SymDNN: Simple & Effective Adversarial Robustness for Embedded Systems

Swarnava Dey   Pallab Dasgupta   Partha P Chakrabarti
Indian Institute of Technology Kharagpur, Kharagpur, India - 721302
swarnava.dey@kgpian.iitkgp.ac.in, {pallab, ppchak}@cse.iitkgp.ac.in

Abstract

We propose SymDNN, a Deep Neural Network (DNN) inference scheme, to segment an input image into small patches, replace those patches with representative symbols, and use the reconstructed image for CNN inference. This approach of deconstruction of images, and the reconstruction from cluster centroids trained on clean images, enhances robustness against adversarial attacks. The input transform used in SymDNN is learned from very large datasets, making it difficult to approximate for adaptive adversarial attacks. For example, SymDNN achieves 23% and 42% robust accuracy at L_∞ attack strengths of $8/255$ and $4/255$ respectively, against BPDA under a complete white box setting, where most input processing based defenses break completely. SymDNN is not a future-proof adversarial defense that can defend any *attack, but it is one of the few readily usable defenses in resource-limited embedded systems that defends against a wide range of attacks. Our code is available at:* https://github.com/swadeykgp/SymDNN.

1. Introduction

Convolutional Neural Networks (CNNs) can automatically learn effective features from images, making those suitable in many computer vision tasks as classifiers and backbone feature extractors. Since the past few years, *Edge Computing* is showing explosive growth [51], with mobile and embedded computer vision being one of the *killer apps* [2], and CNNs dominating that landscape [13].

Albeit being one of the most popular architectures for image tasks, a CNN inference can be forced to generate unexpected output on images that contain visually imperceptible, *well crafted* modifications [8], referred to as adversarial perturbations [57]. Brittleness of CNNs against adversarial attacks make those unsuitable for deployment in safety critical systems. With adversarial attacks very much possible in a real world setting [32], the mobile and embedded vision tasks on resource-limited systems are also affected by this.

Despite significant progress in the research on adversarial robustness, *there are hardly any studies targeting robustness under adversarial attacks on embedded systems*.

Brief Background of Adversarial Robustness. Adversarial attacks aim to perturb a benign input with small changes to create a malicious input, such that the CNN output differs by a significant extent. For instance, in a successful *targeted* attack on the input to a CNN based classifier, the classifier is forced to assign a class that is desired by the adversary. Whereas in a successful *untargeted* attack, a CNN classifier fails to assign the same class to the benign example and the visually similar adversarial example. The difference between a clean and the corresponding adversarial example is often quantified using l_p norms ($p \in \{1, 2, \infty\}$).

Adversarial examples can be generated in a *complete white box* setting, where the model parameters (e.g., architecture, loss function etc.), and the defense parameters (e.g., transform, randomness) are known. Effective adversarial inputs [43] can also be generated in a *complete black box* setting, where the adversary has no access to the model and defense. Defenses against the adversarial attacks often claim robustness in the above two and several different intermediate *threat models*.

In summary, some of the very strong attacks include enhancements of PGD [37] attack (e.g., APGD [19], EOTPGD [5]), translation invariant versions of FGSM [26] attack (e.g., TI-FGSM [23], MI-FGSM [23]), ensemble attacks (e.g., AutoAttacks [19]), and finally customized attacks adapted [58] to each defense.

We refer the reader to a recent survey [48] and a series of works [4,8,9,11,15,19,23,26,37,43,44,57,58] that have shaped the field of adversarial robustness during the past eight years.

The adversarial defenses render a CNN inference robust to the adversarial attacks. It is believed that the process of training a DNN, that tries to generalize on an unconstrained, real-valued input space, based on a finite training set example, leads to imperfect generalization [7, 26, 44].

Adversarial Defenses for Embedded Systems? To address the problem of adversarial attacks, the strongest defenses, that is, the ones that are *provably* robust [15, 16, 40, 56, 63], attempts to identify convex regions that include the non-convex manifolds where the adversarial examples belong to and includes those as a part of the loss function to train a CNN. However, **the scalability of these methods are not yet proven** on larger networks.

Adversarial training [26, 37] is a type of defense that precisely defines the perturbation capabilities of an adversary and harnesses the adversarial examples generated by that adversary into the empirical risk minimization problem [60]. This approach is considered reliable under stringent evaluations [4, 58]. However, this approach also has some limitations e.g., failure to scale to very large datasets [33], failure to resist examples under a different dissimilarity measure [54]. We highlight a different problem associated with this approach. In [37] authors observe that the adversarially trained models use a robust decision boundary to classify adversarial examples, which, in turn, needs a relatively larger model capacity than standard models. We find that adversarially trained models on the CIFAR-10 dataset, available from RobustBench [17], are very large in size, compared to a standard model. For instance, **model sizes for [47] and [53] are** 705**Mb and** 291**Mb** respectively, whereas a ResNet-18 architecture trained on CIFAR-10 in a non-adversarial manner, has a size of 1Mb. It is not feasible to deploy such large footprint models in resource-limited devices. The model sizes remain huge even after pruning [53].

There is another line of defenses that post-processes [10, 24, 27, 39, 42, 55, 61, 67] a tampered image before CNN inference. These defenses are often simple, and require lightweight processing, unless an Autoencoder or another Deep Neural Network is used for the *purification* process. Unfortunately, most of these works are shown to defend only against the gradient based attacks, and **fail completely** against attacks that **bypass *obfuscated gradients*** [4], in a *complete white box* setting.

Based on our study of the current state-of-the-art in adversarial defense strategies, we can conclude that the robust and reliable defenses are either not scalable or have large model footprints, making those unsuitable for embedded vision applications. Some input processing based defenses are computationally efficient, however most of these are broken by recent adaptive attacks [4, 58].

Proposed Approach. In this context we propose SymDNN, yet another input processing based defense that enhances adversarial robustness of a CNN for small attack strengths, and under various threat models. The broad working principle of SymDNN is shown in Fig. 1.

- In the offline training phase we take all the images from the training set, divide them into small patches, accumulate these patches into a single patch dataset, and apply unsupervised clustering on this dataset.
- We associate symbols with the clusters and store these associations in a codebook.
- The image is coded using the codebook by replacing each patch with a symbol. The symbolic coded image is orders of magnitude smaller than the original image, which may be useful for communication and / or storage.
- For CNN inference, an approximation of the image is reconstructed by replacing each symbol with the centroid of the cluster it represents. The reconstructed image is then used as input by CNN.
- When the image is adversarially perturbed, this reconstruction process removes some of the adversarial changes, as the centroids that are used to replace the image patches are learned from the clean images.

