Adversarial Examples for Edge Detection: They Exist, and They Transfer

Christian Cosgrove Alan L. Yuille
Department of Computer Science, The Johns Hopkins University
Baltimore, MD 21218 USA
ccosgro2@jhu.edu alan.l.yuille@gmail.com

Abstract

Convolutional neural networks have recently advanced the state of the art in many tasks including edge and object boundary detection. However, in this paper, we demonstrate that these edge detectors inherit a troubling property of neural networks: they can be fooled by adversarial examples. We show that adding small perturbations to an image causes HED [42], a CNN-based edge detection model, to fail to locate edges, to detect nonexistent edges, and even to hallucinate arbitrary configurations of edges. More importantly, we find that these adversarial examples blindly transfer to other CNN-based vision models. In particular, attacks on edge detection result in significant drops in accuracy in models trained to perform unrelated, high-level tasks like image classification and semantic segmentation.

1. Introduction

Edge and contour detection have long played a major role in computer vision. First studied as a low-level function of biological vision [21, 35], the notion that edge detection can be used to filter out irrelevant lighting and texture information and extract shape information from images dates back to early work in the field [18, 22, 6]. Edge detection has been used as a pre-processing step in many classical vision algorithms [9, 44, 34, 4].

The history of edge detection is substantial, and a wide variety of techniques have been developed. Early approaches used hand-crafted features [22, 6]. Later, data-driven methods like [23, 9] emerged, in which some set of model parameters is automatically tuned on a training dataset in order to reduce false positives. Most recently, convolutional neural networks (CNNs) have been applied to the edge detection problem [36, 42, 5, 28]. One major success of this line of research is Holistically-Nested Edge Detection (HED), a CNN model that achieves near-human edge detection accuracy on standard datasets [42].

Figure 1: Adding a small perturbation (right) to an image causes a state-of-the-art edge detection model to produce a contrived pattern. The same perturbation causes a VGG16 model to misclassify the image (changing the predicted class label from "bighorn sheep" to "Indian elephant"). We set y^{target} to achieve the edge pattern above. Here, $\epsilon = 8$.

This approach has attracted attention for its competitive performance, architectural simplicity, and computational efficiency.

In recent years, automatic feature learning by CNNs has replaced explicit edge detection for higher-level vision tasks like image classification. However, it is well known that CNNs learn edge-like features implicitly [24]. The Gabor-like filters learned by the earliest layers of CNNs emerge

regardless of which dataset or task they are trained on [43]. In this sense, edge detection is a universal visual task that continues to underlie modern vision systems, albeit implicitly.

Despite CNNs' marked gains in accuracy over classical techniques in domains like classification and semantic segmentation, they are vulnerable to *adversarial examples*. In a variety of tasks [38, 40], small perturbations that look like noise to a human can cause the network to produce nonsensical results. In many cases, an attacker can select this perturbation to cause the network to produce any desired output. Worse, some attacks transfer: the same perturbation trained to fool one network sometimes fools similar networks trained on slightly different datasets.

However, it has not yet been shown whether these adversarial examples are limited to networks trained on "complex" visual tasks like classification and semantic segmentation, or whether even a CNN trained to perform a low-level task like edge detection is vulnerable. In this paper, we address this question by investigating the degree to which HED suffers from adversarial examples. Adapting existing methods to HED, we find that it is indeed vulnerable to a particular class of adversarial attacks. Altogether, the following results add yet another example to the list of domains where deep neural networks can be fooled.

Just as edge detection is a universal component of many methods in computer vision, we find that adversarial examples for edge detection affect other models, too: *they transfer to higher-level tasks*. In particular, we show that an attack on edges can transfer to models regardless of architecture, training data, and visual task. Without knowing the parameters of a vision model, we can impair that model's accuracy on an image by attacking the edges of the image. The intuition behind these results should be clear: because edge detection is used in CNNs for downstream processing, the CNN will fail to perform higher-level tasks if we can obfuscate these edges.

2. Related work

Adversarial examples have primarily been studied in the context of image classification [38, 14, 26]. However, they have also been found to affect networks for object detection [40], semantic segmentation [40, 12], and natural language processing [1]. Apart from finding new domains in which adversarial examples exist, much of recent research has focused on devising generic algorithms for generating adversarial examples—*i.e.*, how to synthesize them efficiently and how to improve their success rates. The first work of this kind uses a L-BFGS optimizer to minimize the size of the perturbation subject to the constraint that the network produces the target output [38]. The prevalent fast gradient sign method (FGSM) [14] exploits the linearity of the loss function landscape to generate adversarial examples with only first-order information and a single pass of backpropagation. This method has been improved by iterated updates [26] and momentum [10]. The literature on defending against these adversarial examples is as rich as the study of the attacks themselves; prominent examples are defensive distillation [31], input transformations [15], and adversarial training [38, 29].

While we are the first to develop adversarial attacks for edge detection models, others have investigated the relationship between adversarial attacks and edge information. Harmonic Adversarial Attack Method [17] considers the relationship between edge information and attack quality and transferability. The goal of this work is to maximize the smoothness of the perturbation so that the high-frequency statistics of the image change as little as possible.

