Stereoscopic Universal Perturbations across Different Architectures and Datasets

Zachary Berger[†]
UCLA Vision Lab
zackeberger@g.ucla.edu

Parth Agrawal[†]
UCLA Vision Lab
parthagrawal24@g.ucla.edu

Tian Yu Liu
UCLA Vision Lab
tianyu139@g.ucla.edu

Stefano Soatto
UCLA Vision Lab
soatto@cs.ucla.edu

Alex Wong
UCLA Vision Lab
alexw@cs.ucla.edu

Abstract

We study the effect of adversarial perturbations of images on deep stereo matching networks for the disparity estimation task. We present a method to craft a single set of perturbations that, when added to any stereo image pair in a dataset, can fool a stereo network to significantly alter the perceived scene geometry. Our perturbation images are "universal" in that they not only corrupt estimates of the network on the dataset they are optimized for, but also generalize to different architectures trained on different datasets. We evaluate our approach on multiple benchmark datasets where our perturbations can increase the D1-error (akin to fooling rate) of state-of-the-art stereo networks from 1% to as much as 87%. We investigate the effect of perturbations on the estimated scene geometry and identify object classes that are most vulnerable. Our analysis on the activations of registered points between left and right images led us to find architectural components that can increase robustness against adversaries. By simply designing networks with such components, one can reduce the effect of adversaries by up to 60.5%, which rivals the robustness of networks fine-tuned with costly adversarial data augmentation. Our design principle also improves their robustness against common image corruptions by an average of 70%.

1. Introduction

Deep neural networks are vulnerable to adversarial perturbations, where small changes in the input image(s) can cause large inference errors, for instance in the label of objects portrayed within. Even when the images contain sufficient information for inference, for instance in stereo where the disparity between two calibrated images is used to infer the depth of the scene, adversarial perturbations have been shown to alter the depth map [44]. Such perturbations are ordinarily specific to each individual input, and depend on the particular deep network architecture and the particular dataset on which it is trained.

For classification, [22] showed that a single perturbation can be crafted to disrupt the inference for all images in a dataset with high probability. These are called "universal" adversarial perturbations, even though they are universal to each image in a particular dataset, and usually do not extend to different datasets. In this paper, we show the existence of stereoscopic universal perturbations (SUPs). SUPs are perturbations that can disrupt the depth or disparity estimate of different stereo networks, with different architectures, trained on different datasets, and operating on different images and domains.

Adversarial perturbations arose mainly as a means to study the topology and geometry of the decision boundary of deep networks. Since individual perturbations had to be crafted for each image, security concerns were far fetched. Universal adversarial perturbations, however, revealed vulnerabilities that could be shared among different images. Still, crafting them required knowledge of the architecture and availability of the training set. In contrast, the existence of universal adversarial perturbations for stereo and other spatial inference tasks, common in robotics and autonomy, suggests that such perturbations could present a concern, especially if they can be applied to different images, processed by different neural network models that are trained on different datasets. To the best of our knowledge, we are the first to show, for stereo, that universal perturbations can be applied effectively *even without* knowledge of the trained model, and generalize across domains and datasets. Such perturbations can be optimized on an off-the-shelf model and realized as a filter to be placed on top of a camera lens.

Our main methodological innovation is to design SUPs so that they are approximately space equivariant. We build

[†] denotes authors with equal contributions.
Code: github.com/alexklwong/stereoscopic-universal-perturbations

Figure 1. *Universality across models and datasets*. We optimized a single pair of perturbation images for AANet on the KITTI dataset. When added to a stereo pair from KITTI 2015, it corrupts the disparity estimates of AANet and PSMNet. The same perturbations can be added to stereo pairs in Flyingthings3D to fool AANet and DeepPruner.

the perturbation out of a single tile, repeated periodically. Although the tile is not constrained to have periodic boundary conditions, we notice that the model learns perturbations where boundary artifacts are not obvious, partly because the perturbation itself is designed to be small enough to be quasi-imperceptible. Our design naturally regularizes the tile with data, allowing it to generalize to different image pairs, processed with different architectures trained with different datasets – increasing error from 1% to as much as 87% when added to network inputs.

In our experiments, we observe that the errors in disparity induced by SUPs are more pronounced on certain classes of objects. We conjecture that this is due to said classes exhibiting more homogeneous regions, which are more prone to errors in disparity due to ambiguity. We also found that there is a systematic bias towards closer distance (larger disparity) after perturbations. To study the effect of SUPs on stereo networks, we investigate the activations of left and right feature maps before and after adding perturbations. We validate empirically that the embedding function amplifies the adversarial signal: The embedding of perturbed registered features between the images grows more uncorrelated throughout a forward pass than the embedding of the original or "clean" registered features, which "fools" a stereo network into estimating incorrect correspondences.

Moreover, we use SUPs to improve robustness in stereo networks. We study the effect that different architectural elements (deformable convolutions, and explicit matching modules) have on mitigating perturbations. We observe that by simply designing networks with these elements (and following standard training protocols), one can reduce the effect of adversaries to a similar degree as fine-tuning a model (that lacks such elements) with adversarial data augmentation. While robustness is increased with fine-tuning, it come at a significant cost in time and compute. In contrast, the proposed architectural design choices can mitigate attacks (60.5% error reduction), and only require a few lines of code; they also improve robustness against common image perturbations i.e. lossy compression, noise, blur by an average of 70%. Conclusions are valid for three different architectures, across three datasets. While these are chosen to represent the variety in use today, we cannot exclude that there could be tasks, data and models on which our method to craft perturbations is ineffective, and conversely perturbations that are not mitigated by the methods we propose.

Our contributions include: (i) The design of the first stereoscopic universal perturbations (SUPs) that can not only fool the network they are optimized for, but also other networks across multiple datasets. We perform an empirical analysis on how SUPs affect (ii) the estimated scene geometry, (iii) different object classes, and (iv) the features of registered points in a stereo pair. Our results shed light on how SUPs fool stereo networks and led us to uncover (v) architectural designs, i.e. deformable convolution and explicit feature matching, that mitigate the effect of SUPs to a similar degree as fine-tuning on them. A discussion of potential negative societal impact is available in Supp. Mat.

2. Related Work

Adversarial perturbations. [41] showed that small additive signals can significantly alter the output of a classification network. [11] introduced the fast gradient sign method (FGSM). [8, 17, 19] extended FGSM to iterative optimization to boost its potency. [23] found the minimal perturbation to alter the predicted class while [32] computed the lower bounds on the perturbation magnitudes required to fool a network. [29] showed that unrecognizable noise can yield high confidence outputs and [15] attributed adversaries to non-robust features. [50] improved their transferability across networks with geometric image augmentations. [28] studied their transferability across datasets.