In practice, we implement SymDNN training and index search with a fast similarity search [29], to handle large patch datasets. The theoretical and algorithmic basis for this approach has been detailed in this paper.

SymDNN is a model agnostic pre-processing step that can boost the accuracy of any arbitrary CNN for adversarial images with a limited change per pixel. We observe this gain in robust accuracy under complete black box and partial white box threat models, where the model parameters and dataset are known. We have extensively evaluated SymDNN under these threat models and we show that SymDNN boosts the robust accuracy of a ResNet model in the face of several recent strong attacks, namely enhanced PGD variants [5,19,37], translation invariant attacks [22,23] and ensemble attacks [19], by 30-50% at attack strength of $4/255$ and by at least 10% at attack strength of $8/255$.

In a complete white box setting, where the transformation due to the defense is also known to the adversary, SymDNN exhibits the gradient shattering property [4]. We believe that the transformations used in earlier input processing based defenses [10, 27, 67] were based on some analytical formulation, which made it possible for the adaptive attacks to easily use a surrogate approximation to perform gradient descent and find the adversarial examples even if the gradients were obfuscated. For SymDNN, the computational overhead of building this transformation function from large image datasets, is prohibitive even using a fast indexing and clustering library. For the same reason, **approximating such a function is difficult**, even if gradients are obfuscated [4]. To support this belief, SymDNN accuracy drops to 23%, far better than other defenses that drop to 0% [4, 58] under Backward Pass Differentiable Approximation (BPDA) attack.

We do not claim that SymDNN can defend any adversarial attack, which is what the adversarial research community aims at. The major contributions of SymDNN are as follows: (A) Forming a first line of readily usable defense against a wide range of adversarial attacks with very little computational overhead that embedded vision systems currently lack, and (B) Significant reduction in the size of the images in their symbolic forms without any significant negative impact on their classification post reconstruction.

2. Related Work

The method proposed in this paper is in principle similar to all input processing based defenses, which *cleans* or *purifies* an image before DNN inference.

Previous works [6, 14, 27, 35, 36, 39, 67] have used specialized processing on the input or features to nullify the effects of adversarial perturbations. In [27] total variance minimization is used to reconstruct pixels of an image based on its neighboring pixels, with an aim of smoothing out small and localized perturbations. This work also uses image-quilting, where an image is reconstructed using patches from an image patch dataset. This is very similar to SymDNN. However, neither this paper [27], nor the original paper on patch based texture synthesis [25], elaborate on the patch dataset, resolution of the patches and the computation overhead for texture synthesis. For SymDNN, this patch *K-nearest neighbor graph* is built from the full training dataset, rendering the image reconstruction process hard to approximate.

Another approach is using discretization or quantization of attacked image pixels [10, 62, 69]. In [62, 69] K-means clustering is used to quantize each pixel of an attacked image into 2-5 levels to reduce perturbation and achieve robustness against simple untargeted attacks. However, there are no details on how such a clustering model is trained for quantizing image pixels. In [10] a specialized *thermometer* encoding is proposed that preserves image properties, inference accuracy and enhances adversarial robustness for a limited set of adversarial attacks. However, most of these input processing based defenses are now considered ineffective, as these methods obfuscate gradients and therefore are often only effective against gradient based attacks [4, 58]. These methods are ineffective in the face of strong black-box attacks that are not gradient dependent and against adaptive attacks that can easily approximate the simple transformation function to perform gradient descent.

In [45] it is shown that a series of transforms can prevent adaptive attacks. Another recent input processing defense [42] using an autoencoder based purifier, shows gains in adversarial robustness under various threat models except complete white box. However, a series of transforms or autoencoders are computationally demanding.

SymDNN addresses these problems of adversarial robustness under different threat models, with a single, computationally efficient transformation function that is hard to approximate.

3. The Goal of Symbolic Abstraction

Our goal is to create a symbolic abstraction of an image which is compact, inference preserving, and robust to adversarial attacks. In formal terms, given an input $x \in \mathbb{R}^{I_h \times I_w \times C}$ from an input space $\mathcal{X}$ and a set of labels $y_1, y_2, \ldots, y_k$ from an output space $\mathcal{Y}$, a DNN learns a non-linear mapping $\mathcal{N}_\theta : \mathcal{X} \rightarrow \mathcal{Y}$, where $\boldsymbol{\theta}$ denotes the network parameters, and I_h, I_w, and C are respectively the height (in pixels), width (in pixels), and number of channels of the image, x. We aim to learn a mapping, $\tau : \mathcal{X} \rightarrow \Sigma^*$, where Σ^* is a symbolic abstraction. We also define the mapping: $\tau_R : \Sigma^* \rightarrow \mathcal{X}$, which reconstructs an image from the symbolic abstraction. We address the following requirements:

1. *Compaction.* We show that $\tau(x)$ is orders of magnitude smaller than x. This is useful when the image has to be communicated, say, from an edge device to an edge server.

2. *Inference Preserving.* Our experiments show that the abstraction is almost always inference preserving, that is, $\mathcal{N}_\theta(x) = \mathcal{N}_\theta(\tau_R \tau(x))$.

3. *Enhancing Robustness to Adversarial Attacks.* If an adversarial attack modifies x to x', we show that $\mathcal{N}_\theta(x) = \mathcal{N}_\theta(\tau_R \tau(x'))$ in most cases, even when the attack succeeds on the original image, that is, $\mathcal{N}_\theta(x) \neq \mathcal{N}_\theta(x')$. We propose a model agnostic scheme to defend a DNN inference against adversarial attacks.

In cases where enhancing robustness is the primary goal, we do not need to learn τ and τ_R separately. Instead, it suffices to learn the combined function, $\tau_R \tau : \mathcal{X} \rightarrow \mathcal{X}$, and pass on $\tau_R \tau(x')$ instead of x' to the DNN classifier.

