Shape Defense

Ali Borji
Quintic AI, San Francisco, CA
aliborji@gmail.com

Abstract

Humans rely heavily on shape information to recognize objects. Conversely, convolutional neural networks (CNNs) are biased more towards texture. This fact is perhaps the main reason why CNNs are susceptible to adversarial examples. Here, we explore how shape bias can be incorporated into CNNs to improve their robustness. Two algorithms are proposed, based on the observation that edges are invariant to moderate imperceptible perturbations. In the first one, a classifier is adversarially trained on images with the edge map as an additional channel. At inference time, the edge map is recomputed and concatenated to the image. In the second algorithm, a conditional GAN is trained to translate the edge maps, from clean and/or perturbed images, into clean images. The inference is done over the generated image corresponding to the input's edge map. A large number of experiments with more than 10 data sets demonstrate the effectiveness of the proposed algorithms against FGSM, ℓ_∞ PGD, Carlini-Wagner, Boundary, and adaptive attacks. Further, we show that edge information can a) benefit other adversarial training methods, b) be even more effective in conjunction with background subtraction, c) be used to defend against poisoning attacks, and d) make CNNs more robust against natural image corruptions such as motion blur, impulse noise, and JPEG compression, than CNNs trained solely on RGB images. From a broader perspective, our study suggests that CNNs do not adequately account for image structures and operations that are crucial for robustness. The code is available at: `https://github.com/aliborji/ShapeDefense.git`

1 Introduction

The convolution operation in CNNs is biased towards capturing texture since the number of pixels constituting texture far exceeds the number of pixels that fall on the object boundary. This in turn provides a big opportunity for adversarial image manipulation. Some attempts have been made to emphasize more on edges, for example by utilizing normalization layers (e.g., contrast and divisive normalization [9]). Such attempts, however, have not been fully investigated for adversarial defense. Overall, how shape and texture should be reconciled in CNNs continues to be an open question. Here we propose two solutions that can be easily implemented and integrated in existing defenses. We also investigate possible adaptive attacks against them. Extensive experiments across ten datasets, over which shape and texture have different relative importance, demonstrate the effectiveness of our solutions against strong attacks. Experiments on more than 10 data sets demonstrate the effectiveness of the proposed algorithms against FGSM, ℓ_∞ PGD, substitute, Carlini-Wagner, Boundary, and adaptive attacks (the latter are shown in appendices B, C, D, and E in order).

2 Proposed methods

Edge-guided Adversarial Training (EAT). In this approach, we perform adversarial training over the 2D (Gray+Edge) or 4D (RGB+Edge) input (i.e., number of channels; denoted as Img+Edge). Please see Appx A for illustration of this algorithm (Alg. 1). In another version of the algorithm, first, for each input (clean or adversarial), the old edge map is replaced with the newly extracted one.

I (Still) Can't Believe It's Not Better Workshop at NeurIPS 2021.

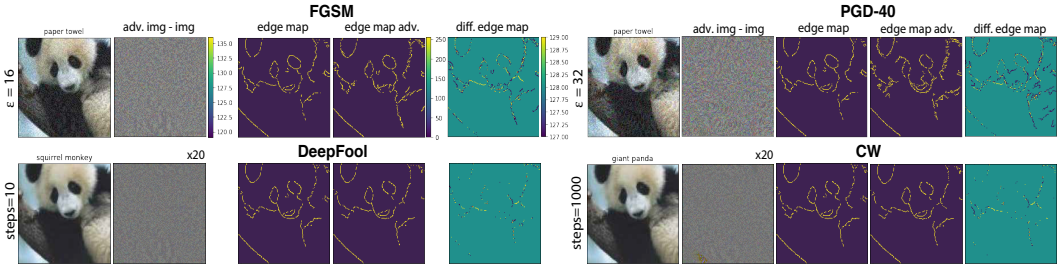

Figure 1: Adversarial attacks against ResNet152 over the giant panda image using FGSM [5], PGD-40 [11] (α=8/255), DeepFool [12] and Carlini-Wagner [3] attacks. The second columns in panels show the difference ($\mathcal{L}_2$) between the original image (not shown) and the adversarial one (values shifted by 128 and clamped). The edge map (using Canny edge detector) remains almost intact at small perturbations. Notice that edges are better preserved for the PGD-40.

