Adversarial Defense by Stratified Convolutional Sparse Coding

Bo Sun
Peking University
bosun@pku.edu.cn

Nian-hsuan Tsai
National Tsinghua University
nianhsuan@gmail.com

Fangchen Liu Ronald Yu Hao Su
UC San Diego
{fliu,ronaldyu,haosu}@eng.ucsd.edu

Abstract

We propose an adversarial defense method that achieves state-of-the-art performance among attack-agnostic adversarial defense methods while also maintaining robustness to input resolution, scale of adversarial perturbation, and scale of dataset size. Based on convolutional sparse coding, we construct a stratified low-dimensional quasi-natural image space that faithfully approximates the natural image space while also removing adversarial perturbations. We introduce a novel Sparse Transformation Layer (STL) between the input image and the first layer of the neural network to efficiently project images into our quasi-natural image space. Our experiments show state-of-the-art performance of our method compared to other attack-agnostic adversarial defense methods in various adversarial settings.

1. Introduction

Existing defense mechanisms against adversarial attacks, although able to achieve robustness in certain adversarial settings, are still unable to achieve true robustness to all adversarial inputs. The most effective existing defense methods modify the network training process to improve robustness against adversarial examples [18, 24, 51, 34]. However, they are trained to defend a specified attack for a specified model, limiting their real-world applications and claims of robustness to *all* adversarial inputs. Ideally, our defense mechanism should be *attack agnostic* and *model agnostic*.

Instead of modifying the network and training process, another line of existing methods achieve the desired property of being attack-agnostic and model-agnostic by modifying adversarial inputs to resemble clean inputs [12, 16, 36, 54, 40, 46, 32, 43]. However, these methods show weaknesses in other adversarial settings such as being unable to handle larger perturbations, unable to simultaneously handle many different resolutions, and not scalable to large datasets.

In this paper, we present an input-transformation based defense method that achieves state-of-the-art performance when compared to previous attack-agnostic and model-agnostic defense methods. Moreover, our method is also far more simultaneously robust to scale of attack perturbation, a variety of different input resolutions, and dataset scale. We achieve our high level of robustness by projecting both clean and adversarially attacked input images into a low-dimensional *quasi-natural image space* that faithfully approximates the natural image space while also removing adversarial perturbations so that adversarial examples will be close to their original inputs in feature space.

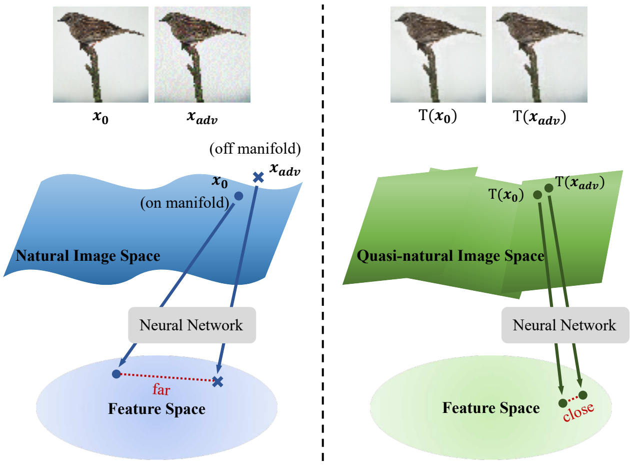

Figure 1: Comparison of feature extraction between natural image space and our learned quasi-natural image space. In the natural image space, a neural network trained from natural images may assign different labels to the adversarial example and the clean image, since they can be far from each other in the feature space. After projection to the quasi-natural image space, they tend to lie closely together in the feature space.

We construct the quasi-natural image space in an unsupervised manner using a convolutional dictionary learning-based [4, 22, 8, 53] method, and we project the input images into our quasi-natural image space by introducing a novel *Sparse Transformation Layer (STL)* between the input and first layer of the network. We can further enhance the robustness of our pipeline by retraining a classifier on the quasi-natural images.

Experimentally, we demonstrate that our method achieves a significant robustness improvement in a variety of different adversarial settings compared with state-of-the-art attack-agnostic defense methods. We also show that our quasi-natural image space is able to provide a better blend of preservation of image details and ability to remove adversarial perturbations compared to other input-transformation-based adversarial defense methods.

In summary, our contributions are:

- We propose a novel and effective attack-agnostic adversarial defensive method that uses a novel *Sparse Transformer Layer* to transform images so that corresponding clean and adversarial images lie close together both in our *quasi-natural image space* and feature space.
- We demonstrate that our defense method achieves state-of-the-art performance among attack-agnostic adversarial defense methods.
- Compared to previous state-of-the-art, our defense method is far more capable of effectively handling a variety of image resolutions, large and small image perturbations, and large-scaled datasets.
- Among image-transformation-based adversarial defenses, our image projection onto quasi-natural image space achieves the best blend of image detail preservation and ability to remove adversarial perturbations.

2. Related Works

Adversarial Attacks Adversarial attacks are inputs that are intentionally slightly perturbed to fool machine learning models. Szegedy et al. [49] first introduce adversarial examples and generate them with the box-constrained L-BFGS method. Goodfellow et al. [18] propose an efficient single step attack called FGSM based on network linearity. Kurakin et al. [25] apply FGSM iteratively and propose BIM. DeepFool [33] finds the smallest perturbation crossing the model decision boundary. CW [7] solves an optimization problem which minimizes both the objective function and difference between adversarial and clean images. Liu et al. [27] generate strongly transferable adversarial examples with an ensemble-based approach. Non-gradient based attacks such as one pixel attack [47] and Zoo [9] do not require knowledge of network parameters and architecture.

Adversarial Defense via Network Modification This type of defense aims to improve the robustness of the target model against adversarial examples. The most common method is adversarial training [18, 24, 51, 34] which adds adversarial examples into training data. This class of methods effectively enhances robustness to the adversarially trained attacks but has poor generalizability to unknown attacks. Other methods like feature squeezing [55], network distillation [39], region-based classifier [6] and saturating networks [35] modify the learning strategy based on gradient masking [38] and smooth the decision boundary, but they are still vulnerable to black-box attacks [7, 37].

Adversarial Defense via Input Transformation Input-transformation defenses aim to remove adversarial perturbation transforming inputs before feeding them to the target network. Some previous methods treat adversarial perturbation as high frequency noise and resort to traditional denoising methods to smooth small perturbations. [12, 16] study the effect of JPEG compression on removing adversarial noise. Osadchy et al. [36] apply a set of filters such as median filter and averaging filter to remove perturbation. Guo et al. [20] test five transformations and find total variation minimization and image quilting obtain good defensive performance. These denoising methods only fix small perturbations and suffer from information loss.

