A Kernelized Manifold Mapping to Diminish the Effect of Adversarial Perturbations

Saeid Asgari Taghanaki*[1], Kumar Abhishek[1], Shekoofeh Azizi[2], and Ghassan Hamarneh[1]

[1]School of Computing Science, Simon Fraser University, Canada
{sasgarit, kabhishe, hamarneh}@sfu.ca
[2]Department of Electrical and Computer Engineering, University of British Columbia, Canada
shazizi@ece.ubc.ca

Abstract

The linear and non-flexible nature of deep convolutional models makes them vulnerable to carefully crafted adversarial perturbations. To tackle this problem, we propose a non-linear radial basis convolutional feature mapping by learning a Mahalanobis-like distance function. Our method then maps the convolutional features onto a linearly well-separated manifold, which prevents small adversarial perturbations from forcing a sample to cross the decision boundary. We test the proposed method on three publicly available image classification and segmentation datasets namely, MNIST, ISBI ISIC 2017 skin lesion segmentation, and NIH Chest X-Ray-14. We evaluate the robustness of our method to different gradient (targeted and untargeted) and non-gradient based attacks and compare it to several non-gradient masking defense strategies. Our results demonstrate that the proposed method can increase the resilience of deep convolutional neural networks to adversarial perturbations without accuracy drop on clean data.

1. Introduction

Deep convolutional neural networks (CNNs) are highly vulnerable to adversarial perturbations [14, 43] produced by two main groups of attack mechanisms: white- and black-box, in which the target model parameters and architecture are accessible and hidden, respectively. In order to mitigate the effect of adversarial attacks, two categories of defense techniques have been proposed: data-level and algorithmic-level. Data-level methods include adversarial training [14, 43], pre-/post-processing methods (e.g., feature squeezing) [55], pre-processing using basis functions [39], and noise removal [16, 28]. Algorithmic-level methods [12, 18, 37] modify the deep model or the training algorithm by reducing the magnitude of gradients [32] or blocking/masking gradients [3, 15, 41]. However, these approaches are not completely effective against several different white- and black-box attacks [28, 37, 47] and pre-processing based methods might sacrifice accuracy to gain resilience to attacks that may never happen. Generally, most of these defense strategies cause a drop in the standard accuracy on clean data [48]. For more details on adversarial attacks and defenses, we refer the readers to [56].

Gradient masking has shown sub-optimal performance against different types of adversarial attacks [31, 47]. Athalye et al. [2] identified *obfuscated gradients*, a special case of gradient masking that leads to a false sense of security in defenses against adversarial perturbations. They showed that 7 out of 9 recent white-box defenses relying on this phenomenon ([3, 8, 15, 25, 37, 41, 53]) are vulnerable to single step or non-gradient based attacks. They finally suggested several symptoms of defenses that rely on obfuscated gradients. Although adversarial training [27] showed reasonable robustness against first order adversaries, this method has two major drawbacks. First, in practice the attack strategy is unknown and it is difficult to choose appropriate adversarial images for training, and second, the method requires more training time [1]. Furthermore, recent studies [23, 40] show that the adversarial training overfits the L_∞ metric while remaining highly susceptible to L_0, L_1, and L_2 perturbations.

As explored in the literature, successful adversarial attacks are mainly the result of models being *too linear* [13, 14, 24, 38] in high dimensional manifolds causing the decision boundary to be close to the manifold of the training data [45] and/or because of the models' *low flexibility* [10]. Another hypothesis is that adversarial examples are off the

*Corresponding author

data manifold [22, 37, 41]. To boost the non-linearity of a model, Goodfellow et al. [14] explored a variety of methods involving utilizing quadratic units and including shallow and deep radial basis function (RBF) networks. They achieved reasonably good performance against adversarial perturbations with shallow RBF networks. However, they found it difficult to train deep RBF models, leading to a high training error using stochastic gradient decent. Fawzi et al. [11] showed that support vector machines with RBF kernels can effectively resist adversarial attacks.

Typically, a single RBF network layer takes a vector of $\mathbf{x} \in \mathbb{R}^n$ as input and outputs a scalar function of the input vector $f(\mathbf{x}) : \mathbb{R}^n \rightarrow \mathbb{R}$ computed as $f(\mathbf{x}) = \sum_{i=1}^{P} w_i e^{-\beta_i D(\mathbf{x}, \mathbf{c}_i)}$, where P is the number of neurons in the hidden layer, $\mathbf{c}_i$ is the center vector for neuron i, $D(\mathbf{x}, \mathbf{c}_i)$ measures the distance between $\mathbf{x}$ and $\mathbf{c}_i$, w_i weights the output of neuron i, and β_i corresponds to the width of the Gaussian. The input can be made linearly separable with a high probability by transforming it to a higher dimensional space (Cover's theorem [7])). The Gaussian basis functions, commonly used in RBF networks, are local to the center vectors, i.e., $\lim_{\|x\| \to \infty} D(\mathbf{x}, \mathbf{c}_i) = 0$, which in turn implies that a small perturbation ϵ added to a sample input $\mathbf{x}$ of the neuron has an increasingly smaller effect on the output response as $\mathbf{x}$ becomes farther from the center of that neuron. Traditional RBF networks are normally trained in two sequential steps. First, an unsupervised method, e.g., K-means clustering, is applied to find the RBF centers [52] and, second, a linear model with coefficients w_i is fit to the desired outputs.