4. The SymDNN Methodology

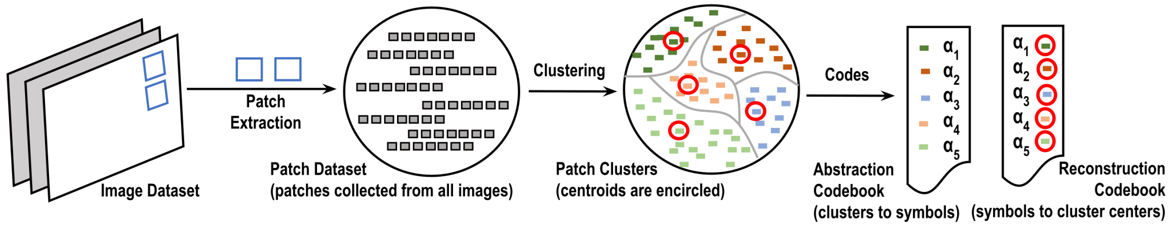
The SymDNN methodology has two broad stages. The first stage learns an alphabet Σ from the training dataset, which forms the basis for defining the mapping $\tau : X \rightarrow \Sigma^*$. In the second stage, the symbolic abstraction is used for new images. The workflow of SymDNN is illustrated in Figure 1.

4.1. Symbolic Abstraction Design

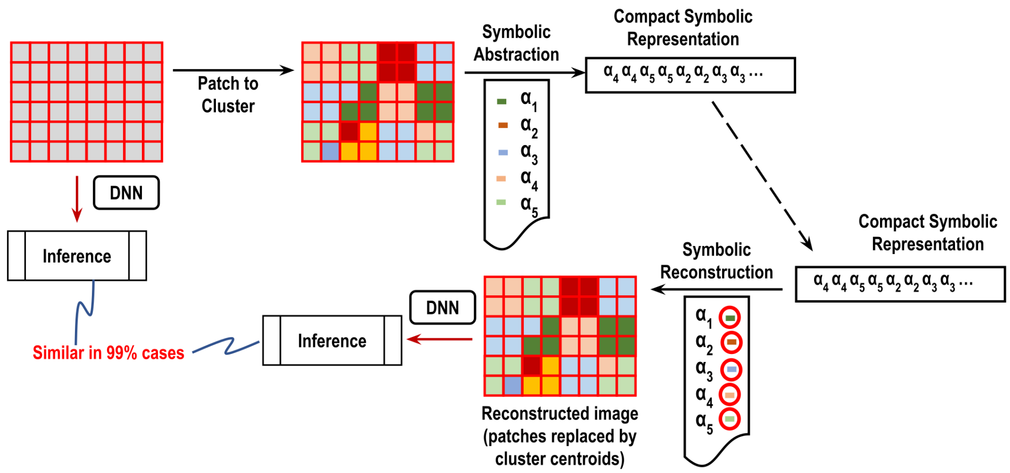
The proposed method for symbolic abstraction design works on a given training set, $\mathcal{D}$, of images. The steps are as follows:

1. We choose a *patch dimension*, P (a parameter of our algorithm). For the chosen, P, we extract patches of dimension $P \times P$ pixels from all the images in $\mathcal{D}$ and populate them into a *patch dataset*, $\mathcal{D}_{patch}$.

2. We use a similarity based clustering algorithm to partition $\mathcal{D}_{patch}$ into disjoint clusters $\{C_1, C_2, \ldots, C_M\}$,



(a) Learning the abstraction & reconstruction codebooks by applying unsupervised clustering on all the patches from a given training set



(b) Inference preserving transformation between input image and compact, symbolically abstracted image for deployment

Figure 1. The two stages of SymDNN methodology: (a) learning of symbols & mapping (b) using the symbolic abstraction for new images.

such that $\mathcal{D}_{patch} = \bigcup_{i=1}^{M} C_i$. We associate a symbol, α_i, with each C_i. We define the alphabet $\Sigma = \{\alpha_1, \ldots, \alpha_M\}$.

3. We identify the cluster centroid, μ_i, for each partition, C_i. Note that μ_i is a $P \times P$ image patch. Let $\mu = \{\mu_1, \ldots, \mu_M\}$ be the set of cluster centroids.

4. We prepare a Codebook containing two mappings, namely:

   - *Patch to symbol.* $\eta : \mathcal{D}_{patch} \rightarrow \Sigma$, such that for each patch $p \in \mathcal{D}_{patch}$, we have $\eta(p) = \alpha_i$ iff $p \in C_i$.
   - *Symbol to patch.* $\eta_R : \Sigma \rightarrow \mu$, such that for each symbol, $\alpha_i \in \Sigma$, we have $\eta_R(\alpha_i) = \mu_i$.

   It may be noted that $\eta_R \eta(p)$ maps the patch p to the centroid of the cluster containing p.

4.1.1 Clustering the Patches

Lloyd's algorithm, also known as, *k-means* clustering can be used to achieve the above partitioning. The basic steps of the above algorithm are as follows:

1. For each cluster, C_j the initial cluster centroid, μ_j is selected. Selected centroids generate the initial partitioning $\mathcal{C}^0$. It may be noted that μ_j is a $P \times P$ image patch.

2. The Euclidean distance of each patch is calculated from all the centroids, that is, $\|p_i - \mu_j\|^2$.

3. Based on the distance computed, the cluster assignment for patch p_i is obtained by finding the nearest centroid, that is, $\underset{\mu_1, \mu_2, \ldots, \mu_M}{\arg\min} \|\mu_j - p_i\|^2$

4. For each partition, a new centroid is obtained by computing the average of all the patches assigned to that partition, that is, $\mu_j = \frac{1}{|\mu_j|} \sum_{p_i \in \mu_j} p_i$. This generates the updated set of partitions $\mathcal{C}^1$.

5. Step 2 to 4 are repeated until a fixed point is reached, that is, $\mathcal{C}^m = \mathcal{C}^{m+1}$.

$$\min_{(C_1 \cup C_2 \cup \ldots \cup C_M = \mathcal{C})} \sum_{j=1}^{M} \sum_{p_i \in \mu_j} \left\| p_i - \frac{1}{|\mu_j|} \sum_{p_i \in \mu_j} p_i \right\|_2^2 \quad (1)$$

The *k-means* objective, stated in Eq. (1) is NP-hard. The iterative approach (Lloyd's algorithm), with suitable initialization runs in polynomial time to converge to a local minima. However, the size of the input space which is in the range of billions, makes the algorithm computationally expensive. Specifically, step 2 of the above iterative algorithm has a time complexity of $\mathcal{O}(MNP^2)$.

The FAISS [29] *k-means* implementation accelerates this step by a large factor, which helps us to experiment with different numbers of clusters, patch sizes and handle around *38.5 million* patches for the CIFAR-10 dataset and *5.6 billion* for ImageNet.

4.2. Inferencing using Symbolic Abstraction

We consider two use cases for inferencing using the proposed symbolic abstraction. In the first case, we study the gain in adversarial robustness by virtue of symbolic abstraction, and reconstruction. In the second case, we study the compaction of the image by virtue of symbolic abstraction, which may be useful for communication and / or storage, and study the loss of inferencing accuracy post reconstruction. This section outlines these flows.

