Stereopagnosia: Fooling Stereo Networks with Adversarial Perturbations

Alex Wong[†], **Mukund Mundhra**[†], **Stefano Soatto**
UCLA Vision Lab
alexw@cs.ucla.edu, mukundmundhra@cs.ucla.edu, soatto@cs.ucla.edu

Abstract

We study the effect of adversarial perturbations of images on the estimates of disparity by deep learning models trained for stereo. We show that imperceptible additive perturbations can significantly alter the disparity map, and correspondingly the perceived geometry of the scene. These perturbations not only affect the specific model they are crafted for, but transfer to models with different architecture, trained with different loss functions. We show that, when used for adversarial data augmentation, our perturbations result in trained models that are more robust, without sacrificing overall accuracy of the model. This is unlike what has been observed in image classification, where adding the perturbed images to the training set makes the model less vulnerable to adversarial perturbations, but to the detriment of overall accuracy. We test our method using the most recent stereo networks and evaluate their performance on public benchmark datasets.

Introduction

Deep Neural Networks are seen as fragile, in the sense that small perturbations of their input, for instance an image, can cause a large change in the output, for instance the inferred class of objects in the scene (Moosavi-Dezfooli, Fawzi, and Frossard 2016) or its depth map (Wong, Cicek, and Soatto 2020). This is not too surprising, since there are infinitely many scenes that are consistent with the given image, so at inference time one has to rely on the complex relation between that image and *different scenes* portrayed in the training set. This is not the case for stereo: Under mild assumptions discussed below, a depth map can be uniquely inferred point-wise from two images. There is no need to *learn* stereo, as the images of a particular scene are sufficient to infer its depth without relying on images of different scenes. (The reason we *do* use learning is to regularize the reconstruction where the assumptions mentioned below are violated, for instance in regions of homogeneous reflectance.) It would therefore be surprising if one could perturb the images in a way that forces the model to overrule the evidence and alter the perceived depth map, especially if such perturbations affect regions of non-uniform reflectance. In this paper, we show that this *can* be done and refer to this phenomenon as *stereopagnosia*, a geometric analogue of prosopagnosia (Damasio, Damasio, and Van Hoesen 1982).

Specifically, we consider stereo networks, that are functions that take as input a calibrated stereo pair and produce a depth map as output. A stereo pair consists of two images captured by cameras in known relative configuration (position and orientation), with non-zero parallax (distance between the optical centers), projectively rectified so that corresponding points (points in the two image planes whose pre-image under perspective projections intersect in space) lie on corresponding scan-lines. A depth map is a function that associates to each pixel in a rectified image a positive real number corresponding to the distance of the point of first intersection in the scene from the optical center.

Equivalently, the network can output *disparity*, the displacement between corresponding points in the two images, from which depth can be computed in closed form. Wherever a point in the scene is supported on a surface that is Lambertian, locally smooth, seen under constant illumination, co-visible from both images, and sporting a *sufficiently exciting* reflectance,[1] its distance from the images can be computed in closed-form (Ma et al. 2012). Where such assumptions are violated, disparity is either not defined, for instance in occluded regions that are visible from one image but not the other, or ill-posed, for instance in regions with constant reflectance where any disparity is equally valid (the so-called "aperture problem"). To impute disparity to these regions, regularization can be either generic (*e.g.*, minimal-surface assumptions (Horn and Schunck 1981)) or data-driven, exploiting known relations between stereo pairs and disparity in scenes other than the one in question. This is where stereo networks come in.

Our first contribution is to show that *stereo networks are vulnerable to adversarial perturbations,* which are small additive changes in the input images (either one or both), designed for a specific image pair in a way that maximally changes the output of a specific trained deep network model. The fact that it is possible to alter the disparity, *even in regions that satisfy the assumptions discussed above* (Fig. 3), where disparity is uniquely defined and computable

[†] denotes authors with equal contributions.

Copyright © 2021, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

[1] There exist region statistics that exhibit isolated extrema, so the region around the point is "distinctive" (Lowe 1999).

in closed form, is surprising since the network is forced to ignore the evidence, rather than simply exploit the unbounded hypothesis space available in an ill-posed problem.

The second contribution is to show that, despite being crafted for a specific model, the perturbations can affect the behavior of other models, with different network architecture, trained with different loss functions and optimization methods. However, transferability is not symmetric, for instance perturbations constructed for AANet (Xu and Zhang 2020) can wreak havoc if used with DeepPruner (Duggal et al. 2019), but not vice-versa. Models that incorporate explicit matching, such as correlation, are more robust than those that are agnostic to the mechanics of correspondence, and are instead based on stacking generic features.

Our third contribution is more constructive, and establishes that adversarial perturbations can be used to beneficial effects by augmenting the dataset and function as regularizers. Unlike in single-image classification and monocular depth perception where such regularization trades off robustness to perturbations with overall accuracy, in our case we obtain models that are more robust while retaining the performance of the original model.

To achieve these results, we extend the Fast Gradient Sign Method (Goodfellow, Shlens, and Szegedy 2014) and its iterative versions (Dong et al. 2018; Kurakin, Goodfellow, and Bengio 2016), developed for for single frame classification, to two-frame stereo disparity estimation. We evaluate the robustness of recent stereo methods (PSMNet, DeepPruner, AANet) on the standard benchmark stereo datasets (KITTI 2012 (Geiger, Lenz, and Urtasun 2012) and 2015 (Menze and Geiger 2015)).

Related Works

Adversarial Perturbations (Szegedy et al. 2013) have been extensively studied for classification (Goodfellow, Shlens, and Szegedy 2014; Szegedy et al. 2013) with many iterative methods to boost the effectiveness of the attacks (Dong et al. 2018; Kurakin, Goodfellow, and Bengio 2016; Moosavi-Dezfooli, Fawzi, and Frossard 2016). (Moosavi-Dezfooli et al. 2017) further extended the attacks to the universal setting, where the same perturbations can be added to each image in a dataset to fool a network; (Nguyen, Yosinski, and Clune 2015) showed that unrecognizable noise can result in high confidence predictions. To defend against such attacks, (Kurakin, Goodfellow, and Bengio 2016; Tramèr et al. 2017) proposed training with adversarial data augmentation and (Xie et al. 2017a) improved it with randomization.

Recently, (Naseer et al. 2019) studied transferability of perturbations across datasets and models and (Xie et al. 2019) improved transferability across networks by deforming the image. (Peck et al. 2017) demonstrated lower bounds on the magnitudes of perturbations needed to fool a network and (Ilyas et al. 2019) showed that the existence of adversarial perturbations can be attributed to non-robust features.