Black-box attacks and transferability have been the subject of extensive study since [38]. In the black-box setting, the attacker does not have access to the model parameters and architecture; however, the model can be queried to generate an attack. An attack *transfers* if it affects a different model without access to parameters, architecture, or input-output pairs. One approach to generating black-box adversarial examples is to attack a surrogate model trained to mimic outputs from the target model [30]. Another is to train a separate network to generate perturbations [32, 3]. Finally, other work studies the transferability of attacks on intermediate layers [19].

3. Methods

3.1. Holistically-Nested Edge Detection [42]

Like many recent models for semantic segmentation, HED uses a fully-convolutional architecture [42]. This means that all of the network's parameters consist of convolution kernels; for this reason, the model is agnostic to input size. HED's convolutional layers are derived from a pretrained VGG16 [37] model and are fine-tuned on the Berkeley Segmentation Data Set (BSDS500) [2]. A multi-scale architecture and deep supervision are two crucial aspects of the HED method. In particular, HED outputs edge predictions from five different layers of the network, each corresponding to a different scale. During training, each of these *side outputs* is encouraged to match the ground-truth edge map [42].

In this paper, we show that despite HED's impressive performance on in-distribution images, this model is easily fooled by adversarial examples. Just like neural network training, the choice of loss function strongly affects the results of an adversarial attack. This is because adversarial attacks are formulated as an optimization problem in the space of images; like learning, generating adversarial examples also uses backpropagation to compute gradients of the loss. Our attack methods optimize a similar cross-entropy

loss to that of HED, except for one crucial difference. Consider the loss for side output m:

$$\ell^m(X, y^{\text{true}}; \theta) = -\frac{1}{2} \sum_{i\,:\,y_i^{\text{true}}=1} \log(\hat{y}_i^m) \\ -\frac{1}{2} \sum_{i\,:\,y_i^{\text{true}}=0} \log(1-\hat{y}_i^m). \tag{1}$$

Here, $\hat{y}_i^m$ denotes the ith pixel of side output m, which is a function of X and θ. Unlike HED, we *do not* weigh edges ($y_i^{\text{true}} = 1$) more strongly than non-edges ($y_i^{\text{true}} = 0$). Instead, the positive and negative classes are penalized equally. This enables additional types of attacks. In particular, in the class-balanced formulation of HED, using $y^{\text{true}} = \mathbf{1}$ causes the first term to vanish, since it is proportional to the number of non-edges in the ground truth y^{true}. This prevents the attack from generating new edges in the image, making so-called *edge activation* attacks impossible. Thus, we use a 1:1 class weighting for all attacks.

Like HED, the overall loss is a linear combination of individual side output losses and a multi-scale fusion term:

$$L(X, y^{\text{true}}; \theta) = \sum_m \alpha_m \ell^m(X, y^{\text{true}}; \theta) \\ -\frac{1}{2} \sum_{i\,:\,y_i^{\text{true}}=1} \log(\hat{y}_i^{\text{fuse}}) - \frac{1}{2} \sum_{i\,:\,y_i^{\text{true}}=0} \log(1-\hat{y}_i^{\text{fuse}}), \tag{2}$$

where $\hat{y}_i^{\text{fuse}} = \text{sigmoid}(\sum_m h_m \hat{y}_i^m)$. At test time, the final edge prediction is a weighted average of the side outputs and $\hat{y}^{\text{fuse}}$ [42].

3.2. Generating adversarial examples

In this paper, we apply attacks in the family of fast gradient sign methods (FGSM). These are some of the most studied attack methods [14, 26, 25, 10], and they require relatively little computation when compared with methods like L-BFGS [14]. In the following section, we describe a few relevant examples of fast gradient sign methods, adopting the notation of [41].

The original FGSM [14] generates an adversarial perturbation using the gradient of the loss

$$X^{\text{adv}} = X + \epsilon \,\text{sign}(\nabla_X L(X, y^{\text{true}}; \theta)), \tag{3}$$

where y^{true} is the ground-truth edge map. FGSM can be extended to the iterative fast gradient sign method (I-FGSM) [26] and the momentum iterative fast gradient sign method (MI-FGSM) [10], the latter of which uses the update rule

$$g_{n+1} = \mu g_n + \frac{\nabla_X L(X_n^{\text{adv}}, y^{\text{true}}; \theta)}{\|\nabla_X L(X_n^{\text{adv}}, y^{\text{true}}; \theta)\|_1} \tag{4}$$

$$X_{n+1}^{\text{adv}} = \text{Clip}_X^{\epsilon}\left[X_n^{\text{adv}} + \alpha \,\text{sign}(g_{n+1})\right], \tag{5}$$

where $\epsilon \geq \left\|X - X^{\text{adv}}\right\|_\infty$ measures the size of the perturbation and the momentum μ and step size α are attack parameters. In this paper, all attacks are based on MI-FGSM.

