Enhancing Cross-Task Black-Box Transferability of Adversarial Examples with Dispersion Reduction

Yantao Lu*
Syracuse University
ylu25@syr.edu

Yunhan Jia*
Bytedance AI Lab
yunhan.jia@bytedance.com

Jianyu Wang
Baidu USA
wjyouch@gmail.com

Bai Li
Duke University
bai.li@duke.edu

Weiheng Chai
Syracuse University
wchai01@syr.edu

Lawrence Carin
Duke University
lcarin@duke.edu

Senem Velipasalar
Syracuse University†
svelipas@syr.edu

Abstract

Neural networks are known to be vulnerable to carefully crafted adversarial examples, and these malicious samples often transfer, i.e., they remain adversarial even against other models. Although significant effort has been devoted to the transferability across models, surprisingly little attention has been paid to cross-task transferability, which represents the real-world cybercriminal's situation, where an ensemble of different defense/detection mechanisms need to be evaded all at once. We investigate the transferability of adversarial examples across a wide range of real-world computer vision tasks, including image classification, object detection, semantic segmentation, explicit content detection, and text detection. Our proposed attack minimizes the "dispersion" of the internal feature map, overcoming the limitations of existing attacks, that require task-specific loss functions and/or probing a target model. We conduct evaluation on open-source detection and segmentation models, as well as four different computer vision tasks provided by Google Cloud Vision (GCV) APIs. We demonstrate that our approach outperforms existing attacks by degrading performance of multiple CV tasks by a large margin with only modest perturbations.

1. Introduction

Recent progress in adversarial machine learning has brought the weaknesses of deep neural networks (DNNs) into the spotlight, and drawn the attention of researchers working on security and machine learning. Given a deep learning model, it is easy to generate adversarial examples (AEs), which are close to the original input, but are easily misclassified by the model [9, 33]. More importantly, their effectiveness sometimes *transfers*, which may severely hinder DNN-based applications especially in security critical scenarios [23, 13, 36]. While such problems are alarming, little attention has been paid to the threat model of commercially deployed vision-based systems, wherein deep learning models across different tasks are assembled to provide fail-safe protection against evasion attacks. Such a threat model is quite different from models that have been intensively studied in the aforementioned research.

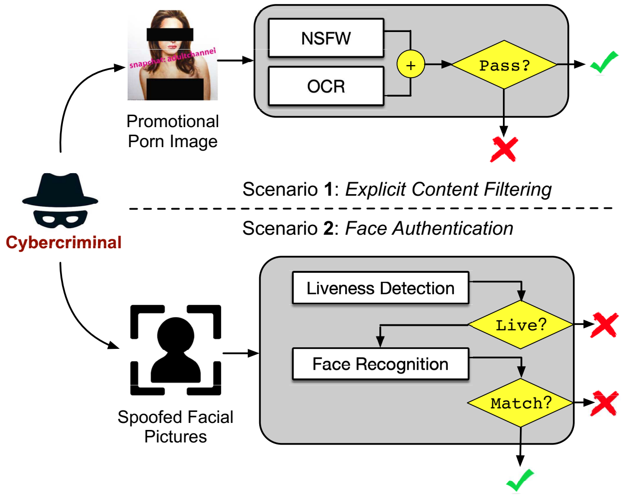


Figure 1: Real-world computer vision systems deployed in safety- and security-critical scenarios usually employ an ensemble of detection mechanisms that are opaque to attackers. Cybercriminals are required to generate adversarial examples that transfer across tasks to maximize their chances of evading the entire detection systems.

Cross-task threat model. Computer vision (CV) based detection mechanisms have been deployed extensively in security-critical applications, such as content censorship and authentication with facial biometrics, and readily available services are provided by cloud giants through APIs

*Equal contribution

†The information, data, or work presented herein was funded in part by National Science Foundation (NSF) under Grant 1739748, Grant 1816732 and by the Advanced Research Projects Agency-Energy (ARPA-E), U.S. Department of Energy, under Award Number DE-AR0000940. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

($e.g.$, Google Cloud Vision [3]). The detection systems have long been targeted by evasive attacks from cybercriminals, and it has resulted in an arms race between new attacks and more advanced defenses. To overcome the weakness of deep learning in an individual domain, real-world CV systems tend to employ an ensemble of different detection mechanisms to prevent evasions. As shown in Fig. 1, underground businesses embed promotional contents such as URLs into porn images with sexual content for illicit online advertising or phishing. A detection system, combining Optical Character Recognition (OCR) and image-based explicit content detection, can thus drop posted images containing either suspicious URLs or sexual content to mitigate evasion attacks. Similarly, a face recognition model that is known to be fragile [32] is usually protected by a liveness detector to defeat spoofed digital images when deployed for authentication. Such ensemble mechanisms are widely adopted in real-world CV deployment.

To evade detection systems with uncertain underlying mechanisms, attackers turn to generating adversarial examples that transfer across CV tasks. Many adversarial techniques on enhancing transferability have been proposed [38, 36, 23, 13]. However, most of them are designed for image classification tasks, and rely on task-specific loss functions ($e.g.$, cross-entropy loss), which limits their effectiveness when transferred to other CV tasks.

To provide a strong baseline attack to evaluate the robustness of DNN models under the aforementioned threat model, we propose a new succinct method to generate adversarial examples, which transfers across a broad class of CV tasks, including classification, object detection, semantic segmentation, explicit-content detection, and text detection and recognition. Our approach, called *Dispersion Reduction* (**DR**) and illustrated in Fig. 2, is inspired by the impact of "contrast" on an image's perceptibility. As lowering the contrast of an image would make the objects indistinguishable, we presume that reducing the "contrast" of an internal feature map would also degrade the recognizability of objects in the image, and thus could evade CV-based detection.

We use *dispersion* as a measure of "contrast" in feature space, which describes how scattered the feature map of an internal layer is. We empirically validate the impact of dispersion on model predictions, and find that reducing the dispersion of internal feature maps significantly affects the activation of subsequent layers. Based on additional observation that lower layers detect simple features [20], we hypothesize that the low-level features extracted by early convolution layers share many similarities across CV models. By reducing the dispersion of an internal feature map, the information that is in the feature output becomes indistinguishable or useless, and thus the following layers are not able to obtain any useful information no matter what kind of CV task is at hand. Thus, the distortions caused by dispersion reduction in feature space are ideally suited to fool any CV model, whether designed for classification, object detection, semantic segmentation, text detection, or other vision tasks.