While there are many adversarial works on classification, there exist only a few for dense-pixel prediction tasks (e.g. semantic segmentation, depth, optical flow). (Xie et al. 2017b) designed attacks for detection and segmentation. (Hendrik Metzen et al. 2017) demonstrated targeted universal attacks for semantic segmentation, where the network is fooled to predict a specific target. (Wong, Cicek, and Soatto 2020) used targeted attacks to provide explainability for single image depth prediction networks; whereas (Dijk and Croon 2019) probed them by inserting vehicles into input images. (Mopuri, Ganeshan, and Babu 2018) examined universal attacks for segmentation and single image depth. (Ranjan et al. 2019) studied patch attacks for optical flow.

Unlike (Mopuri, Ganeshan, and Babu 2018; Wong, Cicek, and Soatto 2020), we study *non-targeted* adversarial perturbations for stereo matching. While (Ranjan et al. 2019) also use multiple frames, they apply the *same visible* patch to the same locations in both images, whereas our attacks are *visually imperceptible* and crafted separately for each image.

Deep Stereo Matching (Zagoruyko and Komodakis 2015; Žbontar and LeCun 2016) leveraged deep networks to extract features and perform matching separately. Recent works implement the entire stereo pipeline as network layers trained end-to-end. (Mayer et al. 2016) used correlation layers to create a 2D cost volume. (Pang et al. 2017) extended (Mayer et al. 2016) to a cascade residual learning framework. AANet (Xu and Zhang 2020) also used correlation, but instead introduced adaptive sampling to replace convolutions when performing cost aggregation to avoid sampling at discontinuities. (Kendall et al. 2017) proposed to concatenate features together to build a 3D cost volume for performing cost aggregation. PSMNet (Chang and Chen 2018) added spatial pyramid pooling layers and introduced a stacked hourglass architecture. DeepPruner (Duggal et al. 2019) followed the 3D cost volume architectures proposed by (Chang and Chen 2018; Kendall et al. 2017) and proposed differentiable patch matching over deep features to construct a sparse 3D cost volume.

In this work, we consider adversaries for PSMNet (Chang and Chen 2018), DeepPruner (Duggal et al. 2019) and AANet (Xu and Zhang 2020). PSMNet is an exemplar of modern stereo networks (stacked hourglass, cost volume, 3D convolutions), but uses feature stacking without explicit matching. DeepPruner shares the general architecture of PSMNet, but performs explicit matching. AANet is the state of the art and represents the 2D convolution and correlation architecture. In choosing these methods, we (i) examine their individual robustness against adversaries, (ii) study the transferability of perturbations between similar and different architectures, and (iii) apply defenses to increase robustness against adversaries. To the best of our knowledge, we are the first to study adversarial perturbations for stereo. As mentioned in the introduction, it is not a given that adversarial perturbations, known to exist for single image reconstruction, would exist for stereo, where the geometry of the scene is uniquely determined from the data, at least in the regions that satisfy the unique correspondence assumptions.

Generating Adversarial Perturbations

Given a pretrained stereo network $f_\theta(x_L, x_R)$ that predicts the disparity between the left x_L and right x_R images of a stereo pair, our goal is to craft perturbations

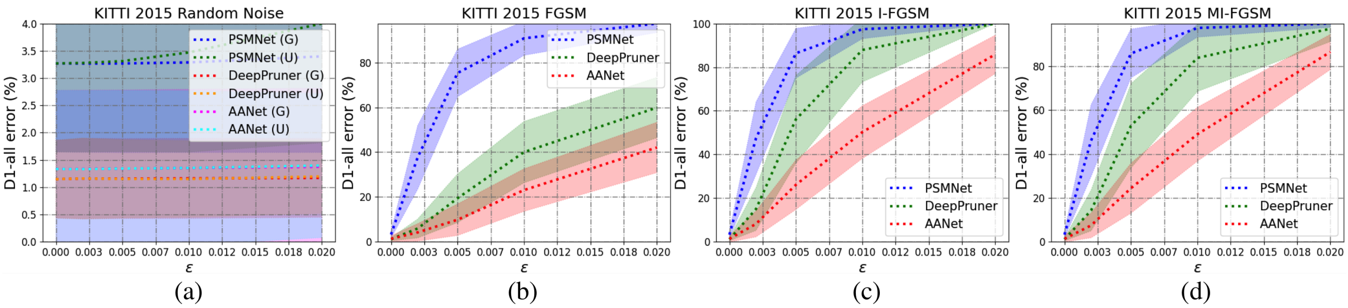


Figure 1: *Attacks on stereo models.* (a) Gaussian (G) and uniform (U) noise with various upper norms (ϵ) are added to input images as a naive attack. All methods are robust to both as performances across ϵ are approximately constant. (b) Even with a single optimization step, FGSM with $\epsilon = 0.02$ is able increase error of PSMNet from ≈3% to ≈97%. (c, d) Iterative methods (I-FGSM, MI-FGSM) are able to further degrade performance, increasing error of AANet from ≈1% to ≈87% and as much as 100% for PSMNet and DeepPruner. AANet is consistently more robust to adversarial noise than PSMNet and DeepPruner.

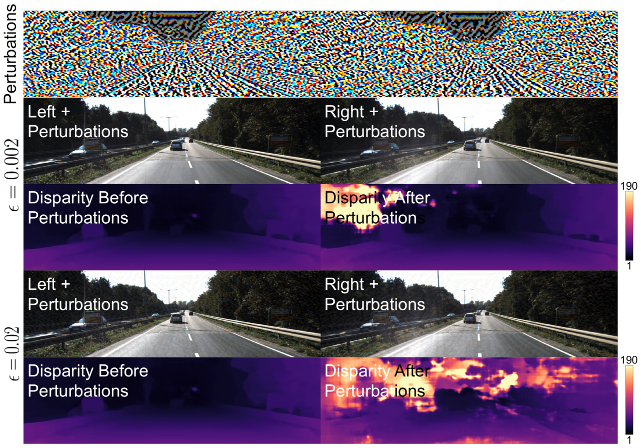


Figure 2: *FGSM with upper norms of 0.002 and 0.02*. With visually imperceptible noise, $\epsilon = 0.002$, PSMNet is fooled to predict much larger disparities (closer depths) in the top left corner region. Using $\epsilon = 0.02$, the perturbations corrupt the geometry of the entire scene.

