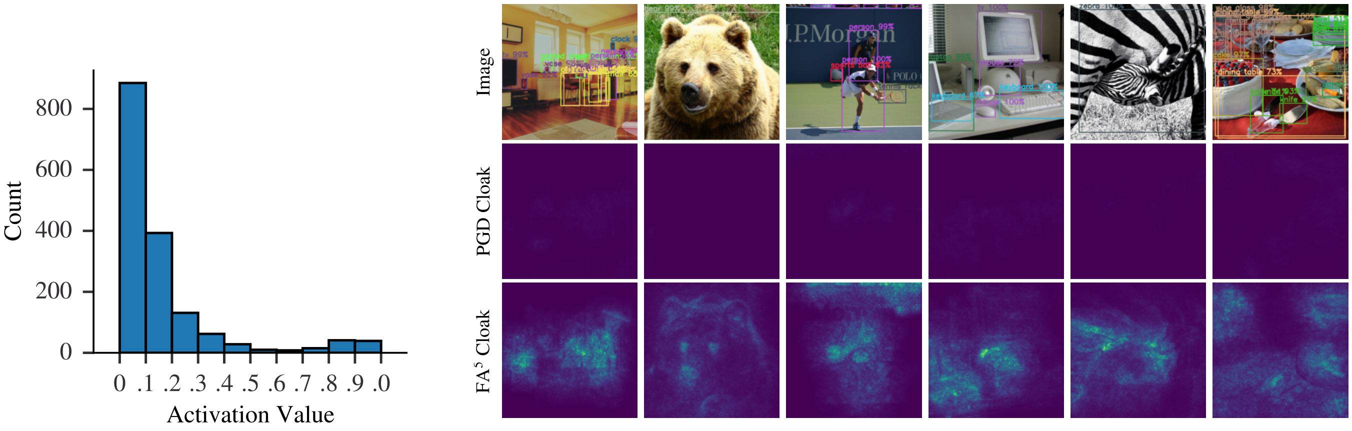
Solution Algorithm

The novelty of our work is the method to find the perturbation δ. We introduce a *focus threshold* t that determines whether a feature of a model's output map should be considered or not to compute the perturbations. The *Focused Activation* FA is defined as the L1 norm of the vector of regions of the feature map that exceed the threshold, and we minimize it with respect to input images to determine the perturbations δ. L and C are the numbers of locations in a feature map and the classes in the detection task, and ϵ is the modulator of the intensity of perturbations.

$$\begin{aligned} FA(\hat{\mathbf{y}}, t) &= \sum_{i}^{A} \sum_{j}^{C} \max(0, \hat{\mathbf{y}}_{ij} - t) \\ \delta &= \epsilon * sign\left(\nabla_{\mathbf{x}} FA(f(\mathbf{x};\theta), t)\right) \end{aligned} \tag{2}$$

The motive behind filtering features is that the majority of detection feature maps are noise. By focusing only on significantly activated regions of the maps, we are able to generate perturbations that specifically target sensitive regions in an image. This drastically reduces the intensity of perturbations in background regions of images and cloaks sensitive regions more effectively. Figure 1b shows how in practice, focused attacks are more targeted to sensitive image parts for object detection.

Two issues with our loss function are that it is not differentiable at $\hat{y}_{ij} - t = 0$, and the derivative is 0 when $\hat{y}_{ij} - t < 0$, meaning there is no information available for how to perform our update. We propose two viable solutions. The first, FA_P, takes advantage of the parallel processing capabilities of modern processors to speed up computations, and the second, FA_I, uses indexing to reduce the volume of operations to execute, which is particularly effective on sparsely activated feature maps. For the remainder of this paper, where the implementation details are not specified, FA_I is assumed to be the one used, because we found it to be faster than FA_P in our experiments. It is advisable to always compare the performance of the two, as either one could be faster depending on the size of feature maps, the



(a) Histogram of the distribution of activations in the 99.98% most activated subset of feature mappings. These are generated by feeding sample 885 from COCO-2017-val to RetinaNet.