Based on these observations, we propose and build the **DR** as a strong baseline attack to evaluate model robustness against black box attacks, which generate adversarial examples using simple and readily-available image classification models ($e.g.$, VGG-16, Inception-V3 and ResNet-152), whose effects extend to a wide range of CV tasks. We evaluate our proposed DR attack on both popular open source detection and segmentation models, as well as commercially deployed detection models on four Google Cloud Vision APIs: classification, object detection, SafeSearch, and Text Detection (see §4). ImageNet, PASCAL VOC2012 and MS COCO2017 datasets are used for evaluations. The results show that our proposed attack causes larger drops on the model performance compared to the state-of-the-art attacks (MI-FGSM [13], DIM [36] and TI [14]) across different tasks. We hope that our findings raise alarms for real-world CV deployment in security-critical applications, and that our simple but effective attack will be used as a benchmark to evaluate model robustness. Code is available at: https://github.com/erbloo/dr_cvpr20.

Contributions. Our contributions include the following:

- This work is the first to study adversarial machine learning for cross-task attacks. The proposed attack, called dispersion reduction, does not rely on labeling systems or task-specific loss functions.
- Evaluations shows that the proposed DR attack beats state-of-the-art attacks in degrading the performance of object detection and semantic segmentation models, and four different GCV API tasks, by a large margin: 52% lower mAP (detection) and 31% lower mIoU (segmentation) compared to the best of the baseline attacks.
- Code and evaluation data are all available at an anonymized GitHub repository [1].

2. Related Work

Adversarial examples [33, 16] have recently been shown to be able to transfer across models trained on different datasets, having different architectures, or even designed for different tasks [23, 35]. This transferability property motivates the research on black-box adversarial attacks.

One notable strategy, as demonstrated in [29, 28], is to perform black-box attacks using a substitute model, which is trained to mimic the behavior of the target model by a distillation technique. They also demonstrated black-box attacks against real-world machine learning services hosted

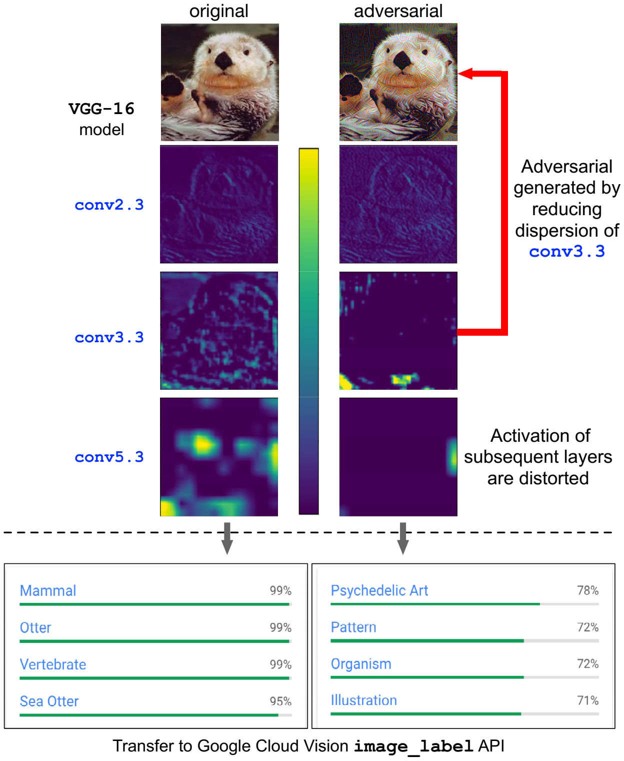


Figure 2: DR attack targets on the dispersion of the feature map at a specific layer of feature extractors. The adversarial example generated by minimizing dispersion at `conv3.3` of VGG-16 model also distorts feature space of subsequent layers (*e.g.*, `conv5.3`), and its effectiveness transfers to commercially deployed GCV APIs.

by Amazon and Google. Another related line of research, called a gradient-free attack, uses feedback on query data, *i.e.*, soft predictions [34, 18] or hard labels [8] to construct adversarial examples.

The limitation of the aforementioned works is that they all require (some form of) feedback from the target model, which may not be practical in some scenarios. Recently, several methods have been proposed to improve transferability, by studying the attack generation process itself; our method falls into this category. In general, an iterative attack [9, 19, 27] achieves a higher attack success rate than a single-step attack [16] in a white-box setting, but performs worse when transferred to other models. The methods mentioned below reduce the overfitting effect by either improving the optimization process or by exploiting data augmentation.

MI-FGSM. Momentum Iterative Fast Gradient Sign Method (MI-FGSM) [13] integrates a momentum term into the attack process, to stabilize update directions and escape poor local maxima. The update procedure is as follows:

$$\begin{aligned} x'_{t+1} &= x'_t + \alpha \cdot sign(g_{t+1}) \\ g_{t+1} &= \mu \cdot g_t + \frac{\bigtriangledown_x J(x'_t, y)}{\| \bigtriangledown_x J(x'_t, y) \|_1} \end{aligned} \qquad (1)$$

The strength of MI-FGSM can be controlled by the momentum and the number of iterations.

DIM. Momentum Diverse Inputs Fast Gradient Sign Method (DIM) combines momentum and an input diversity strategy to enhance transferability [36]. Specifically, DIM applies an image transformation, $T(\cdot)$, to the inputs with a probability p at each iteration of iterative FGSM to alleviate the overfitting phenomenon. The update procedure is similar to MI-FGSM, the only difference being the replacement of (1) by:

$$x'_{t+1} = Clip_x^{\epsilon}\{x'_t + \alpha \cdot sign(\bigtriangledown_x L(T(x'_{t+1}; p), y^{true}))\} \qquad (2)$$

where $T(x'_t, p)$ is a stochastic transformation function that performs input transformation with probability p.