[22] proposed *universal* adversarial perturbations, where the same perturbation can be added to any image in a dataset to fool a network. [24] showed that data independent universal perturbations are transferable across different networks and [25] proposed data-free objectives for craft-

ing them. [13, 26, 33] use generative models to form universal perturbations. [36] proposed universal attacks on graphs, meshes, and point clouds. For those interested, see [5] for an extensive survey. We also study universal perturbations, but unlike past works focused on single image based problems, we consider the deep stereo matching, where the latent variable (disparity) is constrained by the stereo pair.

Efforts to defend against adversarial attack include adversarial data augmentation during training [17, 42], which can be improved with randomization [48]. [27, 40] proposed universal adversarial training, [3, 46] gradient discretization, and [31, 35, 45] randomization. [1, 12, 34, 38] performed purification, and [18] denoising to rectify the image. [47] used batch normalization to mitigate perturbations. [6] used adversarial learning to improve object detection.

Despite many works on adversarial perturbations for classification, few study dense-pixel prediction tasks e.g. segmentation, optical flow, depth estimation. [49] showed adversarial perturbations for object detection and segmentation. [14] proposed universal perturbations for segmentation, while [25] studied them for segmentation and single image depth. [43] showed targeted attacks for single image depth while [7] studied them using images augmented with synthetic vehicles. [57] examined translucent patch attacks for object detection, and [37] visible patch attacks on optical flow. [39] proposed defenses against physical attacks for optical flow. [44] demonstrated adversarial attacks for stereo. Like [44], we also consider stereo, but instead, we study universal perturbations and show that the same perturbations generalize across network architectures and datasets.

Deep Stereo Matching. Early works [52, 53] replaced hand-crafted features with deep features for more robust matching. Recent works realize the entire stereo pipeline as an inductive bias, from feature extraction to cost matching, into 2D and 3D network architectures. 2D architectures leverage correlation layers for matching. For instance, [20] formed a 2D cost volume with correlation over left and right features. [30] extended [20] to a cascade residual learning framework. AANet [51] also used correlation, but proposed deformable convolutions [55] when performing cost aggregation to avoid sampling at discontinuities. 3D architectures use feature concatenation and sparse patch matching. [16] concatenated left and right features together to build a 3D cost volume. PSMNet [4] added spatial pyramid pooling layers and introduced a stacked hourglass architecture. [54] used local and global cost aggregation. DeepPruner [9] proposed differentiable patch matching over deep features to construct their cost volume.

We demonstrate the existence of universal adversarial perturbations on PSMNet, DeepPruner and AANet. We chose them as architectural exemplars for the stereo matching task. PSMNet represents the modern stereo networks (stacked hourglass, cost volume, 3D convolutions), but uses feature stacking without explicit matching. DeepPruner follows the architecture of PSMNet, but performs explicit matching with PatchMatch [2]. AANet represents the state of the art in 2D architecture and uses correlation.

3. Universal Perturbations for Stereo

Formulation. Let $f_\theta(x_L, x_R) \in \mathbb{R}^{H\times W}$ be a pre-trained stereo network that estimates the disparity between the left x_L and right x_R images of a stereo pair and $\mathcal{X}$ be a distribution of stereo pairs that belongs to the set of natural images. Our goal is to craft a single pair of image-agnostic stereoscopic universal perturbation images (SUPs) $v_L, v_R \in [0,1]^{H\times W\times 3}$ that, when added to (x_L, x_R), corrupts the disparity estimate such that $f_\theta(x_L, x_R) \neq f_\theta(\hat{x}_L\ ,\ \hat{x}_R)$ where $\hat{x}_L = x_L + v_L$ and $\hat{x}_R = x_R + v_R$ for $(x_L, x_R) \sim \mathcal{X}$. To ensure that the SUPs are small or quasi-imperceptible, we subject them to the norm constraints $\|v_I\|_\infty \leq \epsilon$ for $I \in \{L, R\}$.

We assume a dataset $X := \{(x_L^{(n)}, x_R^{(n)}), y_{gt}^{(n)}\}_{n=1}^N$ sampled from $\mathcal{X}$ as a "training" set, and access to a stereo network f_θ and its loss function $\ell(f_\theta(\cdot), y_{gt})$ where y_{gt} denotes the ground truth. We note that, unlike classification or segmentation, it is rare for any large scale stereo dataset to provide ground truth for every sample, so instead we use disparity estimated from "clean" stereo pairs, i.e. without any perturbations, as pseudo ground truth, $y^{(n)} = f_\theta(x_L^{(n)}, x_R^{(n)})$.

Algorithm. To craft universal perturbations subject to the norm constraint of $\|v_I\|_\infty \leq \epsilon$, we propose to generate (v_L, v_R) by iterating through X and gradually aggregating small perturbation vectors that are able to fool the stereo network f_θ into altering its output disparity or the perceived scene geometry for a given image pair $(x_L^{(n)}, x_R^{(n)}) \in X$. At each iteration, we compute the gradient of the loss ℓ with respect to each image x_I in the stereo pair for $I \in \{L, R\}$:

$$g_I^{(n)} = \nabla_{x_I^{(n)}} \ell(f_\theta(\hat{x}_L^{(n)}, \hat{x}_R^{(n)}), y^{(n)}). \tag{1}$$

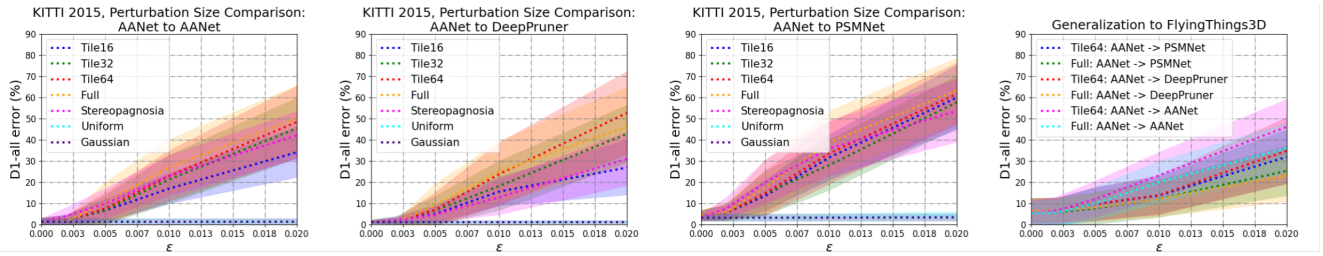
Then, project it onto a smaller (than ϵ) ball with radius α (akin to a learning rate) via the projection operator[1] and aggregate it to the current perturbations:

$$v_I = v_I + \mathbf{p}_{\infty,\alpha}(g_I^{(n)}). \tag{2}$$

Finally, we project v_I onto the ϵ radius ball after each iteration to ensure our perturbations meet the upper norm constraint. The procedure is repeated for all stereo pairs in X. See Alg. 1 for more details.