More recently, other works have tried to purify adversarial images through generative models. Meng et al. [32] propose a two-pronged defense mechanism and use a denoising auto-encoder to remove adversarial perturbation on MNIST digits [26]. Song et al. [46] transform adversarial images into clean images using PixelCNN [42]. Although they achieve good performance on small datasets, these methods do not scale well to higher-resolution or larger datasets.

Pixel manipulation methods are also used to remove small adversarial perturbations. Xie et al. [54] utilize random resizing and padding to mitigate adversarial effects. Prakash et al. [40] locally corrupt adversarial images by redistributing pixel values via a process we term pixel deflection. However, these methods suffer when they encounter perturbations that are not extremely small.

Most similar to our method is D3 [29], which denoises adversarial images by replacing patches with a sparse combination of natural images patches. Further discussion of D3 is reserved for Section 4.

Convolutional Dictionary Learning Convolutional sparse representations are a form of sparse representation learning [31] with a dictionary that has a structure that is equivalent to convolution with a set of linear filters [17, 4]. It is widely and successfully used in signal processing and computational imaging [19, 28, 41, 56, 57, 44]. Many efficient algorithms [22, 4, 8, 53, 11] have been developed to solve this problems. Sung et al. also recently introduced a method that used a deep neural network to learn sparse dictionaries for 3D point clouds [48].

3. Approach

3.1. Method Overview

Let $\mathcal{X}$ be the image space and $\mathcal{Y}$ be the label space. $f_\theta(\cdot) : \mathcal{X} \rightarrow \mathcal{Y}$ is a classifier parameterized by θ. Given the classifier $f_\theta(\cdot)$ and a clean image $\boldsymbol{x_0}$, an adversarial example $\boldsymbol{x}_{adv} = \boldsymbol{x_0} + \boldsymbol{\eta}$ is an image slightly different from $\boldsymbol{x_0}$ but confuses f:

$$d(\boldsymbol{x}_{adv}, \boldsymbol{x_0}) < \epsilon \text{ but } f_\theta(\boldsymbol{x}_{adv}) \neq f_\theta(\boldsymbol{x_0}), \tag{1}$$

where $d(\cdot, \cdot)$ is a distance function between the clean and adversarial images. ϵ is the perturbation scale which is often set to a small number to get almost imperceptible difference between $\boldsymbol{x}_{adv}$ and $\boldsymbol{x_0}$.

Adversarial examples $\boldsymbol{x}_{adv}$ are fabricated images and usually lie out of the natural image manifold. This may cause the network trained from natural images, even with adversarial data augmentation, to map $\boldsymbol{x}_{adv}$ far away from $\boldsymbol{x_0}$ (Figure 1 Left). Our idea is thus to recover $\boldsymbol{x_0}$ as much as possible by projecting $\boldsymbol{x}_{adv}$ to the natural image manifold. However, parameterizing the true natural image manifold is practically infeasible. We instead leverage manifold learning to build a low-dimensional space that approximates the natural image space, which we dub the *quasi-natural space* $\mathcal{P}$ in this paper. Along with $\mathcal{P}$, there is a transformation T that maps an image (natural or spoofed) to $\mathcal{P}$. We require that T satisfies the following constraints:

1. $f_\theta(T(\boldsymbol{x}_{adv})) = f_\theta(T(\boldsymbol{x_0})) = y_{\boldsymbol{x_0}}$;
2. $d(T(\boldsymbol{x}_{adv}), T(\boldsymbol{x_0})) \ll \epsilon$.

Condition 1 requires that the classifier f assigns the same groundtruth label to $\boldsymbol{x}_{adv}$ and $\boldsymbol{x_0}$, which is our final goal. To guarantee Condition 1, other than learning f to optimize classification accuracy, we also introduce Condition 2 (Figure 1 Right). Condition 2 requires that $\boldsymbol{x}_{adv}$ and $\boldsymbol{x_0}$ should be situated closely in $\mathcal{P}$, so that we can learn a quite smooth function f satisfying Condition 1. This is important since our f is a neural network, and learning a smoother map would endow the it better generalization power.

We take an unsupervised approach to build the quasi-natural image space. This space is constructed by stitching multiple low-dimensional linear subspaces together. Practically, we cluster the training data into a few groups and we learn a linear subspace for each group by convolutional sparse coding algorithm [22, 11]. With this quasi-natural space constructed, we are able to project any image to this space by the sparse transformation layer introduced in Section 3.2, which will remove a significant amount of adversarial perturbations. Then in this quasi-natural image space we can retrain a classifier to allow robust prediction over adversarial examples (Section 3.5).

3.2. Sparse Transformation Layer (STL)

Given a classification network f, we add a Sparse Transformation Layer (STL) between the input image and the first layer of f. This STL layer projects the input (adversarial or clean) onto a quasi-natural space, which removes nuisances including adversarial perturbations in the appearance.

Let the projection of $\boldsymbol{x}$ be $T(\boldsymbol{x})$ (assume that $\boldsymbol{x}$ is an image of C channels). The projection in our STL layer follows from the Convolutional Sparse Coding algorithm [11]. This algorithm learns a dictionary in a convolutional manner by solving the following optimization problem:

$$\begin{aligned} \underset{\{\boldsymbol{f}_{i,c}\},\{\boldsymbol{z}_i\}}{\text{minimize}} \quad & \frac{1}{2}\sum_{c=1}^{C} \|\boldsymbol{x}_c - T(x)_c\|_2^2 + \lambda \sum_{i=1}^{K} \|\boldsymbol{z}_i\|_1 \\ \text{subject to} \quad & T(x)_c = \sum_{i=1}^{K} \boldsymbol{f}_{i,c} \otimes \boldsymbol{z}_i \\ & \|\boldsymbol{f}_{i,c}\|_2^2 = 1, 1 \leq i \leq K, 1 \leq c \leq C \end{aligned} \tag{2}$$

where $\otimes$ indicates the convolution operator, C is the number of input channels, K is the number of filters for each input channel, $\boldsymbol{f}_{i,c}|_{i=1,\ldots,K;c=1,\ldots,C}$ denotes a set of filters, and $\boldsymbol{z}_i|_{i=1,\ldots,K}$ are the feature maps for each filter.

Different from standard sparse coding, which learns a dictionary and code for the whole image, as shown by [50], Problem (2) learns to reconstruct image patches by local dictionaries and codes. Here, the local dictionary contains the set of filters $\boldsymbol{f}_{i,c}$, and local codes are stored in the feature map $\boldsymbol{z_i}$. The convolution operation in the constraint essentially computes the linear combination of local filters. In vanilla sparse coding, a small set of bases are selected to reconstruct the image. Similarly, in the convolutional sparse coding formulation, a small set of filters should be selected to reconstruct a local patch. To achieve the filter selection goal, we have to enforce the feature map $\boldsymbol{z_i}$ to be sparse by adding the ℓ_1 regularization term.