TI. Rather than optimizing the objective function at a single point, the Translation-Invariance (TI) [15] method uses a set of translated images to optimize an adversarial example. By approximation, TI calculates the gradient at the untranslated image $\hat{x}$ and then averages all the shifted gradients. This procedure is equivalent to convolving the gradient with a kernel composed of all the weights.

The major difference between our proposed method and the three aforementioned attacks is that **our method does not rely on task-specific loss functions** (*e.g.*, cross-entropy loss or hinge loss). Instead, it focuses on low-level features, that are presumably task-independent and shared across different models. This is especially critical in the scenario for which the attackers do not know the specific tasks of the target models. Our evaluation in §4 demonstrates improved transferability generated by our method across several different real-world CV tasks.

3. Methodology

To construct AEs against a target model, we first establish a source model as the surrogate, to which we have access. Conventionally, the source model is established by training with examples labeled by the target model. That is, the inputs are paired with the labels generated from the target model, instead of the ground truth. In this way, the source model mimics the behavior of the target model. When we construct AEs against the source model, they are likely to transfer to the target model due to this connection.

In our framework, although a source model is still required, there is no need for training new models or querying the target model for labels. Instead, a pretrained public model could simply serve as the source model due to the strong transferability of the AEs generated via our approach. For example, in our experiments, we use pretrained VGG-16, Inception-v3 and Resnet-152, which are publicly available, as the source model f. With f as the source model, we construct AEs against it. Existing attacks perturb input images along gradient directions $\bigtriangledown_x J$ that depend on

Algorithm 1 Dispersion reduction attack

```
Input: A classifier $f$, original sample $\mathbf{x}$, feature map at layer $k$; perturbation budget $\epsilon$
Input: Attack iterations $T$.
Output: An adversarial example $\mathbf{x}'$ with $\parallel \mathbf{x}' - \mathbf{x} \parallel_\infty \leqslant \epsilon$
procedure DISPERSION REDUCTION
    $\mathbf{x}'_0 \leftarrow x$
    for $t = 0$ to $T - 1$ do
        Forward $\mathbf{x}'_t$ and obtain feature map at layer $k$:
            $\mathcal{F}_k = f(\mathbf{x}'_t)|_k$    (3)
        Compute dispersion of $\mathcal{F}_k$: $g(\mathcal{F}_k)$
        Compute its gradient $w.r.t$ the input: $\bigtriangledown_{\mathbf{x}} g(\mathcal{F}_k)$
        Update $\mathbf{x}'_t$:
            $\mathbf{x}'_t = \mathbf{x}'_t - \bigtriangledown_{\mathbf{x}} g(\mathcal{F}_k)$    (4)
        Project $\mathbf{x}'_t$ to the vicinity of $\mathbf{x}$:
            $\mathbf{x}'_{t+1} = clip(\mathbf{x}'_t, \mathbf{x} - \epsilon, \mathbf{x} + \epsilon)$    (5)
    return $\mathbf{x}'_{t+1}$
```

the definition of the task-specific loss function J, which not only limits their cross-task transferability but also requires ground-truth labels that are not always available. To mitigate these issues, we propose a *dispersion reduction* (DR) attack, that formally defines the problem of finding an AE as an optimization problem:

$$\begin{aligned} &\min_{\mathbf{x}'} g(f(\mathbf{x}', \theta)) \\ &s.t. \parallel \mathbf{x}' - \mathbf{x} \parallel_\infty \leqslant \epsilon \end{aligned} \tag{6}$$

where $f(\cdot)$ is a DNN classifier with output of intermediate feature map, and $g(\cdot)$ calculates the dispersion. Our proposed DR attack, detailed in Algorithm 1, takes a multi-step approach that creates an AE by iteratively reducing the dispersion of an intermediate feature map at layer k. Dispersion describes the extent to which a distribution is stretched or squeezed, and there can be different measures of dispersion, such as the standard deviation, and the gini coefficient [26]. In this work, we choose standard deviation as the dispersion metric due to its simplicity, and denote it by $g(\cdot)$.

To explain why reducing dispersion could lead to valid attacks, we propose a similar argument as used in [16]. Consider a simplified model where $f(\mathbf{x}) = \mathbf{a} = (a1, \ldots, a_n)^\top$ is the intermediate feature, and $\mathbf{y} = \mathbf{W}\mathbf{a}$ is an affine transformation of the feature (we omit the bias $\mathbf{b}$ for simplicity), resulting in the final output logits $\mathbf{y} = (y_1, \ldots, y_k)^\top$. In other words, we decompose a DNN classifier into a feature extractor $f(\cdot)$ and an affine transformation. If the correct class is c, the logit y_c of a correctly classified example should be the largest, that is $\mathbf{w}_c\mathbf{a} >> \mathbf{w}_i\mathbf{a}$ for $i \neq c$, where $\mathbf{w}_i$ is the ith row of $\mathbf{W}$. This indicates $\mathbf{w}_c$ and $\mathbf{a}$ are highly aligned.

On the other hand, suppose our attack aims to reduce the standard deviation of the feature $\mathbf{a}$. The corresponding adversarial examples $\mathbf{x}'$ leads to a perturbed feature

$$\begin{aligned} f(\mathbf{x}') = \mathbf{a}' &\approx \mathbf{a} - \alpha \frac{\partial}{\partial \mathbf{a}} Std(\mathbf{a}) \\ &= \mathbf{a} - 2\alpha(\mathbf{a} - \bar{a}\mathbf{1})/(\sqrt{n-1} Std(\mathbf{a})) \end{aligned} \tag{7}$$

Where α depicts the magnitude of the perturbation on $\mathbf{a}$, $\bar{a}$ is the average of the entries of $\mathbf{a}$, and $\mathbf{1}$ is a column vector with 1 in each entry. Therefore, the change of the logit y_c due to adversarial perturbation is essentially

$$\begin{aligned} \Delta y_c &= -2\alpha(\mathbf{w}_c\mathbf{a} - \mathbf{w}_c\mathbf{1}\bar{a})/(\sqrt{n-1} Std(\mathbf{a})) \\ &= -2\alpha(\mathbf{w}_c\mathbf{a} - n\bar{w}_c\bar{a})/(\sqrt{n-1} Std(\mathbf{a})) \\ &= -2\alpha\sqrt{n-1} Cov(\mathbf{w}_c, \mathbf{a})/Std(\mathbf{a}) < 0 \end{aligned} \tag{8}$$

If we think of each entry of $\mathbf{a}$ and $\mathbf{w}_c$ as samples, the $Cov(\mathbf{w}_c, \mathbf{a})$ corresponds to the empirical covariance of these samples. This suggests that as long as $\mathbf{w}_c$ and $\mathbf{a}$ are aligned, our attack can always reduce the logit of the correct class. Note that α is approximately the product of the magnitude of the perturbation on $\mathbf{x}$ and the sensitivity of $f(\cdot)$, therefore the reduction of the logit could be large if $f(\cdot)$ is sensitive, which is often the case in practice.

