# Reconstructive Training for Real-World Robustness in Image Classification

David Patrick, Michael Geyer, Richard Tran, and Amanda Fernandez
University of Texas at San Antonio
amanda.fernandez@utsa.edu

## Abstract

*In order to generalize to real-world data, computer vision models need to be robust to corruptions which may not generally be available in the traditional benchmark datasets. Real world data is diverse and can vary over time - sensors may become damaged, environments may change, or users may provide malicious inputs. While substantial research has focused separately on processing specific image distortions or on defending against types of adversarial attack, some real-world applications will require vision models to generalize to corruptions, while additionally maintaining image quality. We propose a simple training strategy to leverage image reconstruction, with similarities to a GAN training process, to reduce image data corruptions while maintaining the visual integrity of the image. Our approach is demonstrated on several corruptions for the task of image classification, and compared with established approaches, with qualitative and quantitative improvements. Code available at:* `https://github.com/UTSA-VAIL/ReconstructiveTraining`

## 1. Introduction

Deep learning has had a major impact over the last decade on computer vision tasks. However, integrating deep learning into vision comes with a unique requirement for substantial, quality data. This becomes problematic when solving problems for the real world, where data can be less predictable in practice. For instance, computer vision models built for outdoor activities like autonomous driving must be able to handle changing weather. Even models which are used for vision applications in controlled environments, such as in spectroscopy or in manufacturing, encounter changes in data integrity as instrumentation wears, ages, or comes out of alignment.

Even if it is predictable, how can we approach data that changes over time? A wide range of approaches across disciplines are focused on this problem. Rigorous data analysis, looking especially for bias in the data and tracking changes over time, helps identify outliers. Some research in explainability of neural networks aim to help humans understand the features that are learned by the model. Research in out of distribution data aims to help models expand to unseen categories. Many approaches examine the corruption of data, which might be the addition of noise, removal of relevant context, or a deliberate attack. In the case of adversarial examples, the corruption of the data is intentional; however, unintentional corruption may be caused by natural changes in the real world.

In an effort to be as generalizable to real-world data as possible, there have been a number of works aiming to reduce intentional corruption, as well as works reducing unintentional corruption. In this work we focus on the problem of intentional corruptions to input data.

The problem of intentional corruption, adversarial examples, has captured the attention of the research community in the last few years [11, 25, 29]. Adversarial attacks use malicious transformations to cause DNNs and machine learning models to classify inputs incorrectly. Although these adversarial attacks share a similar goal of drifting the target model, each adversarial attack has a unique approach which makes finding effective defenses difficult.

Despite all the challenges surrounding adversarial defense, a number of techniques have shown promise by increasing the robustness of target models against these attacks. When countering adversarial attacks we can either look to condition the model to ignore adversarial perturbations, or remove the effects of the perturbations directly from input images. *Network modification* defenses modify either the architecture or the training set so that the resulting model can ignore adversarial perturbations. These include common techniques such as adversarial training [35, 16, 22], feature cleaning [23, 38] and other methods which change the target model's behavior. Unfortunately while these defenses can be effective against attacks they are trained against, they often struggle generalizing their methods to unknown attacks and can be computationally expensive. *Input transformation* defenses attempt to remove the effects of adversarial perturbations before arriving at the targeted model. This can be achieved by removing the offending noise or by adding counter noise directly to the in-

Figure 1. Perturbations before and after the proposed defense. Perturbations are shown as an absolute difference from the original image. From left: original image, PGD perturbations, perturbations after proposed strategy. (Best viewed in color)

put [24].

Our proposed strategy is a simple training approach most similar to an input transformation. Demonstrated on a simple autoencoder, it strikes a balance between accuracy and visual integrity for both corrupted and non-corrupted image data. Through the feedback of the target classification model, the auto-encoder is able to learn how to apply counter-noise to the relevant areas of the image. Figure 1 examines this effect of our proposed approach on a corrupted image. We demonstrate some improvement over state-of-the-art on popular adversarial attack methods, motivating the use of a corruption-aware training strategy in combination with reconstructive loss.

## 2. Related Works

Every day more digital media is created than ever before. As we aim to create computer vision models to fit that data, we can (and should) utilize simple mathematical models whenever possible. However, some real-world data can be too complex for this type of hands-on feature manipulation to be practical. Deep learning models enable a higher dimensionality of feature representation, which can help to generalize to the diversity of real-world data. Despite this, deep learning approaches are still susceptible to intentional and unintentional corruptions in the data. Recent research leveraging deep learning for vision spans a broad range of applications such as denoising images and video[8], increasing visibility in underwater scenes [40], reduction/removal of watermarking [17], and many others. Each of these works aim to resolve a specific corruption for a dedicated task. In this work, we aim for a more general approach, where the corruption is not known prior to application. To consider this fully, we examine existing *intentional* corruption approaches in the following section, as well as related reconstructions/defenses against such types of corruptions.