(b) Heatmap of the sensitive regions of images found by the focused adversarial examples algorithm (bottom row), compared to the regions affected by standard PGD (middle row). Heatmaps are found with five iterations for each algorithm, and all parameters are the same.

Figure 1: Investigative study of the activations found in Faster R-CNN' feature map.

distribution of activations, and the hardware used.

$$focus(v,t) := \begin{cases} 1 & \text{if } v > t \\ 0 & \text{otherwise} \end{cases}$$
$$FA_P(\hat{\mathbf{y}},t) = \sum_i^A \sum_j^C \hat{\mathbf{y}}_{ij} * focus(\hat{\mathbf{y}}_{ij},t) \tag{3}$$

$$sub(\hat{\mathbf{y}},t) := \{\hat{\mathbf{y}}_{ij} : \hat{\mathbf{y}}_{ij} > t, \forall\, i,j : 0 \le i \le L, 0 \le j \le C\}$$
$$FA_I(\hat{\mathbf{y}},t) = \|sub(\hat{\mathbf{y}},t)\|_1 \tag{4}$$

As with all gradient-based attacks, our focused adversarial attacks can be executed in either a one-shot or iterative fashion. Following is the iterative version of the one-shot attack proposed earlier to compute adversarial perturbations. For the remainder of this paper, this is the exact formulation that we use in all experiments, with $FA = FA_I$ and varying number of iteration steps and per-step magnitude ϵ. Following is its definition, and the complete process is shown in Algorithm 1.

$$\delta^{t+1} = \epsilon * sign\left(\nabla_{\mathbf{x}+\delta}\, FA(f(\mathbf{x}+\delta;\theta),t)\right) \tag{5}$$

Hyperparameter Analysis

In this section, we propose an exploration of three hyperparameters of our methods, which are the total intensity of perturbations, the granularity of perturbation steps, and the focusing threshold used in the Focused Activation function. Finally, we study the performance gain of our focused adversarial attacks in terms of processing speed.[2]

The experiments carried out in this section use a single subset of 500 samples taken uniformly at random from the COCO 2017 validation split and Faster R-CNN as the object detection model. Efficacy is measured with the mean Average Precision (mAP) metric, and because our intent is to hinder object detection models, lower precision values indicate that a method is more effective.

Algorithm 1: Focused Adversarial Attack

Input: an image $\mathbf{x}$; a model f with parameters θ whose outputs lie on $\mathcal{Y} \subset [0,1]^{A \times C}$; the perturbation radius $\varepsilon \in [0,1]$; the number of iterative steps S; the focusing threshold t.

Output: an adversarial image $\mathbf{x}'$

```
function FOCUSED ATTACK($\mathbf{x}, f, \theta, \varepsilon, S, t$)
    $\epsilon \leftarrow \varepsilon / S$                                  ▷ Apply $\epsilon$ for $S$ steps
    $\mathbf{x}' \leftarrow \mathbf{x}$
    for $i \leftarrow 1 \dots S$ do
        $\delta \leftarrow sign\left(\nabla_{\mathbf{x}'} FA_P(f(\mathbf{x}';\theta),t)\right)$     ▷ Implemented in Eq. 3
        $\mathbf{x}' \leftarrow \mathbf{x}' + \epsilon * \delta$     ▷ Ensures $\|\delta\|_\infty \le \epsilon$, thus $\|\mathbf{x}' - \mathbf{x}\|_\infty \le \varepsilon^1$
    end for
    return $\mathbf{x}'$
end function
```

Varying $\mathcal{E}$-ball

The first series of experiments investigates how the intensity of perturbations affects precision. Adversarial examples are defined as $\mathbf{x}' = \mathbf{x} + \delta$, and δ is calculated by optimizing the FA function. Considering using Projected Gradient Descent, the total perturbation δ found after T steps is found iteratively as the summation of T ϵ-magnitude, image-shaped vectors. In order to produce perturbations comparable across image samples and detection models used, we set an upper bound on the L_∞ norm of the total perturbation vector $\mathcal{E}$,

[2] We constrain $0 \le \mathbf{x}' \le 255$ for all dimensions so that these remain valid color values.

such that $\|\delta\|_\infty \leq \mathcal{E}$. Given T iterations of the algorithm, this constraint can be guaranteed by setting $\epsilon = \mathcal{E}/T$. The $\mathcal{E}$-ball of perturbations is, therefore, the space in which all allowable perturbations reside.