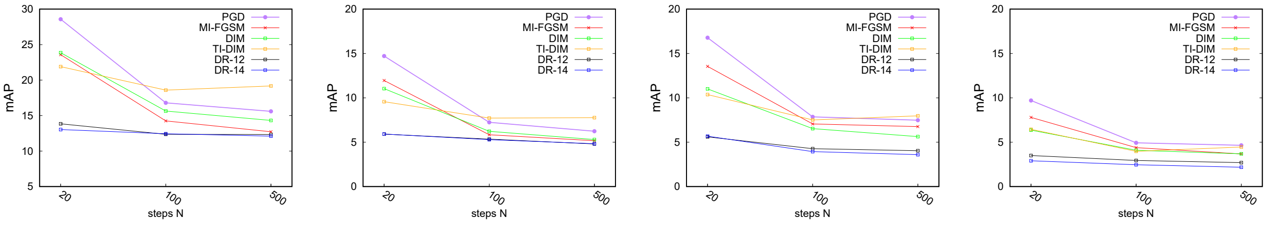
In general, y_c could be any activation that is useful for the task, which may not be classification. As long as y_c is large for natural examples, indicating a certain feature is detected, it is always reduced by our attacks according to the analysis above. Thus, our attack is agnostic to tasks and the choice of loss functions.

4. Experimental Results

We compare our proposed DR attack with the state-of-the-art black-box adversarial attacks on object detection and semantic segmentation tasks (using publicly available models), and commercially deployed Google Cloud Vision (GCV) tasks.

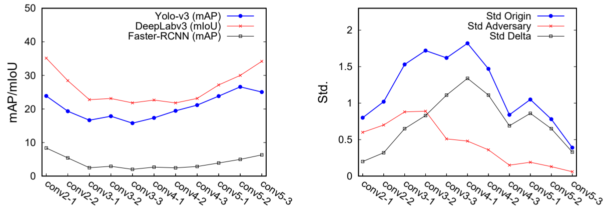
4.1. Experimental Settings

Network Types: We consider Yolov3-DarkNet53 [30], RetinaNet-ResNet50 [21], SSD-MobileNetv2 [22], Faster R-CNN-ResNet50 [31], Mask R-CNN-ResNet50 [17] as the target object detection models and DeepLabv3Plus-ResNet101 [11], DeepLabv3-ResNet101 [10], FCN-ResNet101 [24] as the target semantic segmentation models. All network models are publicly available, and details are provided in the Appendix. The source networks



(a) SSD-ResNet50 (b) RetinaNet-ResNet50 (c) SSD-MobileNet (d) FasterRCNN-ResNet50

Figure 3: **Results of DR attack with different steps N.** The proposed DR attack outperforms all baselines, even starting from small steps ($e.g.$, $N = 20$).



(a) mAP/mIoU results. (b) Std. before and after attack

Figure 4: **Results of DR attack with different attack layers of VGG16.** We see that attacking the middle layers results in higher drop in the performance compared to attacking top or bottom layers. At the same time, in the attacking process, the drop in std of middle layers is also larger than the top and bottom layers. This motivates that we can find a good attack layer by looking at the std drop during the attack.

for generating adversarial examples are VGG16, Inception-v3 and Resnet152 with output image sizes of (224×224), (299×299) and (224×224), respectively. For the evaluation on COCO2017 and PASCAL VOC2012 datasets, the mAP and mIoU are calculated as the evaluation metrics for detection and semantic segmentation, respectively. Due to the mismatch of different models being trained with different labeling systems (COCO / VOC), only 20 classes that correspond to VOC labels are chosen from COCO labels if a COCO pretrained model is tested on the PASCAL VOC dataset, or a VOC pretrained model is tested on the COCO dataset. For the evaluation on ImageNet, since not all test images have the ground truth bounding boxes and pixelwise labels, the mAP and mIoU are calculated as the difference between the outputs of benign/clean images and adversarial images.

Implementation details: We compare our proposed method with projected gradient descent (PGD) [27], momentum iterative fast gradient sign method (MI-FGSM) [12], diverse inputs method (DIM) [37] and translation-invariant attacks (TI) [15]. Concerning the hyperparameters, the maximum perturbation is set to be $\epsilon = 16$ for all the experiments with pixel values in [0, 255]. For the proposed DR attacks, the step size is $\alpha = 4$, and the number of training steps is $N = 100$. For the baseline methods, we first follow the default settings in [37] and [15] with $\alpha = 1$ and $N = 20$ for PGD, MI-FGSM and DIM, $\alpha = 1.6$ and $N = 20$ for TI-DIM. We apply the same hyper-parameters ($\alpha = 4$, $N = 100$) used with the proposed method to all the baseline methods. For MI-FGSM, we adopt the default decay factor $\mu = 1.0$. For DIM and TI-DIM, the transformation probability is set to $p = 0.5$.