$v_L, v_R \in \mathbb{R}^{H \times W \times 3}$, such that when added to (x_L, x_R), $f_\theta(x_L, x_R) \neq f_\theta(x_L + v_L, x_R + v_R)$. To ensure that the perturbations are visually imperceptible, we subject them to the norm constraints $\|v_I\|_\infty \leqslant \epsilon$ for $I \in \{L, R\}$. To demonstrate such perturbations exist, we extend white-box methods Fast Gradient Sign Method (FGSM) (Goodfellow, Shlens, and Szegedy 2014), iterative-FGSM (I-FGSM) (Kurakin, Goodfellow, and Bengio 2016), and its momentum variant (MI-FGSM) (Dong et al. 2018), originally for classification, to the stereo matching task. We note that it is also possible to perturb only one of the two images (e.g. let $v_L = 0$ or $v_R = 0$); the effect is less pronounced, but nonetheless present and shown in the Supp. Mat.

FGSM. Assuming access to the network f_θ and its loss function $\ell(f_\theta(x_L, x_R), y_{gt})$, the perturbations for the left and right images are computed as the sign of gradient with respect to the images separately:

$$v_I = \epsilon \cdot \mathtt{sign}(\nabla_{x_I} \ell(f_\theta(x_L, x_R), y_{gt}), \tag{1}$$

where $y_{gt} \in \mathbb{R}_+^{H \times W}$ is the groundtruth and $I \in \{L, R\}$.

I-FGSM. To craft perturbations v_L and v_R for the stereo pair x_L and x_R using iterative FGSM, we begin with $v_L^0 = 0$ and $v_R^0 = 0$ and accumulate the sign of gradient with respect to each image for N steps:

$$g_I^{n+1} = \nabla_{x_I} \ell(f_\theta(x_L + v_L^n, x_R + v_R^n), y_{gt}), \tag{2}$$

$$v_I^{n+1} = \mathtt{clip}\big(\alpha \cdot \mathtt{sign}(g_I^{n+1}), -\epsilon, \epsilon\big), \tag{3}$$

where n is the step, α is the step size and the $\mathtt{clip}(\cdot, -\epsilon, \epsilon)$ operation sets any value less than $-\epsilon$ to $-\epsilon$ and any value larger than ϵ to ϵ. The output perturbation is obtained after the N-th step, $v_L = v_L^N$ and $v_R = v_R^N$.

MI-FGSM. To leverage gradients from previous steps, we follow (Dong et al. 2018) and replace the gradients (Eqn. 2) with normalized gradients and a momentum term weighted by a positive scalar β for N steps:

$$g_I^{n+1} = \frac{\nabla_{x_I} \ell(f_\theta(x_L + v_L^n, x_R + v_R^n), y_{gt})}{\|\nabla_{x_I} \ell(f_\theta(x_L + v_L^n, x_R + v_R^n), y_{gt})\|_1}, \tag{4}$$

$$m_I^{n+1} = \beta \cdot m_I^n + (1 - \beta) \cdot g_I^{n+1}, \tag{5}$$

$$v_I^{n+1} = \mathtt{clip}\big(\alpha \cdot \mathtt{sign}(m_I^{n+1}), -\epsilon, \epsilon\big), \tag{6}$$

where $m_I^0 = 0$, $v_L = v_L^N$, and $v_R = v_R^N$.

Besides crafting perturbations for specific models, we also study their transferability to different models. To this end, we take $(x_L + v_L, x_R + v_R)$ optimized for one model (e.g. PSMNet) and feed it as input to another (e.g. AANet). However, while iterative methods (I-FGSM, MI-FGSM) are more effective than FGSM at corrupting the target model, their perturbations are unlikely to transfer across models because they tend to overfit to the target model. To increase the transferability across models, we leverage diverse inputs (Xie et al. 2019) as data augmentation when crafting perturbations using I-FGSM and MI-FGSM.

DI^2-FGSM and MDI^2-FGSM. Diverse inputs (DI) for iterative methods aims to reduce overfitting by randomly

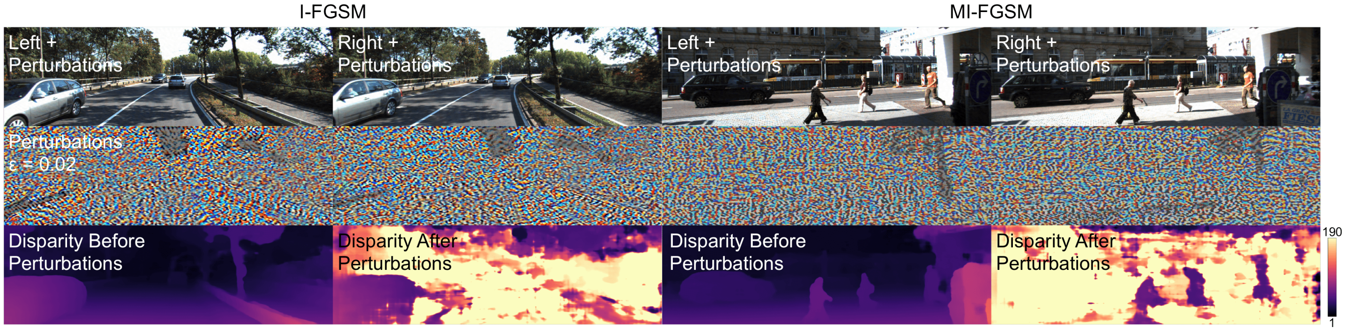

I-FGSM
MI-FGSM
Left + Perturbations
Right + Perturbations
Left + Perturbations
Right + Perturbations
Perturbations ϵ = 0.02
Disparity Before Perturbations
Disparity After Perturbations
Disparity Before Perturbations
Disparity After Perturbations
190
1

Figure 3: *I-FGSM and MI-FGSM on PSMNet.* For $\epsilon = 0.02$, I-FGSM and MI-FGSM can degrade performance much more than FGSM with much smaller perturbation magnitudes. Unlike Fig. 2 where most of the changes in disparity were concentrated on low-texture regions (no disparity signal and the perturbation drives the matching), here the perturbations degrades high texture regions. Still, the shapes of salient objects (e.g. car, human) seem to be preserved (although their disparities are altered).