Algorithm 1 Edge-guided adversarial training (EAT) for T epochs, perturbation budget ϵ, and loss balance ratio α, over a dataset of size M for a network f_θ (performed in minibatches in practice). $\beta \in \{edge, img, imgedge\}$ indicates network type and *redetect_train* means edge redetection during training.

for $t = 1 \dots T$ **do**
  for $i = 1 \dots M$ **do**
    // launch adversarial attack (here FGSM and PGD attacks)
    $\tilde{x}_i = \text{clip}(x_i + \epsilon\ sign(\nabla_x \ell(f_\theta(x_i), y_i)))$
    if β == *imgedge* & *redetect_train* **then**
      $\tilde{x}_i = \text{detect\_edge}(\tilde{x}_i)$ *// recompute and replace the edge map*
    end if
    $\ell = \alpha\ \ell(f_\theta(x_i), y_i) + (1 - \alpha)\ \ell(f_\theta(\tilde{x}_i), y_i)$ *// here* $\alpha = 0.5$
    $\theta = \theta - \nabla_\theta \ell$ *// update model weights with some optimizer, e.g., Adam*
  end for
end for

The edge map can be computed from the average of only image channels or all available channels (i.e., image plus edge). The latter can sometimes improve the results, since the old edge map, although perturbed, still contains unaltered shape structures. Then, adversarial training is performed over the new input. The reason behind adversarial training with redetected edges is to expose the network to possible image structure damage. The loss for training is a weighted combination of loss over clean images and loss over adversarial images. At inference time, first, the edge map is computed and then classification is done over the edge-augmented input. As a baseline model, we also consider first detecting the input's edge map and then feeding it to the model trained on the edges for classification. We refer to this model as Img2Edge.

GAN-based Shape Defense (GSD). Here, first, a conditional GAN is trained to map the edge image, from clean or adversarial images, to its corresponding clean image (Alg. 2). Any image translation method (here pix2pix by [7] using code at `https://github.com/mrzhu-cool/pix2pix-pytorch`) can be employed for this purpose. Next, a CNN is trained over the generated images. At inference time, first, the edge map is computed and then classification is done over the generated image for this edge image. The intuition is that the edge map remains nearly the same over small perturbation budgets (See Appx A). Notice that conditional GAN can also be trained on perturbed images (similar to [14] and [10] or edge-augmented perturbed images (similar to above).

3 Experiments and results

3.1 Datasets and Models

Experiments are spread across 10 datasets covering a variety of stimulus types. Sample images from datasets are given in Fig. 2. Models are trained with cross-entropy loss and Adam optimizer [8] with a batch size of 100, for 20 epochs over MNIST and FashionMNIST, 30 over DogVsCat, and 10 over the remaining. Canny method [2] is used for edge detection over all datasets, except DogBreeds for which Sobel edge detection is used. Edge detection parameters are separately adjusted for each dataset. We did not carry out an exhaustive hyperparameter search, since we are interested in

Algorithm 2 GAN-based shape defense (GSD)

// Training

1. Create a dataset of images $X = \{x_i, y_i\}^{i=1\cdots N}$ including clean and/or perturbed images
2. Extract edge maps (e_i) for all images in the dataset
3. Train a conditional GAN $p_g(x|e)$ to map edge image e to clean image x *// here pix2pix*
4. Train a classifier $p_c(y|x)$ to map generated image x to class label y

// Inference

1. For input image x, clean or perturbed, first compute the edge image e
2. Then, compute $p_c(y|x')$ where x' is the generated image corresponding to e

additional benefits edges may bring rather than training the best possible models. For attacks, we use `https://github.com/Harry24k/adversarial-attacks-pytorch`, except Boundary attack for which we use `https://github.com/bethgelab/foolbox`.

3.2 Results

3.2.1 Edge-guided Adversarial Training

Results over MNIST and CIFAR-10 are shown in Tables 1 and 2, respectively (please see Appx A). In these experiments, edge maps are computed only from the gray-level image (in turn computed from the image channels).

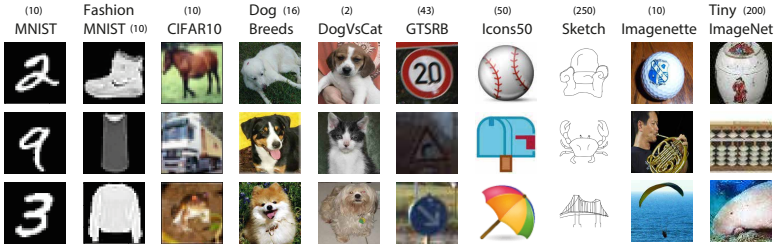

Figure 2: Sample images from the datasets. Numbers in parentheses denote the number of classes.