In transferability studies, [41] showed that introducing *input diversity transformations* makes attack perturbations more likely to transfer across architectures. Like data augmentation, the input image X is randomly resized during the optimization process. Following this approach, we test M-DI^2-FGSM, a modified version of MI-FGSM, in our transfer experiments. This replaces the update in Eq. 4 with

$$g_{n+1} = \mu g_n + \frac{\nabla_X L(T(X_n^{\text{adv}}), y^{\text{true}}; \theta)}{\|\nabla_X L(T(X_n^{\text{adv}}), y^{\text{true}}; \theta)\|_1} \tag{6}$$

where

$$T(X) = \begin{cases} \text{resize}(X) & \text{with probability } 1/2 \\ X & \text{otherwise} \end{cases} \tag{7}$$

The transformation function resize(X) first down-scales the image to a rectangle with random dimensions (w, h)—where $w, h \sim \text{Uniform}(0, 300)$—then randomly pads the boundaries of the image with black pixels to restore it to its original size.

After perturbing the image, it is possible that pixel intensities of X^{adv} leave the valid range [0, 255]. To deal with this, we simply clip pixel intensities to [0, 255] after adding the perturbation. Although this can destroy some of the perturbation, [40] find that the effect is negligible for small ϵ, so we adopt this practice.

3.3. Targeted attacks

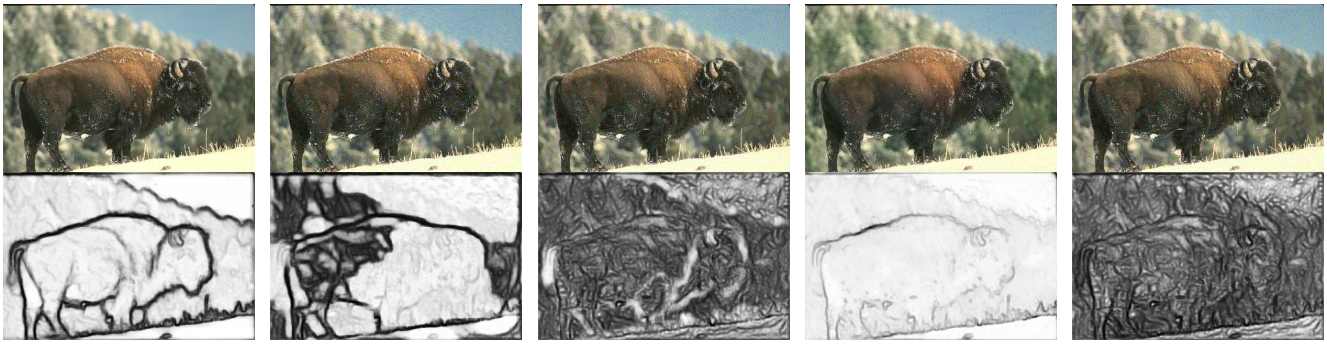
Up to this point, we have only discussed so-called *untargeted* attacks, which maximize the original training loss. Adversarial attacks also come in a *targeted* form that *minimizes*, rather than maximizes, a *modified* loss. For example, targeted MI-FGSM has the update rule

$$g_{n+1} = \mu g_n + \frac{\nabla_X L(X_n^{\text{adv}}, y^{\text{target}}; \theta)}{\|\nabla_X L(X_n^{\text{adv}}, y^{\text{target}}; \theta)\|_1} \tag{8}$$

$$X_{n+1}^{\text{adv}} = \text{Clip}_X^{\epsilon}\left[X_n^{\text{adv}} - \alpha \,\text{sign}(g_{n+1})\right], \tag{9}$$

where y^{target} is the desired output of the network. Note that y^{true} has been changed to y^{target} and the sign in front of α is now negative.

We found that switching from HED's class-balanced loss to a 1:1 class weighting as in Eq. 1 makes it harder for some attacks to suppress edges. To compensate for this, we apply a morphological thickening operation (radius of 3 pixels) to the ground-truth labels y^{true} before using it in an attack. This makes certain attacks, namely MI-FGSM and I-MI-FGSM (see Fig. 2) stronger.



(a) no attack (b) MI-FGSM (c) I-MI-FGSM $y^{\text{target}} = \mathbf{1} - y^{\text{true}}$ (d) S-MI-FGSM $y^{\text{target}} = \mathbf{0}$ (e) A-MI-FGSM $y^{\text{target}} = \mathbf{1}$

Figure 2: Figure 2a is an unaltered image from the BSDS500 test set and the output of HED. Attack 2b uses the untargeted MI-FGSM optimizer (Eqs. 4 and 5). Inverse-target (Figure 2c), suppression (Figure 2d), and activation attacks (Figure 2e) use the targeted MI-FGSM optimizer (Eqs. 8 and 9). Here, we use $\epsilon = 8$ and 10 iterations.

This leads to four main attack variants, the last three of which are targeted:

- $\mathcal{U}$ **Untargeted attack**. Use Eqs. 4 and 5.
- $\mathcal{S}$ **Suppression attack**. The objective is to lower the probability of edges throughout the image. This corresponds to setting $y^{\text{target}} = \mathbf{0}$.
- $\mathcal{A}$ **Activation attack**. The objective is to increase the probability of edges throughout the image. This corresponds to setting $y^{\text{target}} = \mathbf{1}$.
- $\mathcal{I}$ **Inverse-target attack**. The objective is to minimize the loss on the *inverted* ground truth label, using $y^{\text{target}} = \mathbf{1} - y^{\text{true}}$. This is an alternative to untargeted attacks.