resizing the input images by a factor of $h \in [h_{\text{min}}, h_{\text{max}}]$ in height and $w \in [w_{\text{min}}, w_{\text{max}}]$ in width with probability p. To maintain the original resolution, the inputs are randomly padded with zeros on each side such that the total padding along the height is $(H - h \cdot H)$ and along the width $(W - w \cdot W)$, respectively. We denote this procedure as $\phi(x, h, w)$. However, unlike (Xie et al. 2019), where the transformed image maps to a single class label, ground-truth disparity y_{gt} is dense or semi-dense, so a matching transformation must be applied to y_{gt}. Moreover, the scale of y_{gt} also needs to be adjusted with respect to the resized width $(w \cdot W)$ of the image. Hence, we extend diverse inputs to support stereo networks by:

$$\hat{x}_I = \phi(x_I, h, w), \tag{7}$$

$$\hat{v}_I = \phi(v_I, h, w), \tag{8}$$

$$\hat{y}_{gt} = w \cdot \phi(y_{gt}, h, w) \tag{9}$$

To incorporate this into iterative methods, we modify their respective gradient computations, g_I^{n+1}. For I-FGSM, we can re-write Eqn. 2 as:

$$g_I^{n+1} = \nabla_{\hat{x}_I} \ell(f_\theta(\hat{x}_L + \hat{v}_L^n, \hat{x}_R + \hat{v}_R^n), \hat{y}_{gt}), \tag{10}$$

Similarly, for MI-FGSM, we can modify Eqn. 4 to be:

$$g_I^{n+1} = \frac{\nabla_{\hat{x}_I} \ell(f_\theta(\hat{x}_L + \hat{v}_L^n, \hat{x}_R + \hat{v}_R^n), \hat{y}_{gt})}{\|\nabla_{\hat{x}_I} \ell(f_\theta(\hat{x}_L + \hat{v}_L^n, \hat{x}_R + \hat{v}_R^n), \hat{y}_{gt})\|_1}. \tag{11}$$

To evaluate the robustness of stereo networks, we use the official KITTI D1-all (the average number of erroneous pixels in terms of disparity and end-point error) metric:

$$\delta(i,j) = |f_\theta(\cdot)(i,j) - y_{gt}(i,j)|, \tag{12}$$

$$d(i,j) = \begin{cases} 1 & \text{if } \delta(i,j) > 3, \frac{\delta(i,j)}{y_{gt}(i,j)} > 5\%, \\ 0 & \text{otherwise} \end{cases} \tag{13}$$

$$\text{D1-all} = \frac{1}{\|\Omega_{gt}\|} \sum_{i,j \in \Omega_{gt}} d(i,j), \tag{14}$$

where Ω_{gt} is a subset of the image space Ω with valid ground-truth disparity annotations, $y_{gt} > 0$.

Experiment Setup

Datasets. We evaluate adversarial perturbations (robustness, transferability, defense) for recent stereo methods (PSMNet, DeepPruner, AANet) on the standard benchmark datasets: KITTI 2015 stereo (Menze and Geiger 2015) validation set in the main paper and KITTI 2012 (Geiger, Lenz, and Urtasun 2012) validation set in the Supp. Mat.

KITTI 2015 is comprised of 200 training stereo pairs and KITTI 2012 consists of 194 (all at 376 × 1240 resolution) with ground-truth disparities obtained using LiDAR for outdoor driving scenes. Following KITTI validation protocol, the KITTI 2015 training set is divided into 160 for training and 40 for validation, and the KITTI 2012 training set is split into 160 for training and 34 for validation. Due to computational limitations, we downsampled all images to 256×640; hence, there are slight increases in errors (Eqn. 14) compared to those reported by baseline methods.

Hyper-parameters. We study perturbations under four different upper norms, $\epsilon = \{0.02, 0.01, 0.005, 0.002\}$. $\epsilon = 0.002$ is where adversaries have little effect on the networks and $\epsilon = 0.02$ is the norm needed to achieve 100% errors on benchmark datasets. When optimizing with I-FGSM and DI^2-FGSM, we used $N = 40$ and $\alpha = 1/N \cdot \epsilon$ for $\epsilon = \{0.01, 0.005, 0.002\}$ and $\alpha = 0.10\epsilon$ for $\epsilon = 0.02$. For MI-FGSM and MDI^2-FGSM, $\alpha = 1/N \cdot \epsilon$ for all ϵ and chose $\beta = 0.47$ for momentum. More details on hyper-parameters and run-time can be found in Supp. Mat.

Attacking Stereo Networks

We begin with naive attacks on stereo networks (PSMNet, DeepPruner, AANet) by perturbing the input stereo pair (x_L, x_R) with Gaussian $\mathcal{N}(0, (\epsilon/4)^2)$ and uniform $\mathcal{U}(-\epsilon, \epsilon)$ noise for $\epsilon \in \{0.02, 0.01, 0.005, 0.002\}$. Fig. 1-(a) shows that such noise cannot degrade performance as the measured error stayed approximately constant under various ϵ. This demonstrates that the deep features extracted for matching are robust to random noises and fooling a stereo network requires non-trivial perturbations. Hence, we examine the robustness of stereo networks against perturbations specifically optimized for each network using our variants

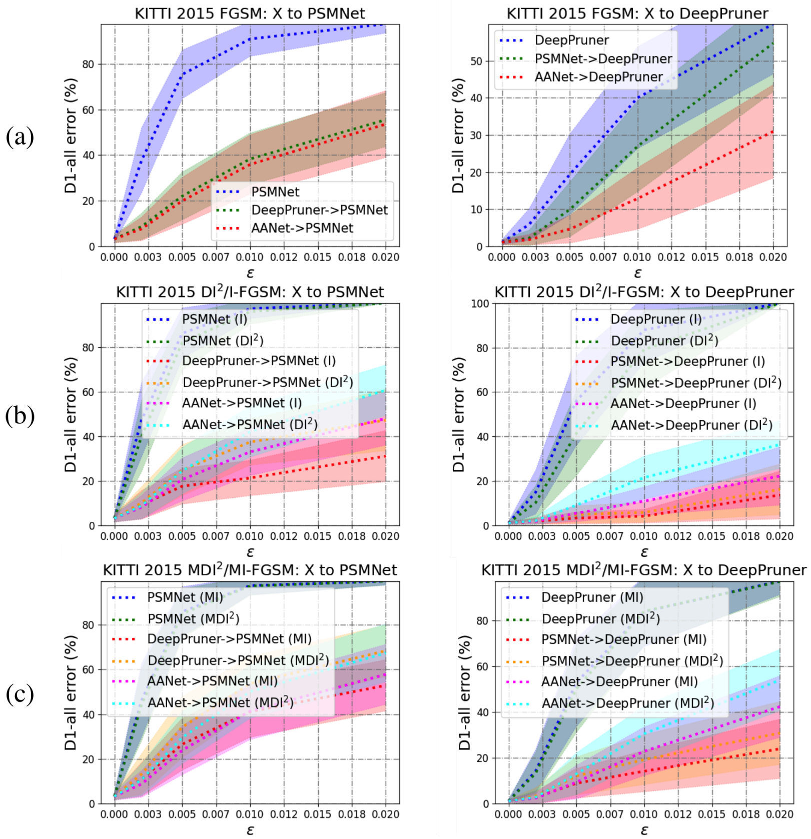


Figure 4: *Transferability.* Images with added adversarial perturbation optimized for various models using (a) FGSM, (b) I-FGSM and DI^2-FGSM, and (c) I-FGSM and MDI^2-FGSM are fed as input to a target model. Transferability is not symmetric. Perturbations crafted for AANet transfer the best. AANet is also the most robust against perturbations from other models.

of FGSM, I-FGSM, MI-FGSM.