Over MNIST and FashionMNIST, robust models trained using edges outperform models trained on gray-level images (the last column). The naturally trained models, however, perform better using gray-level images than edge maps (Orig. model column). Adversarial training with augmented inputs improves the robustness significantly over both datasets, except the FGSM attack on FashionMNIST. Over CIFAR-10, incorporating the edges improves the robustness by a large margin against the PGD-40 attack. At $\epsilon = 32/255$, the performance of the robust model over clean and perturbed images is raised from (0.316, 0.056) to (0.776, 0.392). On average, the robust model shows 64% improvement over the RGB model (last column in Table 2). Over the TinyImageNet dataset, as in CIFAR-10, classification using edge maps is poor perhaps due to the background clutter. Nevertheless, incorporating edges improves the results. We expect even better results with more accurate edge detection algorithms (e.g., supervised deep edge detectors). Over these 4 datasets, the final model (i.e., adversarial training using image + redetected edge, and edge redetection at inference time) leads to the best accuracy. The improvement over the image is more pronounced at larger perturbations, in particular against the PGD-40 attack (as expected; please see Fig. 1).

Over the DogVsCat dataset, as in FashionMNIST, the model trained on the edge map is much more robust than the image-only model (Table 6 in Appx. A). Over the DogBreeds dataset, utilizing edges does not improve the results significantly (compared to the image model). The reason could be that texture is more important than shape in this fine-grained recognition task (Table 7 Appx. A). Over GTSRB, Icons-50, and Sketch datasets, *image+edge* model results in higher robustness than the *image-only* model, but leads to relatively less improvement compared to the *edge-only* model. Please see Tables 8, 10, and 12. Over the Imagenette2-160 dataset (Table 14), classification using images does better than edges since the texture is very important on this dataset.

Average results over 10 datasets is presented in Fig. 3 (left panel). Combining shape and texture (full model) leads to a substantial improvement in robustness over the texture alone (5.24% improvement against FGSM and 28.76% imp. against PGD-40). Also, *image+edge* model is slightly more robust than the *image-only* model. Computing the edge map from all image channels improves the results on some datasets (e.g., GTSRB and Sketch) but hurts on some others (e.g., CIFAR-10) as shown in Appx. A. The right two panels in Fig. 3 show a comparison of natural (Orig. model column in

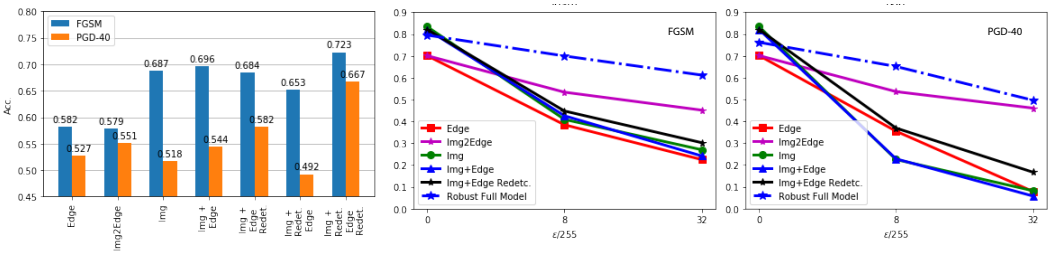

Figure 3: Left) Average results of the EAT defense on all datasets (last cols. in tables). Middle and Right) Comparison of natural (Orig. model column; solid lines) *vs.* adversarial training averaged over all datasets.

tables; solid lines) *vs.* adversarial training. Natural training with *image+edge* and *redetection* at inference time leads to enhanced robustness with little to no harm to standard accuracy. Despite the Edge model only being trained on edges from clean images, the Img2Edge model does better than other naturally-trained models against attacks. The best performance, however, belongs to models trained adversarially. Notice that our results set a new record on adversarial robustness on some of these datasets even without exhaustive parameter search[1].

Robustness against Carlini-Wagner (CW) and Boundary attacks. Performance of our method against l_2 CW attack on MNIST dataset is shown in Appx. C. To make experiments tractable, we set the number of attack iterations to 10. With even 10 iterations, the original Edge and Img models are severely degraded. Img2Edge and Img+(Edge Redetect) models, however, remain robust. Adversarial training with CW attack results in robust models in all cases.

