Improving Transferability of Adversarial Examples with Input Diversity

Cihang Xie[1] Zhishuai Zhang[1] Yuyin Zhou[1] Song Bai[2]
Jianyu Wang[3] Zhou Ren[4] Alan Yuille[1]

[1]Johns Hopkins University [2]University of Oxford [3]Baidu Research [4]Wormpex AI Research

Abstract

Though CNNs have achieved the state-of-the-art performance on various vision tasks, they are vulnerable to adversarial examples — crafted by adding human-imperceptible perturbations to clean images. However, most of the existing adversarial attacks only achieve relatively low success rates under the challenging black-box setting, where the attackers have no knowledge of the model structure and parameters. To this end, we propose to improve the transferability of adversarial examples by creating diverse input patterns. Instead of only using the original images to generate adversarial examples, our method applies random transformations to the input images at each iteration. Extensive experiments on ImageNet show that the proposed attack method can generate adversarial examples that transfer much better to different networks than existing baselines. By evaluating our method against top defense solutions and official baselines from NIPS 2017 adversarial competition, the enhanced attack reaches an average success rate of 73.0%, which outperforms the top-1 attack submission in the NIPS competition by a large margin of 6.6%. We hope that our proposed attack strategy can serve as a strong benchmark baseline for evaluating the robustness of networks to adversaries and the effectiveness of different defense methods in the future. Code is available at https://github.com/cihangxie/DI-2-FGSM.

1. Introduction

Recent success of Convolutional Neural Networks (CNNs) leads to a dramatic performance improvement on various vision tasks, including image classification [15, 32, 13], object detection [10, 28, 40] and semantic segmentation [22, 5]. However, CNNs are extremely vulnerable to small perturbations to the input images, *i.e.*, human-imperceptible additive perturbations can result in failure predictions of CNNs. These intentionally crafted images are known as adversarial examples [36]. Learning how to generate adversarial examples can help us investigate the robustness of different models [1] and understand the insufficiency of current training algorithms [11, 17, 37].

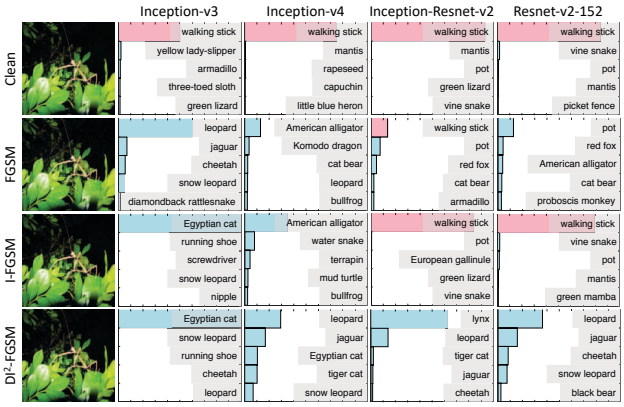

Figure 1. **The comparison of success rates using three different attacks**. The ground-truth "walking stick" is marked as pink in the top-5 confidence distribution plots. The adversarial examples are crafted on Inception-v3 with the maximum perturbation $\epsilon = 15$. From the first row to the the third row, we plot the top-5 confidence distributions of clean images, FGSM and I-FGSM, respectively. The fourth row shows the result of the proposed Diverse Inputs Iterative Fast Gradient Sign Method (DI^2-FGSM), which attacks the white-box model and all black-box models successfully.

Several methods [11, 36, 16] have been proposed recently to find adversarial examples. In general, these attacks can be categorized into two types according to the number of steps of gradient computation, *i.e.*, single-step attacks [11] and iterative attacks [36, 16]. Generally, iterative attacks can achieve higher success rates than single-step attacks in the white-box setting, where the attackers have a perfect knowledge of the network structure and weights. However, if these adversarial examples are tested on a different network (either in terms of network structure, weights or both), *i.e.*, the black-box setting, single-step attacks perform better. This trade-off is due to the fact that iterative attacks tend to overfit the specific network parameters (*i.e.*, have high white-box success rates) and thus making generated adversarial examples rarely transfer to other networks (*i.e.*, have low black-box success rates), while single-step attacks usually underfit to the network parameters (*i.e.*, have

978-1-7281-3293-8/19/$31.00 ©2019 IEEE
DOI 10.1109/CVPR.2019.00284

low white-box success rates) thus producing adversarial examples with slightly better transferability. Observing the phenomenon, one interesting question is whether we can generate adversarial examples with high success rates under both white-box and black-box settings.

In this work, we propose to improve the transferability of adversarial examples by creating diverse input patterns. Our work is inspired by the data augmentation [15, 32, 13] strategy, which has been proven effective to prevent networks from overfitting by applying a set of label-preserving transformations (*e.g.*, resizing, cropping and rotating) to training images. Meanwhile, [38, 12] showed that image transformations can defend against adversarial examples under certain situations, which indicates adversarial examples cannot generalize well under different transformations. These transformed adversarial examples are known as hard examples [30, 31] for attackers, which can then be served as good samples to produce more transferable adversarial examples.

We incorporate the proposed input diversity strategy with iterative attacks, *e.g.*, I-FGSM [17] and MI-FGSM [9]. At each iteration, unlike the traditional methods which maximize the loss function directly w.r.t. the original inputs, we apply random and differentiable transformations (*e.g.*, random resizing, random padding) to the input images with probability p and maximize the loss function w.r.t. these transformed inputs. Note that these randomized operations were previously used to defend against adversarial examples [38], while here we incorporate them into the attack process to create hard and diverse input patterns. Fig. 1 shows an adversarial example generated by our method and compares the success rates to other attack methods under both white-box and black-box settings.

We test the proposed input diversity on several network under both white-box and black-box settings, and single-model and multi-model settings. Compared with traditional iterative attacks, the results on ImageNet (see Sec. 4.2) show that our method gets significantly higher success rates for black-box models and maintains similar success rates for white-box models. By evaluating our attack method w.r.t. the top defense solutions and official baselines from NIPS 2017 adversarial competition [18], this enhanced attack reaches an average success rate of 73.0%, which outperforms the top-1 attack submission in the NIPS competition by a large margin of 6.6%. We hope that our proposed attack strategy can serve as a benchmark for evaluating the robustness of networks to adversaries and the effectiveness of different defense methods in future.