To tackle the *linearity* issue of the current deep CNN models (i.e., which are persistent despite stacking several linear units and using non-linear activation functions [13, 14, 24, 38]), which results in vulnerability to adversarial perturbations, we equip CNNs with radial basis mapping kernels. Radial basis functions with Euclidean distance might not be effective as the activation of each neuron depends only on the Euclidean distance between a pattern and the neuron center. Also, since the activation function is constrained to be symmetrical, all attributes are considered equally relevant. To address these limitations, we can add flexibility to the model by applying asymmetrical quadratic distance functions, such as the Mahalanobis distance, to the activation function in order to take into account the variability of the attributes and their correlations. However, computing the Mahalanobis distance requires complete knowledge of the attributes, which are not readily available in dynamic environments (e.g., features iteratively updated during training CNNs), and are not optimized to maximize the accuracy of the model.

Contributions. In this paper, (I) we propose a new non-linear kernel based Mahalanobis distance-like feature transformation method. (II) Unlike the traditional Mahalanobis formulation, in which a constant, pre-defined covariance matrix is adopted, we propose to *learn* such a "transformation" matrix Ψ. (III) We propose to learn the RBF centers and the Gaussian widths in our RBF transformers. (IV) Our method adds robustness to attacks without reducing the model's performance on clean data, which is not the case for previously proposed defenses. (V) We propose a defense mechanism which can be applied to various tasks, e.g., classification, segmentation (both evaluated in this paper), and object detection.

2. Method

In this section, we introduce a novel asymmetrical quadratic distance function (2.1), and then propose a manifold mapping method using the proposed distance function (2.2). Then we discuss the model optimization and manifold parameter learning algorithm (2.3).

2.1. Adaptive kernelized distance calculation

Given a convolutional feature map $f^{(l)} \in \mathcal{F}^{nm}$ of size $n \times m \times K^{(l)}$ for layer l of a CNN, the goal is to map the features onto a new manifold $g^{(l)} \in \mathcal{G}^{nm}$ of size $n \times m \times P^{(l)}$ where classes are more likely to be linearly separable. Towards this end, we leverage an RBF transformer that takes feature vectors of $f_{K^{(l)}}^{(l)}$ as input and maps them onto a linearly separable manifold by learning a transformation matrix $\Psi^{(l)} \in \mathbb{R}^{K^{(l)} \times P^{(l)}}$ and a non-linear kernel $\kappa : \mathcal{F} \times \mathcal{F} \rightarrow \mathbb{R}$ for which there exists a representation manifold $\mathcal{G}$ and a map $\phi : \mathcal{F} \rightarrow \mathcal{G}$ such that

$$\kappa(x, z) = \phi(x).\phi(z) \qquad \forall x, z \in \mathcal{F} \tag{1}$$

The k^{th} RBF neuron activation function is given by

$$\phi_k^{(l)} = e^{-\beta_k^{(l)} D\left(f_{K^{(l)}}^{(l)}, c_k^{(l)}\right)} \tag{2}$$

where $c_k^{(l)}$ is k^{th} learnable center, and the learnable parameter $\beta_k^{(l)}$ controls the width of the k^{th} Gaussian function. We use a distance metric $D(.)$ inspired by the Mahalanobis distance, which refers to a distance between a convolutional feature vector and a center, and is computed as:

$$D\left(f_{K^{(l)}}^{(l)}, c_k^{(l)}\right) = \left(f_{K^{(l)}}^{(l)} - c_k^{(l)}\right)^T \left(\Psi^{(l)}\right)^{-1} \left(f_{K^{(l)}}^{(l)} - c_k^{(l)}\right) \tag{3}$$

where $\Psi^{(l)}$ refers to the learnable transformation matrix of size $K^{(l)} \times P^{(l)}$. To ensure that the transformation matrix $\Psi^{(l)}$ is positive semi-definite, we set $\Psi^{(l)} = AA^T$ and optimize for A.

Finally, the transformed feature vector $g^{(l)}(f_{K^{(l)}}^{(l)})$ is computed as:

$$g^{(l)}\left(f_{K^{(l)}}^{(l)}\right)=\sum_{k=1}^{P^{(l)}}\left(w_k^{(l)}\cdot\phi_k^{(l)}\right)+b^{(l)} \tag{4}$$

2.2. The proposed manifold mapping

For any intermediate layer l in the network, we compute the transformed feature vector when using the RBF transformation matrix as

$$\begin{aligned}g^{(l)}\left(f^{(l)}\right)&=\sum_{k=1}^{P^{(l)}}\left(w_k^{(l)}\cdot\phi_k^{(l)}\right)+b^{(l)}\\&=\sum_{k=1}^{P^{(l)}}\left({w_k}^{(l)}\cdot\phi_k^{(l)}\left(f_k^{(l)}\left(\Theta\right)\right)\right)+b^{(l)}\\&=\sum_{k=1}^{P^{(l)}}\left(w_k^{(l)}\right)\cdot\exp\left\{-\beta_k^{(l)}\left(f_k^{(l)}\left(\Theta\right)\right.\right.\\&\qquad\left.\left.-c_k^{(l)}\right)^T\left(\Psi^{(l)}\right)^{-1}\left(f_k^{(l)}\left(\Theta\right)-c_k^{(l)}\right)\right\}+b^{(l)}\end{aligned} \tag{5}$$

where $f^{(l)}$ is a function of the CNN parameters Θ.

The detailed diagram of the mapping step is shown in Figure 1. In each layer l of network, output feature maps of the convolutional block are concatenated to the transformed feature maps (Figure 2).