3.4. Evaluation

We evaluate our approach on the test set of the BSDS500 dataset [2]. This consists of 200 images with ground-truth boundary annotations. Like [42], evaluation is performed using the fixed-contour threshold F-score (ODS). We perform the same standard non-maximal-suppression procedure as [6, 42] before evaluating outputs. To measure the effectiveness of the attack, we compare the mean ODS of HED outputs on unattacked images and outputs on attacked images.

4. BSDS500 experiments

In the following experiments, we evaluate these attack variants on BSDS500. Attacks are run for 10 iterations with $\epsilon = 16$, following [10]. We fix $\mu = 0.5$ and $\alpha = 2$. In Table 2, we see that all methods decrease the ODS F-score of HED, with I-MI-FGSM being the most effective attack. For other methods, we find that the combination of side-output averaging and non-maximal suppression protects HED against major drops in accuracy, even though the raw output of the network changes (see Figure 2).

variant	attack name	targeted?	optimizer	y^{target}	input diversity?
$\mathcal{U}$	MI-FGSM		Eqs. 4, 5	n/a	
$\mathcal{S}$	**S**-MI-FGSM	✓	Eqs. 8, 9	$\mathbf{0}$	
$\mathcal{A}$	**A**-MI-FGSM	✓	Eqs. 8, 9	$\mathbf{1}$	
$\mathcal{I}$	**I**-MI-FGSM	✓	Eqs. 8, 9	$\mathbf{1} - y^{\text{true}}$	
$\mathcal{U}$	M-DI^2-FGSM		Eqs. 6, 9	n/a	✓
$\mathcal{S}$	**S**-M-DI^2-FGSM	✓	Eqs. 6, 9	$\mathbf{0}$	✓
$\mathcal{A}$	**A**-M-DI^2-FGSM	✓	Eqs. 6, 9	$\mathbf{1}$	✓
$\mathcal{I}$	**I**-M-DI^2-FGSM	✓	Eqs. 6, 9	$\mathbf{1} - y^{\text{true}}$	✓

Table 1: Attack variants. **No prefix** ($\mathcal{U}$): *untargeted* attacks. **S- prefix** ($\mathcal{S}$): *suppression* attacks. **A- prefix** ($\mathcal{A}$): *activation* attacks. **I- prefix** ($\mathcal{I}$): *inverse-target* attacks. The last four attacks are the same as the first four, except they use input diversity transformations (Eqs. 6, 7).

It is worth noting that these attacks are less successful at suppressing edges than activating non-edges. In Figure 2d, notice that the boundary of the buffalo is still detected, albeit with much lower probability. This may be due to the weighting of the loss function in Eq. 1, as edges are less frequent than non-edges so they contribute less to the loss function. In general, we find that the attacks typically fail to suppress unambiguous edges (*e.g.*, the high-contrast leg in Figure 2) and fool those that require more global context

ϵ	MI-FGSM	I-MI-FGSM	S-MI-FGSM	A-MI-FGSM
0	0.775	0.775	0.775	0.775
1	**0.720**	0.728	0.752	0.756
2	0.680	**0.637**	0.731	0.726
4	0.636	**0.445**	0.702	0.684
8	0.588	**0.332**	0.645	0.642
16	0.545	**0.312**	0.573	0.580

Table 2: BSDS500 test ODS F-score as a function of attack magnitude ϵ for each attack variant.

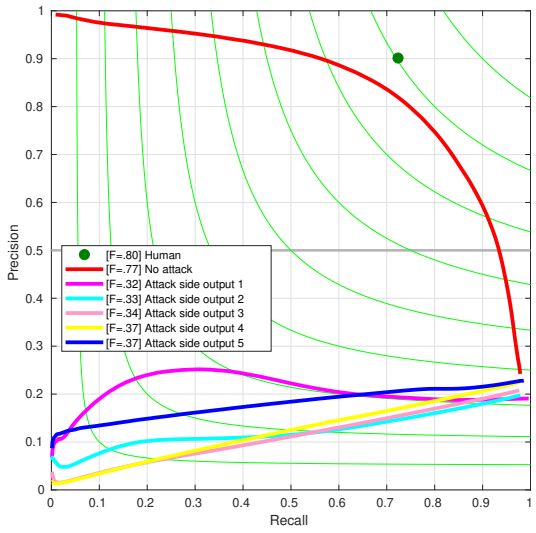


Figure 3: Precision-recall curves from I-MI-FGSM attacks on different side outputs. The "inverted" precision-recall curves are due to the fact that I-MI-FGSM makes the network more likely to classify edges as non-edges and non-edges as edges.

to detect (*e.g.*, the boundary between the mountain and the sky in Figure 2).

We also investigate the effect of attacking HED's individual side outputs. Instead of optimizing the loss in Eq. 2, we simply optimize the single-side-output loss in Eq. 1 for $m = 1, \ldots, 5$. As shown in Figure 3, attacking side outputs 2-5 is more effective than attacking side output 1. Two possible explanations for this are (1) when attacking deeper layers, more of the network's parameters are available for the attack to exploit, and (2) later side outputs have larger receptive fields, so each edge output value is a function of more degrees of freedom in the input. These findings agree with the previous paragraph and show that the deeper, non-local layers of HED are the most vulnerable to attack.