2. Related Work

2.1. Generating Adversarial Examples

Traditional machine learning algorithms are known to be vulnerable to adversarial examples [7, 14, 3]. Recently, Szegedy *et al*. [36] pointed out that CNNs are also fragile to adversarial examples, and proposed a box-constrained L-BFGS method to find adversarial examples reliably. Due to the expensive computation in [36], Goodfellow *et al*. [11] proposed the fast gradient sign method to generate adversarial examples efficiently by performing a single gradient step. This method was extended by Kurakin *et al*. [16] to an iterative version, and showed that the generated adversarial examples can exist in the physical world. Dong *et al*. [9] proposed a broad class of momentum-based iterative algorithms to boost the transferability of adversarial examples. The transferability can also be improved by attacking an ensemble of networks simultaneously [21]. Besides image classification, adversarial examples also exist in object detection [39], semantic segmentation [39, 6], speech recognition [6], deep reinforcement learning [20], etc.. Unlike adversarial examples which can be recognized by human, Nguyen *et al*. [25] generated fooling images that are different from natural images and difficult for human to recognize, but CNNs classify these images with high confidences.

Our proposed input diversity is also related to EOT [2]. These two works differ in several aspects: (1) we mainly focus on the challenging black-box setting while [2] focuses on the white-box setting; (2) our work aims at alleviating overfitting in adversarial attacks, while [2] aims at making adversarial examples robust to transformations, without any discussion of overfitting; and (3) we do not apply expectation step in each attack iteration, while "expectation" is the core idea in [2].

2.2. Defending Against Adversarial Examples

Conversely, many methods have been proposed recently to defend against adversarial examples. [11, 17] proposed to inject adversarial examples into the training data to increase the network robustness. Tramèr *et al*. [37] pointed out that such adversarially trained models still remain vulnerable to adversarial examples, and proposed ensemble adversarial training, which augments training data with perturbations transferred from other models, in order to improve the network robustness further. [38, 12] utilized randomized image transformations to inputs at inference time to mitigate adversarial effects. Dhillon *et al*. [8] pruned a random subset of activations according to their magnitude to enhance network robustness. Prakash *et al*. [27] proposed a framework which combines pixel deflection with soft wavelet denoising to defend against adversarial examples. [24, 33, 29] leveraged generative models to purify adversarial images by moving them back towards the distribution of clean images.

3. Methodology

Let X denote an image, and y^{true} denote the corresponding ground-truth label. We use θ to denote the network parameters, and $L(X, y^{\text{true}}; \theta)$ to denote the loss. To generate

the adversarial example, the goal is to maximize the loss $L(X+r, y^{\text{true}}; \theta)$ for the image X, under the constraint that the generated adversarial example $X^{\text{adv}} = X + r$ should look visually similar to the original image X and the corresponding predicted label $y^{\text{adv}} \neq y^{\text{true}}$. In this work, we use l_∞-norm to measure the perceptibility of adversarial perturbations, *i.e.*, $||r||_\infty \leq \epsilon$. The loss function is defined as

$$L(X, y^{true}; \theta) = -\mathbb{1}_{y^{true}} \cdot \log\left(\text{softmax}(l(X;\theta))\right), \quad (1)$$

where $\mathbb{1}_{y^{true}}$ is the one-hot encoding of the ground-truth y^{true} and $l(X;\theta)$ is the logits output. Note that all the baseline attacks have been implemented in the cleverhans library [26], which can be used directly for our experiments.

3.1. Family of Fast Gradient Sign Methods

In this section, we give an overview of the family of fast gradient sign methods.

Fast Gradient Sign Method (FGSM). FGSM [11] is the first member in this attack family, which finds the adversarial perturbations in the direction of the loss gradient $\nabla_X L(X, y^{\text{true}}; \theta)$. The update equation is

$$X^{\text{adv}} = X + \epsilon \cdot \text{sign}(\nabla_X L(X, y^{\text{true}}; \theta)). \quad (2)$$

Iterative Fast Gradient Sign Method (I-FGSM). Kurakin *et al.* [17] extended FGSM to an iterative version, which can be expressed as

$$X_0^{\text{adv}} = X \quad (3)$$
$$X_{n+1}^{\text{adv}} = \text{Clip}_X^{\epsilon}\left\{X_n^{adv} + \alpha \cdot \text{sign}(\nabla_X L(X_n^{adv}, y^{\text{true}}; \theta))\right\},$$

where Clip_X^{ϵ} indicates the resulting image are clipped within the ϵ-ball of the original image X, n is the iteration number and α is the step size.

Momentum Iterative Fast Gradient Sign Method (MI-FGSM). MI-FGSM [9] proposed to integrate the momentum term into the attack process to stabilize update directions and escape from poor local maxima. The updating procedure is similar to I-FGSM, with the replacement of Eq. (3) by:

$$g_{n+1} = \mu \cdot g_n + \frac{\nabla_X L(X_n^{adv}, y^{\text{true}}; \theta)}{||\nabla_X L(X_n^{adv}, y^{\text{true}}; \theta)||_1} \quad (4)$$
$$X_{n+1}^{\text{adv}} = \text{Clip}_X^{\epsilon}\left\{X_n^{adv} + \alpha \cdot \text{sign}(g_{n+1})\right\},$$

where μ is the decay factor of the momentum term and g_n is the accumulated gradient at iteration n.

3.2. Motivation

Let $\hat{\theta}$ denote the unknown network parameters. In general, a strong adversarial example should have high success rates on both white-box models, *i.e.*, $L(X^{adv}, y^{\text{true}}; \theta) > L(X, y^{\text{true}}; \theta)$, and black-box models, *i.e.*, $L(X^{adv}, y^{\text{true}}; \hat{\theta}) > L(X, y^{\text{true}}; \hat{\theta})$. On one hand, the traditional single-step attacks, *e.g.*, FGSM, tend to underfit to the specific network parameters θ due to inaccurate linear appropriation of the loss $L(X, y^{\text{true}}; \theta)$, thus cannot reach high success rates on white-box models. On the other hand, the traditional iterative attacks, *e.g.*, I-FGSM, greedily perturb the images in the direction of the sign of the loss gradient $\nabla_X L(X, y^{\text{true}}; \theta)$ at each iteration, and thus easily fall into the poor local maxima and overfit to the specific network parameters θ. These overfitted adversarial examples rarely transfer to black-box models. In order to generate adversarial examples with strong transferability, we need to find a better way to optimize the loss $L(X, y^{\text{true}}; \theta)$ to alleviate this overfitting phenomenon.