4.2. Diagnostics

4.2.1 The effect of training steps N

We show the results of attacking SSD-ResNet50, RetinaNet-ResNet50, SSD-MobileNet and Faster RCNN-ResNet50 with a different number of training steps ($N = \{20, 100, 500\}$) based on MS COCO2017 validation set. We also compare the proposed DR attack with multiple baselines, namely PGD, MI-FGSM, DIM, TI-DIM. The results are shown in Fig. 3. In contrast to the classification-based transfer attacks [13, 36, 14], we do not observe over-fitting in cross-task transfer attacks for all the tested methods. Therefore, instead of using $N = 20$, which is the value used by the baseline attacks we compare with, we can employ larger training steps ($N = 100$), and achieve better attack performance at the same time. In addition, we can see that our DR attack outperforms all the state-of-the-art baselines for all the step size settings. It should be noticed that DR attack is able to achieve promising results at $N = 20$, and the results from the DR attack, using 20 steps, are better than those of baseline methods using 500 steps. This shows that our proposed DR attack has higher efficiency than the baselines.

4.2.2 The effect of attack layer

We show the results of attacking different convolutional layers of the VGG16 network with the proposed DR attack based on the PASCAL VOC2012 validation set. Figure 4a shows the mAP for Yolov3 and faster RCNN, and mIoU for Deeplabv3 and FCN. In Fig. 4b we plot the standard deviation (std) values before and after the DR attack, together with the change. As can be seen, attacking the middle layers of VGG16 results in higher drop in the performance compared to attacking top or bottom layers. At the same time,

Detection Results Using Val. Images of		Yolov3	RetinaNet	SSD	Faster-RCNN	Mask-RCNN
COCO and VOC Datasets		DrkNet	ResNet50	MobileNet	ResNet50	ResNet50
		mAP	mAP	mAP	mAP	mAP
		COCO/VOC	COCO/VOC	COCO/VOC	COCO/VOC	COCO/VOC
VGG16	PGD (α=1, N=20)	33.5 / 54.8	14.7 / 31.8	16.8 / 35.9	9.7 / 14.2	10.3 / 15.9
	PGD (α=4, N=100)	21.6 / 38.7	7.2 / 14.6	7.9 / 18.2	4.9 / 6.4	5.7 / 9.7
	MI-FGSM (α=1, N=20)	28.4 / 48.9	12.0 / 23.6	13.6 / 29.6	7.8 / 10.9	8.2 / 12.0
	MI-FGSM (α=4, N=100)	**19.0** / 35.0	5.8 / 10.6	7.0 / 19.1	4.4 / 5.0	4.8 / 7.1
	DIM (α=1, N=20)	26.7 / 46.9	11.0 / 21.9	11.0 / 22.9	6.4 / 8.2	7.2 / 11.6
	DIM (α=4, N=100)	20.0 / 37.6	6.2 / 13.0	6.5 / 14.9	4.1 / 5.0	4.6 / 6.7
	TI-DIM (α=1.6, N=20)	25.8 / 41.4	9.6 / 17.4	10.4 / 19.9	6.5 / 7.5	7.4 / 9.2
	TI-DIM (α=4, N=100)	19.5 / **33.4**	7.7 / 13.1	7.5 / 16.7	4.0 / 5.2	4.8 / 6.6
	DR (α=4, N=100)**(ours)**	19.8 / 38.2	**5.3 / 8.7**	**3.9 / 8.2**	**2.5 / 2.8**	**3.2 / 5.1**
InceptionV3	PGD (α=1, N=20)	46.8 / 67.5	23.9 / 51.8	25.2 / 47.4	27.0 / 45.7	27.5 / 48.7
	PGD (α=4, N=100)	35.3 / 57.1	15.0 / 33.0	14.0 / 31.6	18.2 / 31.7	19.4 / 34.8
	MI-FGSM (α=1, N=20)	42.0 / 63.9	20.0 / 44.3	20.9 / 43.5	22.8 / 39.3	23.7 / 42.9
	MI-FGSM (α=4, N=100)	32.4 / 54.0	12.5 / 27.1	13.1 / 29.2	16.3 / 26.9	17.9 / 30.5
	DIM (α=1, N=20)	32.5 / 54.5	12.9 / 27.5	13.9 / 29.7	14.2 / 24.0	16.3 / 27.7
	DIM (α=4, N=100)	29.1 / 48.3	10.4 / 20.5	10.4 / 22.0	12.2 / 18.2	13.8 / 44.6
	TI-DIM (α=1.6, N=20)	32.1 / 50.2	12.8 / 25.8	13.5 / 28.0	12.5 / 20.4	14.4 / 23.0
	TI-DIM (α=4, N=100)	27.1 / **42.2**	11.0 / 19.8	10.4 / 22.1	9.9 / 14.6	11.1 / 17.5
	DR (α=4, N=100)**(ours)**	**24.2** / 45.1	**8.5 / 18.9**	**9.0 / 19.5**	**8.3 / 14.3**	**9.8 / 17.0**
Resnet152	PGD (α=1, N=20)	39.4 / 62.0	19.1 / 42.9	19.9 / 41.6	13.8 / 19.4	15.0 / 22.0
	PGD (α=4, N=100)	28.8 / 51.5	12.2 / 25.9	11.2 / 24.4	8.2 / 11.3	8.8 / 13.9
	MI-FGSM (α=1, N=20)	35.1 / 58.1	15.8 / 36.2	16.7 / 35.8	11.1 / 16.3	12.2 / 18.1
	MI-FGSM (α=4, N=100)	26.4 / 48.2	11.2 / 23.5	9.9 / 21.3	7.0 / 9.5	8.2 / 11.4
	DIM (α=1, N=20)	28.1 / 50.3	12.2 / 26.3	11.0 / 23.9	7.0 / 10.6	7.9 / 12.6
	DIM (α=4, N=100)	24.7 / 43.2	8.8 / 19.4	7.8 / 16.1	5.1 / 7.1	6.2 / 10.3
	TI-DIM (α=1.6, N=20)	27.9 / 45.6	11.7 / 21.7	11.3 / 22.5	6.8 / 8.7	7.5 / 9.9
	TI-DIM (α=4, N=100)	**22.3 / 36.7**	9.0 / 15.8	8.7 / 19.1	5.0 / 6.6	5.7 / 8.2
	DR (α=4, N=100)**(ours)**	22.7 / 43.8	**6.8 / 12.4**	**4.7 / 7.6**	**2.3 / 2.8**	**3.0 / 4.5**

Table 1: **Detection results using validation images of COCO2017 and VOC2012 datasets.** Our proposed DR attack performs best on 25 out of 30 different cases, and achieves 12.8 mAP on average over all the experiments. It creates 3.9 more drop in mAP compared to the best of the baselines (TI-DIM: 16.7 mAP).

the change in std for middle layers is larger compared to the top and bottom layers. We can infer that for initial layers, the budget ϵ constrains the loss function to reduce the std, while for the layers near the output, the std is already relatively small, and cannot be reduced too much further. Based on this observation, we choose one of the middle layers as the target of the DR attack. More specifically, in the following experiments we attack conv3-3 for VGG16, the last layer of $group - A$ for inception-v3, and the last layer of 2nd group of bottlenecks(conv3-8-3) for ResNet152.