Towards universality across model and data. We aim to optimize a single pair of perturbations that can alter the perceived geometry of a scene, not just for the network and dataset it is optimized for, but for an array of different unseen network architectures across multiple datasets. To this

[1] $\mathbf{p}_{p,\xi}(v) = \arg\min_{v'} \|v - v'\|$ subject to $\|v'\|_p < \xi$



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

Figure 2. *The effect of perturbation size*. All methods are robust to naive attacks of uniform and Gaussian noise, as performance is constant across ϵ. Hence, we optimize a pair of perturbations on KITTI for AANet and attack (a) AANet, (b) DeepPruner, and (c) PSMNet. Amongst the perturbation sizes, full and 64×64 are the most effective at degrading performance on KITTI 2015 validation set. In (d), we show transferability to FlyingThings3D by using the same perturbations optimized on KITTI from (a)-(c) to attack models trained on Scene Flow. The 64×64 perturbations generalize the best across datasets.

Algorithm 1 Computing universal perturbations.

Parameters: Upper norm ϵ, learning rate α.
Inputs: Dataset X, pre-trained stereo network f_θ.
Outputs: Perturbations v_L, v_R.
Initialize: $v_L = \mathbf{0}, v_R = \mathbf{0}$.
for each stereo pair $(x_L^{(n)}, x_R^{(n)}) \in X$ **do**
  Compute $g_L^{(n)}$ and $g_R^{(n)}$ as defined in Eqn. 1
  $v_L = \mathbf{p}_{\infty,\epsilon}(v_L + \mathbf{p}_{\infty,\alpha}(g_L^{(n)}))$
  $v_R = \mathbf{p}_{\infty,\epsilon}(v_R + \mathbf{p}_{\infty,\alpha}(g_R^{(n)}))$
end for

end, the perturbations must be spatially invariant to generalize across different scene distributions i.e. roads are commonly at the center of the image for outdoor driving scenarios, but a variety of shapes may populate the same region for indoors. Hence, rather than optimizing (v_L, v_R) that span the full $H \times W$ image domain, we reduce (v_L, v_R) to a pair of $h \times w$ patches or tiles subject to $h \mid H$ and $w \mid W$. We note that full size $H \times W$ perturbations are a special case.

To apply (v_L, v_R) to (x_L, x_R) over the entire image space, we evenly repeat or tile the perturbation v_I across x_I with no overlap. Formally, we let $x_I(i,j)$ be the $h \times w$ image region that spans from pixel position (i,j) to $(i+h, j+w)$ for $i \in \{0, \frac{H}{h}, ..., \frac{H(h-1)}{h}\}$ and $j \in \{0, \frac{W}{w}, ..., \frac{W(w-1)}{w}\}$. Thus, the perturbed image region is:

$$\hat{x}_I(i,j) = x_I(i,j) + v_I \;\; \forall \;\; i,j. \tag{3}$$

We now modify the the gradient computation step in Alg. 1 for a given stereo pair $(x_L^{(n)}, x_R^{(n)})$ by taking the mean over the gradient with respect to the image $g_I^{(n)}$ for all tiles

$$\bar{g}_I^{(n)} = \frac{h \cdot w}{H \cdot W} \sum_{i,j} g_I^{(n)}(i,j). \tag{4}$$

In doing so, we prevent the perturbations from overfitting to the bias in scene structures induced by the training set e.g. road on bottom of the image and sky on top. We demonstrate in Sec. 4 that this approach yields a single set of universal perturbations that can fool different models across multiple datasets. We note that we can extend our approach to patch attacks by adding the perturbations anywhere on the image, instead of tiling across the image. However, because we constrain our perturbations to be within a small ϵ ball, unlike [37], a visually imperceptible patch attack is limited in its effect on fooling the network.

4. Experiments and Results

We optimized our SUPs on the KITTI raw dataset [10] and evaluated them on KITTI 2012, KITTI 2015 [21] for stereo models [4, 9, 51]. We also show that the same SUPs generalize to FlyingThings3D [20] to disrupt models trained on Scene Flow [20]. Please see Supp. Mat. for details on datasets, hyper-parameters and implementation.

On the effect of perturbation size. We optimize SUPs on AANet using square tiles of $16, 32$, and 64, and the full image size of 256×640. We report results in Fig. 2, which shows the performance of each network on KITTI 2015 when attacked by these perturbations. We compare our results against [44] which uses image-specific perturbations generated with FGSM. We additionally consider two naive attacks that perturb the input stereo pair (x_L, x_R) with uniform $\mathcal{U}(-\epsilon, \epsilon)$ and Gaussian $\mathcal{N}(0, (\epsilon/4)^2)$ noise.

Fig. 2-(a, b, c) show that naive attacks have little effect on stereo networks, as the D1-error is roughly constant for all ϵ. Hence, stereo networks are robust to naive perturbations within ϵ upper norm, and fooling them is non-trivial. Among all square tiles, 64×64 causes the largest error for all networks across all ϵ. We note that, although our SUPs are image-agnostic, we are comparable to [44] on small norms and beat them on larger norms. Fig. 3 shows the 64×64 tiles optimized on KITTI for AANet, DeepPruner, and PSMNet. When added to a stereo pair from KITTI 2015, the disparity estimated by each network is corrupted.