Figure 2a shows how enlarging the $\mathcal{E}$-ball of perturbations makes attacks more effective. This behavior is expected as larger perturbations are likely to be more effective, though more noticeable. Results are presented for the Fast Gradient Sign Method (FGSM) and Projected Gradient Descent (PGD), used as reference scores, while our attacks are FA^S, with S being the number of iterations of the algorithm.

Perturbation Granularity

The second series of experiments on the hyperparameters of focused adversarial attacks is concerned with the granularity of perturbations. Given a constant $\mathcal{E}$-ball of adversarial changes, a different number of iterative steps T can lead to different results. The granularity of perturbations is, therefore, the small epsilon that determines the magnitude of each iteration's perturbation, which is inversely proportional to the number of steps as $\epsilon = \mathcal{E}/T$.

The precision of Faster R-CNN on adversarial examples generated with varying steps is shown in Figure 2b. Given a fairly large perturbation space of 0.1, the effectiveness of attacks increases from 1 to 3 steps and then shows drastically diminishing returns for finer-grained perturbations. On a similar note, when the perturbation space is more constrained, more than three steps also cause diminishing returns, and more than five steps even show a decrease in performance. It is, therefore, safe to assume that many but small perturbations do not perform as well as few, larger ones.

Focusing Factor

The last series of experiments investigates the focusing threshold used in focused adversarial attacks. This hyperparameter dictates which activations in a model's feature map should be considered and which should be excluded. Higher values make the FA function consider very few highly activated features, while lower values expand the search space. As the activations we considered as expressed as probability distributions, we use focus thresholding values in the range $[0, 1)$. All experiments are run with a fixed value of $\mathcal{E} = 0.02$, and PGD attacks run for five iterations.

Figure 2c shows the decrease in precision associated with different focus thresholds. The effectiveness of standard FGSM and PGD attacks is constant because the threshold does not affect them. PGD shows an increase in efficacy for focus values around 0.5 and worse performance as the threshold is raised or lowered. On the other hand, FGSM performs best with high focus threshold values (0.8 to 0.9) and actually exceeds PGD at a fraction of the computational cost when the threshold is above 0.6.

Performance Analysis

The execution time of our method compared to standard FGSM and PGD is presented in Table 1 for different focus thresholding values. We run experiments on a single RTX 3070 GPU and consider only computational time, therefore excluding I/O and main memory to GPU data movement. Results show that our focused adversarial attacks are either equivalent or marginally faster than the other methods. PGD performance is computed for five iterations of the algorithm.

Table 1: Performance of adversarial attack methods. FGSM and PGD show a single value because they are unaffected by the focus threshold. The superscripts for the FA algorithm indicate the number of iterations.

	Focus Threshold				
Attack	0.1	0.3	0.5	0.7	0.9
FGSM	-	-	139 ms	-	-
PGD	-	-	705 ms	-	-
FA^1	139 ms	138 ms	137 ms	137 ms	**136 ms**
FA^5	691 ms	683 ms	677 ms	675 ms	**672 ms**

Evaluation

We evaluate the performance gain of focused adversarial attacks over FGSM and PGD in terms of efficacy and processing speed, with a brief insight on final perturbation magnitudes. Evaluations are carried out on two publicly available datasets: the full COCO 2017 validation split (Lin et al. 2014), and Pascal VOC 2007 test (Everingham et al. 2007). These are publicly available datasets and have no personally identifiable information or offensive images.