Some of the perturbations generated by our method can be perceived by the eye; we quantify this perceptibility using image quality metrics. In Table 3, we see that it is possible to differentiate attacked images from unattacked images using metrics for image quality degradation. This makes sense in the context of earlier findings that FGSM results in loss of SSIM and Edge-SSIM scores [17]. Note that the mean absolute value of the Laplacian is higher for attacked images, but this effect is small. Altogether, these visual and statistical discrepancies suggest that it may be possible to detect if the edges of an image have been attacked. We leave this as an open research question.

	ℓ^2	SSIM	ESSIM	Laplacian
Unattacked	0.000	1.000	1.000	37.234
MI-FGSM	0.090	0.760	0.342	45.441
I-MI-FGSM	0.107	0.684	0.281	47.875
S-MI-FGSM	0.096	0.754	0.318	50.589
A-MI-FGSM	0.106	0.675	0.297	48.708

Table 3: Measurements of image degradation due to various attacks. Each metric is computed on every image in the BSDS500 test set; mean values are reported. ℓ^2 corresponds to the normalized ℓ^2-norm, *i.e.* $\left\|X - X^{\text{adv}}\right\|_2 / \|X\|_2$. *SSIM* [39] is a prevalent measurement of image degradation. *ESSIM* [17], or Edge-SSIM, is obtained by applying SSIM to the Laplacian maps of each image. *Laplacian* corresponds the mean absolute value of the Laplace operator evaluated on the attacked image.

5. Transferability

The following section is concerned with how attacks on HED transfer to higher-level vision tasks and other datasets. These results are perhaps more noteworthy than those of Section 4, since they show that edge-based attacks have implications beyond edge detection. We consider two important tasks: image classification (a high-level task) and semantic segmentation (a low- and high-level task). It is worth investigating both since edges are a low-level feature, and altering edges might have different effects on networks trained to perform high-level tasks compared to those trained to perform low-level tasks. (Edge detection and semantic segmentation differ in that edges may not correspond to object boundaries; the related task of *object boundary detection* [33] corresponds more closely to semantic segmentation.)

5.1. Classification

First, we study the effect of edge-based attacks on ImageNet classification [8]. For each model, we report the top-1 classification accuracy on the validation set *before attack* and *after attack*. All classifiers use a standard imple-

mentation and publicly-available pretrained weights. We test VGG16 [37], the architecture that HED is based on. To investigate cross-architecture transfer, we also consider models from the ResNet family [16]. We use the same momentum and step size parameters from the previous experiments. When not stated, we set $\epsilon = 16$, which yields a slight perturbation. Because we do not have ground-truth edge annotations for the ImageNet dataset, we use the output of HED for y^{true}. This is necessary for untargeted attacks like MI-FGSM (Eqs. 4 and 5).

As shown in Table 4, edge-based attacks do transfer to ImageNet classifiers. The most significant decline is in VGG16, whose accuracy drops from 71.264% to 14.332% on images attacked with A-M-DI^2-FGSM. This is not much of a surprise: VGG16 and HED share the same architecture, and HED is pretrained with VGG16 weights, so one might expect attacks on HED to transfer to VGG16. More unexpectedly, however, the same perturbations also cause the accuracy of ResNet models to drop precipitously. The effect is greater the shallower the model—ResNet18 suffers a 40-point drop from A-M-DI^2-FGSM whereas ResNet152 only has a 34-point drop—but in all cases the reduction is consequential. This drop does not occur when the pixels of the perturbation are randomly permuted (column 3 of Table 6), showing that the structure of the perturbation matters.

Again in Table 4, observe that the edge activation attack A-M-DI^2-FGSM and the inverse-target attack I-M-DI^2-FGSM transfer the best to classification networks (columns 5, 9). In part, this is due to the small boost in transferability that comes from input diversity transformations. Nonetheless, the effect is small, and the same techniques without input diversity transformations transfer almost as well (*e.g.*, compare columns 8 and 9). Overall, it appears that edge activation attacks transfer to classification much better than other types of edge-based attacks.

To give a better understanding of transferability of attacks on edge detection, Table 6 compares the drop in classification accuracy due to edge-based attacks versus white-box attacks on the same models. In particular, we attack VGG16 and ResNet34 directly, using the same attack method (MI-FGSM) and identical parameters ($\epsilon = 16$, $\mu = 0.5$, $\alpha = 2$, 10 iterations). However, instead of using gradients from HED, we use gradients of the cross-entropy loss from the ImageNet ground truth like standard white-box classification attacks [14]. The white-box MI-FGSM attacks on VGG16 and ResNet34 are highly effective on their respective models (both lead to less than 3% accuracy). However, unlike edge attacks, the perturbations from these attacks do not transfer to the other models.

We conduct an additional experiment to see if white-box adversarial examples for classification transfer to edge detection. Using a ResNet18 model, we generate adversarial examples for each image in the BSDS500 test set using MI-FGSM. Here, we set y^{true} to the output of ResNet18 on the BSDS500 images. We find no difference in HED F-scores on the perturbed images versus unperturbed images, indicating that the transferability only works in one direction.