For FlyingThings3D, we consider the full and 64×64 SUPs (both trained on KITTI) as they caused the most corruption on KITTI 2015. Fig. 2-(d) shows that 64×64 generalizes better than full-size SUPs across networks. For $\epsilon = 0.02$, 64×64 achieves 46.14% error on AANet, 34.87%

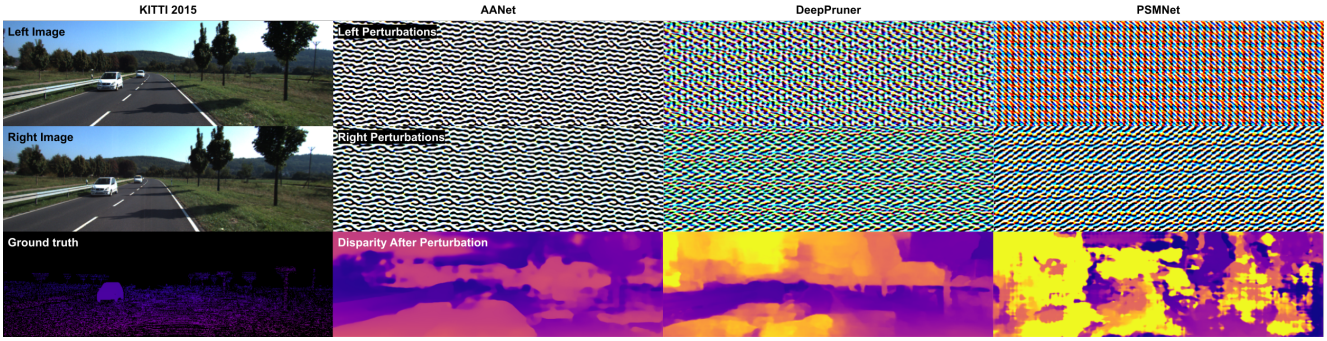
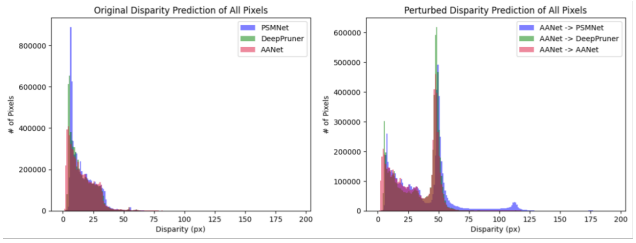


Figure 3. *Attacking stereo networks*. We visualize 64×64 perturbations (tiled across the image domain) optimized for AANet, DeepPruner, and PSMNet on the KITTI dataset. When added to the inputs of the network for which they were optimized, the perturbations can corrupt the estimated disparities. Note: the damage is concentrated on textureless regions e.g. sky, road.



(a) (b)

Figure 4. *Distribution of disparities before and after adding perturbations*. (a) Before adding perturbations, most of the scene is estimated to be ≈2 disparities. (b) The perturbations fool the networks into predicting larger (≈50) disparities.

on DeepPruner and 31.93% on PSMNet, while full achieves 36.09%, 23.28%, and 25.35%, respectively. Thus, our tiling approach can help generalize to different data distributions. As our goal is universality across architectures and datasets, we use 64×64 perturbations for the rest of our experiments.

Generalization across architectures and datasets. We optimized three sets of 64×64 SUPs on KITTI for AANet, DeepPruner, and PSMNet, respectively. In Fig. 5, we attack each network with each set of SUPs on three datasets. We report D1-error for KITTI 2012, and 2015, and EPE for FlyingThings3D (see Supp. Mat. for results for KITTI 2012). For KITTI 2015, Fig. 5-(a) shows that, when trained on the network to be attacked, SUPs with $\epsilon = 0.02$ cause error to rise from 1.47% to 48.43% for AANet, 1.28% to 52.74% for DeepPruner, and 4.25% to 87.72% for PSMNet. While SUPs with $\epsilon = 0.002$ have negligible impact, relaxing ϵ to 0.005 increases the error of AANet to 7.62%, DeepPruner to 8.90%, and PSMNet to 28.97%. Fig. 5-(b) shows that SUPs generalize to other data distributions as well. Adding SUPs optimized on KITTI to FlyingThings3D causes increases in EPE for models trained on Scene Flow – from 1.30px to 9.47px for AANet, 1.25px to 14.77px for DeepPruner, and 1.27px to 18.88px for PSMNet.

For all three datasets, our SUPs also generalize across architectures. For example, SUPs with $\epsilon = 0.02$ optimized for AANet on KITTI can be added to stereo pairs in KITTI 2015 to fool DeepPruner (from 1.28% to 52.66%), and PSMNet (from 4.25% to 61.66%). Similarly, the same SUPs can be added to images in FlyingThings3D to corrupt the outputs of PSMNet (from 1.27 to 6.86px) and DeepPruner (1.25 to 6.60px). Yet, transferability is not symmetric e.g. SUPs optimized for DeepPruner on KITTI only corrupt AANet predictions from 1.30 to 4.49px on Flyingthings3D. Fig. 8 demonstrates the transferability qualitatively, showing corruption against PSMNet on FlyingThings3D.

In our experiments, we found AANet to be the most robust and PSMNet the least. We hypothesize that explicit matching plays a role because DeepPruner shares the same architecture as PSMNet, with the exception of a PatchMatch module, but is significantly more robust. Like DeepPruner, AANet also employs matching, but replaces convolutions with deformable convolutions – we explore the use of these architectural designs as a defense in Sec. 5.

Effect on scene geometry. To quantify how SUPs affect the estimated scene geometry, we compare the disparities estimated for "clean" (no added perturbations, Fig. 4-(a)) and perturbed (optimized on AANet, Fig. 4-(b)) stereo pairs. Fig. 4 shows that the peak of the distribution shifts from ≈2 to ≈50px for all three networks. For PSMNet, we see an additional mode at ≈110px. Depth and disparity are inversely related, so the SUPs fool the network to predict the scene to be closer to the camera. We observe similar trends for DeepPruner and PSMNet (see Supp. Mat.).

Robustness of semantic classes. To analyze their effect on objects populating the scene, we use SDCNet [56] to obtain segmentation maps for the KITTI 2015 validation set. We measure the per class error and found that different semantic classes exhibit different levels of robustness against adversaries. Specifically, Fig. 6 shows that *sky* and *vegetation* are the least robust with 72.96% and 58.52% D1-error, respectively; whereas, *truck* (14.19%) and *car* (16.82%) are the most robust. We observe that the least robust classes are largely homogeneous. We conjecture that these regions are most vulnerable because locally they give little to no information about scene structure, which leads to ambigu-

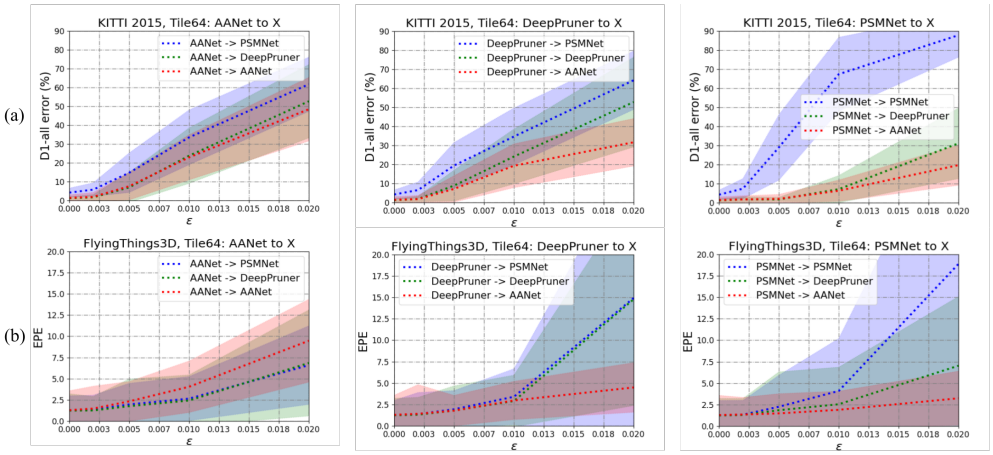


Figure 5. *Generalization across architectures and datasets*. Perturbations were optimized for AANet, DeepPruner, and PSMNet on KITTI and added to stereo pairs of KITTI 2015, and FlyingThings3D. Despite being optimized for a specific model on KITTI, they can corrupt models trained on KITTI for KITTI 2015 and those trained on Scene Flow for FlyingThings3D.