FGSM. Fig. 1-(a) shows errors after attacking the networks with FGSM (Eqn. 1) where perturbations are optimized over a single time step. For large upper norm $\epsilon = 0.02$, the perturbations can degrade performance significantly – from 1.33% (AANet), 1.15% (DeepPruner) and 3.27% (PSMNet) mean error to 42.09%, 59.86%, and 97.33%, respectively. The larger the upper norm, the more potent the attack, but even with small $\epsilon = 0.002$, this attack can still increase AANet to 4.18% error, DeepPruner to 5.93%, and PSMNet to 38.11%. Fig. 2 shows a comparison of FGSM attacks on PSMNet using upper norms of 0.002 and 0.02. For $\epsilon = 0.002$, most of the damage is localized (e.g. top left region of image space); whereas for $\epsilon = 0.02$, the entire predicted scene is corrupted. The localized damage from small norm perturbations can be attributed to the observability of the scene. We hypothesize that training affects inference where the radiance of the surfaces is not sufficiently exciting i.e. the regularizer fills in in a manner that depends on training experience. So, small norm perturbations can corrupt regions where the radiance is less informative (sky, uniform textures, foliage etc.); whereas, other regions require larger norms.

I-FGSM, MI-FGSM. Fig. 1-(b, c) shows that I-FGSM and MI-FGSM both affect performance similarly. Because of the multiple optimization steps, when $\epsilon = 0.02$, even the more robust AANet succumbs to the attacks – increasing error to ≈87%. For PSMNet and DeepPruner, both I-FGSM and MI-FGSM can cause them to reach 100% error. While we cap the upper norm at ϵ, we note that iterative methods are able to introduce more errors while using a much smaller perturbation magnitude. For $\epsilon = 0.02$, FGSM achieves 97.33% error on PSMNet with $\|v_L\|_1 = 0.0569$ and $\|v_R\|_1 = 0.0568$; whereas, I-FGSM achieves 100% error with only $\|v_L\|_1 = 0.0213$ and $\|v_R\|_1 = 0.0196$ – less than half of the L1 norm used by FGSM, making it less perceptible. Fig. 3 shows examples of I-FGSM and MI-FGSM on PSMNet. When compared to rows 4 and 5 of Fig. 2 ($\epsilon = 0.02$), both are less perceptible. Moreover, I-FGSM and MI-FGSM can fool the networks in textured regions where disparity can be obtained simply by matching.

Interesting observations. Even though error reaches 100% for I-FGSM and MI-FGSM, the shape (albeit incorrect) of some salient objects like cars and human still persists (Fig. 3). This is because disparity is largely driven by the data term and hence there exist unique correspondences for such objects with sufficiently exciting texture. However, while the general shape persists, the disparity is incorrect (as

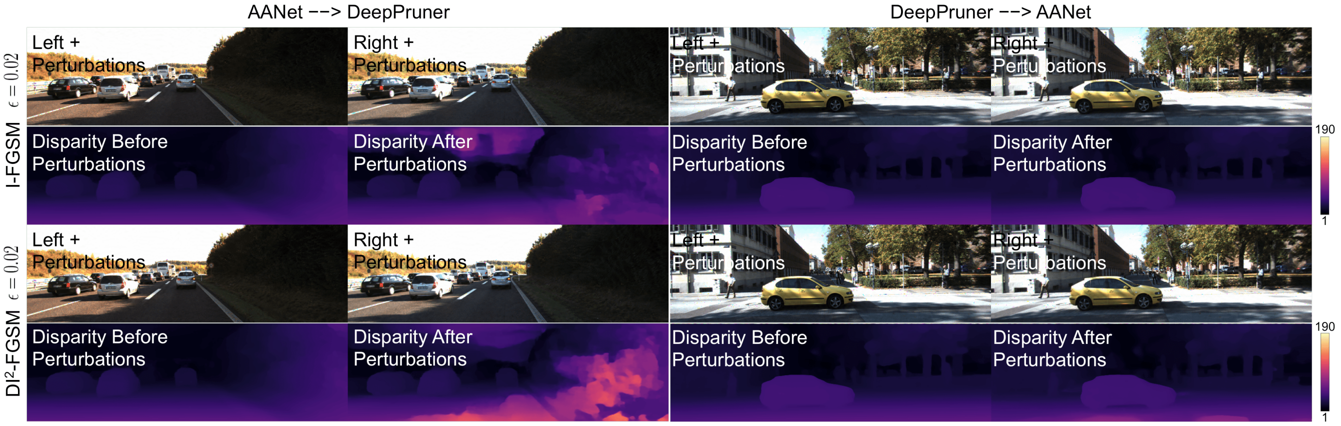


Figure 5: *Transferability between AANet and DeepPruner.* We craft perturbation for AANet and DeepPruner using I-FGSM and DI^2-FGSM with $\epsilon = 0.02$. DI^2-FGSM transfers better than I-FGSM. Perturbations crafted for AANet transfer well to DeepPruner (AANet→DeepPruner), but those for DeepPruner have less effect on AANet (DeepPruner→AANet).

it is filled in by the regularizer). Another phenomenon is that the noise required to perturb white regions (white walls, sky, Fig. 2, 3) is much less than that required to attack other color intensities (e.g. uniform black). While radiance (being less informative) is a factor, we hypothesize that this is a special case due to white regions being on the upper support of image intensities, which results in high activations; hence, the adversary only needs to add small noise to adjust the activations to the needed values to corrupt the scene. We will leave the numerical analysis of this "white-pixel" phenomenon to future work.

Thus, stereo networks are indeed vulnerable to adversarial perturbations. Each architecture exhibits different levels of robustness against adversaries. Feature stacking (PSMNet) is the least robust, followed by patch-matching (DeepPruner) with correlation (AANet) being the most robust. This is because DeepPruner and AANet both find correspondences based on similarity between deep features via explicit matching (well-defined data fidelity, so the perturbations corrupt the regularizer); whereas PSMNet relies purely on learned convolutional filters to produce matches and is more susceptible to the attacks.