In practice, we prefer to use a small number of filters. This forces filters to learn major and expressive local patterns on the natural image manifold. Moreover, from our observation, having too many filters may cause extra filters to learn high frequency components, which can be used to reconstruct arbitrary image patches including adversarial perturbation that should be removed.

3.3. Learning Filters and Feature Maps

Plugging the constraint in Problem (2) into the objective function, we see that Problem (2) is biconvex in $\boldsymbol{f}_{i,c}$ and $\boldsymbol{z_i}$. To solve this biconvex problem, we alternate between (1) learning shared filters from clean images, and (2) learning sparse feature maps for each input image with fixed filters. Next we briefly introduce these two stages.

Dictionary Learning. Given feature maps, Problem (2) becomes convex in $\boldsymbol{f}_{i,c}$. To solve this problem efficiently, we transform to the Fourier domain [52] and use ADMM algorithm as the solver following the framework of [11].

Sparse feature map (code) learning. Given fixed filters $\{\boldsymbol{f}_{i,c}\}$, our objective function is again a convex optimiza-

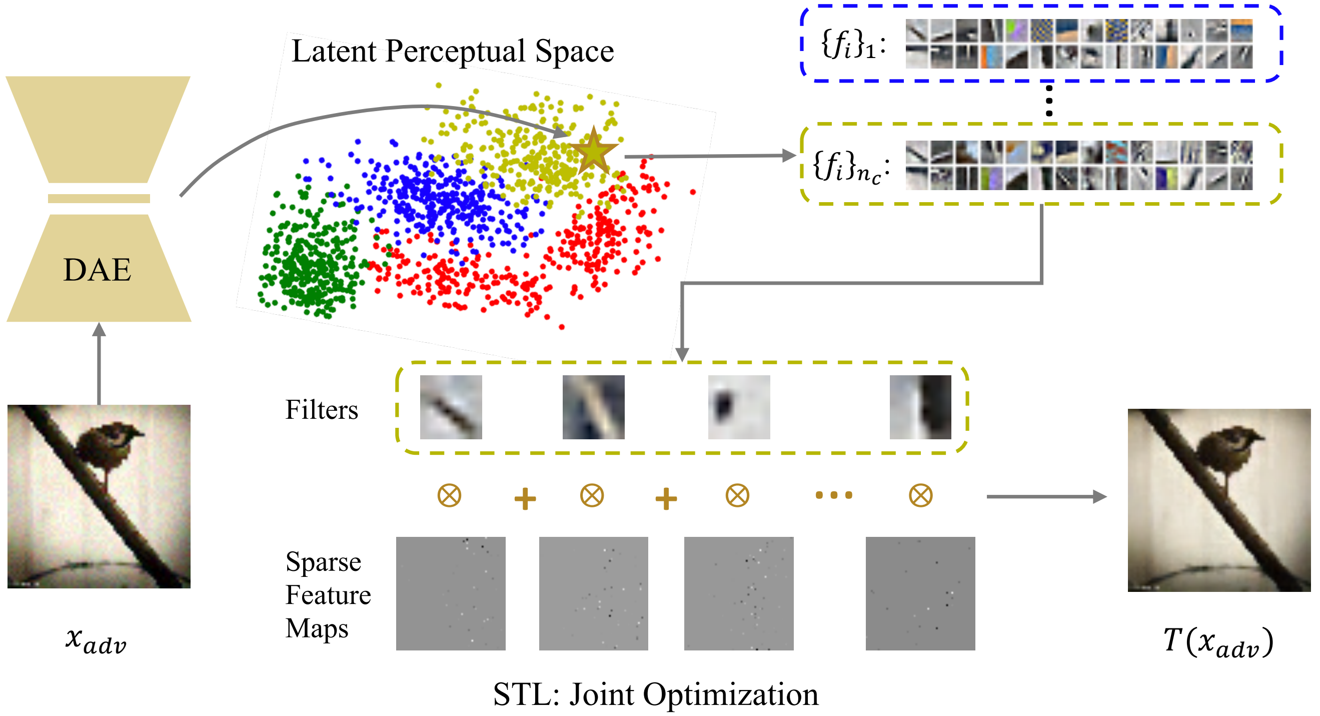

Figure 2: Pipeline of our defensive method. We first feed an image to a pre-trained Denoising Auto-Encoder and find the cluster the image should belong to. Then we select the dictionary corresponding to the selected cluster and jointly optimize the sparse feature maps and filters in this dictionary. In this way, we can project the input to the quasi-natural image space.

tion problem in $\boldsymbol{z_i}$. The problem is also known as Convolutional Basis Pursuit DeNoising (CBPDN) [10] and we use ADMM algorithm [3] to solve it.

3.4. Stratified Quasi-Natural Image Space

Due to the high inherent variation of natural images, it is hard to well reconstruct all images using just a small dictionary. However, as we discussed at the end of Section 3.2, we also do not want to employ a too big dictionary, because the big dictionary will span an excessively high-dimensional space, inevitably covering a significant amount of non-natural images. This would reduce the power of our algorithm to filter out adversarial perturbations.

To circumvent the challenge, we split the data manifold into several regions and learn an individual small dictionary for each region. In this way, each image is still reconstructed by a small dictionary, but we can still reconstruct all images well using their corresponding dictionaries.

In practice, we partition the image space by clustering natural image samples based on their perceptual features. Generative models can learn perceptual features by reconstruction loss. In particular, we find that Denoising Auto-Encoder (DAE) [2] fits the adversarial setting well because it is trained with noisy input and the feature extraction process can modestly tolerate input noise. Specifically, we train a DAE on both natural images and their noise-perturbed versions (Gaussian noise). In practice we find that the original image and adversarial attacked version usually live closely in the latent space learned by DAE. We then use the K-means algorithm to cluster training data [13].

The clusters allow us to partition the natural image manifold. Given an arbitrary input image (adversarial or clean), we can obtain its latent features from the DAE and find the k-nearest neighbors in the training image dataset. Then we vote for the cluster the image should belong to. Once we have found the cluster, we can either update the filters and features maps for dictionary learning, or compute the projection of the image for classification network training/test.

3.5. Classifier Training in the Quasi-Natural Space

To train a classifier for image categorization, we map all the clean training images to $\mathcal{P}$. We simply use their reconstructed version $T(\boldsymbol{x_0})$ to train a user-selected classification network (e.g., AlexNet). To perform defense at test time, we apply the trained classifier on the T-transformed version of the testing image (clean or adversarial).