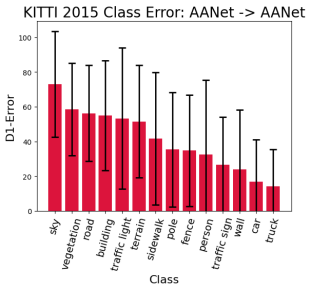


Figure 6. *D1-error for each semantic class for perturbed images*. Each class exhibit different levels of robustness. Homogeneous regions (sky, vegetation) are most vulnerable.

ity when registering points between two images – this is in contrast to sufficiently textured regions where unique correspondences are to be found. Thus, the network must rely on the regularizer (stored in the weights) learned from the training set to fill in the disparity for homogeneous regions.

Effect on feature maps. As DeepPruner and AANet use explicit matching to form their cost volume, there is a well-defined measure of data-fidelity to register the left and right images. So, to alter disparity, SUPs must corrupt the features used in the matching process. Hence, to quantify their effect, we measure the correlation between left and right feature maps before and after perturbing the images.

Let $f_\theta^{(l)}$ be the l-th layer of the encoder shared between the stereo pair and $u \in \Omega$, the image domain. To quantify how SUPs corrupt the feature maps, we compute the correlation between $f_\theta^{(l)}(x_I(u))$ and $f_\theta^{(l)}(\hat{x}_I(u))$ for all l. Fig. 7-(a, b) shows that when SUPs optimized for AANet are added to the input, the correlation between clean and perturbed left and right features grow uncorrelated from 1 to 0.76 during a forward pass i.e. the embedding function amplifies the effect of perturbation. We observe similar trends for DeepPruner and PSMNet (see Supp. Mat.).

While the observations in Fig. 7-(a, b) may be sufficient to fool a classification network, i.e. the feature maps are transformed across a decision boundary, it is not sufficient for stereo matching. To fool a stereo network, SUPs must alter the correspondences between left and right image. In other words, for a pair of registered points $x_L(u)$ and $x_R(u - y_{gt}(u))$, where $y_{gt} \in \mathbb{R}^{H \times W}$ is the true disparity, the perturbations must cause the features of these similar points in the image to be dissimilar in embedding space. To quantify this, we first compute the correlation between the registered clean stereo pair $f_\theta^{(l)}(x_L(u))$ and $f_\theta^{(l)}(x_R(u - y_{gt}(u)))$ in Fig. 7-(c). As expected the feature maps of the registered points are well correlated. In Fig. 7-(d), we compute the correlation between the registered perturbed stereo pair $f_\theta^{(l)}(\hat{x}_L(u))$ and $f_\theta^{(l)}(\hat{x}_R(u - y_{gt}(u)))$. Indeed, the registered perturbed feature maps grow uncorrelated relative to the clean feature maps in the forward pass i.e. the perturbations cause similar regions in the RGB do-

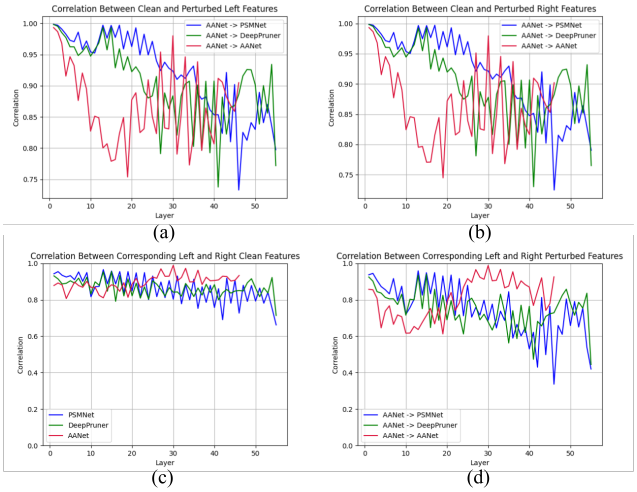


Figure 7. *Effect on features*. Clean and perturbed left (a) and right (b) features grow uncorrelated. Features of clean stereo pairs are correlated (c), but after perturbation, become uncorrelated (d).

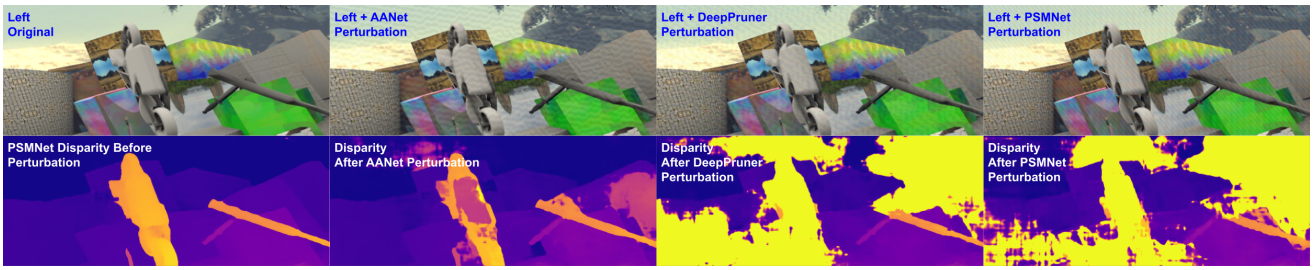
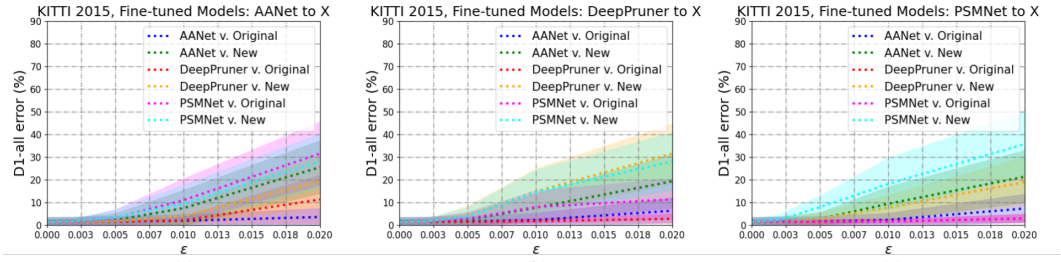


Figure 8. *Transferability to PSMNet*. Stereoscopic universal perturbations optimized on KITTI for AANet, DeepPruner, and PSMNet can generalize to stereo pairs in FlyingThings3D to corrupt the disparity estimation of PSMNet.