Results against the decision-based Boundary attack [1] are shown in Appx. D over MNIST and Fashion MNIST datasets. Edge, Img, and Img+Edge models perform close to zero over adversarial images. Img+(Edge Redetect) model remains robust since the Canny edge map does not change much after the attack, as is illustrated in Fig. 6.

Robustness against substitute model attacks. Following [13], we trained substitute models to mimic the robust models (with the same architecture but with RGB channels) using the cross-entropy loss over the logits of the two networks, for 5 epochs. The adversarial examples crafted for the substitute networks were then fed to the robust networks. Results are shown in *italics* in Tables 1, 2, 3 and 4 (performed only against the edge-redetect models). We find that this attack is not able to knock off the robust models. Surprisingly, it even improves the accuracy in some cases. Please refer to Appx. B for more details.

Robustness against adaptive attacks. So far we have been using the Canny edge detector which is non-differentiable. What if the adversary builds a differentiable edge detector to approximate the Canny edge detector and then utilizes it to craft adversarial examples? To study this, we run two experiments. In the first one, we build the following pipeline using the HED deep edge detector [15]: Img $\longrightarrow$ HED $\longrightarrow$ Classifier^{HED}. A CNN classifier (as above) is trained over the HED edges on the Imagenette2-160 dataset (See Appx. E). Attacking this classifier with FGSM and PGD-5 ($\epsilon = 8/255$) completely fools the network. The original classifier (Img2Edge here) trained on Canny edges, however, is still largely robust to the attacks (i.e., Img$^{adv-HED}$ $\longrightarrow$ Canny $\longrightarrow$ Classifier^{Canny}) as shown in Table 19. Notice that the HED edge maps are continuous in the range [0,1], whereas Canny edge maps are binary, which may explain why it is easy to fool the HED classifier (See Fig. 7).

Above, we used an off the shelf deep edge detector trained on natural scenes. As can be seen in Appx. E, its generated edge maps differ significantly from Canny edges. What if the adversary trains a model with the (*input, output*) pair as (*input image, Canny edge map*) to better approximate the Canny edge detector? In experiment two, we investigate this possibility. We build a pipeline consisting of a convolutional autoencoder followed by a CNN on MNIST. Details regarding architecture and training procedure are given in Appx. E. As results in Fig. 10 reveal, FGSM and PGD-40 attacks against the pipeline are very effective. Passing the adversarial images through Canny and then a trained (naturally or adversarially) classifier on Canny edges (i.e., Img2Edge), still leads to high accuracy, which means that transfer was not successful. We attribute this feat to the binary output of Canny. Two important point deserve attention. First, here we used the Img2Edge model, which as shown above, is less robust compared to the full model (i.e., img+edge and redetection). Thus, adaptive

[1]cf. [6]; the best robust accuracy on CIFAR-10 against PGD attack, ℓ_∞ of size 8/255, is about 67%.

attacks may be even less effective against the full model. Second, proposed methods perform better when edge map is less disturbed. For example, as shown in Fig. 10 (bottom), the PGD-40 adaptive attack is less effective against the shape defense since edges are preserved better.

Analysis of parameter α. By setting $\alpha = 0$, the network will be exposed only to adversarial examples (Alg. 1), which is computationally more efficient. However, it results in lower accuracy and robustness compared to when $\alpha = 0.5$, which means exposing the network to both clean and adversarial images is important (See Table 22; Appx. H). Nevertheless, here again incorporating edges improves the robustness significantly compared to the image-only case.

Speculation behind effectiveness of this method. The main reason is that the edge map acts as a checksum, and the network learns (through adversarial training) to rely more on the redetected edges when other channels are misleading. This aligns with prior observations such as shortcut learning in CNNs [4]. Also, our approach resembles adversarial patch or backdoor/trojan attacks where the goal is to fool a classifier by forcing it to rely on irrelevant cues. Conversely, here we use this trick to make a model more robust. Also, the Img2Edge model can purify the input before classifying it. Any adaptive attack against the EAT defense has to alter the edges which most likely will result in perceptible structural damages.