Models

Focused adversarial attacks can be carried out on any learned machine-learning model. Though we suggest their use on models that output probabilities or probability distributions, with some tuning, any model is compatible. The experiments included in this section are run on four SOTA object detection models: SSD300 (Liu et al. 2016), Faster R-CNN (Ren et al. 2015), RetinaNet (Lin et al. 2020), and DETR (Carion et al. 2020). All models are trained on the COCO 2017 dataset (Lin et al. 2014). We use Viet Nguyen's implementation of **SSD300**[3], to which we manually apply a softmax layer to the outputs to produce probability distributions over the classes available. For **RetinaNet** we use Yann Henon's implementation[4] as is. **Faster R-CNN** is taken from torchvision[5] and we tune its parameters to reduce the amount of detections automatically filtered out. Finally, **DETR** is taken from HuggingFace[6] with all default parameters.

Effectiveness

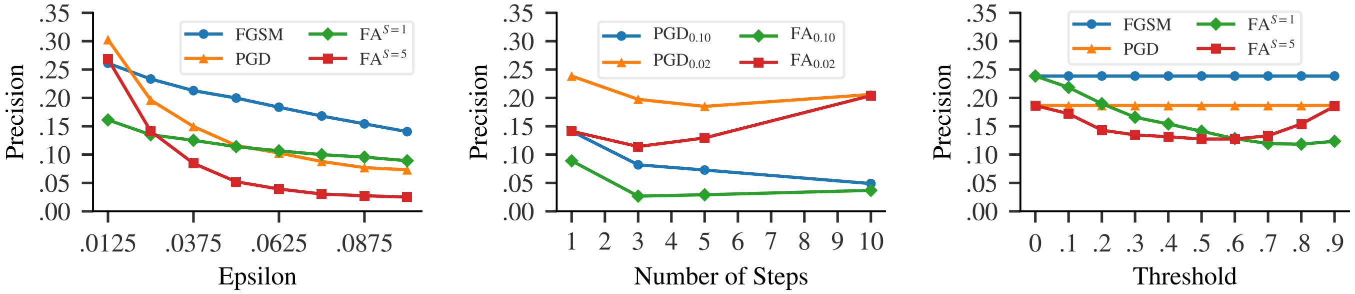
Table 2 shows the mAP scores of the three models on the COCO detection task for $\mathcal{E}$-balls of radius 0.1 and 0.02. In order to avoid overfitting the threshold to each particular model to enhance results artificially, we use a constant focusing threshold $t = 0.5$. In the first column, we reproduce the officially reported precision scores for each model and

[3] https://github.com/uvipen/SSD-pytorch/

[4] https://github.com/yhenon/pytorch-retinanet

[5] https://pytorch.org/vision/stable/models.html

[6] https://huggingface.co/docs/transformers/model_doc/detr



(a) Effect of the $\mathcal{E}$-ball radius. Values are pixel colors in a [0,1] range.

(b) Effect of the perturbation granularity. The footers of the methods indicate the $\mathcal{E}$ radius.

(c) Effect of the focusing threshold. FGSM and PGD are not affected by this hyperparameter.

Figure 2: Effect of various hyperparameters on the precision of detections. The measure is mAP and the baseline for Faster R-CNNis 0.469. *FA* methods' footer is the number of iterations; the exponent is the focusing threshold.

then report the mAP scores for various attacks. While both FGSM and PGD show a significant reduction in mAP, our attacks perform drastically better, even at minimal $\mathcal{E}$ values.

Table 3 compares our focused attacks with other current state-of-the-art adversarial attacks to fool object detectors. To fool SSD300, we use a threshold $T = 0.1$, whereas, for Faster R-CNN, we set it to $T = 0.5$; these values were obtained with a 10-step hyperparameter search and are supported by the fact that SSD300 is more sparsely activated than Faster R-CNN, and thresholding activations at 0.5 often result in all activations being filtered out and the gradients reducing to 0. DETR is, by design, the model with the least sparse feature maps, which are orders of magnitude smaller than in CNNs. The benefits of FAs are partially reduced by the density of activations in DETR's feature map. This can be seen particularly with the smaller budget of $\mathcal{E} = 0.02$. This transformer network is therefore more robust than its CNN counterparts, a result that follows the trends reported in (Mahmood, Mahmood, and van Dijk 2021). Overall, Focused Attacks perform as well or better than other gradient-based attacks and also provide equal or greater performance to GAN-based methods without requiring training any separate network.

