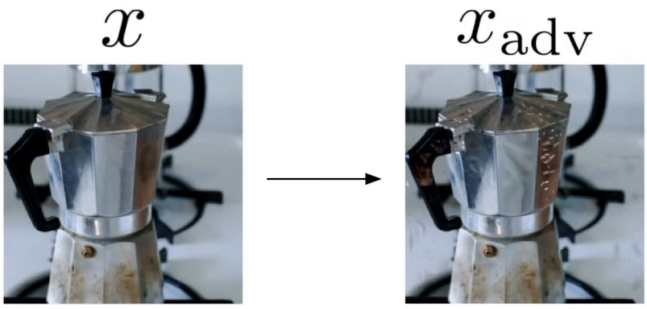
EECS 127 Project Extension

PIXMIX: Dreamlike Pictures Comprehensively Improve Safety Measures

Hendrycks, Zou, et al. (2022)

Introduction

While we can create models that are provably robust to attacks which don't modify images beyond an l-norm bound, such attacks are only a small subset of all attacks. In practice, it is easy to adversarially modify images beyond a small bound while not changing the category of an image:



In the example above, the image is modified noticeably (see the markings on the coffee maker in x_{adv}) but the image still clearly depicts a coffee maker, despite now being classified wrongly by the model[1].

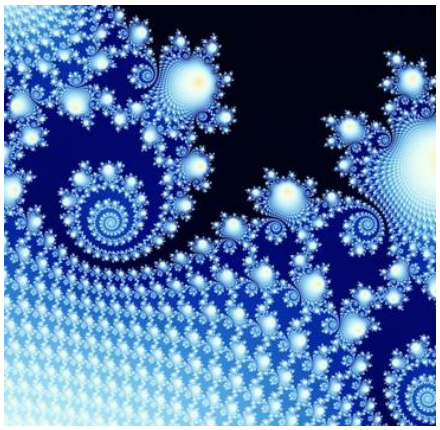
One popular way of improving robustness to this type of attacks is through data augmentation, which involves modifying the inputs to a model in training time. For instance, we can deliberately feed adversarially optimized examples to our model.

The main problem with training successfully against adversarial attacks is that it notably degrades performance metrics like calibration, corruption, consistency, and anomaly detection. While there have been other data augmentation methods proposed in the literature such as AugMix, Outlier Exposure, and AutoAugment that do reasonably well at balancing performance on adversaries and the aforementioned performance metrics, they do not uniformly perform strongly across all the studied metrics. In this paper, Hendrycks, Zou, et al. introduce PiXMiX, a data-augmentation method that reliably improves models across all these criteria.

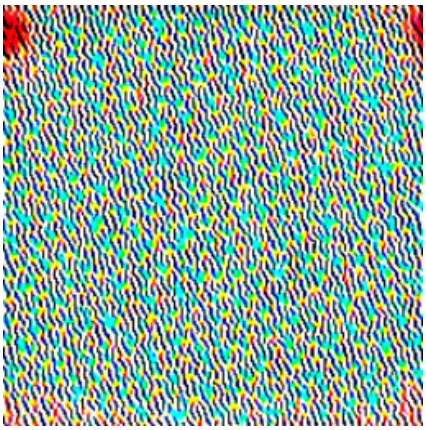
[1] Image taken from Intro to AI Safety Course by Dan Hendrycks.

Formal Description of Proposed Method

The method uses two separate processes to add randomness and structured noise to test set images. The first process is the picture source step (PIX), which at random chooses fractals or feature visualizations to pair with a test image.