(a) (b) (c)

Figure 9. *Adversarial data augmentation*. AANet, DeepPruner, and PSMNet were fine-tuned with adversarial data augmentation. Each model was attacked with a perturbation trained for the original and fine-tuned AANet (a), DeepPruner (b), and PSMNet (c). Fine-tuning with adversarial data augmentation is an effective defense against SUPs trained for the original model, but does not fully mitigate a new adversary. Fine-tuning a model on SUPs optimized for it increases robustness against perturbations optimized for different architectures.

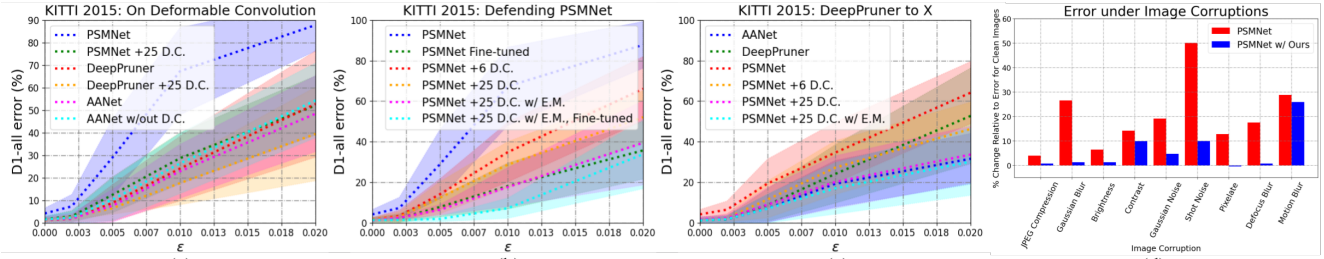
main to be dissimilar in the embedding space, resulting in incorrect points being matched. Note that, like in Fig. 7-(a) and Fig. 7-(b), correlation between left and right AANet features increases from layer 20 to 30; this coincides with deformable convolutions. We conjecture that this may be related to AANet's relative robustness against adversaries.

5. Towards Robust Deep Stereo Networks

Adversarial data augmentation. As shown in [44], fine-tuning with adversarial data augmentation is among the best performing defenses for stereo. Hence, we first fine-tuned each pretrained stereo model on KITTI 2015 with SUPs of $\epsilon \in \{0.002, 0.005, 0.01, 0.02\}$ trained for the model. The SUPs were randomly added the inputs with 50% probability. In Fig. 9, we attack each fine-tuned model with a perturbation trained for the original and fine-tuned variant of each architecture. Fig. 9 shows that adversarial data augmentation improves the robustness of each model. When attacked by the SUPs it is fine-tuned on, AANet reduces in error from 48.43% to 3.62%, DeepPruner from 52.74% to 2.83%, and PSMNet from 87.72% to 2.96% for $\epsilon = 0.02$. New adversaries optimized for the fine-tuned networks are less effective, with AANet dropping to 25.54% error, DeepPruner to 31.54%, and PSMNet to 35.75%.

Fig. 9 also shows that fine-tuning a model on SUPs optimized for it increases robustness against SUPs optimized for different architectures. For example, Fig. 9-(a) shows that fine-tuning reduces AANet in error from 48.43% to 3.62%, DeepPruner from 52.74% to 11.32%, and PSMNet from 87.72% to 31.54% when attacked by SUPs optimized for the original AANet with $\epsilon = 0.02$. Note that AANet has the lowest error against the original AANet adversary because it was fine-tuned on that perturbation, whereas DeepPruner and PSMNet are seeing it as a "new" adversary. Similar results are shown for SUPs optimized for DeepPruner (Fig. 9-(b)) and PSMNet (Fig. 9-(c)). We note that this process of optimizing SUPs and fine-tuning on them is time consuming, and the resulting networks are not fully robust.

On explicit matching (EM) and deformable convolution (DC). Instead, we propose to make simple modifications to the design of stereo networks. From our observations, EM increases robustness as PSMNet (no explicit matching) is more vulnerable than AANet and DeepPruner. Fig. 7 shows that the effect of SUPs is amplified by the embedding function, which ultimately fools the network; we conjecture that EM mitigates this by explicitly registering correspondences based on similarity rather than propagating the local signal. This intuition extends to DCs that learn convolutional offsets to regions locally similar to the element being convolved over and in effect "avoids" the adversarial signal – Fig. 7 shows an increase in AANet's feature correlation that coincides with DCs. While the intent of DC is to minimize artifacts e.g. over-smoothing along occlusion boundaries by sampling features that are robust to local deformation and respect boundary conditions, we hypothesize that this filters out the perturbation signal that causes dissimilarities within a patch, and is the reason DC (and EM) allows AANet to be more robust.



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

Figure 10. *Improving robustness against perturbations*. In (a, b), each variant was attacked by perturbations optimized specifically for them on image pairs. (a) Adding deformable convolution (DC) to PSMNet and DeepPruner improves their robustness, while removing it from AANet decreases its robustness. The least robust model, PSMNet, can achieve comparable performance to the most robust model, AANet, when using DC and explicit matching (EM). (b) Adding DC and EM to PSMNet achieves comparable results to adversarial training. (c) PSMNet with DC and EM is more robust than AANet under black-box attack, where the adversary was optimized for a different model (DeepPruner). (d) By applying our design principles to PSMNet, we improve its robustness to common image corruptions by ≈70%.