Speed

The average adversarial attack execution times are reported in Table 4, and result from using a single RTX 3070 GPU. These times are not limited to the computational time but also include I/O and main memory to GPU (and back) overhead. As this overhead is equivalently present in each of the three models, the results are comparable. In all cases, focused adversarial attacks are equally fast or faster than their baseline counterparts.

Perceptibility

Focused adversarial attacks are constrained by a pixel-wise upper bound $\mathcal{E}$. Within the related $\mathcal{E}$-ball, however, images may be more or less perturbed across all their pixels. As the L_∞ distance metric is unable to capture this difference, we also show the L_1 perturbation magnitude across samples. Given two pixels, their L_1 distance tells how many values they are apart from each other or their absolute difference.

For each image and its adversarial counterpart, $\mathbf{x}, \mathbf{x}' \in [0,1]^{W\times H\times 3}$, their mean L_1 distance is $\frac{\|\mathbf{x}-\mathbf{x}'\|_1}{WHC}$. In contrast with some works in the literature (Wang et al. 2021), we use the *mean* of the norm because of two reasons: it is more interpretable, and it is comparable across images of different sizes.

Figure 4 shows the distribution of mean L_1 norms of the perturbation vectors generated with Faster R-CNN on the COCO 2017 validation dataset, using FGSM or five iterations of PGD. Our focused adversarial examples show a decrease of 10% in the intensity of perturbations applied to images under the same hyperparameter settings. We believe this is due to our perturbations being geared towards cloaking sensitive parts of images while producing randomly-oriented changes in nearby pixels, thus leading to subsequent cloaks often canceling the previous ones out. Visual evidence of this behavior is found in the examples in Figure 3.

Additionally, Table 3 includes a comparison of the Peak Signal to Noise Ratio (PSNR) of cloaked images across methods. The higher this value, the more clear is the image. Focused attacks perform better than baseline FGSM and PGD attacks and equally well to GAN-based adversarial methods.

Limitations and Future Works

In this paper we propose *focused adversarial attacks*, a gradient-based adversarial machine learning attack to break object detectors. By targeting adversarial perturbations only towards sensitive regions of images, focused attacks are more effective and also less visible than other state-of-the-art methods, under the same constraints. We believe this algorithm will be another important tool in ML practitioners' arsenal to evaluate the vulnerability of ML models and design more robust systems.

Being gradient-based, focused attacks require white-box access to the models they attack. While transferability has been shown in a number of similar works, it is usually constrained to special cases or requires significantly invasive

Table 2: mean Average Precision (mAP) on the COCO 2017 val dataset for different $\mathcal{E}$-ball radiuses.

Model	Baseline	$\mathcal{E} = 0.1$				$\mathcal{E} = 0.02$			
		FGSM	PGD	FA^1	FA^5	FGSM	PGD	FA^1	FA^5
RetinaNet	0.345	0.198	0.097	**0.065**	0.074	0.288	0.216	0.081	**0.020**
SSD300	0.244	0.113	0.102	0.056	**0.043**	0.188	0.168	0.088	**0.076**
F. R-CNN	0.469	0.162	0.067	0.078	**0.028**	0.262	0.161	**0.122**	0.181
DETR	0.421	0.189	0.124	0.153	**0.062**	0.321	0.301	0.238	**0.235**

Table 3: mAP resulting of adversarial attacks effectiveness on the Pascal VOC 2007 test set. PSNR metric is included to measure visual perturbation; it is in the range $[0, 100]$, and higher values are better. T is the threshold. DAG refers to (Xie et al. 2017), UEA to (Wei et al. 2019), and MI to (Liang, Wei, and Cao 2021).