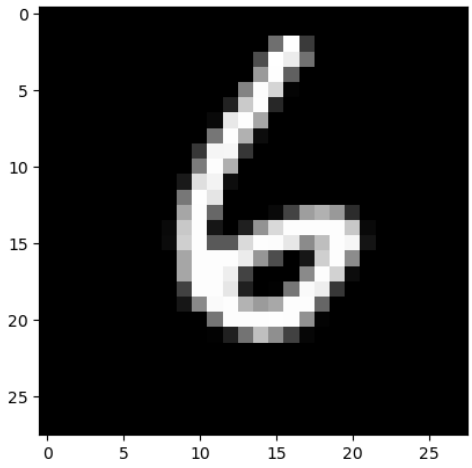
Fractal



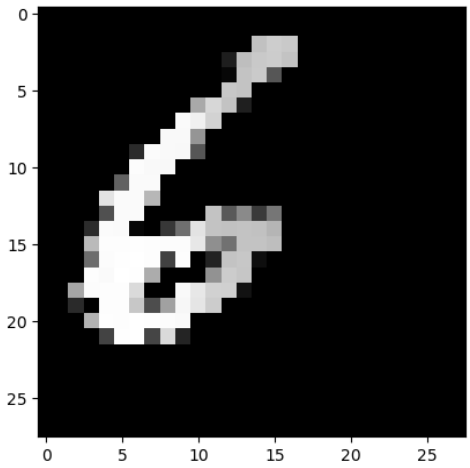
Feature Visualization

In particular, the pictures are either combined additively (i.e. the arithmetic mean of each pixel is used) or multiplicatively (i.e. the geometric mean is used). These image pairs are then processed in the Mixing Pipeline (MIX) i times, where i is a random integer between 0 and k. The mixing pipeline involves either augmenting the image or pairing it again with a fractal or feature visualization. Augmentation involves transforming the image in a way that preserves its meaning (e.g. rotating it slightly or shearing it).

We implemented PixMix on MNIST to compare its performance to the primal and dual networks. Here is an example of an altered digit:



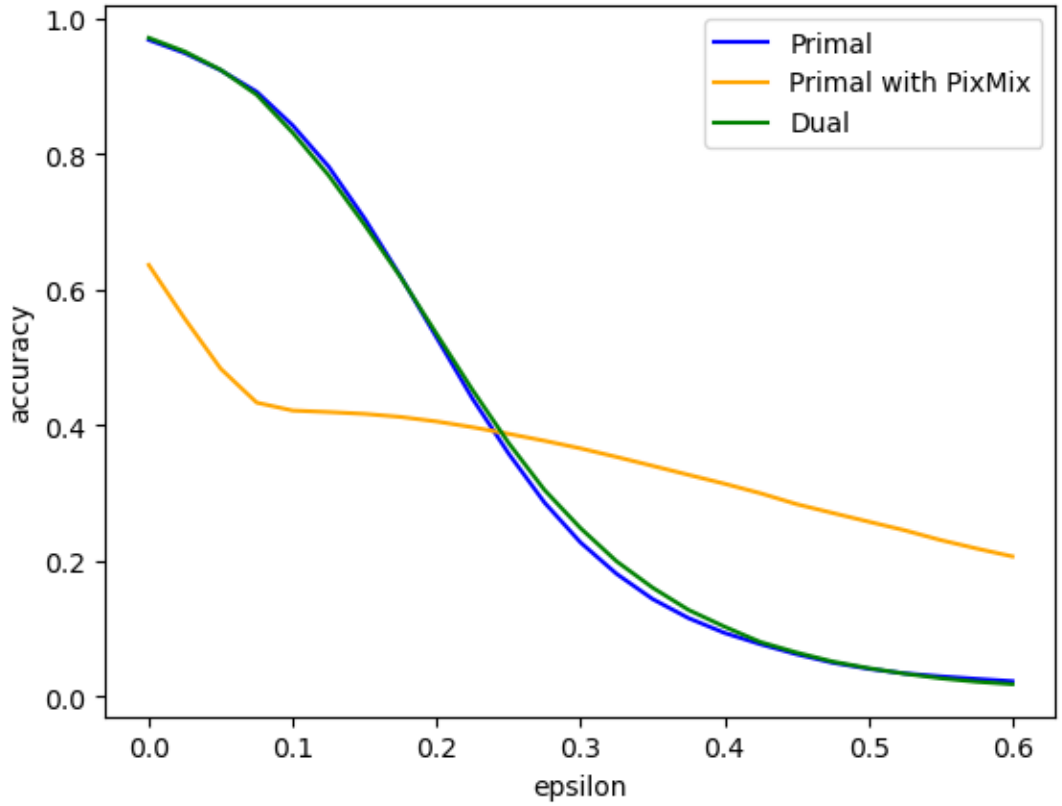
Original Digit



Digit Altered with PixMix

Some of the augmentations had to be removed or adjusted to optimize PixMix for MNIST rather than CIFAR. For example, rotations of 180º were removed, since some digits are not rotation invariant (e.g. an upside-down six is a nine).