After projection to $\mathcal{P}$, $T(\boldsymbol{x}_{adv})$ and $T(\boldsymbol{x_0})$ share close perceptual and semantic features. Therefore, decisions made in this quasi-natural space $\mathcal{P}$ tend to be more reliable for adversarial examples compared to the original space.

4. Discussion

In this section, we discuss our unique advantages over existing adversarial defenses and then analyze possible reasons behind the effectiveness of our method against popular gradient-based attack methods.

Relationship With Existing Methods In contrast to adversarial learning methods [18, 24, 51] that rely on direct knowledge of the attack method and model type, our algorithm only relies on the clean training data at hand. Built without any explicit prior knowledge of the attacker, our design does not overfit to any specific attack strategy and tends to be a generic tool.

Recent attack-agnostic defense methods use generative models to transform images into a low-dimensional space [32, 46, 43]. We choose not to use a network to build our low-dimensional space, since the generative network itself is vulnerable to adversarial attacks. Another disadvantage of these methods is that the limited expressive power of generative models restricts the domain of these methods to datasets small in resolution and scale such as MNIST [26] and CIFAR-10 [23]. Pixel manipulation methods [40, 54] can work on large datasets, but they only achieve good performance under extremely small perturbations. Our method works uniquely well on large adversarial perturbation, complicated datasets, and higher resolutions.

The D3 algorithm proposed in [29] is the most similar to ours. It replaces noisy adversarial image patches by a sparse combination of natural image patches. However, our method provides several advantages. First, D3 reconstructs images poorly on low-resolution datasets like CIFAR-10 [23]. Second, the size of the natural patch dictionary is very large (10K-40K) while we only need a small number of filters (typically 64). The large size of their patch dictionary has two main drawbacks: the excessive number of dictionary elements may lead the dictionary to learn high frequency components, which can be used to wrongly reconstruct adversarial perturbations, and the generic dictionary elements are not as expressive as ours, so D3 generates images that are not as sharp as ours as verified in our experiments.

Robustness to Gradient-Based Attacks There are two main concepts behind the effectiveness of our method against gradient-based attacks: (1) *Gradient Obfuscation:* Obtaining the numerical gradient of the STL is likely to be challenging, because the output of the STL is the solution to a non-convex optimization problem (has the $\arg\min$ form of the input image). Without the gradient of the STL, designing gradient-based attack becomes difficult. (2) *High-frequency Perturbation Removal:* Existing gradient-based attack mechanisms often introduce high-frequency perturbations. With a small dictionary and the sparsity constraint in Problem (2), the learned filters tend to be quite smooth (Figure 2), which could filter out the high-frequency perturbation patterns.

5. Experiments

In this section, we first introduce our experimental settings, and then show a quantitative and qualitative comparison with other attack-agnostic adversarial defenses. We demonstrate that our method outperforms the state-of-the-art. Lastly, we perform an analysis of the intrinsic trade-off between projection image quality and defense robustness of transformation-based defenses.

5.1. Settings

We conduct experiments on CIFAR-10 [23], ImageNet [14], and ImageNet-10, where we manually choose 10 coarse-grained classes from the whole dataset, e.g. bird, car, cat, etc. Every class contains 8000 training and 2000 testing images.

We evaluate our method on VGG-16 [45] and ResNet-50 [21] to defend against FGSM [18], BIM [25], DeepFool [33], and CW [7]. We constrain the perturbation scale $\|\boldsymbol{\eta}\|_2 = \frac{\|\boldsymbol{x}_{adv} - \boldsymbol{x_0}\|}{\|\boldsymbol{x_0}\|}$ to 0.04 (FGSM-0.04) and 0.08 (FGSM-0.08) for FGSM and to 0.04 for BIM, DeepFool, and CW.

By default, we set the filter number $K = 64$, filter size $S = 8$, and sparse constraint $\lambda = 0.2$. We first downsample images to 32×32 to train a DAE, and split the latent space to 4 clusters for CIFAR-10 and ImageNet-10, and 10 clusters for ImageNet.

5.2. Adversarial Defense

We evaluate the defensive effectiveness of our method of retraining a classifier on quasi-natural images and then projecting adversarial examples onto the quasi-natural image space as described in Section 3.5.

Classification accuracy comparison results are in Table 1 for CIFAR-10, Table 2 for ImageNet-10 and Table 3 for ImageNet. In Table 1 and Table 2 we follow our setting as described in Section 5.1. In Table 3 we follow the experimental setting in [20] and [29]. Although we compare with other methods in their preferred resolution and datasets for a fair comparison, we note that one of the unique advantages of our method is that it performs well in various resolutions (in our experiments, from 32 to 224), while others can only work on a limited range of resolutions.

Comparison results show that our method significantly improves the classification robustness against unknown black-box attacks and outperforms state-of-the-art methods in most types of attacks with a large margin. Moreover, our retrained model achieves high accuracy on clean data and is comparable to the clean model, which means we preserve rich fine details that allow the network to learn discriminative features. Furthermore, we also compare our method

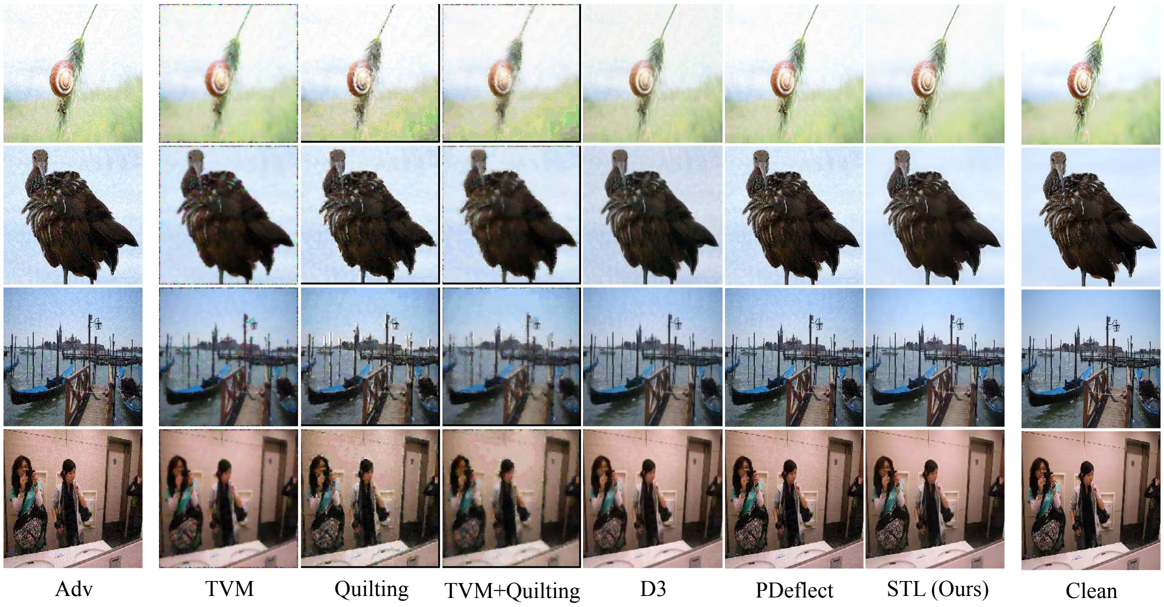

Figure 3: Qualitative comparison of image reconstruction results on ImageNet. The first column is input the adversarial examples generated by FGSM [18] attack with L_2 dissimilarity 0.08. The last column is the corresponding clean images. Visually, our method outperforms others on removing adversarial perturbations and keeping input details. D3 refers to [29] and PDeflect refers to [40].