Transferability Across Models

To study transferability, we (i) optimize perturbations for PSMNet, DeepPruner and AANet separately, (ii) add each set of model-specific perturbations to the associated input stereo pair for another model i.e. add perturbations for PSMNet to the inputs of DeepPruner and AANet, and (iii) measure the resulting error using Eqn. 14.

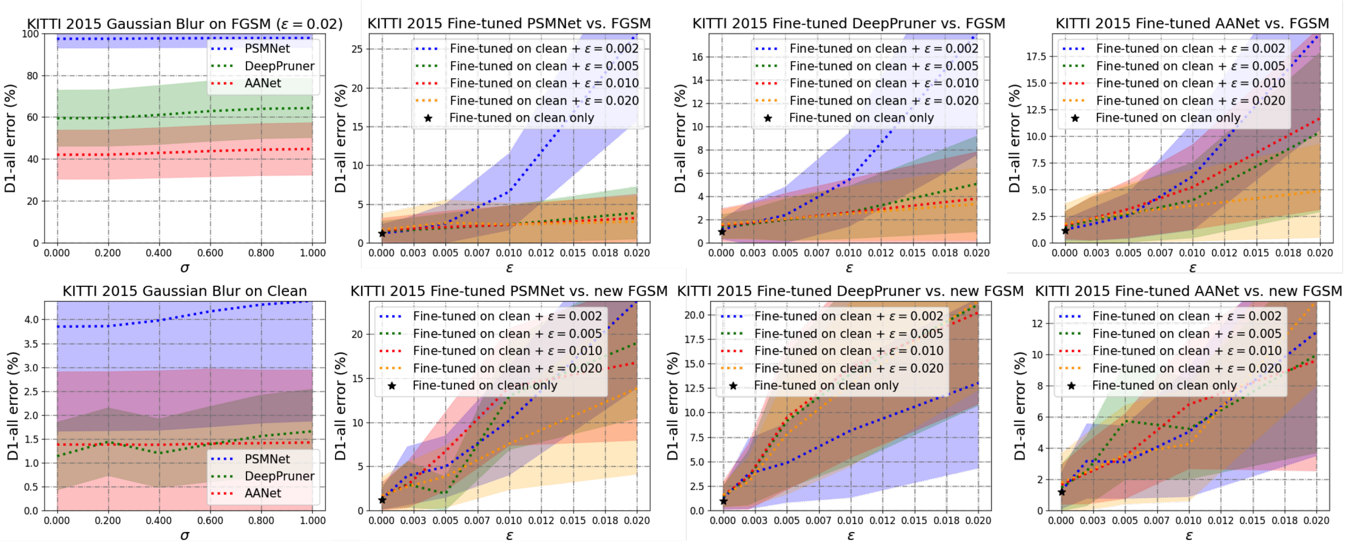
FGSM, I-FGSM, MI-FGSM. We begin with methods studied above. Fig. 4-(a) shows the transferability of FGSM from different models (red, green), for which the perturbations were optimized, to a target model (blue). We found that the perturbations do transfer, but with reduced effects e.g. for $\epsilon = 0.02$, perturbations optimized for DeepPruner and AANet achieve 55.47% and 53.55% error on PSMNet, respectively, while perturbations optimized for PSMNet achieves 97.33%. The potency of the perturbations also grows with the upper norm; hence, one can increase ϵ of an adversary to further degrade new models.

Fig. 4-(b) shows the transferability of I-FGSM and Fig. 4-(c), MI-FGSM. Unlike FGSM, iterative methods transfer much less. For instance, I-FGSM ($\epsilon = 0.02$) perturbations for DeepPruner and AANet achieve 31.20% and 48.08%, respectively, on PSMNet; whereas, FGSM achieves 55.47% and 53.55%, respectively. In general, iterative methods transfer less than FGSM because the perturbations tend to overfit to the model they were optimized for. We note that AANet is the most robust against perturbations from other models and yet has the highest transferability, which interestingly shows that transferability is not symmetric. While perturbations for DeepPruner and AANet achieve 31.20% and 48.08% on PSMNet, PSMNet and AANet achieve 13.74% and 22.25% on DeepPruner, and those for PSMNet and DeepPruner only achieve 8.46% and 5.81% on AANet.

DI^2-FGSM, MDI^2-FGSM. To increase transferability to other stereo networks, we additionally optimized perturbations using DI^2-FGSM, MDI^2-FGSM. Fig. 4-(b) shows that DI^2-FGSM (green) consistently degrades the target model's performance less than I-FGSM (blue). This is largely due to the noise in the gradients introduced by random resizing and padding. Fig. 4-(c) shows that perturbations from MDI^2-FGSM achieve errors similar to MI-FGSM since each iteration still retains the momentum from previous time steps.

Fig. 4-(b, c) shows that DI^2-FGSM and MDI^2-FGSM consistently transfer better to new models than I-FGSM and MI-FGSM (Fig. 5 for visualization). The best performing iterative method is MDI^2-FGSM, which achieves comparable numbers to MI-FGSM on the model it is optimized for, but also transfer well to new models. We note the trends observed in I-FGSM and MI-FGSM are also observed here.

While the mere existence of adversaries indicates (possible common) flaws in stereo networks, the reason that perturbations are transferable is because disparity is generic i.e. a surface 1m away generates the same disparity whether it



(a) (b) (c) (d)

Figure 6: *Defenses against attacks.* (a) Applying Gaussian blur using various σ on perturbed (top) and clean images (bottom). Gaussian blur does not destroy perturbations, but actually degrades performance for both perturbed and clean images. Finetuning (b) PSMNet, (c) DeepPruner and (d) AANet on FGSM perturbations and defending attacks against new FGSM adversaries with various ϵ. Fine-tuning on small ϵ perturbations makes the model robust against both existing and new adversaries without compromising performance on clean images.

belongs to a cat, a dog or a tree. Yet, transferability is not symmetric and AANet is yet again the most robust with the highest transferability. Fig. 5 shows that perturbations optimized for AANet fools DeepPruner, but those optimized for DeepPruner have less effect on AANet. We hypothesize that architectural differences between AANet (2D convolutions) and PSMNet and DeepPruner (3D convolutions) play a role in transferability. A possible reason for why perturbations for AANet transfers better to others (but less the other way around) may be because they are optimized to attack 2D convolutional layers, which PSMNet and Deeppruner also use to build their cost volumes. However, perturbations for PSMNet and Deeppruner are optimized to disrupt 3D convolutions as well, which are not present in AANet.