4.3. Open Source Model Experiments

We compare the proposed DR attack with the state-of-the-art adversarial techniques, to demonstrate the transferability of our method on public object detection and semantic segmentation models. We use validation sets of ImageNet, VOC2012 and COCO2017 for testing object detection and semantic segmentation tasks. For ImageNet, 5000 correctly classified images from the validation set are chosen. For VOC and COCO, 1000 images from the validation set are chosen. The test images are shared in github repository: dispersion_reduction_test_images [2].

The results for detection and segmentation on COCO and VOC datasets are shown in Tables 1 and 2, respectively. The results for detection and segmentation on the ImageNet dataset are provided in the Appendix. We also include the table for average results over all the datasets, including ImageNet, in the Appendix.

As can be seen from Tables 1 and 2, our proposed method (**DR**) achieves the best results on 36 out of 42 experiments by degrading the performance of the target model by a larger margin. For detection experiments, the **DR** attack performs best on 25 out of 30 different cases and for semantic segmentation 11 out of 12 different cases. For detection, our proposed attack achieves 12.8 mAP on average over all the experiments. It creates 3.9 more drop in mAP compared to the best of the baselines (TI-DIM: 16.7 mAP). For semantic segmentation, our proposed attack achieves 20.0 mIoU on average over all the experiments. It achieves 5.9 more drop in mIoU compared to the best of the baselines

(DIM: 25.9 mIoU).

To summarize the results on the ImageNet dataset provided in the Appendix, our proposed method (**DR**) achieves the best results in 17 out of 21 experiments. For detection, our proposed attack achieves 7.4 relative-mAP on average over all the experiments. It creates 3.8 more drop in relative-mAP compared to the best of the baselines (TI-DIM: 11.2). For semantic segmentation, our proposed attack achieves 16.9 relative-mIoU on average over all the experiments. It achieves 4.8 more drop in relative-mIoU compared to the best of the baselines (TI-DIM: 21.7).

Seg. Results Using Val. Images of COCO and VOC Datasets		DeepLabv3 ResNet-101 mIoU	FCN ResNet-101 mIoU
		COCO/VOC	COCO/VOC
VGG16	PGD (α=1, N=20)	37.8 / 42.6	26.7 / 29.1
	PGD (α=4, N=100)	22.3 / 24.0	17.1 / 18.1
	MI-FGSM (α=1, N=20)	32.8 / 36.2	22.7 / 25.0
	MI-FGSM (α=4, N=100)	19.9 / 21.6	22.0 / 16.5
	DIM (α=1, N=20)	30.3 / 33.2	15.5 / 22.4
	DIM (α=4, N=100)	21.2 / 23.7	16.2 / 16.9
	TI-DIM (α=1.6, N=20)	29.9 / 31.1	21.9 / 23.0
	TI-DIM (α=4, N=100)	23.8 / 24.7	18.9 / 19.2
	DR (α=4, N=100)**(ours)**	**17.2 / 21.8**	**12.9 / 14.4**
IncV3	PGD (α=1, N=20)	49.4 / 56.0	36.8 / 40.1
	PGD (α=4, N=100)	37.1 / 41.3	26.1 / 28.3
	MI-FGSM (α=1, N=20)	44.2 / 51.1	32.4 / 35.4
	MI-FGSM (α=4, N=100)	33.7 / 39.1	24.0 / 35.4
	DIM (α=1, N=20)	35.7 / 40.4	24.9 / 27.2
	DIM (α=4, N=100)	30.4 / 33.9	21.3 / 22.3
	TI-DIM (α=1.6, N=20)	35.3 / 37.0	26.4 / 27.7
	TI-DIM (α=4, N=100)	29.0 / 29.8	22.5 / 23.5
	DR (α=4, N=100)**(ours)**	**23.2 / 29.2**	**17.1 / 20.9**
Res152	PGD (α=1, N=20)	45.2 / 50.2	30.7 / 34.6
	PGD (α=4, N=100)	31.5 / 35.1	21.6 / 24.0
	MI-FGSM (α=1, N=20)	39.9 / 43.9	26.4 / 29.9
	MI-FGSM (α=4, N=100)	28.2 / 32.2	19.9 / 22.1
	DIM (α=1, N=20)	31.3 / 35.5	22.3 / 23.9
	DIM (α=4, N=100)	25.9 / 28.8	19.0 / 19.9
	TI-DIM (α=1.6, N=20)	31.8 / 33.9	23.7 / 25.2
	TI-DIM (α=4, N=100)	26.6 / **26.6**	20.3 / 21.4
	DR (α=4, N=100)**(ours)**	**22.7** / 27.0	**16.4 / 17.6**

Table 2: **Semantic Segmentation results using validation images of the COCO2017 and VOC2012 datasets.** Our proposed DR attack performs best on 11 out of 12 cases and achieves 20.0 mIoU on average over all the experiments. It achieves 5.9 more drop in mIoU compared to the best of the baselines (DIM: 25.9 mIoU).

4.4. Cloud API Experiments

We compare the proposed DR attack with the state-of-the-art adversarial techniques to enhance transferability on commercially deployed Google Cloud Vision (GCV) tasks [1]:

- Image Label Detection (**`Labels`**) classifies image into broad sets of categories.