Table 5 shows the transferability of attacks on different HED side outputs. Here, we choose A-M-DI^2-FGSM, which we found has the highest transferability (Table 4). We observe that attacking side output 3 transfers the best to classification (*i.e.*, optimizing the loss in Eq. 1 for $m = 3$). This result mimics findings that attacking intermediate layers of classifiers—rather than output layers—leads to greater transferability of adversarial examples [19]. However, attacking the multi-scale loss of Eq. 2 still transfers better than attacking any individual side output.

5.2. Reducing texture bias: a successful defense against edge-based transfer attacks?

It has been shown that CNNs trained on ImageNet rely heavily on texture to classify images [13]. To reduce this reliance, [13] train classifiers on ImageNet training examples that have transformed using AdaIN style transfer [20]. This style transfer removes low-level texture information but preserves global shape. Augmenting the dataset with these stylized images reduces a CNN's reliance on texture [13]. Moreover, training on stylized examples improves robustness to distortions such as uniform noise, contrast changes, and high-pass and low-pass filtering [13].

In this experiment, we investigate whether training on texture-free images also improves robustness to our edge-based attacks. We compare the ImageNet validation accuracy of a ResNet50 model trained on ImageNet and a ResNet50 model trained jointly on ImageNet and Stylized-ImageNet [13] under various edge-based attacks. According to the first row of Table 7, both models have roughly the same performance on unattacked images (75.586% versus 74.074%). However, on edge-based adversarial examples, the two models' accuracy differs considerably.

In Table 7, note that training on stylized images improves robustness to edge suppression and activation attacks, but it decreases robustness to untargeted MI-FGSM edge attacks. In particular, the Stylized-ImageNet model achieves a 6-point lower accuracy on MI-FGSM adversarial examples than the ImageNet-trained model. Since this model is biased towards shape, this suggests that MI-FGSM targets shape information more than S-MI-FGSM or A-M-DI^2-FGSM. On the other hand, on adversarial examples generated with S-MI-FGSM, the shape-biased model achieves a 6-point higher accuracy, and on those generated with A-M-DI^2-FGSM, it improves by 13 points. (This may suggest that the edge suppression and edge activation attacks obfuscate texture more than untargeted edge attacks.)

Thus, augmenting the training set with stylized images is not enough to defend against all edge-based attacks.

	none	MI-FGSM	M-DI^2-FGSM	I-MI-FGSM	I-M-DI^2-FGSM	S-MI-FGSM	S-M-DI^2-FGSM	A-MI-FGSM	A-M-DI^2-FGSM
VGG16	71.264	25.184	25.698	15.330	14.830	26.554	26.042	15.856	**14.332**
ResNet18	68.932	36.748	36.944	29.380	28.902	36.480	35.140	30.310	**28.662**
ResNet34	72.766	43.910	44.696	35.630	**34.202**	43.448	42.720	36.374	34.544
ResNet50	75.586	46.202	46.818	36.772	35.574	45.518	45.554	37.152	**35.002**
ResNet101	77.122	50.354	51.128	41.132	**40.094**	49.704	49.206	42.346	40.644
ResNet152	78.018	53.418	54.030	45.106	**43.722**	53.146	52.816	45.870	43.734

Table 4: The top-1 ImageNet accuracy of classification models under various edge-based attacks.

Side output:	VGG16	ResNet18	ResNet34	ResNet50	ResNet101	ResNet152
1	39.042	46.376	51.456	52.33	55.262	58.832
2	25.67	40.244	48.122	47.146	51.97	54.83
3	**17.352**	**32.87**	40.584	**38.896**	**42.946**	**46.758**
4	20.884	34.642	**40.38**	40.94	46.288	49.984
5	22.896	36.424	41.19	44.03	48.146	51.996
All	**14.332**	**28.662**	**34.544**	**35.002**	**40.644**	**43.734**

Table 5: Classification transferability results when A-M-DI^2-FGSM is applied to one of the side outputs of HED. Of all of the individual side outputs, the third one is the best to attack (with the exception of ResNet34). However, attacking all side outputs simultaneously (by optimizing Eq. 2 directly) still transfers the best.

	unattacked	A-MI-FGSM	A-MI-FGSM (permuted)	VGG16 MI-FGSM	ResNet34 MI-FGSM
VGG16	71.264	15.856	65.112	**2.274**	65.774
ResNet18	68.932	**30.31**	64.262	63.97	59.784
ResNet34	72.766	36.374	68.982	68.546	**0.536**
ResNet50	75.586	**37.152**	71.424	71.378	67.87
ResNet101	77.122	**42.346**	73.718	73.7	70.34
ResNet152	78.018	**45.87**	74.878	74.946	72.046

Table 6: A comparison of top-1 ImageNet classification accuracies on images attacked with A-MI-FGSM and white-box MI-FGSM. The third column is obtained by directly attacking VGG16 using MI-FGSM, then evaluating all six models on the perturbed images. As shown, white-box attacks on VGG16 and ResNet34 are more effective than edge-based A-MI-FGSM, but they do not transfer as well to the other models, unlike edge-based attacks. The third column, which highlights the importance of the structure of the perturbation, is obtained by permuting the pixels returned by A-MI-FGSM, in a similar manner to [40].