### 2.1. Adversarial Attacks

Adversarial attacks are small perturbations applied to input data with the intention of fooling DNNs. Since DNNs almost exclusively rely on gradient descent for training, gradient-based attacks are extremely effective against most modern computer vision systems. Fast Gradient Signed Method (FGSM)[10] is considered one of the first successful gradient-based attacks. The core idea of FGSM is to use the sign of the gradient from a loss function in order to create adversarial perturbations. Although FGSM[10] is still effective, state-of-the-art attacks often replace this single-step approach with an iterative approach which allows them to fine tune and strengthen their attacks. Projected Gradient Decent (PGD)[19] extended the ideas presented by FGSM by introducing iterative steps. By ensuring each step is a valid attack, PGD is able to see how the target model reacts to its perturbations and adjust accordingly, resulting in more robust attacks. Carlini-Wagner (CW)[3] introduced an objective function in combination with a distance metric for the attack to minimize on, creating an effective optimization-based attack. By combining this objective function with a hyper-parameter tuning search step, CW is able to produce extremely effective attacks, although at the cost of computation time. DeepFool[21] uses a linear approximation of the target model, focusing on minimizing the number of perturbations needed to cross the decision boundary. DeepFool attacks are not as effective against denoising strategies due to the comparatively small number of perturbations introduced to the image, however this also means they are much more difficult to detect. More recently, SmoothFool[5] applies smoothed DeepFool[21] adversarial perturbations to an image. This technique of smoothing the adversarial perturbations across the entire image renders many of the common denoising techniques ineffective and enhances the attack's transferability. While this attacks is effective at fooling many state-of-the-art defenses, it often can be visually seen. Backward Pass Differentiable Approximation (BPDA)[1] computes the gradient after applying the target model's defense method. This attack has been shown to be extremely effective in bypassing the defense of the target model; however, it not only requires full information from the target model but also full information about its defense. Some adversarial attacks, such as one-pixel attack[32], have even started to use non-gradient-based methods in order to attack the target models.

### 2.2. Network Modification Defenses

Network modification defenses are designed to directly improve the robustness of the target model. The most common network modification defense is adversarial training[35, 22, 10, 16] where the targeted model is purposely fed adversarial images during the training process. These works show that while adversarial training increases the model's robustness against known attacks, it often has difficulty generalizing the defense to unknown attacks. While adversarial training makes changes to the target model's training set, other network modification tech-

niques make architectural changes to improve adversarial robustness. For example, feature denoising[37] and feature squeezing[38] modify the network's feature extraction to include denoising techniques. Other network modification techniques including region-based classifiers[2] and saturating networks[23] modify the decision boundary of the network to account for adversarial examples. While these defenses are effective in defending against adversarial attacks, they may not be applicable to all models as they often require unique architectural modifications. In addition to not being easily applicable, network modification defenses require the target model to be retrained, which can be an expensive process.

### 2.3. Input Transformation Defenses

Rather than making changes directly to the target models, input transformation defenses apply changes to images before classification is performed. This approach is model agnostic and allows cleaned images to be saved; allowing applications which require human readable input, or employ multiple models to save computation time. The ideal input transformation defense is one which, when given a potentially corrupted image, produces an identical image without any perturbations and does not reduce the classification accuracy of images which have not been corrupted. We divide the scope of input transformation defenses into two main categories: *static* and *learned* defenses.

*Static defenses* involve applying fixed transformations to the target model's input, usually through the use of a denoising technique. Many traditional input transformation methods such as JPEG compression[9] and bit-depth reduction[12] fall into this category. More recently, there has been some work in redesigning and creating new static methods to defend against adversaries. Feature Distillation (FD) [18] redesigned the JPEG compression with a quantization process which, when applied to an image, filters out adversarial perturbations. While these static methods can be effective against small adversarial perturbations, they often struggle against stronger attacks and can sometimes have adverse effects on the visual integrity of the images.