Data augmentation [15, 32, 13] is shown as an effective way to prevent networks from overfitting during the training process. Meanwhile, [38, 12] showed that adversarial examples are no longer malicious if simple image transformations are applied, which indicates these transformed adversarial images can serve as good samples for better optimization. Those facts inspire us to apply random and differentiable transformations to the inputs for the sake of the transferability of adversarial examples.

3.3. Diverse Input Patterns

Based on the analysis above, we aim at generating more transferable adversarial examples via diverse input patterns.

DI^2-FGSM. First, we propose the Diverse Inputs Iterative Fast Gradient Sign Method (DI^2-FGSM), which applies image transformations $T(\cdot)$ to the inputs with the probability p at each iteration of I-FGSM [17] to alleviate the overfitting phenomenon.

In this paper, we consider random resizing, which resizes the input images to a random size, and random padding, which pads zeros around the input images in a random manner [38], as the instantiation of the image transformations $T(\cdot)$[1]. The transformation probability p controls the trade-off between success rates on white-box models and success rates on black-box models, which can be observed from Fig. 4. If $p = 0$, DI^2-FGSM degrades to I-FGSM and leads to overfitting. If $p = 1$, *i.e.*, only transformed inputs are used for the attack, the generated adversarial examples tend to have much higher success rates on black-box models but lower success rates on white-box models, since the original inputs are not seen by the attackers.

In general, the updating procedure of DI^2-FGSM is similar to I-FGSM, with the replacement of Eq. (3) by

$$X_{n+1}^{\text{adv}} = \text{Clip}_X^{\epsilon}\{X_n^{adv} + \alpha \cdot \text{sign}\left(\nabla_X L(T(X_n^{adv}; p), y^{\text{true}}; \theta)\right)\}, \quad (5)$$

[1]We have also experimented with other image transformations, *e.g.*, rotation or flipping, to create diverse input patterns, and found random resizing & padding yields adversarial examples with the *best* transferability.

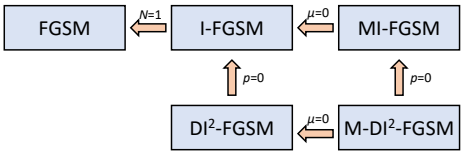

Figure 2. **Relationships between different attacks**. By setting setting values of the transformation probability p, the decay factor μ and the total iteration number N, we can relate these different attacks in the family of Fast Gradient Sign Methods.

where the stochastic transformation function $T(X_n^{adv}; p)$ is

$$T(X_n^{adv}; p) = \begin{cases} T(X_n^{adv}) & \text{with probability } p \\ X_n^{adv} & \text{with probability } 1-p \end{cases}. \quad (6)$$

M-DI^2-FGSM. Intuitively, momentum and diverse inputs are two completely different ways to alleviate the overfitting phenomenon. We can combine them naturally to form a much stronger attack, *i.e.*, Momentum Diverse Inputs Iterative Fast Gradient Sign Method (M-DI^2-FGSM). The overall updating procedure of M-DI^2-FGSM is similar to MI-FGSM, with the only replacement of Eq. (4) by

$$g_{n+1} = \mu \cdot g_n + \frac{\nabla_X L(T(X_n^{adv}; p), y^{\text{true}}; \theta)}{||\nabla_X L(T(X_n^{adv}; p), y^{\text{true}}; \theta)||_1}. \quad (7)$$

3.4. Relationships between Different Attacks

The attacks mentioned above all belong to the family of Fast Gradient Sign Methods, and they can be related via different parameter settings as shown in Fig. 2. To summarize,

- If the transformation probability $p = 0$, M-DI^2-FGSM degrades to MI-FGSM, and DI^2-FGSM degrades to I-FGSM.
- If the decay factor $\mu = 0$, M-DI^2-FGSM degrades to DI^2-FGSM, and MI-FGSM degrades to I-FGSM.
- If the total iteration number $N = 1$, I-FGSM degrades to FGSM.

3.5. Attacking an Ensemble of Networks

Liu *et al*. [21] suggested that attacking an ensemble of multiple networks simultaneously can generate much stronger adversarial examples. The motivation is that if an adversarial image remains adversarial for multiple networks, then it is more likely to transfer to other networks as well. Therefore, we can use this strategy to improve the transferability even further.

We follow the ensemble strategy proposed in [9], which fuse the logit activations together to attack multiple networks simultaneously. Specifically, to attack an ensemble of K models, the logits are fused by:

$$l(X; \theta_1, ..., \theta_K) = \sum_{k=1}^{K} w_k l_k(X; \theta_k) \quad (8)$$

where $l_k(X; \theta_k)$ is the logits output of the k-th model with the parameters θ_k, w_k is the ensemble weight with $w_k \geq 0$ and $\sum_{k=1}^{K} w_k = 1$.

4. Experiment

4.1. Experiment Setup

Dataset. It is less meaningful to attack the images that are already classified wrongly. Therefore, we randomly choose 5000 images from the ImageNet validation set that are classified correctly by all the networks which we test on, to form our test dataset. All these images are resized to $299 \times 299 \times 3$ beforehand.

Networks. We consider four normally trained networks, *i.e.*, Inception-v3 (Inc-v3) [35], Inception-v4 (Inc-v4) [34], Resnet-v2-152 (Res-152) [13] and Inception-Resnet-v2 (IncRes-v2) [34], and three adversarially trained networks [37], *i.e.*, ens3-adv-Inception-v3 (Inc-v3$_{ens3}$), ens4-adv-Inception-v3 (Inc-v3$_{ens4}$) and ens-adv-Inception-ResNet-v2 (IncRes-v2$_{ens}$). All networks are publicly available[2,3].