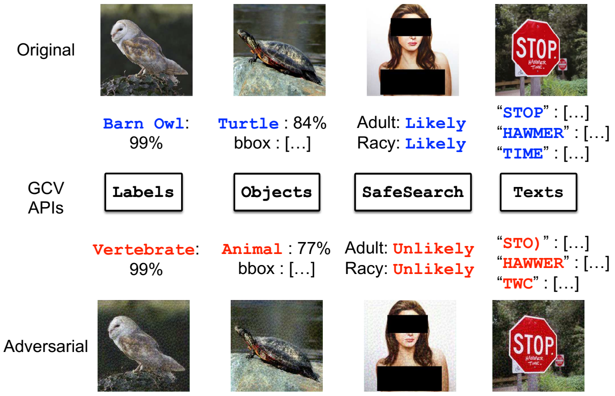

Original
Barn Owl: 99%
Turtle : 84% bbox : [...]
Adult: Likely Racy: Likely
"STOP" : [...] "HAWMER" : [...] "TIME" : [...]
GCV APIs
Labels
Objects
SafeSearch
Texts
Vertebrate: 99%
Animal : 77% bbox : [...]
Adult: Unlikely Racy: Unlikely
"STO)" : [...] "HAWWER" : [...] "TWC" : [...]
Adversarial

Figure 5: Visualization of images chosen from the testing set and their corresponding AEs generated by DR. All the AEs are generated on VGG-16 `conv3.3` layer, with perturbations clipped by $l_\infty \leq 16$, and they effectively fool the four GCV APIs as indicated by their outputs.

- Object Detection (**`Objects`**) detects multiple objects with their labels and bounding boxes in an image.
- Image Texts Recognition (**`Texts`**) detects and recognize text within an image, which returns their bounding boxes and transcript texts.
- Explicit Content Detection (**`SafeSearch`**) detects explicit content such as adult or violent content within an image, and returns the likelihood.

Datasets. We use ImageNet validation set for testing `Labels` and `Objects`, and the NSFW Data Scraper [7] and COCO-Text [4] dataset for evaluating against `SafeSearch` and `Texts`, respectively. We randomly choose 100 images from each dataset for our evaluation, and Fig. 5 shows sample images in our test set. Please note that due to the API query fees, larger scale experiments could not be performed for this part.

Experiment setup. To generate the AEs, We use normally trained VGG-16 and Resnet-152 as our source models, since Resnet-152 is commonly used by MI-FGSM and DIM for generation [36, 13]. Since the DR attack targets a specific layer, we choose `conv3.3` for VGG-16 and `conv3.8.3` for Resnet-152 as per the profiling result in Table 3 and discussion in Sec. 4.2.2.

Attack parameters. We follow the default settings in [13] with the momentum decay factor $\mu = 1$ when implementing the MI-FGSM attack. For the DIM attack, we set probability $p = 0.5$ for the stochastic transformation function $T(x; p)$ as in [36], and use the same decay factor $\mu = 1$ and total iteration number $N = 20$ as in the vanilla MI-FGSM. For our proposed DR attack, we do not rely on the FGSM method, and instead use the Adam optimizer ($\beta_1 = 0.98$, $\beta_2 = 0.99$) with learning rate of $5e^{-2}$ to reduce the dispersion of target feature map. The maximum

[1] https://cloud.google.com/vision/docs

Model	Attack	**Labels**	**Objects**	**SafeSearch**	**Texts**	
		acc.	mAP (IoU=0.5)	acc.	AP (IoU=0.5)	C.R.W[2]
baseline (SOTA)[1]		82.5%	73.2	100%	69.2	76.1%
VGG-16	MI-FGSM	41%	42.6	62%	38.2	15.9%
	DIM	39%	36.5	57%	29.9	16.1%
	DR (**Ours**)	**23%**	**32.9**	**35%**	**20.9**	**4.1%**
Resnet-152	MI-FGSM	37%	41.0	61%	40.4	17.4%
	DIM	49%	46.7	60%	**34.2**	15.1%
	DR (**Ours**)	**25%**	**33.3**	**31%**	34.6	**9.5%**

[1] The baseline performance of GCV models cannot be measured due to the mismatch between the original labels and labels used by Google. We use the GCV prediction results on original images as ground truth, thus the baseline performance should be 100% for all accuracy and 100.0 for mAP and AP. Here we provide state-of-the-art performance [5, 6, 4, 7] for reference.

[2] Correctly recognized words (C.R.W) [4].

Table 3: **The degraded performance of four Google Cloud Vision models, where we attack a single model from the left column.** Our proposed DR attack degrades the accuracy of **Lables** and **SafeSearch** to 23% and 35%, the mAP of **Objects** and **Texts** to 32.9 and 20.9, the word recognition accuracy of **Texts** to only 4.1%, which outperform existing attacks.

perturbation of all attacks in the experiments are limited by clipping at $l_\infty = 16$, which is still considered less perceptible for human observers [25].

Evaluation metrics. We perform adversarial attacks only on a single network and test them on the four black-box GCV models. The effectiveness of attacks is measured by the model performance under attacks. As the labels from original datasets are different from labels used by GCV, we use the prediction results of GCV APIs on the original data as the ground truth, which gives a baseline performance of 100% relative accuracy or 100.0 relative mAP and AP respectively.

Results. We provide the state-of-the-art results on each CV task as reference in Table 3. As shown in Table 3, DR outperforms other baseline attacks by degrading the target model performance by a larger margin. For example, the adversarial examples crafted by DR on VGG-16 model brings down the accuracy of **Labels** to only 23%, and **SafeSearch** to 35%. Adversarial examples created with the DR also degrade the mAP of **Objects** to 32.9% and AP of text localization to 20.9%, and with barely 4.1% accuracy in recognizing words. Strong baselines like MI-FGSM and DIM, on the other hand, only cause 38% and 43% success rate, respectively, when attacking SafeSearch, and are less effective compared with DR when attacking all other GCV models. The results demonstrate the better cross-task transferability of the dispersion reduction attack.