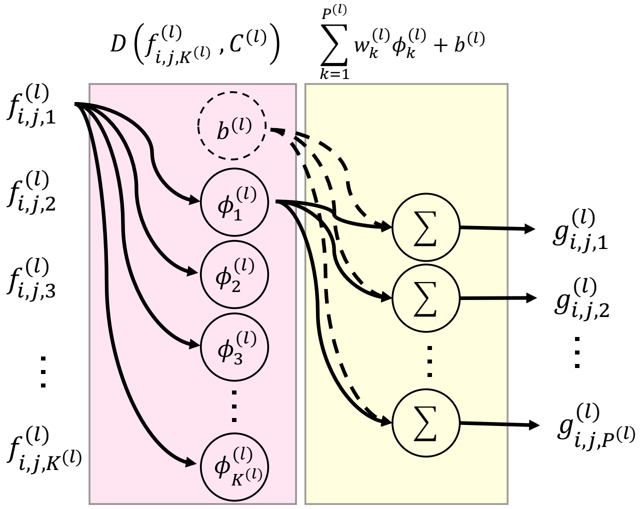


Figure 1: Detailed diagram of the proposed radial basis mapping block $g^{(l)}(f^{(l)})$. In the figure, $i = 1:n$; $j = 1:m$; $l = 1:L$ where n, m, $K^{(l)}$, and $P^{(l)}$ are the width, the height, and the number of channels of the input and the output feature maps respectively for layer l.

For the last layer (layer L) of the network, as shown in the green bounding box in Figure 2, the expression becomes

$$\begin{aligned}g^{(L)}\left(f^{(L)}\right)=&\sum_{k=1}^{P^{(L)}}\left(w_k^{(L)}\right)\cdot\exp\left\{-\beta_k^{(L)}\left(f_k^{(L)}\left(\Theta\right)\right.\right.\\&\left.\left.-c_k^{(L)}\right)^T\left(\Psi^{(L)}\right)^{-1}\left(f_k^{(L)}\left(\Theta\right)-c_k^{(L)}\right)\right\}\\&+b^{(L)}\end{aligned} \tag{6}$$

The concatenation output of the last layer is given by

$$y=\left[g^{(L)}(f^{(L)}),f^{(L)}\right] \tag{7}$$

2.3. Model optimization and mapping parameter learning

All the RBF parameters, i.e., the transformation matrix $\Psi^{(l)}$, the RBF centers $C^{(l)}=\{c_1^{(l)},c_2^{(l)},\cdots,c_{N^{(l)}}^{(l)}\}$ (where $N^{(l)}$ denotes the number of RBF centers in layer l), and the widths of the Gaussians $\beta_i^{(l)}$, along with all the CNN parameters Θ, are learned end-to-end using back-propagation. This approach results in the local RBF centers being adjusted optimally as they are updated based on the whole network parameters.

The categorical cross entropy loss is calculated using the softmax activation applied to the concatenation output of the last layer. The softmax activation function for the i^{th} class, denoted by $\xi(y)_i$, is defined as

$$\xi(y)_i=\frac{e^{y_i}}{\sum_j^Q e^{y_j}} \tag{8}$$

where Q represents the total number of classes. The loss is therefore defined by

$$\mathcal{L}=-\sum_i^Q t_i\,log\left(\xi(y)_i\right) \tag{9}$$

Since the multi-class classification labels are one-hot encoded, the expression for the loss contains only the element of the target vector $\mathbf{t}$ which is not zero, i.e., t_p, where p denotes the positive class. This expression can be simplified by discarding the summation elements which are zero because of the target labels, and we get

$$\mathcal{L}=-\,log\left(\frac{e^{y_p}}{\sum_j^Q e^{y_j}}\right) \tag{10}$$

where y_p represents the network output for the positive class.

Therefore, we define a loss function $\mathcal{L}$ encoding the classification error in the transformed manifold and seek A^*, β^*, C^*, and Θ^* that minimize $\mathcal{L}$:

$$A^*,C^*,\beta^*,\Theta^*=\arg\min_{A,C,\beta,\Theta}\ \mathcal{L}\left(A,C,\beta,\Theta\right). \tag{11}$$

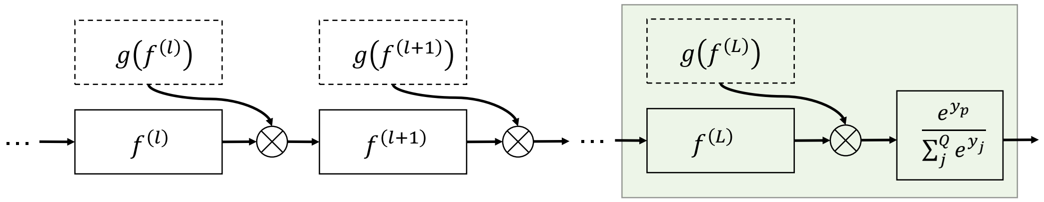


Figure 2: Placing radial basis mapping blocks in a CNN network. $\otimes$ shows concatenation operation, and $f^{(l)}$ is the output of convolution layer l. The output of each RBF block contains $n \in [1, N]$ channels that are concatenated to convolutional feature maps.