3.2.2 GAN-based Shape defense

We trained the pix2pix model for 10 epochs over MNIST and FashionMNIST datasets, and for 100 epochs over Icons-50 dataset. Sample generated images are shown in Fig. 11 (Appx. F). A CNN (same architecture as before) was trained for 10 epochs to classify the generated images. Results are shown in Fig. 4. The model trained over the images generated by pix2pix (solid lines in the figure) is compared to the model trained over the original clean training set (denoted by the dashed lines). Both models are tested over the clean and perturbed versions of the original test sets of the four datasets. Over MNIST and FashionMNIST datasets, GSD performs on par with the original model on clean test images. It is, however, much more robust than the original model against the attacks. When we trained the pix2pix over the edge maps from the perturbed images, the new CNN models became even more robust (stars in Fig. 4; top panels). We expect even better results with training over edge maps from both intact and perturbed images[2].

Over Icons-50 dataset, generated images are poor. Consequently, GSD underperforms the original model on clean images. Over the adversarial inputs, however, GSD wins, especially at high perturbation budgets and against the PGD-40 attack. With better edge detection and image generation methods (e.g., using perceptual loss), better results are expected.

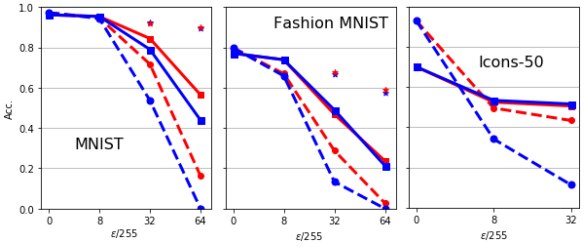

Figure 4: Results of GSD method.

Speculation behind effectiveness of this method. The main reason is that cGAN learns a function f that is invariant to adversarial perturbations. Since the edge map is not completely invariant to (especially large) perturbations, one has to train the cGAN on the augmented dataset composed of clean and perturbed images. One advantage of this approach is it computational efficiency since there is no need for adversarial training. Any adaptive attack against this defense has to fool the cGAN which is perhaps not feasible since it will be noticed from the generated images (i.e., cGAN will fail to generate decent images).

4 Summary and Discussion

Two algorithms are proposed to use shape bias and background subtraction to strengthen CNNs and defend against adversarial attacks and backdoor attacks. To fool these defenses, one has to perturb the image such that the new edge map is significantly different from the old one while preserving image shape and geometry, which does not seem to be trivial at low perturbation budgets. Our results without exhaustive parameter search (model architecture, epochs, edge detection, cGAN training, etc.) are very promising. Comparison with other more complicated defenses remains to be done.

[2]Similarly, the edge map classifier used in the Img2Edge model in the previous section (EAT defense) can be trained on edge maps from both clean and adversarial examples to improve performance.

References

[1] W. Brendel, J. Rauber, and M. Bethge. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. *CoRR*, abs/1712.04248, 2017.

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

[3] N. Carlini and D. A. Wagner. Towards evaluating the robustness of neural networks. In *IEEE Symposium on Security and Privacy*, pages 39–57, 2017.

[4] R. Geirhos, J.-H. Jacobsen, C. Michaelis, R. Zemel, W. Brendel, M. Bethge, and F. A. Wichmann. Shortcut learning in deep neural networks. *arXiv preprint arXiv:2004.07780*, 2020.

[5] I. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. In *Proc. ICLR*, 2015.

[6] S. Gowal, C. Qin, J. Uesato, T. Mann, and P. Kohli. Uncovering the limits of adversarial training against norm-bounded adversarial examples. *arXiv preprint arXiv:2010.03593*, 2020.

[7] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.

[8] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.

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

[10] X. Li and S. Ji. Defense-vae: A fast and accurate defense against adversarial attacks. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 191–207. Springer, 2019.

[11] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. Towards deep learning models resistant to adversarial attacks. *CoRR*, abs/1706.06083, 2017.

[12] S. Moosavi-Dezfooli, A. Fawzi, and P. Frossard. Deepfool: A simple and accurate method to fool deep neural networks. In *Proc. CVPR*, pages 2574–2582, 2016.

[13] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. Berkay Celik, and A. Swami. Practical black-box attacks against deep learning systems using adversarial examples. *CoRR*, abs/1602.02697, 2016.

[14] P. Samangouei, M. Kabkab, and R. Chellappa. Defense-gan: Protecting classifiers against adversarial attacks using generative models. *arXiv preprint arXiv:1805.06605*, 2018.

[15] S. Xie and Z. Tu. Holistically-nested edge detection. In *IEEE International Conference on Computer Vision*, 2015.