Table 1: CIFAR-10 classification accuracy for adversarial examples on VGG-16 after defense by methods in comparison. All methods are trained and tested on their transformed data. "Clean" means accuracy of transformed clean data on each method. "STL" denotes STL transformation with a single universal set of filters. "STL (cluster)" denotes STL filters are chosen through latent space clustering.

Defense	Clean	FGSM-0.08	FGSM-0.04	BIM	DeepFool	CW
No Defense	0.9298	0.5816	0.6523	0.1803	0.1760	0.0936
MagNet[32]	**0.9206**	0.7393	0.8552	0.7707	0.8770	0.8594
PixelDefend[46]	0.9041	0.8316	**0.8799**	0.8515	0.8827	0.8845
STL	0.9002	0.8515	0.8732	0.8754	0.8838	0.8880
STL (cluster)	0.9011	**0.8567**	0.8715	**0.8803**	**0.8890**	**0.8904**

with the widely used adversarial training [24] and show that we achieve better results on unknown attacks (Appendix B).

Table 2: ImageNet-10 classification accuracy for adversarial examples on VGG-16 after defense by methods in comparison at resolution 64 (Table 2.A) and 128 (Table 2.B). All methods are trained and tested on their transformed data by their defense method. Here Crop-Ens denotes Crop+TVM+Quilting in [20] and PD-Ens denotes PD+R-CAM+DWT in [40].

Table 2.A Resolution 64.

Defense	Clean	FGSM-0.08	FGSM-0.04	BIM	DeepFool	CW
No Defense	0.8665	0.2816	0.3080	0.1883	0.0811	0.0751
TVM[20]	0.7555	0.5997	0.6930	0.7156	0.7210	0.7187
Quilting[20]	0.7741	0.7304	0.7418	0.7642	0.7646	0.7662
Crop-Ens[20]	0.7508	0.6968	0.7221	0.7369	0.7401	0.7304
PD-Ens[40]	0.8250	0.6634	0.7607	0.7903	0.7955	0.7813
STL	**0.8438**	0.7275	0.8002	**0.8164**	0.8163	0.8058
STL (cluster)	0.8421	**0.7514**	**0.8038**	0.8103	**0.8221**	**0.8122**

Table 2.B Resolution 128.

Defense	Clean	FGSM-0.08	FGSM-0.04	BIM	DeepFool	CW
No Defense	0.8991	0.2123	0.2409	0.1790	0.0584	0.0504
TVM[20]	0.8567	0.7302	0.8181	0.8183	0.8221	0.8101
Quilting[20]	0.8354	**0.7612**	0.7914	0.8048	0.8164	0.8093
Crop-Ens[20]	0.8382	0.7640	0.7969	0.8033	0.8071	0.7955
PD-Ens[40]	0.8603	0.6740	0.8011	0.8273	0.8320	0.8262
STL	**0.8784**	0.7202	0.8308	0.8320	**0.8560**	**0.8449**
STL (cluster)	0.8721	0.7421	**0.8356**	**0.8385**	0.8494	0.8421

5.3. White Box Attacks

Our defense is designed primarily for black/grey-box attacks, and like other methods, is highly susceptible to white-box attacks, especially on ImageNet [14]. Nevertheless, we show that our method is significantly less susceptible to the white-box attack Backward Pass Differentiable Approximation (BPDA) on CIFAR-10 [1]. BPDA specifically targets defenses in which the gradient does not optimize the loss; this is the case for our method since our STL is non-differentiable. Table 4 shows that although our defense accuracy is hurt by obfuscated gradient-based attacks, it is much more robust than other defenses with this phenomenon on CIFAR-10 dataset.

On ImageNet [14], all defense methods in their case study ([20] and [54]) get 0% defense accuracy. Under the

Table 3: Top-1 ImagetNet classification accuracy for adversarial examples on ResNet-50 after defense by methods in comparison. We follow experimental settings in [20] and [29] where all attacks are in an average normalized L_2-dissimilarity of 0.06. All methods are trained and tested on their transformed data.

Defense	Clean	FGSM	BIM	DeepFool	CW	UAP
No Defense	0.761	0.107	0.012	0.010	0.019	0.133
quilt[20]	0.701	0.655	0.656	0.652	0.641	-
TVM+quilt[20]	**0.724**	0.657	0.658	0.658	0.640	-
Crop-Ens[20]	0.721	0.667	0.670	0.671	0.635	-
D3 (40K-5)[29]	0.718	0.686	-	0.631	-	**0.715**
D3 (10K-5)[29]	0.708	0.683	-	0.646	-	0.703
D3 (10K-4)[29]	0.690	0.671	-	0.648	-	0.689
PD-Ens[40]	0.719	0.637	0.633	0.638	0.643	0.667
STL (cluster)	0.721	**0.693**	**0.678**	**0.685**	**0.677**	0.712

Table 4: Backward Pass Differentiable Approximation (BPDA) [1] attack results on CIFAR-10, VGG-16. All methods are attacked at distance $L_\infty = 0.031$. Defenses denoted with $*$ propose combining adversarial training.

Defense	SAP [15]	TE [5]	LID [30]	PD [46]	MagNet [32]	STL	STL (cluster)
Accuracy	0.00	0.00*	0.05	0.09*	0.10	0.38*	**0.42***

same settings, our defense accuracy similarly collapses to 1%. We further analyze our method's robustness to other simple custom-made white-box attacks with full knowledge of our model (including dictionary coefficients) in Appendix C.

5.4. Input Transformation Effectiveness

Since STL has a strong reconstruction capacity, the projected images still faithfully preserve information from the input data space. This is a useful property since it allows us to use a vanilla model to partially defend against adversarial examples when we are not able train our own classifier on quasi-natural images due to limitations such as access to the entire dataset.