5.3. Semantic segmentation

In addition to classification, we also study whether adversarial examples for edge detection transfer to semantic segmentation. Like those for classification, adversarial examples for semantic segmentation have also been shown to transfer between deep network architectures [40]. DeepLabv3+ [7] is a state-of-the-art CNN model for semantic segmentation. We test a model provided by the authors of the paper [7] that is pretrained on MS-COCO [27] and on augmented training examples from PASCAL VOC 2012 [11]. To evaluate the transferability of edge-based attacks, we compare the mean intersection over union (mIOU) of DeepLabv3+ on unperturbed and perturbed images from the PASCAL VOC 2012 validation set.

As shown in Table 8, attacks on edge detection also transfer to DeepLabv3+, albeit to a lesser degree. The degradation in this model is smaller than in classification; in Figure 4—a typical example of semantic segmentation—many objects in the scene are still detected. However, like in

	ImageNet (texture-biased)	ImageNet + Stylized-ImageNet (shape-biased)
unattacked	**75.586**	74.074
MI-FGSM	**46.202**	40.952
S-MI-FGSM	45.518	**51.778**
A-M-DI^2-FGSM	35.002	**48.786**

Table 7: Top-1 ImageNet validation accuracies of a ResNet50 model trained on ImageNet and both the ImageNet and the Stylized-ImageNet datasets. Decreasing texture bias by training on Stylized-ImageNet improves robustness to suppression and activation attacks like S-MI-FGSM and A-M-DI^2-FGSM, but it reduces robustness to MI-FGSM.

attack	mIOU
none	0.822
MI-FGSM	0.648
S-MI-FGSM	0.681
A-MI-FGSM	**0.553**
S-M-DI^2-FGSM	0.735
A-M-DI^2-FGSM	0.603
A-MI-FGSM (permuted)	0.775

Table 8: Performance of DeepLabv3+ model on the validation set of PASCAL VOC 2012. The model, `xception65_coco_voc_train_aug`, was trained on the COCO and VOC 2012 training datasets (with data augmentation).

classification experiments, when we randomly permute the perturbation like [40], we observe a much smaller degradation in performance (0.047 drop with permutation and 0.269 without). This demonstrates that the structure of perturbation still matters; the attack cannot be replicated with random noise.

6. Conclusions

In this paper, we have added to the wealth of existing evidence that, regardless of task or domain application, undefended deep neural networks are susceptible to adversarial attacks. In particular, we have shown that even a network trained to perform a low-level, "straightforward" task like edge detection can be confused and manipulated by slight perturbations. This lends further credence to the notion that adversarial examples are intrinsic to current neural networks (or their optimization process) rather than a mere artifact of training data and task.

However, at the same time, attacks on edge detection

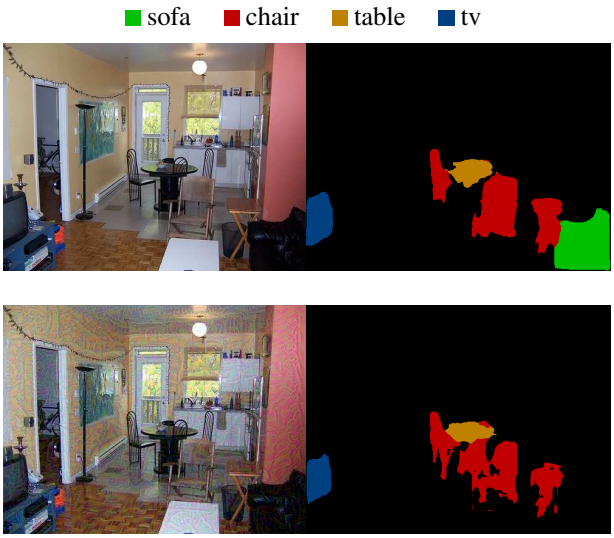


Figure 4: **Top row**: output of DeepLabv3+ model [7] on an image from the Pascal VOC 2012 validation set. **Bottom row**: output on the same image attacked with A-MI-FGSM. Under an edge-based attack, the segmentation model fails to recognize the sofa.

are unique in that they blindly *transfer*: the same attacks that fool an edge detection network also fool deep networks trained to perform classification and, to a lesser extent, semantic segmentation. These attacks transfer despite significant differences in network architecture and training data.

Still, unresolved questions remain. The exact reasons why edge-based adversarial attacks transfer to classification and segmentation are unclear. Perhaps the low-level cues learned by HED are shared by ImageNet classifiers and semantic segmentation networks, and when these cues are disrupted, all models suffer. Perhaps ImageNet classifiers' reliance on texture information makes them especially prone to certain types of attacks (edge suppression and activation), a question we began to address in Section 5.2.

As the defense we explored in Section 5.2 was weak, we see potential in protecting HED and similar models against white-box adversarial examples and defending higher-level vision models against edge-based transfer attacks. It is possible that existing defense techniques (*e.g.*, adversarial training [14, 29]) are effective here; otherwise, new defenses may need to be explored.

Acknowledgements

We thank Cihang Xie, Yingwei Li, and Wei Shen for their helpful comments. We thank the Office of Naval Research with grant N00014-15-1-2356. Christian Cosgrove was supported by the 2018 Pistritto Fellowship from the Johns Hopkins Department of Computer Science.