Defenses against Adversaries

We begin with a basic defense, Gaussian blur, against adversaries. Fig. 6-(a) shows Gaussian blurring (3×3 kernel) with various σ does not mitigate the effect of adversaries, but exacerbates them – further degrading performance. In addition, simply applying Gaussian blur on clean images also decreases performance. Hence, we aim to learn more robust representations by harnessing adversarial examples to fine-tune the models. Fig. 6-(b, c, d) show the performance of stereo methods after fine-tuning on a combination of clean and perturbed images (using FGSM with various ϵ). As a sanity check, we also fine-tuned on just clean images ($\star$) to ensure that differences are result of adversarial examples.

Adversarial data augmentation increases robustness for all models. For FGSM $\epsilon = 0.02$, PSMNet decreases error from 97.33% (Fig. 2) to 2.74% against the adversary it is trained on. Moreover, training on a smaller norm ($\epsilon = 0.002$) can increase robustness against larger norm ($\epsilon = 0.02$) attacks e.g. FGSM $\epsilon = 0.02$ can degrade PSMNet to only 27.03% error. Also the models are more robust against new adversaries. For this, we attacked each fine-tuned model and found that a new adversary (FGSM $\epsilon = 0.02$) can only degrade a PSMNet trained on FGSM $\epsilon = 0.02$ to 13.84% error and 23.74% when PSMNet is trained on FGSM $\epsilon = 0.002$. We also observe these trends in DeepPruner and AANet (Fig. 6-(c, d)).

Contrary to findings reported in classification (Goodfellow, Shlens, and Szegedy 2014; Kurakin, Goodfellow, and Bengio 2016), augmenting the training set with adversarial examples have little negative effect on performance of stereo models for clean images. When training with $\epsilon = 0.002$ (blue), performance for PSMNet and AANet are essentially unchanged (compared to $\star$); for the largest $\epsilon = 0.02$ (orange), errors increased by $\approx$0.4%. The smaller the norm, the less it affects performance on clean images. This is likely due to the mis-match in image intensity distributions between natural and adversarial examples. To avoid loss in performance, one can train on $\epsilon = 0.002$ and still observe the benefits on robustness. Fig. 6-(b, c, d) shows that all models are (i) robust to perturbations at $\epsilon = 0.002$ and 0.005, (ii) comparable to leveraging larger norm perturbations when facing new adversaries, and (iii) retains original performance on clean images.

While training on larger norms increases robustness against both existing and new adversaries, the model generally performs worse against a new adversary. For $\epsilon = 0.02$, a fine-tuned PSMNet achieves 2.74% against the adversary it is trained on and 13.84% against a new adversary; similarly, DeepPruner achieves 3.33% and 21.39% respectively. In contrast, when training on smaller norms ($\epsilon = 0.002$), the model keeps the same level robustness against existing and new adversaries. In fact, both PSMNet and DeepPruner perform better against new adversaries. For FGSM $\epsilon = 0.02$, PSMNet fine-tuned on $\epsilon = 0.002$ achieves 27.03% against the existing adversary and 23.74% against a new adversary; similarly, DeepPruner achieves 17.99% and 13.01%, respectively. This phenomenon is likely because the network is overfitting to the intensity patterns of the large norm noise (also related to the slight degrade in performance). But for small norms, the network learns the underlying pattern without needing to alter its decision boundaries significantly since the intensity distribution is closer to natural images – resulting in a more regularized model. Perhaps a strategy to learning robust stereo networks is to iteratively craft various small norm perturbations and train on them with a mixture of clean images.

Discussion

Stereo networks are indeed vulnerable to adversarial perturbations. This is unexpected because the problem setting is quite unlike adversarial perturbations for single image tasks where there is no unique minimizer (a single image does not constraint the latent, the training set does). Here, the geometry of the scene can be directly observed in co-visible regions as the data term is well defined and would have a unique minimizer under mild conditions. This means that stereo matching does not require learning; learning affects regularization. So it is surprising that, despite a uniquely identifiable latent variable (disparity), the training manages to produce such a strong bias that makes the overall system susceptible to perturbations, and local perturbations to boot. What is more interesting is that, not only can these perturbations drastically alter predictions on the stereo models they are optimized for, they can also transfer across models (although with reduced potency). However, given that it is rare for a malicious agent to have full access to a network and its loss, these attacks are not feasible in practice. Yet, the fact they exist gives us an opportunity to leverage them offline and train more robust stereo networks.

Previous works in single image based tasks have demonstrated that augmenting the training set with adversarial examples can improve robustness, but at the expense of performance on clean images. Yet, for stereo networks, we show that adversarial data augmentation can increase robustness *without* compromising performance on clean images – critical for designing robust and accurate systems. This too is likely related to the observability of the scene geometry from images where texture is sufficiently exciting. So, whereas in single image based tasks, training with adversarial perturbations alters the low-level filters to the point of hampering precision, in stereo precision is dictated by the data term, which is largely unaffected by training (correlation is an architectural inductive bias in deep stereo matching networks and is precisely why DeepPruner and AANet use it). While indeed, adversarial perturbations wreck havoc (reaching as much as 100% in D1-all error) on networks trained only on clean images, stereo networks can recover by learning the distribution of adversarial noise through data augmentation with adversarially perturbed images and the matching process takes care of the rest.

Our work here is just a first step. We only studied transferability across models and not datasets. We also do not consider the universal setting, where a constant additive image can degrade performance across all images within a dataset. Computationally, crafting perturbations using iterative methods adds an average of $\approx$29s on top of the time needed for forward passes; hence, they cannot be computed in real time. Amongst our findings, we also observed the "white-pixel" phenomenon, where very little perturbations are needed to alter regions with white pixels. This is an interesting phenomenon that is present across all methods. We believe this is due to white being on the upper support of image intensities; we leave the numerical analysis of this to future work. While there is still much to do, we hope that our work can lay the foundation for harnessing adversarial perturbations to train more robust stereo models.

Acknowledgements

This work was supported by ONR N00014-17-1-2072 and ARO W911NF-20-1-0158.

Ethical Impact

As deep learning models are widely deployed for various tasks, adversarial examples have been treated as a threat to the security of such applications. While demonstrating that adversaries exist for stereo seems to add to this belief (since stereo is widely used in autonomous agents), we want to assure the reader that these perturbations cannot (and should not) cause harm outside of the academic setting. Cameras are not the only sensors on an autonomous agent, they are generally equipped with range sensors as well. Hence, corrupting the depth or disparity map will not cause the system to fail since it can still obtain depth information from other sources. Also, as mentioned in our Introduction section, crafting these perturbations is computationally expensive and cannot be done in real time. Thus, we see little opportunities for negative ethical implications, but of course where there is a will there is a way.