Hence, we also evaluate the accuracy of using STL to project adversarial examples of a vanilla model that was pre-trained only on clean data. To perform the defense, we simply project the input into quasi-natural space and feed the projected image back into the vanilla model.

We compare with other input-transformation methods applied to attacked vanilla models in Table 5 for CIFAR-10, Table 6 for ImageNet-10 and Table 7). Qualitative comparisons of our input transformations are shown in Figure 4 for CIFAR-10 and Figure 3 for ImageNet. More results are in Appendix E.

Under relatively large perturbations (e.g. FGSM-0.08), all competing methods fail to successfully overcome adversarial attacks while our method significantly outperforms them. On slightly perturbed adversarial examples (e.g. DeepFool and CW), we achieve a strong defense and also maintain accuracy on clean data. We see that our method can effectively defend against adversarial attacks even using a vanilla clean model.

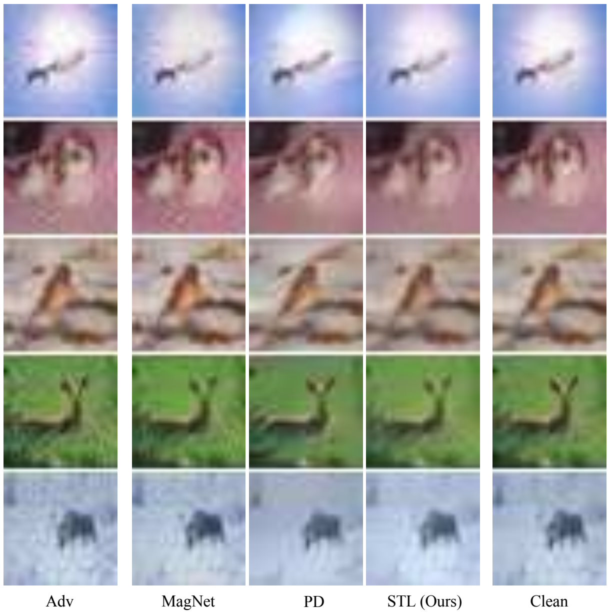

Figure 4: Qualitative comparison on CIFAR-10 [23] with MagNet [32] and PixelDefend (PD) [46]. The first column is adversarial images generated by FGSM [18] with L_2-dissimilarity = 0.08. The last column is corresponding clean images. We can observe that MagNet cannot fully remove adversarial perturbation, while PixelDefend oversmooths images, causing large information loss and sometimes introducing colorful artifacts.

Table 5: CIFAR-10 classification accuracy of transformed clean and adversarial examples on the attacked vanilla VGG-16 model.

Defense	Clean	FGSM-0.08	FGSM-0.04	BIM	DeepFool	CW
No Defense	0.9298	0.5816	0.6523	0.1803	0.1760	0.0936
MagNet[32]	**0.9035**	0.6145	0.6521	0.4312	0.6535	0.4845
PixelDefend[46]	0.8526	0.6810	0.7329	**0.7729**	0.7414	0.7579
STL	0.8285	0.7099	0.7487	0.7462	0.7854	0.7765
STL (cluster)	0.8360	**0.7103**	**0.7547**	0.7531	**0.7959**	**0.7906**

5.5. Trade-off Between Quality and Robustness

In transformation-based adversarial defenses, we typically aim to remove adversarial perturbations while preserving useful details. However, this is hard to achieve, as important details and adversarial perturbations are usually removed together. Thus, we examine the inherent trade-off between transformation quality and defensive robustness in our method.

In our method, the key parameter controlling the projection quality is the sparsity constraint weight λ: a larger λ causes more blurry results. We gradually in-

Table 6: ImageNet-10 classification accuracy of transformed clean and adversarial examples on an attacked vanilla VGG-16 model at resolution 64 (Table 6.A) and 128 (Table 6.B).

Table 6.A Resolution 64.

Defense	Clean	FGSM-0.08	FGSM-0.04	BIM	DeepFool	CW
No Defense	0.8665	0.2816	0.3080	0.1883	0.0811	0.0751
TVM[20]	**0.8172**	0.3403	0.4744	0.6595	0.6943	0.6823
Quilting[20]	0.6318	0.4541	0.5312	0.5696	0.5436	0.5563
Crop-Ens[20]	0.5590	0.4570	0.5328	0.5369	0.5429	0.5320
PD-Ens[40]	0.7946	0.3388	0.5526	0.6568	0.6919	0.6827
STL	0.7925	0.5472	0.6825	**0.7245**	0.7562	0.7414
STL (cluster)	0.8017	**0.5729**	**0.6914**	0.7234	**0.7652**	**0.7521**

Table 6.B Resolution 128.

Defense	Clean	FGSM-0.08	FGSM-0.04	BIM	DeepFool	CW
No Defense	0.8991	0.2123	0.2409	0.1790	0.0584	0.0504
TVM[20]	0.8591	0.2568	0.4386	0.6586	0.6360	0.6129
Quilting[20]	0.8149	0.3903	0.5889	0.6434	0.6242	0.5922
Crop-Ens[20]	0.7730	0.4622	0.6447	0.6876	0.7060	0.6888
PD-Ens[40]	**0.8789**	0.2333	0.4286	0.7221	**0.7359**	0.7272
STL	0.8654	0.4552	0.6418	**0.7332**	0.7308	0.7212
STL (cluster)	0.8759	**0.4733**	**0.6606**	0.7323	0.7301	**0.7432**

Table 7: Top-1 ImageNet classification accuracy of transformed clean and adversarial examples on an attacked vanilla ResNet-50 model.

Defense	Clean	FGSM-0.08	FGSM-0.04	BIM	DeepFool	CW
No Defense	0.7613	0.0862	0.1140	0.0131	0.0106	0.0201
TVM[20]	0.6205	0.3123	0.4256	0.4923	0.5232	0.5012
Quilting[20]	0.4168	0.3787	0.3865	0.3823	0.3859	0.3783
Crop-Ens[20]	0.6432	0.4623	0.5546	0.5965	0.6023	0.5980
PD-Ens[40]	0.6821	0.3846	0.5691	0.6089	0.6220	**0.6371**
STL	0.6728	0.5348	0.6032	0.6253	0.6233	0.6158
STL (cluster)	**0.6921**	**0.5588**	**0.6053**	**0.6348**	**0.6468**	0.6220

crease λ and explore this trade-off (Figure 5). We denote $\mathrm{Acc}(\boldsymbol{x})$ as the accuracy on the vanilla model of input $\boldsymbol{x}$. Higher $\mathrm{Acc}(T(\boldsymbol{x_0}))$ means higher transformation quality because the projected images still preserve useful information. Small $\|\mathrm{Acc}(T(\boldsymbol{x_0})) - \mathrm{Acc}(T(\boldsymbol{x}_{adv}))\|$ means the clean and adversarial examples are similar in feature space. The decision can be robust if both $\mathrm{Acc}(T(\boldsymbol{x_0}))$ and $\mathrm{Acc}(T(\boldsymbol{x}_{adv}))$ are high. In Figure 5, we see that as λ increases, $\mathrm{Acc}(T(\boldsymbol{x_0}))$ decreases and the gap between $\mathrm{Acc}(T(\boldsymbol{x_0}))$ and $\mathrm{Acc}(T(\boldsymbol{x}_{adv}))$ shrinks.