Implementation details. For the parameters of different attackers, we follow the default settings in [16] with the step size $\alpha = 1$ and the total iteration number $N = \min(\epsilon + 4, 1.25\epsilon)$. We set the maximum perturbation of each pixel to be $\epsilon = 15$, *which is still imperceptible for human observers* [23]. For the momentum term, decay factor μ is set to be 1 as in [9]. For the stochastic transformation function $T(X; p)$, the probability p is set to be 0.5, *i.e.*, attackers put equal attentions on the original inputs and the transformed inputs. For transformation operations $T(\cdot)$, the input X is first randomly resized to a $rnd \times rnd \times 3$ image, with $rnd \in [299, 330)$, and then padded to the size $330 \times 330 \times 3$ in a random manner.

4.2. Attacking a Single Network

We first perform adversarial attacks on a single network. We craft adversarial examples only on normally trained networks, and test them on all seven networks. The success rates are shown in Table 1, where the diagonal blocks indicate white-box attacks and off-diagonal blocks indicate black-box attacks. We list the networks that we attack on in rows, and networks that we test on in columns.

From Table 1, a first glance shows that M-DI^2-FGSM outperforms all other baseline attacks by a large margin on all black-box models, and maintains high success rates on all white-box models. For example, if adversarial examples

[2] https://github.com/tensorflow/models/tree/master/research/slim

[3] https://github.com/tensorflow/models/tree/master/research/adv_imagenet_models

Model	Attack	Inc-v3	Inc-v4	IncRes-v2	Res-152	Inc-v3$_{ens3}$	Inc-v3$_{ens4}$	IncRes-v2$_{ens}$
Inc-v3	FGSM	64.6%	23.5%	21.7%	21.7%	8.0%	7.5%	3.6%
	I-FGSM	**99.9%**	14.8%	11.6%	8.9%	3.3%	2.9%	1.5%
	DI^2-FGSM (**Ours**)	**99.9%**	35.5%	27.8%	21.4%	5.5%	5.2%	2.8%
	MI-FGSM	**99.9%**	36.6%	34.5%	27.5%	8.9%	8.4%	4.7%
	M-DI^2-FGSM (**Ours**)	**99.9%**	**63.9%**	**59.4%**	**47.9%**	**14.3%**	**14.0%**	**7.0%**
Inc-v4	FGSM	26.4%	49.6%	19.7%	20.4%	8.4%	7.7%	4.1%
	I-FGSM	22.0%	**99.9%**	13.2%	10.9%	3.2%	3.0%	1.7%
	DI^2-FGSM (**Ours**)	43.3%	99.7%	28.9%	23.1%	5.9%	5.5%	3.2%
	MI-FGSM	51.1%	**99.9%**	39.4%	33.7%	11.2%	10.7%	5.3%
	M-DI^2-FGSM (**Ours**)	**72.4%**	99.5%	**62.2%**	**52.1%**	**17.6%**	**15.6%**	**8.8%**
IncRes-v2	FGSM	24.3%	19.3%	39.6%	19.4%	8.5%	7.3%	4.8%
	I-FGSM	22.2%	17.7%	97.9%	12.6%	4.6%	3.7%	2.5%
	DI^2-FGSM (**Ours**)	46.5%	40.5%	95.8%	28.6%	8.2%	6.6%	4.8%
	MI-FGSM	53.5%	45.9%	**98.4%**	37.8%	15.3%	13.0%	8.8%
	M-DI^2-FGSM (**Ours**)	**71.2%**	**67.4%**	96.1%	**57.4%**	**25.1%**	**20.7%**	**14.9%**
Res-152	FGSM	34.4%	28.5%	27.1%	75.2%	12.4%	11.0%	6.0%
	I-FGSM	20.8%	17.2%	14.9%	99.1%	5.4%	4.6%	2.8%
	DI^2-FGSM (**Ours**)	53.8%	49.0%	44.8%	**99.2%**	13.0%	11.1%	6.9%
	MI-FGSM	50.1%	44.1%	42.2%	99.0%	18.2%	15.2%	9.0%
	M-DI^2-FGSM (**Ours**)	**78.9%**	**76.5%**	**74.8%**	**99.2%**	**35.2%**	**29.4%**	**19.0%**

Table 1. **The success rates on seven networks where we attack a single network**. The diagonal blocks indicate white-box attacks, while the off-diagonal blocks indicate black-box attacks which are much more challenging. Experiment results demonstrate that our proposed input diversity strategy substantially improve the transferability of generated adversarial examples.

Figure 3. **Visualization of randomly selected clean images and their corresponding adversarial examples**. All these adversarial examples are generated on Inception-v3 using our proposed DI^2-FGSM with the maximum perturbation of each pixel $\epsilon = 15$.

are crafted on IncRes-v2, M-DI^2-FGSM has success rates of 67.4% on Inc-v4 (normally trained black-box model) and 25.1% on Inc-v3$_{ens3}$ (adversarially trained black-box model), while strong baselines like MI-FGSM only obtains the corresponding success rates of 45.9% and 15.3%, respectively. This convincingly demonstrates the effectiveness of the combination of input diversity and momentum for improving the transferability of adversarial examples.

We then compare the success rates of I-FGSM and DI^2-FGSM to see the effectiveness of diverse input patterns solely. By generating adversarial examples with input diversity, DI^2-FGSM significantly improves the success rates of I-FGSM on challenging black-box models, regardless whether this model is adversarially trained, and maintains high success rates on white-box models. For example, if adversarial examples are crafted on Res-152, DI^2-FGSM has success rates of 99.2% on Res-152 (white-box model), 53.8% on Inc-v3 (normally trained black-box model) and 11.1% on Inc-v3$_{ens4}$ (adversarially trained black-box model), while I-FGSM only obtains the corresponding success rates of 99.1%, 20.8% and 4.6%, respectively. Compared with FGSM, DI^2-FGSM also reaches much higher success rates on the normally trained black-box models, and comparable performance on the adversarially trained black-box models. Besides, we visualize 5 randomly selected pairs of such generated adversarial images and their clean counterparts in Figure 3. These visualization results show that these generated adversarial perturbations are human imperceptible.

It should be mentioned that the proposed input diversity is not merely applicable to fast gradient sign methods. To demonstrate the generalization, we also incorporate C&W attack [4] with input diversity. The experiment is conducted on 1000 correctly classified images. For the parameters of C&W, the maximal iteration is 250, the learning rate is 0.01 and the confidence is 10. As Table 2 suggests, our method D-C&W obtains a significant performance improvement over C&W on black-box models.