3. Data

We conduct two sets of experiments: image (i) classification and (ii) segmentation. (i) For the image classification experiments, we use the MNIST [21] and the NIH Chest X-Ray-14 [51] (hereafter referred to as CHEST) datasets, where the latter comprises of 112,120 gray-scale images with 14 disease labels and one 'no (clinical) finding' label. We treat all the disease classes as positive and formulate a binary classification task. We randomly selected 90,000 images for training: 45,000 images with "positive" label and the remaining 45,000 with "negative" label. The validation set comprised of 13,332 images with 6,666 images of each label. We randomly picked 200 unseen images as the test set, with 93 images of positive and 107 images of negative class, respectively. These clean (test) images are used for carrying out different adversarial attacks and the models trained on clean images are evaluated against them. (ii) For the image segmentation task experiments, we use the 2D RGB skin lesion dataset from the 2017 IEEE ISBI International Skin Imaging Collaboration (ISIC) Challenge [6] (hereafter referred to as SKIN). We trained on a set of 2,000 images and test on an unseen set of 150 images.

4. Experiments and results

In this section, we report the results of several experiments for two tasks of classification and segmentation. We first start with MNIST as it has extensively been used for evaluating adversarial attacks and defenses. Next, we show how the proposed method is applicable to another classification dataset and segmentation task.

4.1. Evaluation on classification task

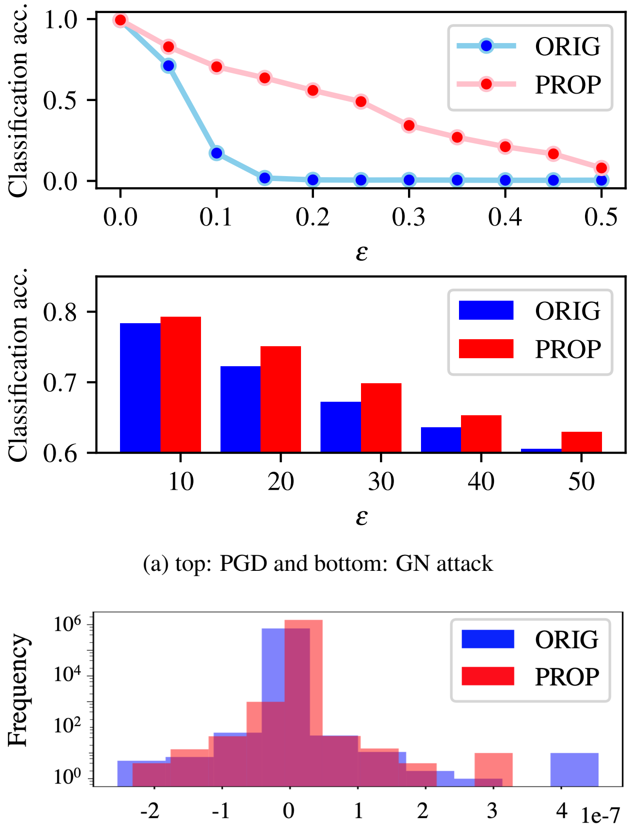
In this section, we analyze the performance of the proposed method on two different classifications datasets MNIST and CHEST. In Table 1, we report the results of the proposed method on MNIST dataset when attacked by different targeted and un-targeted attacks i.e., fast gradient sign method (FGSM) [19], basic iterative method (BIM) [19], projected gradient descent (PGD) [27], Carlini & Wagner method (C&W) [5], and momentum iterative method (MIM) [9] (the winner of NIPS 2017 adversarial attacks competition). The proposed method (i.e., PROP) successfully resists all the attacks (with both L_∞ and L_2 perturbations) for which the 3-layers CNN (i.e., ORIG) network almost completely fails e.g., for the strongest attack (i.e., MIM) the proposed method achieves 64.25% accuracy while the original CNN network obtains almost zero (0.58%) accuracy. Further, we test the proposed method with two non-gradient based attacks: simultaneous perturbation stochastic approximation (SPSA) [49] and Gaussian additive noise (GN) [33] to show that the robustness of the proposed method is not because of gradient masking. We compare our results to other defenses e.g., Binary CNN, Nearest Neighbour (NN) model, Analysis by Synthesis (ABS), Binary ABS [23] and Fortified Networks [20]. Looking at the Binary ABS and ABS results in Table 1, the former generally outperforms the latter, but it should be noted that Binary ABS is applicable only to simple datasets, e.g., MNIST, as it leverages binarization. Although Fortified Net outperforms our method for the FGSM attack, it has been tested only on gradient-based attacks, and therefore, it is unclear how it would perform against gradient-free attacks such as SPSA and GN.

To ensure that the robustness of the proposed method is not due to masked/obfuscated gradient, as suggested by [2], we test the proposed feature mapping method based on several characteristic behaviors of defenses that cause obfuscated gradients to occur. a) As reported in Table 1, one-step attacks (e.g., FGSM) did not perform better than iterative attacks (e.g., BIM, MIM); b) According to Tables 4 and 5, black-box attacks did not perform better than white-box ones; c) as shown in Figure 3 (a and b), larger distortion factors monotonically increase the attack success rate; d) the proposed method performs well against gradient-free attacks e.g., GN and SPSA. The subplot (b) in Figure 3 also indicates that the robustness of the proposed method is not because of numerical instability of gradients.

Next, to quantify the compactness and separability of different clusters/classes, we evaluate the features produced ORIG and PROP methods with clustering evaluation techniques such as mutual information based score [50], homo-

Table 1: Classification accuracy under different attacks tested on MNIST dataset. FGSM: $\epsilon = 0.3$; BIM: $\epsilon = 0.3$ and iterations = 5; MIM: $\epsilon = 0.3$, iterations = 10, and decay factor = 1; PGD: $\epsilon = 0.1$, iterations = 40; C&W: iterations = 50, GN: $\epsilon = 20$; SPSA: $\epsilon = 0.3$. "n/a" denotes that the corresponding entry was not reported in the respective paper.