Model	Baseline	Gradient Attacks					GAN Attacks	
		FGSM	PGD	FA^1	FA^5	DAG	UEA	MI
SSD300	0.686	0.564	0.531	0.200	0.048	0.640	0.200	0.160
F. R-CNN	0.778	0.431	0.388	0.204	0.056	0.050	0.050	0.060
PSNR		21	28	22	30	31	28	30

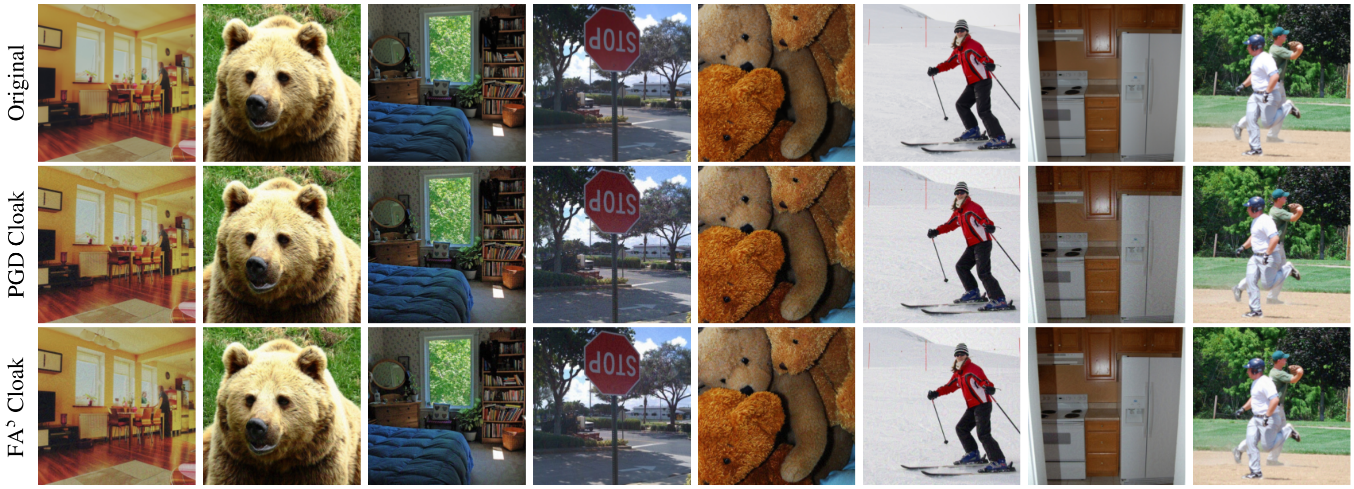


Figure 3: Adversarial examples generated with standard PGD and with our method. Cloaks are computed with Faster R-CNN, with 5 steps of gradient descent, and within an $\mathcal{E}$-ball of radius $\mathcal{E} = 0.1$ on a $[0, 1]$ scale.

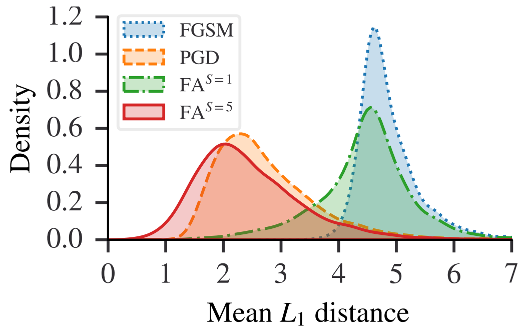


Figure 4: Distribution of L_1 norms of the perturbation masks generated with Faster R-CNN. The dashed lines indicate the mean of each distribution. Values on the abscissa are average pixel differences in the image and refer to a $[0, 255]$ range.

Table 4: Average time to cloak a COCO image.

Model	FGSM	PGD	FA^1	FA^5
RetinaNet	130 ms	645 ms	117 ms	585 ms
SSD300	95 ms	468 ms	34 ms	167 ms
F. R-CNN	183 ms	919 ms	171 ms	817 ms

perturbations to be effective. At the same time, ensemble attacks have shown good transferability and our system can easily be incorporated into an ensemble model.

In the future, we intend to optimize the implementation of focused attacks to reduce the processing time per frame down to at most $40ms$. This will allow us to perform real-time object cloaking in video feeds (at 24 FPS), and ideally adapt the attacks to the physical world.