*Learned defenses*, unlike static defenses, are typically more complex and require some amount of training in order to properly function as a defense. Magnet[20] runs the image through a reformer network, which could be a single autoencoder or a series of autoencoders, and reconstructs the image closer to that natural image manifold. Rather than training on adversarial examples, Magnet focuses on learning from natural examples and being able to detect and reform examples that stray from the natural image statistics. Sparse Transformation Layer (STL)[33] projects the image onto a quasi-natural space, which reconstructs the image back together based on its best features. By learning and projecting the image onto this quasi-natural image space, STL is able to reduce the space between features of original and adversarial images, allowing for more accurate classifications. Some learned input transformations use generative adversarial networks (GANs) in order to reconstruct an attacked image. DefenseGAN[28] proposed a GAN reconstruction method that when tested on the MNIST dataset mitigated adversarial attacks. DefenseGAN uses a seed vector for each image in order to create a new image with the same classification. This vector is created by backpropagating a distance metric through a generator, resulting in a generated image closely related to the input image.

Many input transformation defenses, both static and learned, utilize randomized elements in order to defend against adversarial attacks. These randomized elements help disrupt attacks and make it difficult for attackers to recreate their defense. Random Resize and Padding (RRP)[36] takes each image and applies it with a transformation of re-scaling and padding in order to distort the adversarial noise before sending it to the classification network. The hope is that by shifting the location of attacked pixels, their effectiveness will be reduced. Total variation minimization (TVM)[13] takes inspiration from pixel dropout and randomly chooses a small set pixels within the attacked image to reconstruct. The selected pixels are then reconstructed in such a way that it recreates the "simplest" image that is consistent with the selected pixels, disrupting the adversarial perturbations. Pixel Deflection (PD)[24] reverts the image to its' natural image statistics by randomly redistributing the pixel values across the image using nearest neighbors. By randomly redistributing the pixel values, PD is able to convert the image's adversarial perturbations into natural noise which many image classifiers are inherently more robust to. SHIELD[6] randomly applies different levels of JPEG compression in patches to denoise the attacked image. In order to effectively combat the lossy nature of JPEG compression, SHIELD also introduced a technique they refer to as "vaccinating" the model. This technique requires the target model to be trained with compressed images in order to increase its robustness towards image compression techniques. ME-Net [39] combines pixel dropout with matrix estimation to reconstruct the input image without adversarial perturbations.

## 3. Proposed Approach

In this work, we propose a learned input transformation strategy which uses a fully-convolutional autoencoder to remove adversarial perturbations through image reconstruction. This is trained much like a GAN where the target classifier treated as a frozen discriminator. However, unlike a traditional GAN, the objective function to optimize is based on both classification accuracy and visual distance of images. During training, the model is tasked with learning image reconstruction that not only restores attacked im-

ages back to their original state but also does not modify the classification of images that are not attacked. Providing the attacked image, original image and ground truth label to the network enables learning an effective corruption-free version of the image, while maintaining visual integrity. The following subsections describe first the auto-encoder for this approach, then the proposed training strategy.

## 3.1. Auto-encoder

We propose a simple autoencoder to demonstrate the effectiveness of the described strategy. Images are reduced through strided convolutions and then upsampled through transposed convolutions, reconstructing the image from limited features.

### 3.1.1 Encoder & Decoder Blocks

Each encoder block consists of two convolution layers, reducing the number of filters on each consecutive block by half. For consistency, the decoder blocks mirror their respective encoder blocks, both in convolution layers and number of filters. In addition to matching the number of filters, each convolution within a block is paired with a Batch Normalization layer and Rectified Linear Unit activation. This consistency enables connection of feature maps from the encoder blocks to the decoder blocks. Pooling and upsampling interpolation between all blocks is performed using strided convolution and transposed convolution respectively. This technique has been shown to create more accurate image reconstructions, and will often improve the overall performance of the network as opposed to max or average pooling and nearest neighbor upsampling [31, 26].

### 3.1.2 Skip Connection

Our AE uses a low number of filters in combination with a small bottleneck between the encoder and decoder to filter and remove pixel-level corruptions. This process is effective for removing perturbation from the reconstructed image, however it results in loss of visual detail. In order to overcome this bottleneck and improve the reconstruction's visual quality, skip layers are introduced, connecting the output of the first encoder block to the input of the last decoding block. This skip connection allows the last decoder block to use additional information in the final reconstruction of the image, creating a more accurate representation of the input. Additional skip layer connections are not incorporated elsewhere as we do not want give additional chances for the attack to transfer to the final image.