Models	Clean	L_2		L_∞				
		C&W [5]	GN [33]	FGSM [19]	BIM [19]	MIM [9]	PGD [27]	SPSA [49]
ORIG [30]	0.9930	0.1808	0.7227	0.0968	0.0070	0.0051	0.1365	0.3200
Binary CNN [23]	0.9850	n/a	0.9200	0.7100	0.7000	0.7000	n/a	n/a
NN [23]	0.9690	n/a	0.9100	0.6800	0.4300	0.2600	n/a	n/a
Binary ABS [23]	0.9900	n/a	0.8900	0.8500	**0.8600**	**0.8500**	n/a	n/a
ABS [23]	0.9900	n/a	**0.9800**	0.3400	0.1300	0.1700	n/a	n/a
Fortified Net [20]	0.9893	0.6058	n/a	**0.9131**	n/a	n/a	0.7954	n/a
PROP	**0.9942**	**0.9879**	0.7506	0.8582	0.7887	0.6425	**0.8157**	**0.7092**



(a) top: PGD and bottom: GN attack

(b) Gradient distribution

Figure 3: Gradient masking analysis: (a) accuracy-distortion plots and (b) gradient distribution for the proposed method and the original CNN

geneity and completeness [35], Silhouette coefficient [36], and Calinski-Harabaz index [4]. Both Silhouette coefficient and Calinski-Harabaz index quantify how well clusters are separated from each other and how compact they are without taking into account the ground truth labels, while mutual information based score, homogeneity, and completeness scores evaluate clusters based on labels. As reported in Table 2, when the original CNN network applies radial basis feature mapping it achieves considerably higher scores (for all the metrics, higher values are better). As both the original and the proposed method achieved high classification test accuracy i.e., $\sim 99\%$, the difference in scores for label based metrics, i.e., mutual information based, homogeneity, and completeness, scores are small.

Table 2: Feature mapping analysis via intra-class compactness and inter-class separability measures of the MNIST dataset for original 3-layer CNN versus the proposed method. The abbreviated column headers are Silhouette, Calinski, Mutual Information, Homogeneity, and Completeness metrics, respectively.

	Sil.	Cal.	MI	Homo.	Comp.
ORIG [30]	0.2612	1658.20	0.9695	0.9696	0.9721
PROP	**0.4284**	**2570.42**	**0.9720**	**0.9721**	**0.9815**

In Figure 4, we visualize the feature spaces of each layer in a simple 3-layer CNN using t-SNE [26] and PCA [17] methods by reducing the high dimensional feature space into two dimensions. As can be seen, the proposed radial basis feature mapping reduces intra-class and increases inter-class distances.

To test the robustness of the proposed method on CHEST, we follow the strategy of Taghanaki et al. [44]. We select Inception-ResNet-v2 [42] and modify it by the proposed radial basis function blocks. According to the study done by Taghanaki et al. [44], we focus on the most effective attacks in term of imperceptibility and power i.e., gradient-based attacks (basic iterative method [19]: BIM and L1-BIM). We also compare the proposed method with two defense strategies: Gaussian data augmentation

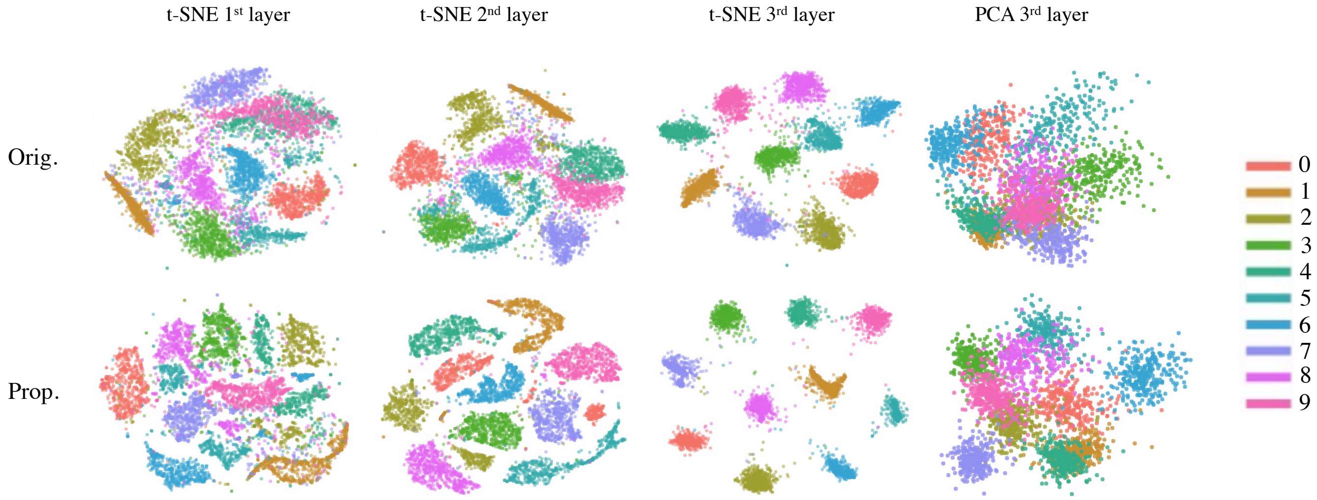


Figure 4: Feature space visualization at different layers via t-SNE and PCA of MNIST dataset produced via original 3-layer CNN network and the proposed method.