References

Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; and Zagoruyko, S. 2020. End-to-End Object Detection with Transformers. In Vedaldi, A.; Bischof, H.; Brox, T.; and Frahm, J.-M., eds., *Computer Vision – ECCV 2020*, 213–229. Cham: Springer International Publishing. ISBN 978-3-030-58452-8.

Carlini, N.; and Farid, H. 2020. Evading Deepfake-Image Detectors With White- and Black-Box Attacks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*.

Cilloni, T.; Wang, W.; Walter, C.; and Fleming, C. 2022. Ulixes: Facial Recognition Privacy with Adversarial Machine Learning. *Proceedings on Privacy Enhancing Technologies*, 2022(1): 148–165.

Everingham, M.; Van Gool, L.; Williams, C. K. I.; Winn, J.; and Zisserman, A. 2007. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative Adversarial Nets. In Ghahramani, Z.; Welling, M.; Cortes, C.; Lawrence, N.; and Weinberger, K., eds., *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc.

Gu, Z.; Hu, W.; Zhang, C.; Lu, H.; Yin, L.; and Wang, L. 2021. Gradient Shielding: Towards Understanding Vulnerability of Deep Neural Networks. *IEEE Transactions on Network Science and Engineering*, 8(2): 921–932.

Liang, S.; Wei, X.; and Cao, X. 2021. Generate More Imperceptible Adversarial Examples for Object Detection. In *ICML 2021 Workshop on Adversarial Machine Learning*.

Lin, T.; Goyal, P.; Girshick, R.; He, K.; and Dollar, P. 2020. Focal Loss for Dense Object Detection. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 42(02): 318–327.

Lin, T.; Maire, M.; Belongie, S. J.; Bourdev, L. D.; Girshick, R. B.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft COCO: Common Objects in Context. *CoRR*, abs/1405.0312.

Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; and Berg, A. C. 2016. SSD: Single Shot MultiBox Detector. In Leibe, B.; Matas, J.; Sebe, N.; and Welling, M., eds., *Computer Vision – ECCV 2016*, 21–37. Cham: Springer International Publishing. ISBN 978-3-319-46448-0.

Mahmood, K.; Mahmood, R.; and van Dijk, M. 2021. On the Robustness of Vision Transformers to Adversarial Examples. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 7838–7847.

Nguyen, T. T.; Nguyen, C. M.; Nguyen, D. T.; Nguyen, D. T.; and Nahavandi, S. 2019. Deep Learning for Deepfakes Creation and Detection. *CoRR*, abs/1909.11573.

Ren, S.; He, K.; Girshick, R.; and Sun, J. 2015. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In Cortes, C.; Lawrence, N.; Lee, D.; Sugiyama, M.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

Shan, S.; Wenger, E.; Zhang, J.; Li, H.; Zheng, H.; and Zhao, B. Y. 2020. Fawkes: Protecting Privacy against Unauthorized Deep Learning Models. In *29th USENIX Security Symposium (USENIX Security 20)*, 1589–1604. USENIX Association. ISBN 978-1-939133-17-5.

Wang, D.; Li, C.; Wen, S.; Han, Q.-L.; Nepal, S.; Zhang, X.; and Xiang, Y. 2021. Daedalus: Breaking Nonmaximum Suppression in Object Detection via Adversarial Examples. *IEEE Transactions on Cybernetics*, 1–14.

Wei, X.; Liang, S.; Chen, N.; and Cao, X. 2019. Transferable Adversarial Attacks for Image and Video Object Detection. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, 954–960. International Joint Conferences on Artificial Intelligence Organization.

Xiao, T.; Tsai, Y.; Sohn, K.; Chandraker, M.; and Yang, M. 2020. Adversarial Learning of Privacy-Preserving and Task-Oriented Representations. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, 12434–12441. AAAI Press.

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 (ICCV)*.

Xu, K.; Liu, S.; Zhao, P.; Chen, P.-Y.; Zhang, H.; Fan, Q.; Erdogmus, D.; Wang, Y.; and Lin, X. 2019. Structured Adversarial Attack: Towards General Implementation and Better Interpretability. In *International Conference on Learning Representations*.