## 3.2. Training

While the fully convolutional auto-encoder plays an integral part to the success of the proposed model, we believe the most important contribution of this work is its unique training process. While it takes inspiration from adversarial training, in contrast it aims to restore an altered image back to its original state. To be effective, we aim to restore corrupted images back to their original state while also not affecting the classification of images that were not corrupted. In order to facilitate this, both the original image and its corrupted counterpart are leveraged during the training process. During each epoch, cleaned images are generated from corrupted images using the fully convolutional autoencoder. These images are then compared to the original images. Once the autoencoder has established a starting point, a training strategy that resembles a GAN begins. In this strategy, the autoencoder is treated as a generator and the target model is treated as a frozen discriminator. Training is performed using an image dataset $S = \{(x_1, \tilde{x}_1, y_1), ..., (x_N, \tilde{x}_N, y_N)\}$ where $\tilde{x_i}$ denotes a corrupted image corresponding to its original image $x_i$. Furthermore, in this example *N* refers to the number of images in the dataset and $y_i$ refers to ground truth labels. Algorithm 1 outlines the overall process for training a model once its reached this stage. The generated images are passed to the target model in order to get feedback regarding the classification of the reconstructed images. This allows for the autoencoder to not only focus on minimizing the distance between the reconstruction and original image, but also allows it to shift the focus to fixing the classification of the image reconstruction. In addition to allowing the focus shift, it incorporates the target model directly into its own defense process without having to modify the classifier, unlike most network modification defenses.

Our network also utilizes a custom loss function which can be represented in two parts: a visualization loss and a classification loss.

$$\alpha \cdot \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \hat{x}_i)^2} + \beta \cdot (C + \log(\sum_{i=1}^{N}(x_i - \hat{x}_i)^2)) \quad (1)$$

$$\sum_{i=1}^{N} y_i \cdot \log \hat{y}_i \quad (2)$$

The visualization loss, represented by Equation 1, is a custom loss function which combines the root mean squared error (RMSE) with the peak signal-to-noise ratio (PSNR) metric. The goal of this visualization loss is to minimize the distance between the original and generated images, represented by $x$ and $\hat{x}$ respectively, to help reconstruct a more visually accurate image. In order to make PSNR a loss function that could be both minimized on and combined with RMSE, it was scaled using a constant C, which for our experiments was set to 100. Finally, the visualization loss function uses $\alpha$ and $\beta$ to weigh in favor of the much smaller RMSE. For our experiments, $\alpha$ was set to 1 while $\beta$ was

**Algorithm 1** Proposed training strategy: classification loss improves discriminator performance while visual loss improves visual clarity.

**Require:**
$g_\theta$: Generator function parameterized by model weights  $f$: Discriminator function

**for** each epoch **do**
  **for** $(x, \tilde{x}, y)$ in $S$ **do**
    $\hat{x} \leftarrow g_\theta(\tilde{x})$ ▷ Generate cleaned images from adversarial examples based on current model weights.
    $\hat{y} \leftarrow f(\hat{x})$ ▷ Discriminator is queried using generated images.
    $\mathcal{L}_d \leftarrow H(\hat{y}, y)$ ▷ Discriminator loss is calculated using categorical cross entropy (Eq. 2).
    $\mathcal{L}_v \leftarrow V(\hat{x}, x)$ ▷ Visual loss is calculated using Eq. 1 on generated and original images.
    Update $\theta$ using $\Delta\mathcal{L}_d + \Delta\mathcal{L}_v$ ▷ Model weights are updated based on both visual and classification loss.
  **end for**
**end for**

set to 0.01. Scaling the individual loss values in this way allows a model to focus on reducing the distance between the generated and original images while placing the visualization loss in a similar range as the classification loss. The classification loss, represented by Equation 2, is the standard implementation of categorical cross-entropy where $\hat{y}$ is the target model's predictions and $y$ is the dataset labels. The total loss function for our model is the summation of the visualization loss and the classification loss.

## 4. Experiments & Results

In evaluation of our proposed strategy, we apply our reconstruction strategy as an adversarial defense, and compare it with state-of-the-art defense approaches against established attacks. As discussed, some recent work in both attack and defense are limited in scalability, and better demonstrated on smaller datasets with few classes [4]. Similar strategies proposed in these types of related works are constructed for smaller, shallow neural networks [20]. In an effort to maintain our focus on real-world data, we design experiments around data with more classes - the ImageNet Large Scale Visual Recognition Challenge 2012 (ILSVRC2012) dataset[27] - and the models which can accommodate real-world vision applications.

### 4.1. Data and Classifier Set-up

For this experiment, we leverage the ILSVRC2012 Validation Dataset[27]. This contains 50,000 images which were used to evaluate the effectiveness of ImageNet[7] models InceptionV3[34] and VGG19[30]. In order to maintain a fair comparison between a variety of defenses, the images were sequentially split up into three parts; a training set of 45,000, a testing set of 2,500 and a validation set of 2,500. The training and testing sets were used exclusively for training our model. The validation set was used to analyze the effectiveness of each defense and it remained unseen to our model during the training process.

| Attack Name | Parameters |
|---|---|
| FGSM[10] | $\epsilon = 0.01$ |
| PGD[19] | $\epsilon = 0.01$, $\alpha = 0.004$, Iterations 10 |
| DeepFool[21] | Overshoot 0.02, Max Iterations 50 |

Table 1. Attacks and parameters used for experiments. Parameters scaled for images within a 0 to 1 range.