To assess DC as an inherent defense against adversarial perturbations, we trained (i) PSMNet and (ii) DeepPruner, each with 25 DCs, and (iii) AANet without DCs. We optimized six pairs of SUPs on KITTI for vanilla AANet, DeepPruner, PSMNet, and their variants. In Fig. 10-(a), we show results on KITTI 2015, where each network was attacked by SUPs optimized specifically for them. We found that DCs do improve robustness as both DeepPruner and PSMNet produced lower D1-errors across all norms. For $\epsilon = 0.02$, DeepPruner drops in error from 52.74% to 39.47%, and PSMNet drops from 87.72% to 52.10%. Conversely, replacing DCs with regular convolutions can make a model more susceptible to adversaries – AANet without DCs is less robust, as D1-error increase from 48.43% to 54.32%. In summary, simply designing a network with DC, the least robust model, PSMNet, can become comparable in performance to AANet, the most robust model.

Next, we assessed how EM and DC compare to adversarial training. We trained variants of PSMNet with (i) 6 DCs, (ii) 25 DCs, and (iii) 25 DCs and EM i.e. DeepPruner with 25 DCs. We performed adversarial fine-tuning on PSMNet and (iii). In Fig. 10-(b), we observe for $\epsilon = 0.02$ that increasing the number of DCs and then adding explicit matching drops the D1-error of PSMNet from 87.72% to 66.10%, 52.10%, and finally to 39.47%. PSMNet fine-tuned on SUPs also performs well; however, with a D1-error of 35.75%, it only marginally beats PSMNet with 25 DCs and explicit matching on $\epsilon = 0.02$. For all other norms, the two are comparable. The best performing variant is PSMNet with DCs and EM fine-tuned on SUPs, achieving D1-error of 33.85%. So, simply using DCs and EM and following standard training protocols can yield more robust networks with no explicit intent for defense. Moreover, they can also be used in conjunction with existing defenses (i.e. adversarial fine-tuning) to yield even more robust networks.

In Fig. 10-(c), we simulate the realistic black-box scenario where an attacker does not have access to a network (PSMNet or AANet) and crafts SUPs with an off-the-shelf model (DeepPruner). Replacing convolutions in PSMNet with DCs leads to immediate improvements in robustness with no loss in accuracy on clean images. With just 6 DCs, PSMNet becomes more robust than DeepPruner and with 25 it is on par with AANet. Incorporating PatchMatch into PSMNet (i.e. DeepPruner) with 25 DCs improves it to the most robust method. Note: fine-tuning on SUPs as data augmentation can further improve its robustness (Fig. 10-(b)).

Designing networks with DCs and inductive biases like EM not only improves robustness against SUPs, but also against common image corruptions i.e. lossy compression, blur and noise. Fig. 10-(d) shows that PSMNet (red) is susceptible to blurring and shot noise where the latter can increase error by 50%. Our design improves its robustness across all common corruption. Particularly, Gaussian and defocus blur, and pixelation have little effect – we improve by as much as 80% on shot noise and 70% on average.

6. Discussion

Stereoscopic universal perturbations (SUPs) exist and can generalize across architectures and datasets. SUPs can be partly mitigated by fine-tuning with adversarial data augmentation. However, doing so is costly in time and compute. Instead, we propose to address the robustness problem starting from the the design of deep networks. We have identified architectural elements, i.e. deformable convolutions and explicit matching, which can be easily incorporated into stereo networks with few lines of code and trained with standard protocol. The resulting networks are comparable in robustness and performance to those without these elements, but fine-tuned on adversarial examples. Admittedly, SUPs do not exist in nature; nonetheless, our design is also applicable to common image corruptions. While the our scope is limited to stereo, many geometry problems i.e. optical flow share similar architectural designs. So we hope this work can contribute to robust systems in related fields.

Acknowledgements. We thank ARL W911NF-20-1-0158, ONR N00014-19-1-2229 and ARO W911NF-17-1-0304.

References

[1] Naveed Akhtar, Jian Liu, and Ajmal Mian. Defense against universal adversarial perturbations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3389–3398, 2018. 3

[2] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. Patchmatch: A randomized correspondence algorithm for structural image editing. *ACM Trans. Graph.*, 28(3):24, 2009. 3

[3] Jacob Buckman, Aurko Roy, Colin Raffel, and Ian Goodfellow. Thermometer encoding: One hot way to resist adversarial examples. In *International Conference on Learning Representations*, 2018. 3

[4] Jia-Ren Chang and Yong-Sheng Chen. Pyramid stereo matching network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5410–5418, 2018. 3, 4

[5] Ashutosh Chaubey, Nikhil Agrawal, Kavya Barnwal, Keerat K Guliani, and Pramod Mehta. Universal adversarial perturbations: A survey. *arXiv preprint arXiv:2005.08087*, 2020. 3

[6] Xiangning Chen, Cihang Xie, Mingxing Tan, Li Zhang, Cho-Jui Hsieh, and Boqing Gong. Robust and accurate object detection via adversarial learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16622–16631, 2021. 3

[7] Tom van Dijk and Guido de Croon. How do neural networks see depth in single images? In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2183–2191, 2019. 3

[8] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9185–9193, 2018. 2

[9] Shivam Duggal, Shenlong Wang, Wei-Chiu Ma, Rui Hu, and Raquel Urtasun. Deeppruner: Learning efficient stereo matching via differentiable patchmatch. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4384–4393, 2019. 3, 4

[10] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3354–3361. IEEE, 2012. 4

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

[12] Chuan Guo, Mayank Rana, Moustapha Cisse, and Laurens Van Der Maaten. Countering adversarial images using input transformations. *arXiv preprint arXiv:1711.00117*, 2017. 3

[13] Jamie Hayes and George Danezis. Learning universal adversarial perturbations with generative models. In *2018 IEEE Security and Privacy Workshops (SPW)*, pages 43–49. IEEE, 2018. 3

[14] Jan Hendrik Metzen, Mummadi Chaithanya Kumar, Thomas Brox, and Volker Fischer. Universal adversarial perturbations against semantic image segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2755–2764, 2017. 3

[15] Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. In *Advances in Neural Information Processing Systems*, pages 125–136, 2019. 2

[16] Alex Kendall, Hayk Martirosyan, Saumitro Dasgupta, Peter Henry, Ryan Kennedy, Abraham Bachrach, and Adam Bry. End-to-end learning of geometry and context for deep stereo regression. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 66–75, 2017. 3

[17] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial machine learning at scale. *arXiv preprint arXiv:1611.01236*, 2016. 2, 3

[18] Fangzhou Liao, Ming Liang, Yinpeng Dong, Tianyu Pang, Xiaolin Hu, and Jun Zhu. Defense against adversarial attacks using high-level representation guided denoiser. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1778–1787, 2018. 3

[19] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017. 2

[20] Nikolaus Mayer, Eddy Ilg, Philip Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. 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*, pages 4040–4048, 2016. 3, 4

[21] Moritz Menze and Andreas Geiger. Object scene flow for autonomous vehicles. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3061–3070, 2015. 4