(GDA) [57] and feature squeezing (FSM) [55]. GDA is a data augmentation technique that augments a dataset with copies of the original samples to which Gaussian noise has been added. FSM method reduces the precision of the components of the original input by encoding them with a smaller number of bits. In Table 3, we report the classification accuracy of different attacks and defenses (including PROP) on CHEST.

Table 3: Classification accuracy on CHEST for different attacks and defenses.

		Defense			
Attack	Iteration	ORIG	GDA	FSM	PROP
L_1 BIM [19]	5	0	0	0.55	**0.63**
L_∞ BIM [19]	5	0	0	0.54	**0.65**
Clean	-	0.74	**0.75**	0.57	0.74

4.2. Evaluation on segmentation task

To assess the segmentation vulnerability to adversarial attacks, we apply the dense adversary generation (DAG) method proposed by Xie et al. [54] to two state-of-the-art segmentation networks: U-Net [34] and V-Net [29] under both white- and black-box conditions. We compare the proposed feature mapping method to other defense strategies e.g., Gaussian and median feature squeezing [55] (FSG and FSM, respectively) and adversarial training [14] (ADVT) on SKIN. From Table 4, it can be seen that the proposed method is more robust to adversarial attacks and when applied to U-Net, its performance deteriorates much lesser than the next best method (ADVT) 1.60% and 6.39% vs 9.44% and 13.47% after 10 and 30 iterations of the attack with $\gamma = 0.03$. Similar performance was also observed for V-Net, where the accuracy drop of the proposed method using the same γ for 10 iterations was 8.50%, while the next best method (ADVT) dropped by 11.76%. It should also be noted that applying the feature mapping led to an improvement in the segmentation accuracy on clean (non-attacked/unperturbed) images, and the performance increased to 0.7780 from the original 0.7743 for U-Net, and to 0.8213 from the original 0.8070 for V-Net.

Figure 5 visualizes the segmentation results of a few samples from SKIN for different defense strategies. As shown, the proposed method obtains the closest results to the ISIC ground truth (GT) than all other methods. Although adversarial training (ADVT) also produces promising segmentation results, it requires knowledge of the adversary in order to perform robust optimization which is almost impossible to obtain in practice since the attack is unknown. However, the proposed method does not have such a dependency.

As reported in Table 5, under black-box attack, the proposed method is the best performing method across all 12 experiments except for one in which the accuracy of the best method was just 0.0022 higher (i.e., 0.7284 ± 0.2682 vs 0.7262 ± 0.2621). However, it should be noted that the standard deviation of the winner is larger than the proposed method.

Next, we analyze the usefulness of learning the transformation matrix (Ψ) and width of the Gaussian (β) in our Mahalanobis-like distance calculation. As can be seen in Table 6, in all the cases, i.e., testing with clean images and images 10 and 30 iterations of attack, our method with Ψ and β achieved higher performance.

Table 4: Segmentation results (average DICE ± standard error) of different defense mechanisms compared to the proposed radial basis feature mapping method for V-Net and U-Net under DAG attack. $10i$ and $30i$ refer to 10 and 30 iterations of attack, respectively.

Network	Method	Clean	$10i$ (% Accuracy drop)	$30i$ (% Accuracy drop)
U-Net [34]	ORIG [34]	0.7743 ± 0.0202	$0.5594 \pm 0.0196 (27.75\%)$	$0.4396 \pm 0.0222 (43.23\%)$
	FSG [55]	0.7292 ± 0.0229	$0.6382 \pm 0.0206 (15.58\%)$	$0.5858 \pm 0.0218 (24.34\%)$
	FSM [55]	0.7695 ± 0.0198	$0.6039 \pm 0.0199 (22.01\%)$	$0.5396 \pm 0.0211 (30.31\%)$
	ADVT [14]	0.6703 ± 0.0273	$0.7012 \pm 0.0255 (9.44\%)$	$0.6700 \pm 0.0260 (13.47\%)$
	PROP	$\mathbf{0.7780 \pm 0.0209}$	$\mathbf{0.7619 \pm 0.0208\ (1.60\%)}$	$\mathbf{0.7248 \pm 0.0226\ (6.39\%)}$
V-Net [29]	ORIG [34]	0.8070 ± 0.0189	$0.5320 \pm 0.0207 (34.10\%)$	$0.3865 \pm 0.0217 (52.10\%)$
	FSG [55]	0.7886 ± 0.0205	$0.6990 \pm 0.0189 (13.38\%)$	$0.6840 \pm 0.0188 (15.24\%)$
	FSM [55]	0.8084 ± 0.0189	$0.5928 \pm 0.0209 (26.54\%)$	$0.5144 \pm 0.0218 (36.26\%)$
	ADVT [14]	0.7924 ± 0.0162	$0.7121 \pm 0.0174 (11.76\%)$	$\mathbf{0.7113 \pm 0.0179\ (11.85\%)}$
	PROP	$\mathbf{0.8213 \pm 0.0177}$	$\mathbf{0.7384 \pm 0.0169\ (8.50\%)}$	$0.6944 \pm 0.0178 (13.95\%)$

Table 5: Segmentation DICE ± standard error scores of black-box attacks; adversarial images were produced with methods in first left column and tested with methods in the first row. U-PROP and V-PROP refer to equipped U-Net and V-Net with our mapping method.