In these experiments, the proposed approach leveraged the VGG19[30] model in order to help it learn how to improve classification, and demonstrate usefulness on an established model. While we chose to pair it with VGG19[30] model, it is important to note that the discriminator can be easily swapped out with another pre-trained classification model, such as InceptionV3[34], ResNet50[14], and many others, as the discriminator remains frozen in its initial state. As we will show in our second experiment, the proposed strategy should ideally be paired with the target model you wish to defend.

### 4.2. Attacker Set-up

All attacks that were used in the experiments were set-up as untargeted black-box attacks where the attacker has full knowledge of the classification model and its parameters but no knowledge of the defense method that is being used. Since the models had full knowledge of the classification models, the images were preprocessed with respect to the relevant classification model before applying the attack. Table 1 provides the parameters used in each attack. Parameters for attacks were chosen with the intention of reducing the classification rates of the target model by at least 50%. Beyond this criteria, a subjective visual inspection was performed to ensure the perturbations were not easily visible to the naked eye. It is important to note that incorrectly classified images were not removed from the data set, resulting in an initial classification accuracy of 72% for VGG19[30] and

initial classification accuracy of 71% for InceptionV3[34]. For DeepFool[21], rather than using the model labels in order to to attack the images, the attack was given the ground truth labels instead. While this left images that were originally misclassified as "unattacked", it ensured that we did not have a case where an incorrectly classified image was accidentally changed to be correctly classified after the attack due to the nature of an untargeted DeepFool[21] attack.

### 4.3. Adversarial Defense

To test the effectiveness of our defense we trained two separate models; one where the training set was attacked by FGSM[10] and the other where the training set was attacked by PGD[19], denoted by ours$_{FGSM}$ and ours$_{PGD}$ respectively. The results of this experiment are shown in Table 2. Comparison with aforementioned related works is provided in this table, with the exceptions of MagNet[20] and ME-Net[39], which were designed for significantly smaller data and with more perturbations than what is analyzed in this experiment.

A visual comparison of our input transformation in regards to other input transformations are shown in Figure 2. It is important to note that while ours$_{FGSM}$ has the best performance in defending against FGSM attacks, we consider it to have an unfair advantage as it was exposed to FGSM attacks during training. Despite having never been exposed to DeepFool[21] or PGD[19] attacks, ours$_{FGSM}$ was able to improve classification performance against these attacks significantly. ours$_{PGD}$, which was trained on PGD alone and was never exposed to FGSM or DeepFool, similarly performed well. This provides evidence that our model is able to defend against attacks not in the training set. Another point of interest is that, ours$_{FGSM}$ was able to push the discriminator above the initial baseline accuracy when defending against FGSM attacks. We believe this is due to the autoencoder learning FGSMs attack patterns. Because adversarial attacks are created using the gradient of the image in relation to its ground-truth classification, there is an inherent amount of information leakage. This extra information, which is not normally available, is being used to push the classification above its baseline accuracy and can be considered a form of overfitting. While our approach was able to successfully defend against the attacks, it did suffer a penalty when defending non-attacked images. The penalty incurred by the image reconstruction is in part due to the AE expecting every image to contain corruptions, and is therefore applying defensive transformations to every image. While these transformations defend against adversarial perturbations, they also hinder classification of non-attacked images.

Many defenses are able to defend the image, however they lose critical visual details. Figure 3 shows a closer inspection of an image after state-of-the-art defenses have been applied to it. Ours focuses on maintaining the perceptual quality of the salient object, which is often crucial to object classification/detection tasks. Due to the fact that ours focuses on the object important for classification, we note an occasionally perceptible loss of quality when reconstructing the background of the image.

| | No Attack | FGSM [10] | PGD [19] | DeepFool [21] |
|---|---|---|---|---|
| No Defense | 0.722 | 0.060 | 0.007 | 0.052 |
| PD[24] | 0.622 | 0.140 | 0.072 | 0.436 |
| RRP[36] | 0.642 | 0.159 | 0.062 | 0.453 |
| FD[18] | **0.707** | 0.108 | 0.041 | 0.475 |
| STL[33] | 0.650 | 0.235 | 0.224 | 0.650 |
| ours$_{FGSM}$ | 0.665 | *0.752** | **0.709** | **0.677** |
| ours$_{PGD}$ | 0.644 | **0.669** | *0.681** | 0.662 |