4.2.1 Resistance Against Adversarial Attacks

SymDNN follows the same defense model as other input processing based defenses: given a pre-trained classifier $\mathcal{N}_\theta(\cdot)$, the preprocessor $\tau_R\tau(\cdot)$ is almost always inference preserving, that is, $\mathcal{N}_\theta(x) \approx \mathcal{N}_\theta(\tau_R\tau(x))$, and the symbolic reconstruction removes the adversarial perturbation.

The seminal work on *gradient obfuscation* [4, 58] and gradient masking [43] have highlighted that in a complete white box setting, these types of non-differentiable defenses cannot be backpropagated through to generate adversarial examples. In [4], BPDA attack creates a differentiable approximation of these transformation functions and uses that to backpropagate and generate effective adversarial examples.

SymDNN, being a non-differentiable defense, also exhibits the same behavior. However, the success of BPDA or other adaptive attacks depend on how easily and precisely the transformation function can be approximated. Compared to the transformations used in earlier input processing based defenses [10, 27, 67], which are based on some analytical formulation, SymDNN's approximation function $\tau_R\tau(\cdot)$ is very difficult to approximate. It is learned from a large image dataset, and the computational overhead of building this transformation function is prohibitive even using a fast indexing and clustering library.

In [4], defenses that employ randomized transformations to the input are attacked using Expectation over Transformation (EOT) [5]. Samples of transforms used by a defense is used by EOT to iteratively approximate an expected transform for generating adversarial examples. SymDNN also uses a test time randomness, where instead of replacing patch p_{i_j} in image x with $\eta(p_{i_j})$, we replace it with a randomly chosen symbol from among the k nearest centroids. We name this mechanism ***MSR**: Multi-Sym Randomized* inference, and evaluate the Top-1 and Top-5 accuracy of our models based on it. In this paper we use *25* most similar symbols ($k = 25$) for a given patch from the k-nearest neighbor graph of centroid patches to achieve the best clean vs. robust accuracy trade-off. To optimize over the k-nearest centroid patches for a given centroid patch, and generate an expected transform is as computationally demanding as learning the k-NN graph itself. This makes the test time randomness of SymDNN difficult to break by an adversary using EOT attack variants, within practically feasible computation capability and time.

Thus, the *key to withstand the adaptive attacks in [4, 58], is to design the transformation as hard as possible*, when proposing an input processing based defense. For deploying in embedded systems, such transformation needs to be computationally efficient as well. Similar results, where BPDA is less effective can be observed in cases where the transform is difficult to approximate [45, 50]. In [45] a series of transforms are used, and in [50] a projection into a Generative Adversarial Network manifold is performed to achieve the robustness. In contrast, SymDNN executes a single, computationally efficient transform to achieve similar robustness.

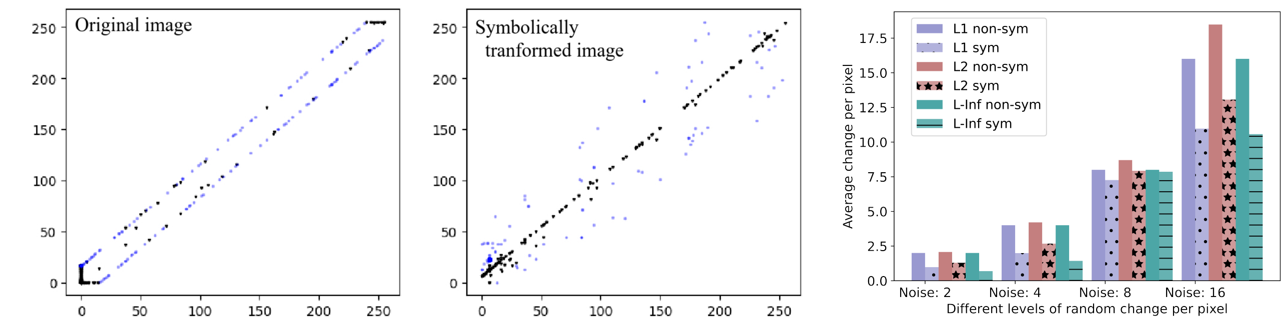
SymDNN's robustness is not only limited to complete white-box setting, against gradient based attacks. The symbolic reconstruction process replaces a bunch of pixels of patch resolution, with a clean centroid patch from the clustering model. This operation reinstates the values of most of the pixels as shown in Fig. 2a. Although it throws off some of the values randomly, our experiments show that the inference accuracy is preserved in most cases. As shown in Fig. 2b, this reduction of change per pixel happens at different noise levels, although it is most effective at lower attack strengths.

4.2.2 Compaction and Reconstruction

We define the following mappings for compaction and reconstruction of an image:

- *Symbolic Abstraction.* The given image $x \in \mathbb{R}^{I_h \times I_w \times C}$ is partitioned into patches of size $P \times P$ pixels (with suitable zero padding if the dimensions are not multiples of P), and arranged into a sequence π of patches, $p_{i_1}, \ldots, p_{i_n}$. We define $\tau(x) = s$, where $s \in \Sigma^*$ is the string $\alpha_{i_1}, \ldots, \alpha_{i_n}$, such that $\eta(p_{i_j}) = \alpha_{i_j}$. We shall refer to s as the *code* for x.

- *Symbolic Reconstruction.* Given the code, s, for the image x, we define $\tau_R(s) = x'$, where the image x' is obtained by replacing each $\alpha_i \in s$ by $\eta_R(\alpha_i)$ and rearranging the patches in the same order as in the original image, x. In other words, in the reconstructed image, each patch



(a) Reinstating of pixel values by symbolic transform (x & y axes: pixel values 0-255) (b) Resistance at different change magnitudes

Figure 2. Mapping back of pixel changes & Resistance to perturbations due to symbolic abstraction on images

p_{i_j} in x is replaced by $\eta_R \eta(p_{i_j})$, namely the centroid of the cluster that contains p_{i_j}.

The reduction due to abstraction, Δ^a, can be expressed as follows:

$$\mathbf{\Delta^a} = \left(1 - \frac{\log M}{P^2 \times B}\right) \times 100\% \quad (2)$$

where B denotes the bits per pixel representation of an image and M is the number of clusters. For storage / communication entropy encoding e.g., Huffman encoding can be used for reducing the amortized number of bits per pixel.