4.3. Attacking an Ensemble of Networks

Though the results in Table 1 show that momentum and input diversity can significantly improve the transferability of adversarial examples, they are still relatively weak at attacking an adversarially trained network under the black-box setting, *e.g.*, the highest black-box success rate on IncRes-v2$_{ens}$ is only 19.0%. Therefore, we follow the strat-

Model	Attack	Inc-v3	Inc-v4	IncRes-v2	Res-152	Inc-v3$_{ens3}$	Inc-v3$_{ens4}$	IncRes-v2$_{ens}$
Inc-v3	C&W	**100.0%**	5.7%	5.3%	5.1%	3.0%	2.5%	1.1%
	D-C&W (**Ours**)	**100.0%**	**16.8%**	**13.0%**	**11.2%**	**5.8%**	**3.9%**	**2.1%**
Inc-v4	C&W	15.1%	**100.0%**	9.2%	7.8%	4.4%	3.5%	1.9%
	D-C&W (**Ours**)	**29.3%**	**100.0%**	**20.1%**	**15.4%**	**7.1%**	**5.3%**	**3.1%**
IncRes-v2	C&W	15.8%	11.2%	99.9%	8.6%	6.3%	3.6%	3.4%
	D-C&W (**Ours**)	**33.9%**	**25.6%**	**100.0%**	**19.4%**	**11.2%**	**7.3%**	**4.0%**
Res-152	C&W	11.4%	6.9%	6.1%	**100.0%**	4.4%	4.1%	2.3%
	D-C&W (**Ours**)	**33.0%**	**27.7%**	**24.4%**	**100.0%**	**13.1%**	**9.3%**	**5.7%**

Table 2. **The success rates on seven networks where we attack a single network using C&W attack**. Experiment results demonstrate that the proposed input diversity strategy can enhance C&W attack for generating more transferable adversarial examples.

Model	Attack	-Inc-v3	-Inc-v4	-IncRes-v2	-Res-152	-Inc-v3$_{ens3}$	-Inc-v3$_{ens4}$	-IncRes-v2$_{ens}$
Ensemble	I-FGSM	96.6%	**96.9%**	98.7%	96.2%	**97.0%**	**97.3%**	94.3%
	DI^2-FGSM (**Ours**)	88.9%	89.6%	93.2%	87.7%	91.7%	91.7%	93.2%
	MI-FGSM	**96.9%**	**96.9%**	**98.8%**	**96.8%**	96.8%	97.0%	94.6%
	M-DI^2-FGSM (**Ours**)	90.1%	91.1%	94.0%	89.3%	92.8%	92.7%	**94.9%**
Hold-out	I-FGSM	43.7%	36.4%	33.3%	25.4%	12.9%	15.1%	8.8%
	DI^2-FGSM (**Ours**)	69.9%	67.9%	64.1%	51.7%	36.3%	35.0%	30.4%
	MI-FGSM	71.4%	65.9%	64.6%	55.6%	22.8%	26.1%	15.8%
	M-DI^2-FGSM (**Ours**)	**80.7%**	**80.6%**	**80.7%**	**70.9%**	**44.6%**	**44.5%**	**39.4%**

Table 3. **The success rates of ensemble attacks**. Adversarial examples are generated on an ensemble of six networks, and tested on the ensembled network (*white-box setting*) and the hold-out network (*black-box setting*). The sign "-" indicates the hold-out network. We observe that the proposed M-DI^2-FGSM significantly outperforms *all* other attacks on *all* black-box models.

egy in [21] to attack multiple networks simultaneously in order to further improve transferability. We consider all seven networks here. Adversarial examples are generated on an ensemble of six networks, and tested on the ensembled network and the hold-out network, using I-FGSM, DI^2-FGSM, MI-FGSM and M-DI^2-FGSM, respectively. FGSM is ignored here due to its low success rates on white-box models. All ensembled models are assigned with equal weight, *i.e.*, $w_k = 1/6$.

The results are summarized in Table 3, where the top row shows the success rates on the ensembled network (white-box setting), and the bottom row shows the success rates on the hold-out network (black-box setting). Under the challenging black-box setting, we observe that M-DI^2-FGSM always generates adversarial examples with better transferability than other methods on all networks. For example, by keeping Inc-v3$_{ens3}$ as a hold-out model, M-DI^2-FGSM can fool Inc-v3$_{ens3}$ with an success rate of 44.6%, while I-FGSM, DI^2-FGSM and MI-FGSM only have success rates of 12.9%, 36.3% and 22.8%, respectively. Besides, compared with MI-FGSM, we observe that using diverse input patterns alone, *i.e.*, DI^2-FGSM, can reach a much higher success rate if the hold-out model is an adversarially trained network, and a comparable success rate if the hold-out model is a normally trained network.

Under the white-box setting, we see that DI^2-FGSM and M-DI^2-FGSM reach slightly lower (but still very high) success rates on ensemble models compared with I-FGSM and MI-FGSM. This is due to the fact that attacking multiple networks simultaneously is much harder than attacking a single model. However, the white-box success rates can be improved if we assign the transformation probability p with a smaller value, increase the number of total iteration N or use a smaller step size α (see Sec. 4.4).

4.4. Ablation Studies

In this section, we conduct a series of ablation experiments to study the impact of different parameters. We only consider attacking an ensemble of networks here, since it is much stronger than attacking a single network and can provide a more accurate evaluation of the network robustness. The max perturbation of each pixel ϵ is set to 15 for all experiments.

Transformation probability p. We first study the influence of the transformation probability p on the success rates under both white-box and black-box settings. We set the step size $\alpha = 1$ and the total iteration number $N = \min(\epsilon + 4, 1.25\epsilon)$. The transformation probability p varies from 0 to 1. Recall the relationships shown in Fig. 2, M-DI^2-FGSM (or DI^2-FGSM) degrades to MI-FGSM (or I-FGSM) if $p = 0$.