[22] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Universal adversarial perturbations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1765–1773, 2017. 1, 2

[23] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2574–2582, 2016. 2

[24] KR Mopuri, U Garg, and R Venkatesh Babu. Fast feature fool: A data independent approach to universal adversarial perturbations. In *British Machine Vision Conference 2017, BMVC 2017*. BMVA Press, 2017. 2

[25] Konda Reddy Mopuri, Aditya Ganeshan, and R Venkatesh Babu. Generalizable data-free objective for crafting universal adversarial perturbations. *IEEE transactions on pattern analysis and machine intelligence*, 41(10):2452–2465, 2018. 2, 3

[26] Konda Reddy Mopuri, Utkarsh Ojha, Utsav Garg, and R Venkatesh Babu. Nag: Network for adversary generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 742–751, 2018. 3

[27] Chaithanya Kumar Mummadi, Thomas Brox, and Jan Hendrik Metzen. Defending against universal perturbations with

shared adversarial training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4928–4937, 2019. 3

[28] Muhammad Muzammal Naseer, Salman H Khan, Muhammad Haris Khan, Fahad Shahbaz Khan, and Fatih Porikli. Cross-domain transferability of adversarial perturbations. In *Advances in Neural Information Processing Systems*, pages 12905–12915, 2019. 2

[29] Anh Nguyen, Jason Yosinski, and Jeff Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 427–436, 2015. 2

[30] Jiahao Pang, Wenxiu Sun, Jimmy SJ Ren, Chengxi Yang, and Qiong Yan. Cascade residual learning: A two-stage convolutional neural network for stereo matching. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 887–895, 2017. 3

[31] Tianyu Pang, Kun Xu, and Jun Zhu. Mixup inference: Better exploiting mixup to defend adversarial attacks. *arXiv preprint arXiv:1909.11515*, 2019. 3

[32] Jonathan Peck, Joris Roels, Bart Goossens, and Yvan Saeys. Lower bounds on the robustness to adversarial perturbations. In *Advances in Neural Information Processing Systems*, pages 804–813, 2017. 2

[33] Omid Poursaeed, Isay Katsman, Bicheng Gao, and Serge Belongie. Generative adversarial perturbations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4422–4431, 2018. 3

[34] Aaditya Prakash, Nick Moran, Solomon Garber, Antonella DiLillo, and James Storer. Deflecting adversarial attacks with pixel deflection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8571–8580, 2018. 3

[35] Edward Raff, Jared Sylvester, Steven Forsyth, and Mark McLean. Barrage of random transforms for adversarially robust defense. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6528–6537, 2019. 3

[36] Arianna Rampini, Franco Pestarini, Luca Cosmo, Simone Melzi, and Emanuele Rodola. Universal spectral adversarial attacks for deformable shapes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3216–3226, 2021. 3

[37] Anurag Ranjan, Joel Janai, Andreas Geiger, and Michael J Black. Attacking optical flow. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2404–2413, 2019. 3, 4

[38] Pouya Samangouei, Maya Kabkab, and Rama Chellappa. Defense-gan: Protecting classifiers against adversarial attacks using generative models. *arXiv preprint arXiv:1805.06605*, 2018. 3

[39] Simon Schrodi, Tonmoy Saikia, and Thomas Brox. What causes optical flow networks to be vulnerable to physical adversarial attacks. *arXiv preprint arXiv:2103.16255*, 2021. 3

[40] Ali Shafahi, Mahyar Najibi, Zheng Xu, John Dickerson, Larry S Davis, and Tom Goldstein. Universal adversarial training. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 5636–5643, 2020. 3

[41] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013. 2

[42] Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. Ensemble adversarial training: Attacks and defenses. *arXiv preprint arXiv:1705.07204*, 2017. 3

[43] Alex Wong, Safa Cicek, and Stefano Soatto. Targeted adversarial perturbations for monocular depth prediction. *Advances in Neural Information Processing Systems*, 33, 2020. 3

[44] Alex Wong, Mukund Mundhra, and Stefano Soatto. Stereopagnosia: Fooling stereo networks with adversarial perturbations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021. 1, 3, 4, 7

[45] Chang Xiao and Changxi Zheng. One man's trash is another man's treasure: Resisting adversarial examples by adversarial examples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 412–421, 2020. 3

[46] Chang Xiao, Peilin Zhong, and Changxi Zheng. Enhancing adversarial defense by k-winners-take-all. *arXiv preprint arXiv:1905.10510*, 2019. 3

[47] Cihang Xie, Mingxing Tan, Boqing Gong, Jiang Wang, Alan L Yuille, and Quoc V Le. Adversarial examples improve image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 819–828, 2020. 3

[48] Cihang Xie, Jianyu Wang, Zhishuai Zhang, Zhou Ren, and Alan Yuille. Mitigating adversarial effects through randomization. *arXiv preprint arXiv:1711.01991*, 2017. 3

[49] Cihang Xie, Jianyu Wang, Zhishuai Zhang, Yuyin Zhou, Lingxi Xie, and Alan Yuille. Adversarial examples for semantic segmentation and object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1369–1378, 2017. 3

[50] Cihang Xie, Zhishuai Zhang, Yuyin Zhou, Song Bai, Jianyu Wang, Zhou Ren, and Alan L Yuille. Improving transferability of adversarial examples with input diversity. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2730–2739, 2019. 2

[51] Haofei Xu and Juyong Zhang. Aanet: Adaptive aggregation network for efficient stereo matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1959–1968, 2020. 3, 4

[52] Sergey Zagoruyko and Nikos Komodakis. Learning to compare image patches via convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4353–4361, 2015. 3

[53] Jure Žbontar and Yann LeCun. Stereo matching by training a convolutional neural network to compare image patches. *The journal of machine learning research*, 17(1):2287–2318, 2016. 3

[54] Feihu Zhang, Victor Prisacariu, Ruigang Yang, and Philip HS Torr. Ga-net: Guided aggregation net for end-to-end stereo matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 185–194, 2019. 3

[55] Xizhou Zhu, Han Hu, Stephen Lin, and Jifeng Dai. Deformable convnets v2: More deformable, better results. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9308–9316, 2019. 3

[56] Yi Zhu, Karan Sapra, Fitsum A Reda, Kevin J Shih, Shawn Newsam, Andrew Tao, and Bryan Catanzaro. Improving semantic segmentation via video propagation and label relaxation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8856–8865, 2019. 5

[57] Alon Zolfi, Moshe Kravchik, Yuval Elovici, and Asaf Shabtai. The translucent patch: A physical and universal attack on object detectors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15232–15241, 2021. 3