5. Experimental Evaluation

5.1. Experimental Setup

Architectures & Datasets. For ImageNet [49] dataset we use publicly available pre-trained models: WideResNet [68], ResNet-152 [28] and ResNeXt-101 [66] from PyTorch model repository[1]. We use ResNet-20 architecture from [70] for CIFAR-10 [31], and LeNet-5 [21] for MNIST [34] dataset.

Adversarial Robustness. We choose Torchattacks [30] for attack implementations. For demonstrating efficacy of SymDNN on known robust models, we use certified CIFAR-10 models from RobustBench [17]. We use FoolBox [46] to perform adversarial attacks on ImageNet pre-trained models. We specify the threat models, attack iterations and attack strengths corresponding to each result in the result section.

5.2. Adversarial Robustness

Tab. 1 shows the robust accuracy gains of SymDNN under all possible attack models described in Sec. 1 and Sec. 4.2.1, compared to the standard inference. SymDNN performs best under the white-box model, black-box defense attack model (columns 3-4 of Tab. 1), where the adversary has no knowledge about the defense being used.

[1] https://pytorch.org/vision/stable/models.html

The accuracy gain in the complete white-box setting (last column of Tab. 1) is possibly due to obfuscated gradients [4, 58], except for the BPDA attack (top 3 rows of Tab. 1). However SymDNN's transform is difficult to approximate, and hence it does not break completely against BPDA. SymDNN is only effective to complete black-box attacks under lower attack strengths. This is currently a limitation, although this is the best we have in an embedded resource limited setting. We also hypothesize that a more strongly separable clustering may enable SymDNN to resist attacks with larger values of ϵ. We discuss this aspect further in Sec. 5.4.

Tab. 2 presents a comparison of SymDNN with NRP [42] and DefenseGAN [50], two state-of-the-art input defenses. SymDNN provides a better defense than NRP in 12 out of the 15 attacks. Specifically, NRP's defense fails completely against a recent attack (Jitter [52]). DefenseGAN [50] performance is highly dependent on the generator training. We trained a generator with $10k$ iterations, which did not perform well against the wide range of attacks we used.

NRP uses a DNN of 16.5 million parameters for the *purification* step. DefenseGAN uses a resource intensive Generative Adversarial Network generator to generate cleaned images. In the case of the adversarially trained model [47] (last column of Tab. 2), the model size is 705Mb. Compared to these methods, SymDNN is suitable for resource-limited embedded systems. It has very low computation overhead. SymDNN, with 2×2 patches, takes 0.5 milliseconds to lookup clusters & encode/decode images, with a peak memory load of 44 Mb.

For MNIST, we observe a minimum of 93% and 68% robust SymDNN accuracy at ($\epsilon = 16/255$) and ($\epsilon = 32/255$) respectively, for the 7 attack we tried.

For ImageNet, preliminary experiments using FoolBox [46] show approximately 60% boost in robust accuracy for C&W [12], PGD [38], and FGSM [26]. Detailed results and visualizations for MNIST & ImageNet are presented in the supplementary material.

Table 1. SymDNN accuracy(%) under different attacks: Abbreviations used:"M" - Model, "D" - Defense, "W" - White-box, and "B" - Black-box. The attack models are expressed as combinations of these. SymDNN performs best under the white-box model / black-box defense combination. Different row colors denote different attack strengths (col 2). '^{s}' indicates SymDNN accuracy with 2048 clusters. The case of gradient obfuscating [4,58] happens under the complete white-box attack model(last column). SymDNN's input transform is very difficult to approximate, even if gradients are shattered. This is evident in the BPDA [4] attack, against which most input defenses break completely.

Attacks	**Strength**	**M:W, D:B Acc.**	**M:W, D:B Acc.^{s}**	**M:B, D:B Acc.**	**M:B, D:B Acc.^{s}**	**M:W, D:W Acc.^{s}**
BPDA [4]	$\epsilon = 8/255$	**8**	**48**	32	45	23
TI-FGSM [23]	$\epsilon = 8/255$	**11**	**28**	83	81	45
AutoAttack [20]	$\epsilon = 8/255$	**0**	**10**	32	58	21
DI-FGSM [65]	$\epsilon = 8/255$	**0**	**8**	32	51	40
MI-FGSM [65]	$\epsilon = 8/255$	**0**	**12**	36	55	44
RFGSM [59]	$\epsilon = 8/255$	**0**	**10**	33	55	46
EOTPGD [71]	$\epsilon = 8/255$	**0**	**10**	35	57	51
APGD(CE) [20]	$\epsilon = 8/255$	**1**	**15**	32	57	30
APGD (DLR) [20]	$\epsilon = 8/255$	**0**	**34**	64	75	30
APGDT [20]	$\epsilon = 8/255$	**0**	**33**	63	75	28
Jitter [59]	$\epsilon = 8/255$	**0**	**30**	59	71	51
BPDA [4]	$\epsilon = 4/255$	**8**	**49**	61	71	42
TI-FGSM [23]	$\epsilon = 4/255$	**50**	**58**	72	69	63
AutoAttack [20]	$\epsilon = 4/255$	**0**	**37**	8	18	26
DI-FGSM [65]	$\epsilon = 4/255$	**0**	**39**	1	11	58
MI-FGSM [65]	$\epsilon = 4/255$	**0**	**38**	5	18	63
RFGSM [59]	$\epsilon = 4/255$	**0**	**40**	4	18	62
EOTPGD [71]	$\epsilon = 4/255$	**0**	**41**	5	16	63
APGD(CE) [20]	$\epsilon = 4/255$	**1**	**51**	10	21	38
APGD (DLR) [20]	$\epsilon = 4/255$	**0**	**34**	35	52	30
APGDT [20]	$\epsilon = 4/255$	**0**	**33**	30	51	28
Jitter [59]	$\epsilon = 4/255$	**0**	**30**	29	47	51

5.3. Compaction and Clean Symbolic Accuracy

As summarized in Tab. 3, our proposed SymDNN has less than 0.4% accuracy drop in Top-1 and Top-5 accuracy metrics, for different pre-trained models on 50,000 ImageNet testset. The Top-5 metric, popularized in ILSVRC [49], can be useful for building ensemble models [64].

The data reduction can be calculated directly from Eq. (2). For instance, with a patch size of 2×2, considering 8 bits per pixel representation and 2048 symbols, SymDNN achieves around 68% compaction on the ImageNet test set. On the other hand, using 512 symbols brings down the compaction to 74%, with 1% drop in accuracy. Using more than 2048 symbols does not seem to be profitable, as it increases the computation load for cluster index training and patch extraction, with negligible benefits in terms of accuracy.