We additionally propose a metric to measure this trade-off. Specifically, we use PSNR between $T(\boldsymbol{x}_{adv})$ and $\boldsymbol{x_0}$ to measure reconstruction quality. For each method in comparison, let $a_0 = \mathrm{Acc}(T(\boldsymbol{x_0}))$ and $a_{adv} = \mathrm{Acc}(T(\boldsymbol{x}_{adv}))$, then we associate it with a characteristic interval $[\min(a_0, a_{adv}), \max(a_0, a_{adv})]$ to represent its overall prediction quality. Apparaently, a strong method should have an interval that is short (good robustness) and high (good accuracy). We plot a 2D PSNR vs. prediction quality map, where the top right corner indicates highest robustness and prediction quality. In Figure 6, we show comparison results of occupied regions on this map. Our method achieves both the highest PSNR and most preferable characteristic interval, demonstrating its superior ability to achieve robustness, accuracy, and maintain image quality.

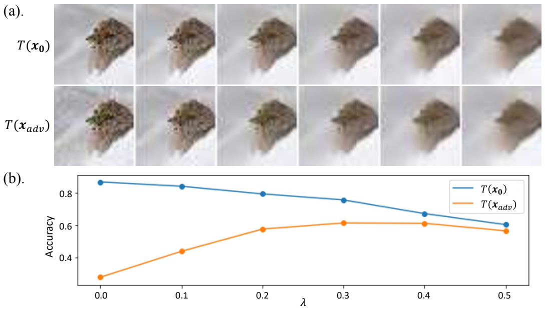

Figure 5: Intrinsic tradeoff between image reconstruction quality and defensive robustness. (a). Transformation results of each corresponding λ. (b). Accuracy of $T(\boldsymbol{x}_{adv}$ and $T(\boldsymbol{x_0})$ on attacked vanilla model. (Setting: FGSM-0.08, ImageNet-10, VGG-16, resolution 64).

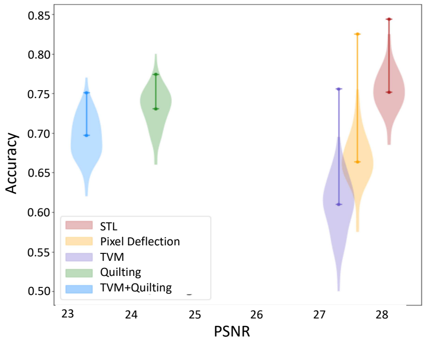

Figure 6: The PSNR, $\mathrm{Acc}(T(\boldsymbol{x}_{adv}))$ and $\mathrm{Acc}(T(\boldsymbol{x_0}))$ of different methods (Setting: FGSM-0.08, ImageNet-10, VGG-16, resolution 64). For both axes, the higher number the better. And less difference between $\mathrm{Acc}(T(\boldsymbol{x}_{adv}))$ and $\mathrm{Acc}(T(\boldsymbol{x_0}))$ means higher robustness.

6. Conclusion

We have proposed a novel state-of-the-art attack-agnostic adversarial defense method with additional increased robustness to input resolution, perturbation scale, and dataset scale. Inspired by convolutional sparse coding, we design a novel sparse transformation layer (STL) to project the inputs to a low-dimensional quasi-natural space, wherein a retrained classifier can make more reliable decisions. We evaluate the proposed method on CIFAR-10 and ImageNet and show that our defense mechanism provide state-of-the-art results. We have also provided an analysis of the trade-off between the projection image quality and defense robustness.

Acknowledgements We thank Bo Li for providing key discussions on white box attacks.

References

[1] A. Athalye, N. Carlini, and D. Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples, 2018. In ICML.

[2] Y. Bengio, L. Yao, G. Alain, and P. Vincent. Generalized denoising auto-encoders as generative models, 2013. In NIPS.

[3] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122,, 2010.

[4] H. Bristow, A. Eriksson, and S. Lucey. Fast convolutional sparse coding, 2013. In CVPR.

[5] J. Buckman, A. Roy, C. Raffel, and I. Goodfellow. Thermometer encoding: One hot way to resist adversarial examples. In *International Conference on Learning Representations(ICLR)*, 2018.

[6] X. Cao and N. Z. Gong. Mitigating evasion attacks to deep neural networks via region-based classication, 2017. In Proceedings of the 33rd Annual Computer Security Applications Conference. ACM.

[7] N. Carlini and D. A. Wagner. Towards evaluating the robustness of neural networks, 2017. 2017 IEEE Symposium on Security and Privacy (SP).

[8] R. Chalasani, J. C. Principe, and N. Ramakrishnan. A fast proximal method for convolutional sparse coding, 2013. In IJCNN.

[9] P.-Y. Chen, H. Zhang, Y. Sharma, J. Yi, and C.-J. Hsieh. Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models, 2017. arXiv preprint arXiv:1708.03999.

[10] S. S. Chen, D. L. Donoho, and M. A. Saunders. Atomic decomposition by basis pursuit. *SIAM J. Sci. Comput.*, 20(1):31–61, 1998.

[11] B. Choudhury, R. Swanson, F. Heide, G. Wetzstein, and W. Heidrich. Consensus convolutional sparse coding, 2017. In ICCV.

[12] N. Das, M. Shanbhogue, S.-T. Chen, F. Hohman, L. Chen, M. E. Kounavis, and D. H. Chau. Keeping the bad guys out: Protecting and vaccinating deep learning with jpeg compression, 2017. arXiv preprint arXiv:1705.02900.

[13] A. David and S. Vassilvitskii. k-means++: The advantages of careful seeding, 2007. Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms.

[14] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and F. fei Li. Imagenet: A large-scale hierarchical image database, 2009. In CVPR.

[15] G. S. Dhillon, K. Azizzadenesheli, J. D. Bernstein, J. Kossaifi, A. Khanna, Z. C. Lipton, and A. Anandkumar. Stochastic activation pruning for robust adversarial defense. In *International Conference on Learning Representations(ICLR)*, 2018.