We show the success rates on various networks in Fig. 4. We observe that both DI^2-FGSM and M-DI^2-FGSM achieve a higher black-box success rates but lower white-box success rates as p increase. Moreover, for all attacks, if p is small, *i.e.*, only a small amount of transformed inputs are utilized, black-box success rates can increase significantly, while white-box success rates only drop a little. This phenomenon reveals the importance of adding transformed inputs into the attack process.

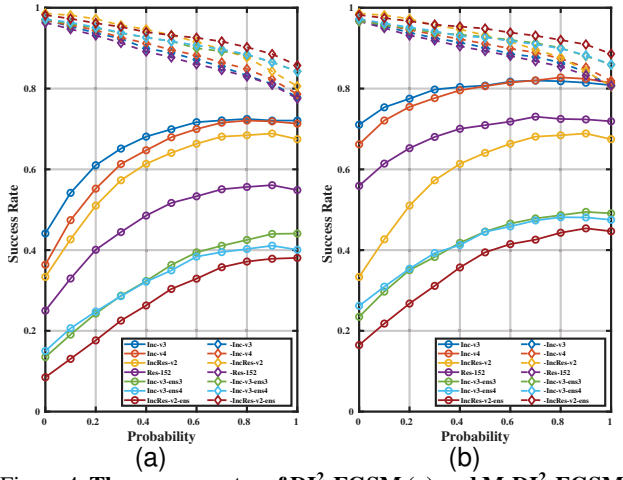

Figure 4. **The success rates of DI^2-FGSM (a) and M-DI^2-FGSM (b) when varying the transformation probability** p. "Ensemble" (white-box setting) is with dashed lines and "Hold-out" (black-box setting) is with solid lines.

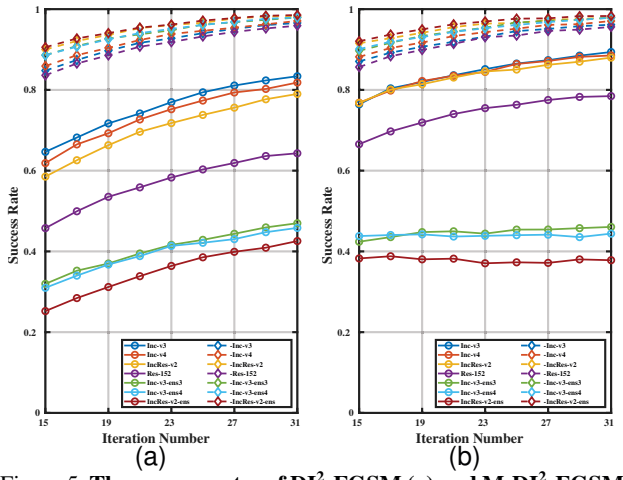

Figure 5. **The success rates of DI^2-FGSM (a) and M-DI^2-FGSM (b) when varying the total iteration number** N. "Ensemble" (white-box setting) is with dashed lines and "Hold-out" (black-box setting) is with solid lines.

The trends shown in Fig. 4 also provide useful suggestions of constructing strong adversarial attacks in practice. For example, if you know the black-box model is a new network that totally different from any existing networks, you can set $p = 1$ to reach the maximum transferability. If the black-box model is a mixture of new networks and existing networks, you can choose a moderate value of p to maximize the black-box success rates under a pre-defined white-box success rates, *e.g.*, white-box success rates must greater or equal than 90%.

Total iteration number N**.** We then study the influence of the total iteration number N on the success rates under both white-box and black-box settings. We set the transformation probability $p = 0.5$ and the step size $\alpha = 1$. The total iteration number N varies from 15 to 31, and the results are plotted in Fig. 5. For DI^2-FGSM, we see that the black-box success rates and white-box success rates always increase as the total iteration number N increase. Similar trends can also be observed for M-DI^2-FGSM except for the black-box success rates on adversarially trained models, *i.e.*, performing more iterations cannot bring extra transferability on adversarially trained models. Moreover, we observe that the success rates gap between M-DI^2-FGSM and DI^2-FGSM is diminished as N increases.

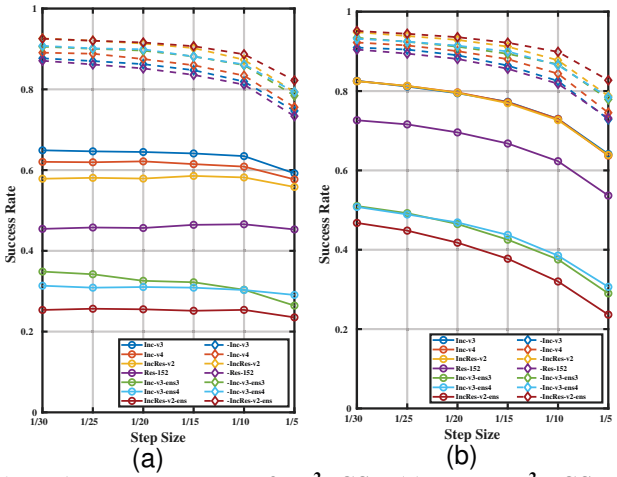

Figure 6. **The success rates of DI ^{2}FGSM (a) and M-DI^2-FGSM (b) when varying the step size** α. "Ensemble" (white-box setting) is with dashed lines and "Hold-out" (black-box setting) is with solid lines.

Step size α**.** We finally study the influence of the step size α on the success rates under both white-box and black-box settings. We set the transformation probability $p = 0.5$. In order to reach the maximum perturbation ϵ even for a small step size α, we set the total iteration number be proportional to the step size, *i.e.*, $N = \epsilon/\alpha$. The results are plotted in Fig. 6. We observe that the white-box success rates of both DI^2-FGSM and M-DI^2-FGSM can be boosted if a smaller step size is provided. Under the black-box setting, the success rates of DI^2-FGSM is insensitive to the step size, while the success rates of M-DI^2-FGSM can still be improved with smaller step size.

4.5. NIPS 2017 Adversarial Competition

In order to verify the effectiveness of our proposed attack methods in practice, we here reproduce the top defense entries and official baselines from NIPS 2017 adversarial competition [18] for testing transferability. Due to the resource limitation, we only consider the top-3 defense entries, *i.e.*, *TsAIL* [19], *iyswim* [38] and *Anil Thomas*[4], as well 3 official baselines, *i.e.*, Inc-v3$_{adv}$, IncRes-v2$_{ens}$ and Inc-v3.