Table 2. SymDNN accuracy(%) comparison with state-of-the-art: tested on 2000 randomly selected images from CIFAR-10 testset. '^{s}' indicates SymDNN inference; codebook size: 2048. Attack magnitudes:- $\epsilon = 8/255$, Threat model: whitebox model & black-box defense, iterations: 100. NRP [42] and **DefenseGAN (GD)** [50] are input defenses. Rice2020Overfitting [47] is an **Adversarially Trained (AT)** model. In comparison, SymDNN has much less compute & memory overhead.

Attacks	Acc.	**Acc.^{s}**	**NRP** [42]	**DG** [50]	**AT** [47]
BPDA [4]	0	23	5	16	60
C & W [12]	0	67	51	0	20
FAB [18]	0	83	84	26	52
Square [3]	11	59	78	30	60
DeepFool [41]	3	82	81	20	0
TI-FGSM [23]	12	20	63	25	63
AutoAttack [20]	0	49	19	34	50
DI-FGSM [65]	16	52	40	22	58
MI-FGSM [22]	2	50	27	23	56
RFGSM [59]	3	50	30	54	56
APGDT [20]	0	67	50	30	50
APGD(CE) [20]	0	48	20	24	56
APGD (DLR) [20]	0	62	46	26	52
EOTPGD [71]	4	52	29	24	57
Jitter [52]	6	59	0	21	53

Table 3. ImageNet SymDNN accuracy (%) & compaction (%) : Top-1 & Top-5 accuracies reported on full ImageNet testset. '^{s}' indicates SymDNN inference that uses a 2048 Symbols (# Syms). For MSR (defined in Sec. 4.2.1), Top-5 accuracy is reported here.

# Syms	Model	Top-1	**Top-1^s**	Top-5	**Top-5^s**	**MSR^s**	$\Delta^a \downarrow$
2048	WideResnet	71.95	71.73	89.54	89.21	88.66	68
	Resnet152	71.2	71.07	89.2	89.08	88.78	
	ResNext	72.79	72.64	90.10	89.76	89.46	
512	WideResnet	71.95	70.61	89.54	88.45	87.75	74
	Resnet152	71.19	70.34	89.2	88.62	87.54	
	ResNext	72.79	71.97	90.10	89.56	88.65	

5.4. Discussion

We observe that the clustering model for ImageNet can be used to obtain 88.73% and 99.07% classification accuracy on CIFAR-10 & MNIST test sets, respectively. These values are comparable to the inference accuracy with their respective clustering models. This shows that a pre-trained clustering model, learned from a large image dataset, can be fairly generalized and useful for symbolic inference on other datasets.

The adversarial robustness of SymDNN depends on the

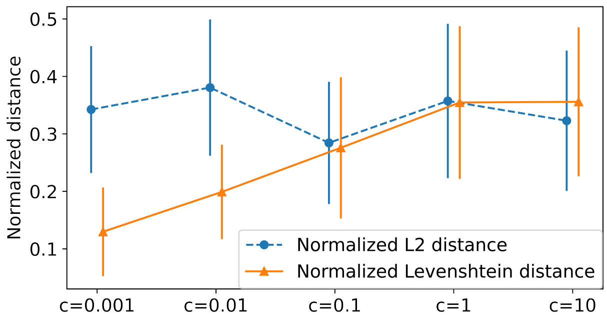


Figure 3. Investigation on C&W attack [12]: comparison of L2 distance between clean and attacked image vs. Levenshtein distance between clean and attacked symbolic image, for varying c.

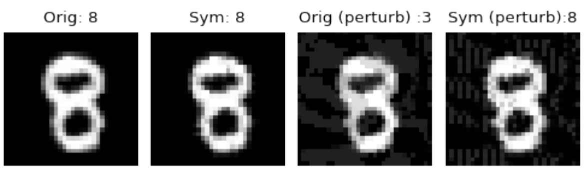


Figure 4. Resistance of SymDNN: The symbolic reconstruction possibly reinstates the faded out patterns of the perturbed image (3rd from left), by replacing with patch centroids in the reconstructed image (4th from left). EOTPGD [5] with $\epsilon = 32/255$.

underlying model used for clustering the patches. When the clustering algorithm is robust to small, potentially adversarial noise, it can map locally altered regions to the same representative patch (centroid), as illustrated in Fig. 4. When the symbolic image is reconstructed from the representative patches, it remains noise-free. Previous works investigate noise robustness of clustering algorithms in the context of random and potentially adversarial noise. In [1], it is proven in a theoretical framework that *k-means* clustering is robust to small set noise if the data is clusterable. For SymDNN, we find that the cluster centroids are mostly well separated based on the distances obtained from [29] library. (Please refer to the supplementary material).

To do further analyses, we choose Carlini & Wagner L_2 attack [12] on CIFAR-10 dataset for these experiments. In C & W attack, a constant c is introduced for trading between generation of visually similar adversarial examples (effectiveness) and large number of adversarial examples (success rate). We identify the cases where the non-symbolic inference failed, but the symbolic one survived against the adversarial attack. We find 1300 such cases (mean: 1293, std:220) out of 2000 randomly selected images from CIFAR-10 testset, for varying values of c. We measure (A) the normalized L_2 norm between clean and attacked image, and (B) the normalized Levenshtein (edit) distance between the reconstructed clean symbolic image and the symbolic image after the attack.

Fig. 3 shows a visual comparison between the two distance measures, for different values of c. We observe that the mean value of the normalized edit distance remains significantly lower than the L_2 norm, till the c value reaches 1. The edit distance remains low as small changes in the image do not result in significant change to the cluster assignments. We believe that the *effective adversarial examples* that C & W attack generates, are thwarted by this clustering robustness, resulting in a boost in robust symbolic accuracy. However, the performance degrades as the value of c reaches 1 and beyond, i.e., when the perturbations are larger, the symbol map corresponding to the attacked image changes significantly. It may be noted that this is a defense black-box experiment and hence the gradient obfuscation does not happen here.

It is also evident from Fig. 3 that the edit distance curve is more aligned to the step-wise increment of the c value, compared to the L_2 distance. When using a symbolic inference scheme, we believe that the edit distances between clean and attacked symbolic images have the potential to reveal further insights about adversarial attacks.