Table 2. Classification accuracy (top 1) for VGG19[30]. (*) Denotes model has prior knowledge of attack.

| | No Attack | FGSM [10] | PGD [19] | DeepFool [21] |
|---|---|---|---|---|
| No Defense | 0.705 | 0.132 | 0.012 | 0.010 |
| PD[24] | 0.636 | 0.251 | 0.267 | 0.515 |
| RRP[36] | 0.685 | 0.343 | 0.340 | 0.576 |
| FD[18] | **0.694** | 0.199 | 0.154 | 0.509 |
| STL[33] | 0.640 | 0.341 | 0.430 | 0.587 |
| ours$_{FGSM}$ | 0.666 | *0.466** | **0.528** | **0.623** |
| ours$_{PGD}$ | 0.647 | **0.380** | *0.474** | 0.611 |

Table 3. Classification accuracy (top 1) for InceptionV3[34]. Both instances of our approach are trained using VGG19[30]. (*) Denotes model has prior knowledge of attack.

### 4.4. Defense Generalization

In order to extensively test the robustness of our model we designed an experiment to explore how well our approach generalizes across discriminators. This was done to ensure that the model was not overfitting on the specific latent space of the given discriminator.

In this experiment, we trained using VGG19[30], but evaluated using InceptionV3[34]. Furthermore all attacks were performed targeting InceptionV3[34]. Since ours$_{FGSM}$ and ours$_{PGD}$ used VGG19[30] as their pre-trained discriminator during the training process, and all attacks provided during training time were targeting VGG19, the only common ground to leverage was the dataset. Table 3 show the results of these experiments. From these results we can see that while some performance was lost, the defense was still effective compared to other methods.

This result is significant as it demonstrates that our approach is, in part, translating attacked images back to the distribution of the original dataset instead of overfitting on