-	U-Net [34]	U-PROP	V-Net [29]	V-PROP
U-Net [34]	-	$\mathbf{0.7341 \pm 0.0205}$	0.6364 ± 0.0189	0.7210 ± 0.0189
U-PROP	$\mathbf{0.7284 \pm 0.0219}$	-	0.6590 ± 0.0218	0.7262 ± 0.0241
V-Net [29]	0.7649 ± 0.0168	$\mathbf{0.7773 \pm 0.0167}$	-	0.7478 ± 0.2090
V-PROP	0.7922 ± 0.0188	$\mathbf{0.7964 \pm 0.0192}$	0.6948 ± 0.0171	-

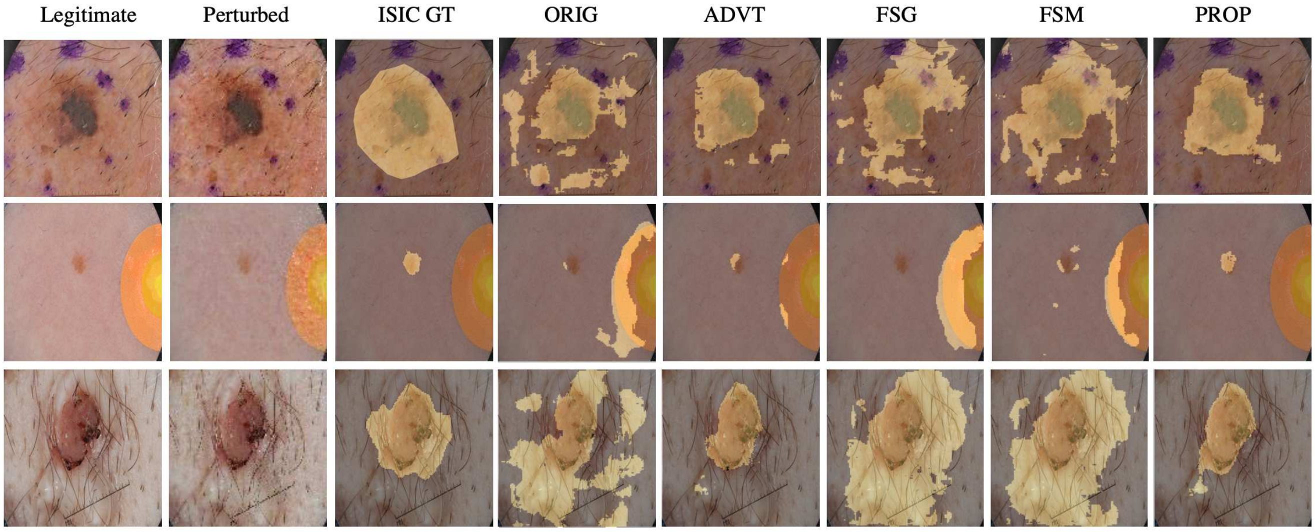


Figure 5: Lesion segmentation sample results on SKIN for different defense strategies applied with U-Net.

Figure 6 shows how the Ψ of a single layer converges after a few epochs on the MNIST dataset. The Y-axis is the Frobenius norm of the change in Ψ between two consecutive epochs. Adopting the value Ψ converges to results in a

Table 6: Ablation study over the usefulness of learning the transformation matrix Ψ and β on SKIN for V-Net; mean ± standard error

	Ψ	β	Dice	FPR	FNR
Clean	✗	✓	0.7721 ± 0.0210	0.0149 ± 0.0022	0.2041 ± 0.0245
	✓	✗	0.8200 ± 0.0163	0.0177 ± 0.0026	0.1547 ± 0.0188
	✗	✗	0.8002 ± 0.0184	0.0137 ± 0.0022	0.1883 ± 0.0211
	✓	✓	**0.8213** ± **0.0177**	0.0141 ± 0.0020	0.1706 ± 0.0200
$10i$	✗	✓	0.6471 ± 0.0213	0.0437 ± 0.0052	0.1992 ± 0.0261
	✓	✗	0.7010 ± 0.0161	0.0606 ± 0.0054	0.1020 ± 0.0166
	✗	✗	0.6740 ± 0.0187	0.0458 ± 0.0037	0.1472 ± 0.0217
	✓	✓	**0.7384** ± **0.0169**	0.0444 ± 0.0041	0.1234 ± 0.0186
$30i$	✗	✓	0.6010 ± 0.0221	0.0371 ± 0.0030	0.2304 ± 0.0273
	✓	✗	0.6458 ± 0.0180	0.0633 ± 0.0042	0.1164 ± 0.0183
	✗	✗	0.6188 ± 0.0188	0.0615 ± 0.0040	0.1384 ± 0.0205
	✓	✓	**0.6944** ± **0.0179**	0.0418 ± 0.0038	0.1489 ± 0.0205

superior performance (e.g., mean Dice 0.8213 vs 0.7721, as reported in Table 6) compared to when we do not optimize for Ψ, i.e., hold it constant (identity matrix).

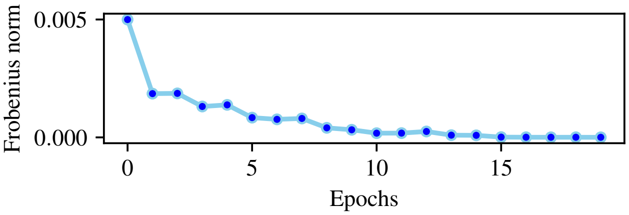


Figure 6: Convergence: Plotting the Frobenius norm of the change in Ψ between two consecutive epochs.