For an epsilon greater than roughly 0.24, the PixMix-trained network outperformed both other networks, maintaining accuracy above 0.2 for very large epsilon as the accuracy of other networks went to 0:



Discussion on Benefits of Method

As we have seen, PixMix allows for robust training of machine learning models against adversarial attacks even when the nature of the attack (e.g. that it is an FGSM attack within a certain bound) is not known. As the paper outlines, this is done while preserving high performance on metrics like calibration, corruption, consistency, and anomaly detection.

More broadly, with machine learning becoming more prevalent in real-world applications such as self-driving cars, it will become increasingly critical to have dependable and high performing methods to train such models. In the context of self-driving cars for example, being robust to adversarial attacks can help prevent potentially life-threatening misclassifications of objects such as stop signs.

"Fighting Gradients with Gradients: Dynamic Defenses against Adversarial Attacks"
Wang et al. (2021)

Introduction:
Many current robustly-designed deep networks are robust only in reference to training data, but not necessarily in test data, leaving them weak to dynamic adversarial attacks. Strong attacks like those using gradient optimization iteratively update to break down models, and models with static defenses are prone to failure. Previous research in this area has explored how models can adapt to attacks during the training step of their development as well as using stochastic defenses. The predominant flaw in the models remains after these model adjustments, as the high dependency on the training data, leaving the model incapable to defend itself dynamically when put to the test.

Formal Description of Proposed Method:
Dent, or defensive entropy minimization adapts the model during training loops to defend against the adversary, and most notably, with the last move advantage. Specifically, the model tries to optimize its parameters on inputs than an adversary might be perturbing towards its own optimum.

Mathematically, the following equations demonstrate the differences between static and dynamic modeling.

Static model prediction: $H(\hat{y}^t) = f(x + \delta^t; \theta)$
Where $x + \delta$ *is the adversarial sample*

Dynamic model prediction: $H(\hat{y}^t) = f(g(x + \delta^t; \Sigma^t); \theta + \Delta^t)$
Where $x + \delta$ *is the adversarial sample, and*
$g(x + \delta; \Sigma)$ *is the input transform*

We can see with this modeling that there are dynamic updates applied to the model prediction at each iteration whilst the static modeling does not perform this. Without any adaptations, the static model is left vulnerable to iterative and dynamic attacks.

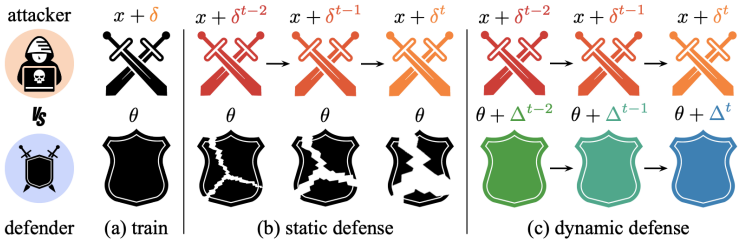


Illustration of the prediction modelings compared [Wagner et al.]

Discussion on Benefits of Method:

By performing an input transform on the adversarial sample, the model effectively minimizes the entropy of the prediction. As such, the dent method strongly improves the strength of models to ensure against powerful attackers and obtain a last move advantage, albeit at an efficiency tradeoff. It is noted in the paper that it is more valuable for a method to be highly accurate and slow as opposed to quick and incorrect. Applications of such a model could be defending against an adversary that a model has never seen before in training. In the case of image classification, this could be seen as a self-driving car learning how snow and dust storms affect its image classification abilities if not trained on such augmented data before. This kind of dynamic defense allows the classification algorithm to perform better with time in the assumption that the model could not have been trained on all potential adversaries.