6. Conclusions

In this paper we have presented the algorithmic basis and comprehensive evaluation of SymDNN, a scheme for abstraction & reconstruction of an input image before DNN inference. We highlight that the DNNs deployed on embedded systems lack defense mechanisms against a wide variety of adversarial attacks. Adversarial research community strives for defenses that are robust against *any* adversarial attacks, in a complete white-box setting. There are only a small number of methods that achieve that aim. However those methods are not suitable for resource-limited embedded systems. Thus embedded vision tasks are unprotected against *any* adversarial attacks. We show that SymDNN has capability to undo adversarial perturbations for a wide range of recent attacks, under black-box and white-box attack models, when the defense remains a black-box to the adversary. Under a complete white-box one attack model, we show that the key to resist the recent strong adaptive attacks, is designing transformation functions that are as hard as possible. Our proposed SymDNN employs such an input transformation, which is computationally efficient, and results in image compression.

Along with these concrete benefits, we also report several interesting aspects of SymDNN inference, *e.g.* the fundamental visual symbols learned by the ImageNet codebook, the potential of Levenshtein distance as a measure of dissimilarity between clean and attacked images, etc. We believe that this paper will encourage other researchers to study the larger potential of *discretized* and *abstracted* images in computer vision. Our future work will be to strengthen the confluence between symbolic abstraction and DNN inference.

References

[1] Margareta Ackerman, Shai Ben-David, David Loker, and Sivan Sabato. Clustering oligarchies. In Carlos M. Carvalho and Pradeep Ravikumar, editors, *Proceedings of the Sixteenth International Conference on Artificial Intelligence and Statistics*, volume 31 of *Proceedings of Machine Learning Research*, pages 66–74, Scottsdale, Arizona, USA, 29 Apr–01 May 2013. PMLR. 8

[2] Ganesh Ananthanarayanan, Paramvir Bahl, Peter Bodík, Krishna Chintalapudi, Matthai Philipose, Lenin Ravindranath, and Sudipta Sinha. Real-time video analytics: The killer app for edge computing. *Computer*, 50(10):58–67, 2017. 1

[3] Maksym Andriushchenko, Francesco Croce, Nicolas Flammarion, and Matthias Hein. Square attack: A query-efficient black-box adversarial attack via random search. In *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIII*, page 484–501, Berlin, Heidelberg, 2020. Springer-Verlag. 7

[4] Anish Athalye, Nicholas Carlini, and David A. Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *ICML*, pages 274–283, 2018. 1, 2, 3, 5, 6, 7

[5] Anish Athalye, Logan Engstrom, Andrew Ilyas, and Kevin Kwok. Synthesizing robust adversarial examples. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 284–293. PMLR, 10–15 Jul 2018. 1, 2, 5, 8

[6] Yassine Bakhti, Sid Ahmed Fezza, Wassim Hamidouche, and Olivier Déforges. Ddsa: A defense against adversarial attacks using deep denoising sparse autoencoder. *IEEE Access*, 7:160397–160407, 2019. 3

[7] Yoshua Bengio. Learning deep architectures for ai. *Found. Trends Mach. Learn.*, 2(1):1–127, jan 2009. 1

[8] Battista Biggio, Igino Corona, Davide Maiorca, Blaine Nelson, Nedim Srndic, Pavel Laskov, Giorgio Giacinto, and Fabio Roli. Evasion attacks against machine learning at test time. In H. Blockeel, editor, *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD)*, pages 387–402. Springer-Verlag Berlin Heidelberg, Springer-Verlag Berlin Heidelberg, 2013. 1

[9] W. Brendel, J. Rauber, M. Kümmerer, I. Ustyuzhaninov, and M. Bethge. Accurate, reliable and fast robustness evaluation. In *Advances in Neural Information Processing Systems 32, 2019*, Dec 2019. 1

[10] Jacob Buckman, Aurko Roy, Colin Raffel, and Ian Goodfellow. Thermometer encoding: One hot way to resist adversarial examples. In *International Conference on Learning Representations*, 2018. 2, 3, 5

[11] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 39–57, 2017. 1

[12] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 39–57, 2017. 6, 7, 8

[13] Jiasi Chen and Xukan Ran. Deep learning with edge computing: A review. *Proceedings of the IEEE*, 107(8):1655–1674, 2019. 1

[14] Ka-Ho Chow, Wenqi Wei, Yanzhao Wu, and Ling Liu. Denoising and verification cross-layer ensemble against blackbox adversarial attacks, 2019. 3

[15] Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. Certified adversarial robustness via randomized smoothing. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 1310–1320. PMLR, 09–15 Jun 2019. 1

[16] Francesco Croce, Maksym Andriushchenko, and Matthias Hein. Provable robustness of relu networks via maximization of linear regions. In Kamalika Chaudhuri and Masashi Sugiyama, editors, *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, volume 89 of *Proceedings of Machine Learning Research*, pages 2057–2066. PMLR, 16–18 Apr 2019. 1

[17] Francesco Croce, Maksym Andriushchenko, Vikash Sehwag, Edoardo Debenedetti, Nicolas Flammarion, Mung Chiang, Prateek Mittal, and Matthias Hein. Robustbench: a standardized adversarial robustness benchmark, 2021. 2, 6

[18] Francesco Croce and Matthias Hein. Minimally distorted adversarial examples with a fast adaptive boundary attack. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 2196–2205. PMLR, 13–18 Jul 2020. 7

[19] Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *ICML*, 2020. 1, 2

[20] Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 2206–2216. PMLR, 13–18 Jul 2020. 7

[21] Y. Le Cun, B. Boser, J. S. Denker, R. E. Howard, W. Habbard, L. D. Jackel, and D. Henderson. *Handwritten Digit Recognition with a Back-Propagation Network*, page 396–404. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1990. 6

[22] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 2, 7

[23] Yinpeng Dong, Tianyu Pang, Hang Su, and Jun Zhu. Evading defenses to transferable adversarial examples by translation-invariant attacks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 1, 2, 7

[24] Abhimanyu Dubey, Laurens van der Maaten, Zeki Yalniz, Yixuan Li, and Dhruv Mahajan. Defense against adversarial images using web-scale nearest-neighbor search. In

2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 8759–8768, 2019. 2

[25] Alexei A. Efros and William T. Freeman. Image quilting for texture synthesis and transfer. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '01, page 341–346, New York, NY, USA, 2001. Association for Computing Machinery. 3

[26] Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015. 1, 2, 6

[27] Chuan Guo, Mayank Rana, Moustapha Cisse, and Laurens van der Maaten. Countering adversarial images using input transformations, 2018. 2, 3, 5

[28] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 6

[29] Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 7(3):535–547, 2021. 2, 5, 8