References

[1] M. Alzantot, Y. Sharma, A. Elgohary, B.-J. Ho, M. Srivastava, and K.-W. Chang. Generating natural language adversarial examples. In *Conference on Empirical Methods in Natural Language Processing*, 2018.

[2] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(5):898–916, May 2011.

[3] S. Baluja and I. Fischer. Adversarial transformation networks: Learning to generate adversarial examples. *arXiv preprint arXiv:1703.09387*, 2017.

[4] S. Belongie, G. Mori, and J. Malik. Matching with shape contexts. In *Statistics and Analysis of Shapes*, pages 81–105. Springer, 2006.

[5] G. Bertasius, J. Shi, and L. Torresani. Deepedge: A multi-scale bifurcated deep network for top-down contour detection. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[6] J. Canny. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, (6):679–698, 1986.

[7] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, 2018.

[8] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. Ieee, 2009.

[9] P. Dollár and C. L. Zitnick. Structured forests for fast edge detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1841–1848, 2013.

[10] Y. Dong, F. Liao, T. Pang, H. Su, J. Zhu, X. Hu, and J. Li. Boosting adversarial attacks with momentum. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[11] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html.

[12] V. Fischer, M. C. Kumar, J. H. Metzen, and T. Brox. Adversarial examples for semantic image segmentation. In *International Conference on Learning Representations Workshop*, 2018.

[13] R. Geirhos, P. Rubisch, C. Michaelis, M. Bethge, F. A. Wichmann, and W. Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. *International Conference on Learning Representations*, 2019.

[14] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. *CoRR*, abs/1412.6572, 2014.

[15] C. Guo, M. Rana, M. Cisse, and L. van der Maaten. Countering adversarial images using input transformations. In *International Conference on Learning Representations Workshop*, 2018.

[16] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[17] W. Heng, S. Zhou, and T. Jiang. Harmonic adversarial attack method. *CoRR*, abs/1807.10590, 2018.

[18] B. K. Horn. The binford-horn line-finder. 1973.

[19] Q. Huang, Z. Gu, I. Katsman, H. He, P. Pawakapan, Z. Lin, S. J. Belongie, and S. Lim. Intermediate level adversarial attack for enhanced transferability. *CoRR*, abs/1811.08458, 2018.

[20] X. Huang and S. Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[21] D. H. Hubel and T. N. Wiesel. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of physiology*, 160(1):106–154, 1962.

[22] J. Kittler. On the accuracy of the sobel edge detector. *Image and Vision Computing*, 1(1):37–42, 1983.

[23] S. Konishi, A. L. Yuille, J. M. Coughlan, and S. C. Zhu. Statistical edge detection: Learning and evaluating edge cues. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(1):57–74, 2003.

[24] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

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

[26] A. Kurakin, I. Goodfellow, and S. Bengio. Adversarial machine learning at scale. *International Conference on Learning Representations*, 2017.

[27] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[28] Y. Liu and M. S. Lew. Learning relaxed deep supervision for better edge detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 231–240, 2016.

[29] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. Towards deep learning models resistant to adversarial attacks. *International Conference on Learning Representations*, 2018.

[30] N. Papernot, P. McDaniel, and I. Goodfellow. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. *arXiv preprint arXiv:1605.07277*, 2016.

[31] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In *2016 IEEE Symposium on Security and Privacy (SP)*, pages 582–597. IEEE, 2016.

[32] O. Poursaeed, I. Katsman, B. Gao, and S. Belongie. Generative adversarial perturbations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4422–4431, 2018.

[33] V. Premachandran, B. Bonev, and A. L. Yuille. Pascal boundaries: A class-agnostic semantic boundary dataset. *arXiv preprint arXiv:1511.07951*, 2015.

[34] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47(1):7–42, Apr 2002.

[35] R. Shapley and D. Tolhurst. Edge detectors in human vision. *The Journal of physiology*, 229(1):165–183, 1973.

[36] W. Shen, X. Wang, Y. Wang, X. Bai, and Z. Zhang. Deepcontour: A deep convolutional feature learned by positive-sharing loss for contour detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3982–3991, 2015.

[37] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[38] 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.

[39] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[40] C. Xie, J. Wang, Z. Zhang, Y. Zhou, L. Xie, and A. Yuille. Adversarial examples for semantic segmentation and object detection. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[41] C. Xie, Z. Zhang, Y. Zhou, S. Bai, J. Wang, Z. Ren, and A. Yuille. Improving transferability of adversarial examples with input diversity. In *Computer Vision and Pattern Recognition*. IEEE, 2019.

[42] S. Xie and Z. Tu. Holistically-nested edge detection. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, ICCV '15, pages 1395–1403, Washington, DC, USA, 2015. IEEE Computer Society.

[43] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. How transferable are features in deep neural networks? In *Advances in neural information processing systems*, pages 3320–3328, 2014.

[44] L. L. Zhu, C. Lin, H. Huang, Y. Chen, and A. Yuille. Unsupervised structure learning: Hierarchical recursive composition, suspicious coincidence and competitive exclusion. In *European Conference on Computer Vision*, pages 759–773. Springer, 2008.