[16] G. K. Dziugaite, Z. Ghahramani, and D. M. Roy. A study of the effect of jpg compression on adversarial images, 2016. CoRR, abs/1608.00853.

[17] C. Garcia-Cardona and B. Wohlberg. Convolutional dictionary learning: A comparative review and new algorithms, 2018. arXiv preprint arXiv:1709.02893.

[18] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples, 2014. CoRR, abs/1412.6572.

[19] S. Gu, W. Zuo, Q. Xie, D. Meng, X. Feng, and L. Zhang. Convolutional sparse coding for image super-resolution, 2015. In ICCV.

[20] C. Guo, M. Rana, M. Cisse, and L. van der Maaten. Countering adversarial images using input transformations, 2018. In ICLR.

[21] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition, 2017. In CVPR.

[22] F. Heide, W. Heidrich, and G. Wetzstein. Fast and exible convolutional sparse coding, 2015. In CVPR.

[23] A. Krizhevsky, V. Nair, and G. Hinton. Cifar-10 (canadian institute for advanced research).

[24] A. Kurakin, I. Goodfellow, and S. Bengio. Adversarial machine learning at scale., 2016. arXiv preprint arXiv:1611.01236.

[25] A. Kurakin, I. J. Goodfellow, and S. Bengio. Adversarial examples in the physical world. *CoRR*, abs/1607.02533, 2016.

[26] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition., 1998. Proceedings of the IEEE, 86(11):22782324.

[27] Y. Liu, X. Chen, C. Liu, and D. X. Song. Delving into transferable adversarial examples and black-box attacks. *CoRR*, abs/1611.02770, 2016.

[28] Y. Liu, X. Chen, R. K. Ward, and Z. J. Wang. Image fusion with convolutional sparse representation, 2016. IEEE Signal Process. Lett.

[29] S.-M. M.-Dezfooli, A. Shrivastava, and O. Tuzel. Divide, denoise, and defend against adversarial attacks, 2018. arXiv preprint, arXiv:1802.06806v1.

[30] X. Ma, B. Li, Y. Wang, S. M. Erfani, S. Wijewickrema, G. Schoenebeck, M. E. Houle, D. Song, and J. Bailey. Characterizing adversarial subspaces using local intrinsic dimensionality. In *International Conference on Learning Representations(ICLR)*, 2018.

[31] J. Mairal, F. Bach, and J. Ponce. Sparse modeling for image and vision processing. *Foundations and Trends in Computer Graphics and Vision*, 8(2-3):85–283, 2014.

[32] D. Meng and H. Chen. Magnet: A two-pronged defense against adversarial examples, 2017. In CCS.

[33] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard. Deepfool: A simple and accurate method to fool deep neural networks, 2016. In CVPR.

[34] T. Na, J. H. Ko, and S. Mukhopadhyay. Cascade adversarial machine learning regularized with a unified embedding, 2018. In ICLR.

[35] A. Nayebi and S. Ganguli. Biologically inspired protection of deep networks from adversarial attacks, 2017. arXiv preprint arXiv:1703.09202.

[36] M. Osadchy, J. Hernandez-Castro, S. Gibson, O. Dunkelman, and D. Perez-Cabo. No bot expects the deepcaptcha! introducing immutable adversarial examples with applications to captcha, 2017. In IEEE Transactions on Information Forensics and Security.

[37] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami. Practical black-box attacks against machine learning, 2017. In ACM Asia Conference on Computer and Communications Security,.

[38] N. Papernot, P. McDaniel, A. Sinha, and M. Wellman. Towards the science of security and privacy in machine learning, 2016. arXiv preprint arXiv:1611.03814.

[39] N. Papernot, P. D. McDaniel, X. Wu, S. Jha, and A. Swami. Distillation as a defense to adversarial perturbations against deep neural networks, 2016. 2016 IEEE Symposium on Security and Privacy (SP).

[40] A. Prakash, N. Moran, S. Garber, A. DiLillo, and J. Storer. Deecting adversarial attacks with pixel deection, 2018. In CVPR 2018.

[41] T. M. Quan and W.-K. Jeong. Compressed sensing reconstruction of dynamic contrast enhanced mri using gpu-accelerated convolutional sparse coding, 2016. In ISBI.

[42] T. Salimans, A. Karpathy, X. Chen, and D. P. Kingma. Pixelcnn++: Improving the pixelcnn with discretized logistic mixture likelihood and other modifications, 2017. arXiv:1701.05517.

[43] P. Samangouei, M. Kabkab, and R. Chellappa. Defense-gan: Protecting classifiers against adversarial attacks using generative models, 2018. In ICLR.

[44] A. Serrano, F. Heide, D. Gutierrez, G. Wetzstein, and B. Masia. Convolutional sparse coding for high dynamic range imaging. *Computer Graphics Forum*, 35(2):153–163, 2016.

[45] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition, 2014. arXiv preprint arXiv:1409.1556.

[46] Y. Song, T. Kim, S. Nowozin, S. Ermon, and N. Kushman. Pixeldefend: Leveraging generative models to understand and defend against adversarial examples, 2018. In ICLR.

[47] J. Su, D. V. Vargas, and K. Sakurai. One pixel attack for fooling deep neural networks. *CoRR*, abs/1710.08864, 2017.

[48] M. Sung, H. Su, R. Yu, and L. Guibas. Deep functional dictionaries: Learning consistent semantic structures on 3d models from functions. *arXiv preprint arXiv:1805.09957*, 2018.

[49] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. J. Goodfellow, and R. Fergus. Intriguing properties of neural networks, 2014. In ICLR.

[50] I. Tosic and P. Frossard. Dictionary learning. *IEEE Signal Process. Mag.*, 28(2):27–38, 2011.

[51] F. Tramr, A. Kurakin, N. P. abd D. Boneh, and P. D. McDaniel. Ensemble adversarial training: Attacks and defenses., 2018. In ICLR.

[52] B. Wohlberg. Efficient convolutional sparse coding. *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7173–7177, 2014.

[53] B. Wohlberg. Efcient convolutional sparse coding, 2014. In ICASSP.

[54] C. Xie, J. Wang, Z. Zhang, Z. Ren, and A. Yuille. Mitigating adversarial effects through randomization. In *International Conference on Learning Representations*, 2018.

[55] W. Xu, D. Evans, and Y. Qi. Feature squeezing: Detecting adversarial examples in deep neural networks, 2018. In Network and Distributed Systems Security Symposium (NDSS).

[56] H. Zhang and V. Patel. Convolutional sparse coding-based image decomposition, 2016. In BMVC.

[57] H. Zhang and V. M. Patel. Convolutional sparse and low-rank coding-based rain streak removal, 2017. In WACV.