[30] Hoki Kim. Torchattacks: A pytorch repository for adversarial attacks, 2021. 6

[31] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical Report 0, University of Toronto, Toronto, Ontario, 2009. 6

[32] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world, 2017. 1

[33] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial machine learning at scale, 2017. 2

[34] Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2, 2010. 6

[35] Sungyoon Lee and Jaewook Lee. Defensive denoising methods against adversarial attack, 2018. 3

[36] Fangzhou Liao, Ming Liang, Yinpeng Dong, Tianyu Pang, Xiaolin Hu, and Jun Zhu. Defense against adversarial attacks using high-level representation guided denoiser. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 1778–1787. Computer Vision Foundation / IEEE Computer Society, 2018. 3

[37] 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. 1, 2

[38] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. 6

[39] 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*, CCS '17, page 135–147, New York, NY, USA, 2017. Association for Computing Machinery. 2, 3

[40] Matthew Mirman, Timon Gehr, and Martin Vechev. Differentiable abstract interpretation for provably robust neural networks. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 3578–3586. PMLR, 10–15 Jul 2018. 1

[41] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks, 2016. 7

[42] Muzammal Naseer, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Fatih Porikli. A self-supervised approach for adversarial robustness. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 2, 3, 6, 7

[43] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z. Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*, ASIA CCS '17, page 506–519, New York, NY, USA, 2017. Association for Computing Machinery. 1, 5

[44] Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z. Berkay Celik, and Ananthram Swami. The limitations of deep learning in adversarial settings. In *2016 IEEE European Symposium on Security and Privacy (EuroS P)*, pages 372–387, 2016. 1

[45] Edward Raff, Jared Sylvester, Steven Forsyth, and Mark McLean. Barrage of random transforms for adversarially robust defense. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6521–6530, 2019. 3, 5

[46] Jonas Rauber, Wieland Brendel, and Matthias Bethge. Foolbox: A python toolbox to benchmark the robustness of machine learning models. In *Reliable Machine Learning in the Wild Workshop, 34th International Conference on Machine Learning*, 2017. 6

[47] Leslie Rice, Eric Wong, and J. Zico Kolter. Overfitting in adversarially robust deep learning, 2020. 2, 6, 7

[48] Wenjie Ruan, Xinping Yi, and Xiaowei Huang. *Adversarial Robustness of Deep Learning: Theory, Algorithms, and Applications*, page 4866–4869. Association for Computing Machinery, New York, NY, USA, 2021. 1

[49] 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. 6, 7

[50] Pouya Samangouei, Maya Kabkab, and Rama Chellappa. Defense-GAN: Protecting classifiers against adversarial attacks using generative models. In *International Conference on Learning Representations*, 2018. 5, 6, 7

[51] Mahadev Satyanarayanan. The emergence of edge computing. *Computer*, 50(1):30–39, 2017. 1

[52] Leo Schwinn, René Raab, An Nguyen, Dario Zanca, and Bjoern Eskofier. Exploring misclassifications of robust neural networks to enhance adversarial attacks, 2021. 6, 7

[53] Vikash Sehwag, Shiqi Wang, Prateek Mittal, and Suman Jana. Hydra: Pruning adversarially robust neural networks, 2020. 2

[54] Yash Sharma and Pin-Yu Chen. Attacking the madry defense model with l_1-based adversarial examples, 2018. 2

[55] Changhao Shi, Chester Holtz, and Gal Mishne. Online adversarial purification based on self-supervised learning. In *International Conference on Learning Representations*, 2021. 2

[56] Gagandeep Singh, Timon Gehr, Markus Püschel, and Martin Vechev. An abstract domain for certifying neural networks. *Proc. ACM Program. Lang.*, 3(POPL), jan 2019. 1

[57] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In Yoshua Bengio and Yann LeCun, editors, *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014. 1

[58] Florian Tramèr, Nicholas Carlini, Wieland Brendel, and Aleksander Madry. On adaptive attacks to adversarial example defenses. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. 1, 2, 3, 5, 6, 7

[59] Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian J. Goodfellow, Dan Boneh, and Patrick D. McDaniel. Ensemble adversarial training: Attacks and defenses. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. 7

[60] V.N. Vapnik. An overview of statistical learning theory. *IEEE Transactions on Neural Networks*, 10(5):988–999, 1999. 2

[61] Gunjan Verma and Ananthram Swami. Error correcting output codes improve probability estimation and adversarial robustness of deep neural networks. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. 2

[62] Fu Wang, Liu He, Wenfen Liu, and Yanbin Zheng. Harden deep convolutional classifiers via k-means reconstruction. *IEEE Access*, 8:168210–168218, 2020. 3

[63] Eric Wong and Zico Kolter. Provable defenses against adversarial examples via the convex outer adversarial polytope. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 5286–5295. PMLR, 10–15 Jul 2018. 1

[64] Yanzhao Wu, Ling Liu, Zhongwei Xie, Ka-Ho Chow, and Wenqi Wei. Boosting ensemble accuracy by revisiting ensemble diversity metrics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16469–16477, June 2021. 7

[65] Cihang Xie, Zhishuai Zhang, Yuyin Zhou, Song Bai, Jianyu Wang, Zhou Ren, and Alan Yuille. Improving transferability of adversarial examples with input diversity, 2019. 7

[66] Saining Xie, Ross Girshick, Piotr Dollar, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 6

[67] Weilin Xu, David Evans, and Yanjun Qi. Feature squeezing: Detecting adversarial examples in deep neural networks. *Proceedings 2018 Network and Distributed System Security Symposium*, 2018. 2, 3, 5

[68] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In Edwin R. Hancock Richard C. Wilson and William A. P. Smith, editors, *Proceedings of the British Machine Vision Conference (BMVC)*, pages 87.1–87.12. BMVA Press, September 2016. 6

[69] Yanbin Zheng, Hongxu Yun, Fu Wang, Yong Ding, Yongzhong Huang, and Wenfen Liu. Defence against adversarial attacks using clustering algorithm. In Xiaohui Cheng, Weipeng Jing, Xianhua Song, and Zeguang Lu, editors, *Data Science*, pages 323–333, Singapore, 2019. Springer Singapore. 3

[70] Chen Zhu, Renkun Ni, Zheng Xu, Kezhi Kong, W. Ronny Huang, and Tom Goldstein. Gradinit: Learning to initialize neural networks for stable and efficient training. *CoRR*, abs/2102.08098, 2021. 6

[71] Roland S. Zimmermann. Comment on "adv-bnn: Improved adversarial defense through robust bayesian neural network", 2019. 7