[4] https://github.com/anlthms/nips-2017/tree/master/mmd

Attack	TsAIL	iyswim	Anil Thomas	Inc-v3$_{adv}$	IncRes-v2$_{ens}$	Inc-v3	**Average**
I-FGSM	14.0%	35.6%	30.9%	**98.2%**	**96.4%**	**99.0%**	62.4%
DI^2-FGSM (**Ours**)	**22.7%**	58.4%	48.0%	91.5%	90.7%	97.3%	68.1%
MI-FGSM	14.9%	45.7%	46.6%	97.3%	95.4%	98.7%	66.4%
MI-FGSM*	13.6%	43.2%	43.9%	94.4%	93.0%	97.3%	64.2%
M-DI^2-FGSM (**Ours**)	20.0%	**69.8%**	**64.4%**	93.3%	92.4%	97.9%	**73.0%**

Table 4. **The success rates on top defense solutions and official baselines from NIPS 2017 adversarial competition [18]**. * indicates the official results reported in the competition. Our proposed M-DI^2-FGSM reaches an average success rate of 73.0%, which outperforms the top-1 attack submission in the NIPS competition by a large margin of 6.6%.

We note that the No.1 solution and the No.3 solution apply significantly different image transformations (compared to random resizing & padding used in our attack method) for defending against adversarial examples. For example, the No.1 solution, *TsAIL*, applies an image denoising network for removing adversarial perturbations, and the No.3 solution, *Anil Thomas*, includes a series of image transformations, *e.g.*, JPEG compression, rotation, shifting and zooming, in the defense pipeline. The test dataset contains 5000 images which are all of the size $299 \times 299 \times 3$, and their corresponding labels are the same as the ImageNet labels.

Generating adversarial examples. When generating adversarial examples, we follow the procedure in [18]: (1) split the dataset equally into 50 batches; (2) for each batch, the maximum perturbation ϵ is randomly chosen from the set $\{\frac{4}{255}, \frac{8}{255}, \frac{12}{255}, \frac{16}{255}\}$; and (3) generate adversarial examples for each batch under the corresponding ϵ constraint.

Attacker settings. For the settings of attackers, we follow [9] by attacking an ensemble eight diferent models, *i.e.*, Inc-v3, Inc-v4, IncRes-v2, Res-152, Inc-v3$_{ens3}$, Inc-v3$_{ens4}$, IncRes-v2$_{ens}$ and Inc-v3$_{adv}$ [17]. The ensemble weights are set as $1/7.25$ equally for the first seven models and $0.25/7.25$ for Inc-v3$_{adv}$. The total iteration number N is 10 and the decay factor μ is 1. This configuration for MI-FGSM won the 1-st place in the NIPS 2017 adversarial attack competition. For DI^2-FGSM and M-DI^2-FGSM, we choose $p = 0.4$ according to the trends shown in Fig. 4.

Results. The results are summarized in Table 4. We also report the official results of MI-FGSM (named MI-FGSM*) as a reference to validate our implementation. The performance difference between MI-FGSM and MI-FGSM* is due to the randomness of the max perturbation magnitude introduced in the attack process. Compared with MI-FGSM, DI^2-FGSM have higher success rates on top defense solutions while slightly lower success rates on baseline models, which results in these two attack methods having similar average success rates. By integrating both diverse inputs and momentum term, this enhanced attack, M-DI^2-FGSM, reaches an average success rate of 73.0%, which is far better than other methods. For example, the top-1 attack submission, MI-FGSM, in the NIPS competition only gets an average success rate of 66.4%. We believe this superior transferability can also be observed on other defense submissions which we do not evaluate on.

4.6. Discussion

We provide a brief discussion of why the proposed diverse input patterns can help to generate adversarial examples with better transferability. One hypothesis is that the decision boundaries of different networks share similar inherent structures due to the same training dataset, *e.g.*, ImageNet. For example, as shown in Fig 1, different networks make similar mistakes in the presence of adversarial examples. By incorporating diverse patterns at each attack iteration, the optimization produces adversarial examples that are more robust to small transformations. These adversarial examples are malicious in a certain region at the network decision boundary, thus increasing the chance to fool other networks, *i.e.*, they achieve better black-box success rate than existing methods. In the future, we plan to validate this hypothesis theoretically or empirically.

5. Conclusions

In this paper, we propose to improve the transferability of adversarial examples with input diversity. Specifically, our method applies random transformations to the input images at each iteration in the attack process. Compared with traditional iterative attacks, the results on ImageNet show that our proposed attack method gets significantly higher success rates for black-box models, and maintains similar success rates for white-box models. We improve the transferability further by integrating momentum term and attacking multiple networks simultaneously. By evaluating this enhanced attack against the top defense submissions and official baselines from NIPS 2017 adversarial competition [18], we show that this enhanced attack reaches an average success rate of 73.0%, which outperforms the top-1 attack submission in the NIPS competition by a large margin of 6.6%. We hope that our proposed attack strategy can serve as a benchmark for evaluating the robustness of networks to adversaries and the effectiveness of different defense methods in future. Code is publicly available at https://github.com/cihangxie/DI-2-FGSM.

Acknowledgement: This work was supported by a gift grant from YiTu and ONR N00014-12-1-0883.

References

[1] A. Arnab, O. Miksik, and P. H. Torr. On the robustness of semantic segmentation models to adversarial attacks. *arXiv preprint arXiv:1711.09856*, 2017. 1

[2] A. Athalye, L. Engstrom, A. Ilyas, and K. Kwok. Synthesizing robust adversarial examples. In *International Conference on Machine Learning*, pages 284–293, 2018. 2

[3] B. Biggio, I. Corona, D. Maiorca, B. Nelson, N. Šrndić, P. Laskov, G. Giacinto, and F. Roli. Evasion attacks against machine learning at test time. In *Joint European conference on machine learning and knowledge discovery in databases*, pages 387–402, 2013. 2

[4] N. Carlini and D. Wagner. Towards evaluating the robustness of neural networks. In *IEEE Symposium on Security and Privacy*, 2017. 5

[5] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017. 1

[6] M. Cisse, Y. Adi, N. Neverova, and J. Keshet. Houdini: Fooling deep structured prediction models. *arXiv preprint arXiv:1707.05373*, 2017. 2