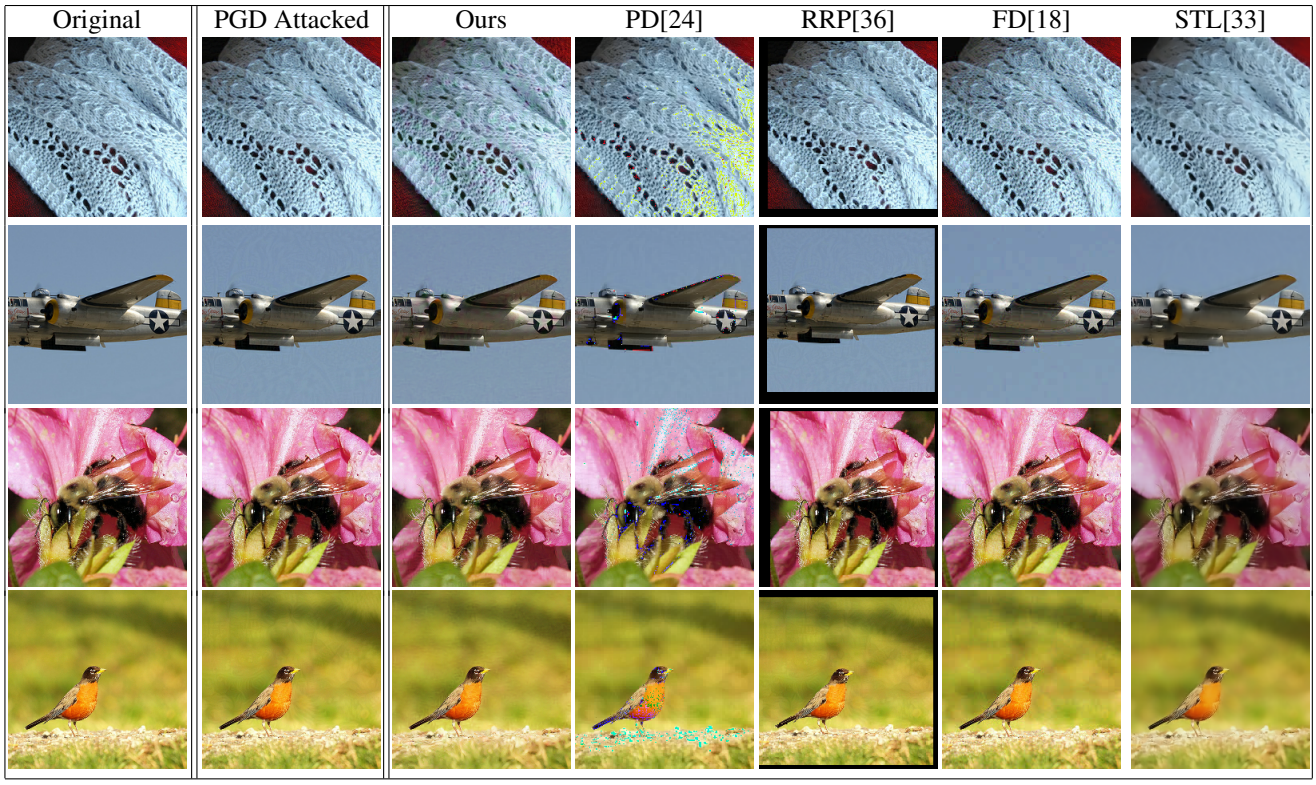


Figure 2. Comparison of 4 state-of-the-art defenses and our proposed approach. Images from ILSVRC2012[27] validation set. (Best viewed in color.)

the latent space of the training discriminator. We hypothesize that its ability to transfer across models is due in part to the ability of attacks to transfer across models [10]. This property of adversarial attacks, should allow for a defense targeted toward a specific model to transfer to other models.

Although our defense was not as effective as it was against VGG19[30] attacks, it was still the top performer in most of the categories. This shows that both ours$_{FGSM}$ and ours$_{PGD}$ were able to highlight features in the dataset, allowing the same model to defend networks that were trained using similar data. It also highlighted the importance of using the network you want to defend as the the pretrained discriminator during the proposed training process. Since we use a training process similar to that of a GAN, the feedback from the target model is crucial to creating more tailored image reconstructions which highlight features that are important to its classification. Without the target models feedback during the training process, our model could only leverage image similarity and would be unable to guarantee improvement on classification accuracy.

## 5. Conclusions

In this work, we present a novel generative input transformation strategy which improves classification performance on corrupted image data. We utilize a fully convolutional autoencoder to remove corruptions through image reconstruction, and propose a training strategy wherein both corrupted images and their original counterparts are provided to facilitate the reconstruction cleaning process. In doing so, we are able to reduce overfitting while maintaining visual quality for the reconstructed image. Finally, we showed the transferability across models without having to retrain it, and examined the penalty in terms of qualitative and quantitative metrics. Our proposed approach's ability to defend against attack types not represented in the training set, and its ability to translate across models without re-training demonstrates that it is not fixating on specific adversarial patterns but is translating images back to the original dataset distribution.

In future work, we aim to apply these findings to instrumentation corruptions found in experimental image data from the physical sciences. We will look to improve performance on non-corrupted images, while maintaining the performance demonstrated on corruption removal. To further examine performance prior to applying to experimental data, we aim to evaluate on new non-adversarial datasets, such as ImageNet-C[15] which incorporates real-world corruptions such as fog and frost.

This work focused on improvement of image classifica-

Figure 3. Visual quality comparison between defense approaches. Row 1: original, zoomed original, PGD[24] attacked, ours. Row 2:FD[18], STL[33], RRP[36], PD[24]. (Best viewed in color)

tion models exclusively from a top-1 perspective, and we will explore in future work the applicability of our proposed strategy on multi-label classification tasks. This is a particularly important extension needed for vision models dealing with real-world data, where there are a large number of objects to classify/localize, visual clutter, and objects in unusual contexts.

Finally, we note that to be effective against all types of adversarial example, a good defense should be extended to non-gradient based attacks, such as BPDA[1]. Given the prevalence of gradient-based training approaches, we have focused on these in this work, however BPDA, EoT, and others will be another category of corruptions to consider in future work in this field.

## Acknowledgement

This work was supported by the National Nuclear Security Administration, Minority Serving Institutions Partnership Program DE-NA0003948. This project was funded in part by the University of Texas at San Antonio, Office of the Vice President for Research, Economic Development & Knowledge Enterprise.

## References

[1] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *International Conference on Machine Learning*, pages 274–283, 2018.

[2] Xiaoyu Cao and Neil Zhenqiang Gong. Mitigating evasion attacks to deep neural networks via region-based classification. In *Proceedings of the 33rd Annual Computer Security Applications Conference*, pages 278–287, 2017.

[3] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pages 39–57. IEEE, 2017.

[4] Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *International conference on machine learning*, pages 2206–2216. PMLR, 2020.

[5] Ali Dabouei, Sobhan Soleymani, Fariborz Taherkhani, Jeremy Dawson, and Nasser Nasrabadi. Smoothfool: An efficient framework for computing smooth adversarial perturbations. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, March 2020.

[6] Nilaksh Das, Madhuri Shanbhogue, Shang-Tse Chen, Fred Hohman, Siwei Li, Li Chen, Michael E Kounavis, and Duen Horng Chau. Shield: Fast, practical defense and vaccination for deep learning using jpeg compression. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 196–204, 2018.