More importantly, we see adversarial perturbations as a vehicle to develop better understanding of the behavior of black-box models. By identifying the input signals to which the output is most sensitive, we can ascertain properties of the map, as others have recently begun doing by using them to compute the curvature of the decision boundaries, and therefore the fragility of the networks and the reliability of their output.

What we want to stress in this work is that the mere existence of these perturbations tells us that there is a problem with the robustness of stereo networks. Therefore, we treat them as an opportunity to investigate and ultimately to improve stereo models. Our findings in our Defenses against Adversaries section shed light on the benefits of harnessing adversarial examples and potential direction towards more robust representations.

References

Chang, J.-R.; and Chen, Y.-S. 2018. Pyramid stereo matching network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5410–5418.

Damasio, A. R.; Damasio, H.; and Van Hoesen, G. W. 1982. Prosopagnosia: anatomic basis and behavioral mechanisms. *Neurology* 32(4): 331–331.

Dijk, T. v.; and Croon, G. d. 2019. How do neural networks see depth in single images? In *Proceedings of the IEEE International Conference on Computer Vision*, 2183–2191.

Dong, Y.; Liao, F.; Pang, T.; Su, H.; Zhu, J.; Hu, X.; and Li, J. 2018. Boosting adversarial attacks with momentum. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 9185–9193.

Duggal, S.; Wang, S.; Ma, W.-C.; Hu, R.; and Urtasun, R. 2019. Deeppruner: Learning efficient stereo matching via differentiable patchmatch. In *Proceedings of the IEEE International Conference on Computer Vision*, 4384–4393.

Geiger, A.; Lenz, P.; and Urtasun, R. 2012. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 3354–3361. IEEE.

Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572* .

Hendrik Metzen, J.; Chaithanya Kumar, M.; Brox, T.; and Fischer, V. 2017. Universal adversarial perturbations against semantic image segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, 2755–2764.

Horn, B. K.; and Schunck, B. G. 1981. Determining optical flow. In *Techniques and Applications of Image Understanding*, volume 281, 319–331. International Society for Optics and Photonics.

Ilyas, A.; Santurkar, S.; Tsipras, D.; Engstrom, L.; Tran, B.; and Madry, A. 2019. Adversarial examples are not bugs, they are features. In *Advances in Neural Information Processing Systems*, 125–136.

Kendall, A.; Martirosyan, H.; Dasgupta, S.; Henry, P.; Kennedy, R.; Bachrach, A.; and Bry, A. 2017. End-to-end learning of geometry and context for deep stereo regression. In *Proceedings of the IEEE International Conference on Computer Vision*, 66–75.

Kurakin, A.; Goodfellow, I.; and Bengio, S. 2016. Adversarial machine learning at scale. *arXiv preprint arXiv:1611.01236* .

Lowe, D. G. 1999. Object recognition from local scale-invariant features. In *Proceedings of the seventh IEEE international conference on computer vision*, volume 2, 1150–1157. Ieee.

Ma, Y.; Soatto, S.; Kosecka, J.; and Sastry, S. S. 2012. *An invitation to 3-d vision: from images to geometric models*, volume 26. Springer Science & Business Media.

Mayer, N.; Ilg, E.; Hausser, P.; Fischer, P.; Cremers, D.; Dosovitskiy, A.; and Brox, T. 2016. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4040–4048.

Menze, M.; and Geiger, A. 2015. Object scene flow for autonomous vehicles. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3061–3070.

Moosavi-Dezfooli, S.-M.; Fawzi, A.; Fawzi, O.; and Frossard, P. 2017. Universal adversarial perturbations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1765–1773.

Moosavi-Dezfooli, S.-M.; Fawzi, A.; and Frossard, P. 2016. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2574–2582.

Mopuri, K. R.; Ganeshan, A.; and Babu, R. V. 2018. Generalizable data-free objective for crafting universal adversarial perturbations. *IEEE transactions on pattern analysis and machine intelligence* 41(10): 2452–2465.

Naseer, M. M.; Khan, S. H.; Khan, M. H.; Khan, F. S.; and Porikli, F. 2019. Cross-domain transferability of adversarial perturbations. In *Advances in Neural Information Processing Systems*, 12905–12915.

Nguyen, A.; Yosinski, J.; and Clune, J. 2015. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 427–436.

Pang, J.; Sun, W.; Ren, J. S.; Yang, C.; and Yan, Q. 2017. Cascade residual learning: A two-stage convolutional neural network for stereo matching. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 887–895.

Peck, J.; Roels, J.; Goossens, B.; and Saeys, Y. 2017. Lower bounds on the robustness to adversarial perturbations. In *Advances in Neural Information Processing Systems*, 804–813.

Ranjan, A.; Janai, J.; Geiger, A.; and Black, M. J. 2019. Attacking optical flow. In *Proceedings of the IEEE International Conference on Computer Vision*, 2404–2413.

Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; and Fergus, R. 2013. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199* .

Tramèr, F.; Kurakin, A.; Papernot, N.; Goodfellow, I.; Boneh, D.; and McDaniel, P. 2017. Ensemble adversarial training: Attacks and defenses. *arXiv preprint arXiv:1705.07204* .

Wong, A.; Cicek, S.; and Soatto, S. 2020. Targeted Adversarial Perturbations for Monocular Depth Prediction. *arXiv preprint arXiv:2006.08602* .

Xie, C.; Wang, J.; Zhang, Z.; Ren, Z.; and Yuille, A. 2017a. Mitigating adversarial effects through randomization. *arXiv preprint arXiv:1711.01991* .

Xie, C.; Wang, J.; Zhang, Z.; Zhou, Y.; Xie, L.; and Yuille, A. 2017b. Adversarial examples for semantic segmentation and object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, 1369–1378.

Xie, C.; Zhang, Z.; Zhou, Y.; Bai, S.; Wang, J.; Ren, Z.; and Yuille, A. L. 2019. Improving transferability of adversarial examples with input diversity. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2730–2739.

Xu, H.; and Zhang, J. 2020. AANet: Adaptive Aggregation Network for Efficient Stereo Matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1959–1968.

Zagoruyko, S.; and Komodakis, N. 2015. Learning to compare image patches via convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4353–4361.

Žbontar, J.; and LeCun, Y. 2016. Stereo matching by training a convolutional neural network to compare image patches. *The journal of machine learning research* 17(1): 2287–2318.