Figure 5 shows example of each GCV model's output for the original and adversarial examples. The performance of **Labels** and **SafeSearch** are measured by the accuracy of classification. More specifically, we use *top1* accuracy for **Labels**, and use the accuracy for detecting the given porn images as LIKELY or VERY_LIKELY being adult for **SafeSearch**. The performance of **Objects** is given by the mean average precision (mAP) at IoU=0.5. For **Texts**, we follow the bi-fold evaluation method of ICDAR 2017 Challenge [4]. We measure text localization accuracy using average precision (AP) of bounding boxes at IoU=0.5, and evaluate the word recognition accuracy with correctly recognized words (C.R.W) that are case insensitive.

When comparing the effectiveness of attacks on different generation models, the results demonstrate that DR generates adversarial examples that transfer better across these four commercial APIs. The visualization in Fig. 5 shows that the perturbed images with $l_\infty \leq 16$ well maintain their visual similarities with the original images, but fool the real-world computer vision systems.

5. Discussion and Conclusion

We have proposed a *Dispersion Reduction* (DR) attack to improve the cross-task transferability of adversarial examples. Specifically, our method reduces the dispersion of intermediate feature maps. Compared to existing black-box attacks, results show that our proposed method performs better on attacking black-box cross-CV-task models. One intuition behind the DR attack is that by minimizing the dispersion of feature maps, images become "featureless." This is because few features can be detected if neuron activations are suppressed by perturbing the input (Fig. 2). Moreover, with the observation that low-level features bear more similarities across CV models, we hypothesize that the DR attack would produce transferable adversarial examples when one of the middle convolution layers is targeted. Evaluation on different CV tasks shows that this enhanced attack greatly degrades model performance compared to prior state-of-the-art attacks, and thus would facilitate evasion attacks against a different task model or even an ensemble of CV-based detection mechanisms.

References

[1] Github repository for our code. https://github.com/erbloo/dr_cvpr20. 2

[2] Github repository for our evaluation data. https://github.com/erbloo/dr_images_cvpr20. 6

[3] Google Cloud Vision. Link. 2

[4] ICDAR2017 Robust reading challenge on COCO-Text. Link. 7, 8

[5] ImageNet Challenge 2017. Link. 8

[6] Keras Applications. Link. 8

[7] NSFW Data Scraper. Link. 7, 8

[8] Wieland Brendel, Jonas Rauber, and Matthias Bethge. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. *arXiv preprint arXiv:1712.04248*, 2017. 3

[9] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 39–57. IEEE, 2017. 1, 3

[10] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *CoRR*, abs/1706.05587, 2017. 4

[11] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. *arXiv:1802.02611*, 2018. 4

[12] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Xiaolin Hu, and Jun Zhu. Discovering adversarial examples with momentum. *CoRR*, abs/1710.06081, 2017. 5

[13] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9185–9193, 2018. 1, 2, 3, 5, 7

[14] Yinpeng Dong, Tianyu Pang, Hang Su, and Jun Zhu. Evading defenses to transferable adversarial examples by translation-invariant attacks. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2019. 2, 5

[15] Yinpeng Dong, Tianyu Pang, Hang Su, and Jun Zhu. Evading defenses to transferable adversarial examples by translation-invariant attacks. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2019. 3, 5

[16] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014. 2, 3, 4

[17] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask r-cnn. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988, 2017. 4

[18] Andrew Ilyas, Logan Engstrom, Anish Athalye, and Jessy Lin. Black-box adversarial attacks with limited queries and information. *arXiv preprint arXiv:1804.08598*, 2018. 3

[19] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial machine learning at scale. *arXiv preprint arXiv:1611.01236*, 2016. 3

[20] Honglak Lee, Roger Grosse, Rajesh Ranganath, and Andrew Y Ng. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In *Proceedings of the 26th annual international conference on machine learning*, pages 609–616. ACM, 2009. 2

[21] Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2999–3007, Oct 2017. 4

[22] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott E. Reed, Cheng-Yang Fu, and Alexander C. Berg. SSD: single shot multibox detector. *CoRR*, abs/1512.02325, 2015. 4

[23] Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. Delving into transferable adversarial examples and black-box attacks. *arXiv preprint arXiv:1611.02770*, 2016. 1, 2

[24] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. *CoRR*, abs/1411.4038, 2014. 4

[25] Yan Luo, Xavier Boix, Gemma Roig, Tomaso A. Poggio, and Qi Zhao. Foveation-based mechanisms alleviate adversarial examples. *arXiv preprint arXiv:1511.06292*, 2015. 8

[26] Chris A Mack. *NIST,SEMATECH e-Handbook of Statistical Methods*. 2007. 4

[27] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks, 2017. 3, 5

[28] 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*, 2016. 2

[29] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia conference on computer and communications security*, pages 506–519. ACM, 2017. 2

[30] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv*, 2018. 4

[31] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 91–99. Curran Associates, Inc., 2015. 4

[32] Mahmood Sharif, Sruti Bhagavatula, Lujo Bauer, and Michael K Reiter. Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 1528–1540. ACM, 2016. 2

[33] 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. 1, 2

[34] Jonathan Uesato, Brendan O'Donoghue, Aaron van den Oord, and Pushmeet Kohli. Adversarial risk and the dangers of evaluating against weak attacks. *arXiv preprint arXiv:1802.05666*, 2018. 3

[35] Cihang Xie, Jianyu Wang, Zhishuai Zhang, Yuyin Zhou, Lingxi Xie, and Alan Yuille. Adversarial examples for semantic segmentation and object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1369–1378, 2017. 2

[36] Cihang Xie, Zhishuai Zhang, Jianyu Wang, Yuyin Zhou, Zhou Ren, and Alan Yuille. Improving transferability of adversarial examples with input diversity. *arXiv preprint arXiv:1803.06978*, 2018. 1, 2, 3, 5, 7

[37] Cihang Xie, Zhishuai Zhang, Jianyu Wang, Yuyin Zhou, Zhou Ren, and Alan L. Yuille. Improving transferability of adversarial examples with input diversity. *CoRR*, abs/1803.06978, 2018. 5

[38] Wen Zhou, Xin Hou, Yongjun Chen, Mengyun Tang, Xiangqi Huang, Xiang Gan, and Yong Yang. Transferable adversarial perturbations. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 452–467, 2018. 2