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[8] Valery Dewil, Jeremy Anger, Axel Davy, Thibaud Ehret, Gabriele Facciolo, and Pablo Arias. Self-supervised training for blind multi-frame video denoising. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 2724–2734, January 2021.

[9] Gintare Karolina Dziugaite, Zoubin Ghahramani, and Daniel M Roy. A study of the effect of jpg compression on adversarial images. *arXiv preprint:1608.00853*, 2016.

[10] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *stat*, 1050:20, 2015.

[11] Kathrin Grosse, Nicolas Papernot, Praveen Manoharan, Michael Backes, and Patrick McDaniel. Adversarial exam-

ples for malware detection. In *European Symposium on Research in Computer Security*, pages 62–79. Springer, 2017.

[12] Shuangchi Gu, Ping Yi, Ting Zhu, Yao Yao, and Wei Wang. Detecting adversarial examples in deep neural networks using normalizing filters. *UMBC Student Collection*, 2019.

[13] Chuan Guo, Mayank Rana, Moustapha Cisse, and Laurens van der Maaten. Countering adversarial images using input transformations. In *International Conference on Learning Representations*, 2018.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[15] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations*, 2018.

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

[17] Yang Liu, Zhen Zhu, and Xiang Bai. Wdnet: Watermark-decomposition network for visible watermark removal. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 3685–3693, January 2021.

[18] Zihao Liu, Qi Liu, Tao Liu, Nuo Xu, Xue Lin, Yanzhi Wang, and Wujie Wen. Feature distillation: Dnn-oriented jpeg compression against adversarial examples. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 860–868. IEEE, 2019.

[19] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018.

[20] Dongyu Meng and Hao Chen. Magnet: a two-pronged defense against adversarial examples. In *Proceedings of the 2017 ACM SIGSAC conference on computer and communications security*, pages 135–147, 2017.

[21] 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.

[22] Taesik Na, Jong Hwan Ko, and Saibal Mukhopadhyay. Cascade adversarial machine learning regularized with a unified embedding. *arXiv preprint arXiv:1708.02582*, 2017.

[23] Aran Nayebi and Surya Ganguli. Biologically inspired protection of deep networks from adversarial attacks. *arXiv preprint arXiv:1703.09202*, 2017.

[24] 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.

[25] Yao Qin, Nicholas Carlini, Garrison Cottrell, Ian Goodfellow, and Colin Raffel. Imperceptible, robust, and targeted adversarial examples for automatic speech recognition. In *International Conference on Machine Learning*, pages 5231–5240, 2019.

[26] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[27] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[28] Pouya Samangouei, Maya Kabkab, and Rama Chellappa. Defense-gan: Protecting classifiers against adversarial attacks using generative models. In *International Conference on Learning Representations*, 2018.

[29] Mahmood Sharif, Sruti Bhagavatula, Lujo Bauer, and Michael K Reiter. Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 1528–1540. ACM, 2016.

[30] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[31] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*, 2014.

[32] Jiawei Su, Danilo Vasconcellos Vargas, and Kouichi Sakurai. One pixel attack for fooling deep neural networks. *IEEE Transactions on Evolutionary Computation*, 2019.

[33] Bo Sun, Nian-hsuan Tsai, Fangchen Liu, Ronald Yu, and Hao Su. Adversarial defense by stratified convolutional sparse coding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11447–11456, 2019.

[34] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[35] Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick Drew McDaniel. Ensemble adversarial training: Attacks and defenses. In *6th International Conference on Learning Representations, ICLR 2018*, 2018.

[36] Cihang Xie, Jianyu Wang, Zhishuai Zhang, Zhou Ren, and Alan Yuille. Mitigating adversarial effects through randomization. In *International Conference on Learning Representations*, 2018.

[37] Cihang Xie, Yuxin Wu, Laurens van der Maaten, Alan L Yuille, and Kaiming He. Feature denoising for improving adversarial robustness. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 501–509, 2019.

[38] Weilin Xu, David Evans, and Yanjun Qi. Feature squeezing: Detecting adversarial examples in deep neural networks. *arXiv preprint arXiv:1704.01155*, 2017.

[39] Yuzhe Yang, Guo Zhang, Dina Katabi, and Zhi Xu. ME-Net: Towards effective adversarial robustness with matrix estimation. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, 2019.

[40] Peder Georg Olofsson Zwilgmeyer, Mauhing Yip, Andreas Langeland Teigen, Rudolf Mester, and Annette Stahl. The varos synthetic underwater data set: Towards realistic multi-sensor underwater data with ground truth. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3722–3730, 2021.