[7] N. Dalvi, P. Domingos, S. Sanghai, D. Verma, et al. Adversarial classification. In *ACM SIGKDD international conference on Knowledge discovery and data mining*, 2004. 2

[8] G. S. Dhillon, K. Azizzadenesheli, J. D. Bernstein, J. Kossaifi, A. Khanna, Z. C. Lipton, and A. Anandkumar. Stochastic activation pruning for robust adversarial defense. In *International Conference on Learning Representations*, 2018. 2

[9] Y. Dong, F. Liao, T. Pang, H. Su, X. Hu, J. Li, and J. Zhu. Boosting adversarial attacks with momentum. *arXiv preprint arXiv:1710.06081*, 2017. 2, 3, 4, 8

[10] R. Girshick. Fast r-cnn. In *International Conference on Computer Vision*, 2015. 1

[11] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015. 1, 2, 3

[12] C. Guo, M. Rana, M. Cissé, and L. van der Maaten. Countering adversarial images using input transformations. In *International Conference on Learning Representations*, 2018. 2, 3

[13] K. He, X. Zhang, S. Ren, and J. Sun. Identity mappings in deep residual networks. In *European Conference on Computer Vision*, 2016. 1, 2, 3, 4

[14] L. Huang, A. D. Joseph, B. Nelson, B. I. Rubinstein, and J. Tygar. Adversarial machine learning. In *ACM workshop on Security and artificial intelligence*, 2011. 2

[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, 2012. 1, 2, 3

[16] A. Kurakin, I. Goodfellow, and S. Bengio. Adversarial examples in the physical world. In *International Conference on Learning Representations Workshop*, 2017. 1, 2, 4

[17] A. Kurakin, I. Goodfellow, and S. Bengio. Adversarial machine learning at scale. In *International Conference on Learning Representations*, 2017. 1, 2, 3, 8

[18] A. Kurakin, I. Goodfellow, S. Bengio, Y. Dong, F. Liao, M. Liang, T. Pang, J. Zhu, X. Hu, C. Xie, et al. Adversarial attacks and defences competition. *arXiv preprint arXiv:1804.00097*, 2018. 2, 7, 8

[19] F. Liao, M. Liang, Y. Dong, T. Pang, X. Hu, and J. Zhu. Defense against adversarial attacks using high-level representation guided denoiser. In *Computer Vision and Pattern Recognition*, 2018. 7

[20] Y.-C. Lin, Z.-W. Hong, Y.-H. Liao, M.-L. Shih, M.-Y. Liu, and M. Sun. Tactics of adversarial attack on deep reinforcement learning agents. In *International Joint Conference on Artificial Intelligence*, 2017. 2

[21] Y. Liu, X. Chen, C. Liu, and D. Song. Delving into transferable adversarial examples and black-box attacks. In *International Conference on Learning Representations*, 2017. 2, 4, 6

[22] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Computer Vision and Pattern Recognition*, 2015. 1

[23] Y. Luo, X. Boix, G. Roig, T. Poggio, and Q. Zhao. Foveation-based mechanisms alleviate adversarial examples. *arXiv preprint arXiv:1511.06292*, 2015. 4

[24] D. Meng and H. Chen. Magnet: a two-pronged defense against adversarial examples. *arXiv preprint arXiv:1705.09064*, 2017. 2

[25] A. Nguyen, J. Yosinski, and J. Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Computer Vision and Pattern Recognition*, 2015. 2

[26] N. Papernot, F. Faghri, N. Carlini, I. Goodfellow, R. Feinman, A. Kurakin, C. Xie, Y. Sharma, T. Brown, A. Roy, A. Matyasko, V. Behzadan, K. Hambardzumyan, Z. Zhang, Y.-L. Juang, Z. Li, R. Sheatsley, A. Garg, J. Uesato, W. Gierke, Y. Dong, D. Berthelot, P. Hendricks, J. Rauber, and R. Long. cleverhans v2.1.0: an adversarial machine learning library. *arXiv preprint arXiv:1610.00768*, 2018. 3

[27] A. Prakash, N. Moran, S. Garber, A. DiLillo, and J. Storer. Deflecting adversarial attacks with pixel deflection. *arXiv preprint arXiv:1801.08926*, 2018. 2

[28] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, 2015. 1

[29] P. Samangouei, M. Kabkab, and R. Chellappa. Defense-GAN: Protecting classifiers against adversarial attacks using generative models. In *International Conference on Learning Representations*, 2018. 2

[30] A. Shrivastava, A. Gupta, and R. Girshick. Training region-based object detectors with online hard example mining. In *Computer Vision and Pattern Recognition*, 2016. 2

[31] E. Simo-Serra, E. Trulls, L. Ferraz, I. Kokkinos, P. Fua, and F. Moreno-Noguer. Discriminative learning of deep convolutional feature point descriptors. In *International Conference on Computer Vision*, 2015. 2

[32] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015. 1, 2, 3

[33] Y. Song, T. Kim, S. Nowozin, S. Ermon, and N. Kushman. Pixeldefend: Leveraging generative models to understand and defend against adversarial examples. *arXiv preprint arXiv:1710.10766*, 2017. 2

[34] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI*, 2017. 4

[35] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *Computer Vision and Pattern Recognition*, 2016. 4

[36] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations*, 2014. 1, 2

[37] F. Tramèr, A. Kurakin, N. Papernot, D. Boneh, and P. McDaniel. Ensemble adversarial training: Attacks and defenses. *arXiv preprint arXiv:1705.07204*, 2017. 1, 2, 4

[38] C. Xie, J. Wang, Z. Zhang, Z. Ren, and A. Yuille. Mitigating adversarial effects through randomization. In *International Conference on Learning Representations*, 2018. 2, 3, 7

[39] C. Xie, J. Wang, Z. Zhang, Y. Zhou, L. Xie, and A. Yuille. Adversarial Examples for Semantic Segmentation and Object Detection. In *International Conference on Computer Vision*, 2017. 2

[40] Z. Zhang, S. Qiao, C. Xie, W. Shen, B. Wang, and A. L. Yuille. Single-shot object detection with enriched semantics. *arXiv preprint arXiv:1712.00433*, 2017. 1