5. Implementation details

5.1. Classification experiments

- MNIST experiments: For both the original CNN with 3-layers (i.e., ORIG) and the proposed method (i.e., PROP), we used a batch size of 128 and Adam optimizer with learning rate of 0.001.
- CHEST: Inception-ResNet-v2 network was trained with a batch size of 4 with RMSProp optimizer [46] with a decay of 0.9 and $\epsilon = 1$ and an initial learning rate of 0.045, decayed every 2 epochs using an exponential rate of 0.94.

For all the gradient based attacks applied in the classification part, we used the CleverHans library [30], and for the Gaussian additive noise attack, we used FoolBox [33].

5.2. Segmentation experiments

For both U-Net and V-Net, we used a batch size of 16, ADADELTA optimizer with learning rate of 1.0, rho = 0.95, and decay = 0.0. We tested the DAG method with 10 and 30 iterations and perturbation factor $\gamma = 0.03$. For FSM and FSG defenses, we applied a window size of 3×3 and a standard deviation of 1.0, respectively.

6. Conclusion

We proposed a nonlinear radial basis feature mapping method to transform layer-wise convolutional features into a new manifold, where improved class separation reduces the effectiveness of perturbations when attempting to fool the model. We evaluated the model under white- and black-box attacks for two different tasks of image classification and segmentation and compared our method to other non-gradient based defenses. We also performed several tests to ensure that the robustness of the proposed method is neither because of numerical instability of gradients nor because of gradient masking. In contrast to previous methods, our proposed feature mapping improved the classification and segmentation accuracy on both clean and perturbed images.

Acknowledgement

Partial funding for this project is provided by the Natural Sciences and Engineering Research Council of Canada (NSERC). The authors are grateful to the NVIDIA Corporation for donating a Titan X GPU used in this research.

References

[1] S. R. Adnan, H. Zhezhi, G. Boqing, and F. Deliang. Blind pre-processing: A robust defense method against adversarial examples. *arXiv preprint arXiv:1802.01549*, 2018.

[2] A. Athalye, N. Carlini, and D. Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. *arXiv preprint arXiv:1802.00420*, 2018.

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

[4] T. Caliński and J. Harabasz. A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods*, 3(1):1–27, 1974.

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

[6] N. C. Codella, D. Gutman, M. E. Celebi, B. Helba, M. A. Marchetti, S. W. Dusza, A. Kalloo, K. Liopyris, N. Mishra, H. Kittler, et al. Skin lesion analysis toward melanoma detection: A challenge at the 2017 international symposium on biomedical imaging (isbi). *the International Skin Imaging Collaboration (ISIC). arXiv preprint arXiv:1710.05006*, 2017.

[7] T. M. Cover. Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. *IEEE Transactions on Electronic Computers*, EC-14(3):326–334, June 1965.

[8] G. S. Dhillon, K. Azizzadenesheli, Z. C. Lipton, J. Bernstein, J. Kossaifi, A. Khanna, and A. Anandkumar. Stochastic activation pruning for robust adversarial defense. *arXiv preprint arXiv:1803.01442*, 2018.

[9] Y. Dong, F. Liao, T. Pang, H. Su, X. Hu, J. Li, and J. Zhu. Boosting adversarial attacks with momentum. *arXiv preprint arXiv:1710.06081*, 2017.

[10] A. Fawzi, O. Fawzi, and P. Frossard. Fundamental limits on adversarial robustness. In *Proc. ICML, Workshop on Deep Learning*, 2015.

[11] A. Fawzi, O. Fawzi, and P. Frossard. Analysis of classifiers robustness to adversarial perturbations. *Machine Learning*, 107(3):481–508, 2018.

[12] J. Folz, S. Palacio, J. Hees, D. Borth, and A. Dengel. Adversarial defense based on structure-to-signal autoencoders. *arXiv preprint arXiv:1803.07994*, 2018.

[13] J. Gilmer, L. Metz, F. Faghri, S. S. Schoenholz, M. Raghu, M. Wattenberg, and I. Goodfellow. Adversarial spheres. *arXiv preprint arXiv:1801.02774*, 2018.

[14] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples (2015). *arXiv preprint arXiv:1412.6572*, 2015.

[15] C. Guo, M. Rana, M. Cisse, and L. van der Maaten. Countering adversarial images using input transformations. *arXiv preprint arXiv:1711.00117*, 2017.

[16] D. Hendrycks and K. Gimpel. Early methods for detecting adversarial images. *arXiv preprint arXiv:1608.00530*, 2016.

[17] I. Jolliffe. Principal component analysis. In *International encyclopedia of statistical science*, pages 1094–1096. Springer, 2011.

[18] J. Z. Kolter and E. Wong. Provable defenses against adversarial examples via the convex outer adversarial polytope. *arXiv preprint arXiv:1711.00851*, 2017.

[19] A. Kurakin, I. Goodfellow, and S. Bengio. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*, 2016.

[20] A. Lamb, J. Binas, A. Goyal, D. Serdyuk, S. Subramanian, I. Mitliagkas, and Y. Bengio. Fortified networks: Improving the robustness of deep networks by modeling the manifold of hidden representations. *arXiv preprint arXiv:1804.02485*, 2018.

[21] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[22] H. Lee, S. Han, and J. Lee. Generative adversarial trainer: Defense to adversarial perturbations with gan. *arXiv preprint arXiv:1705.03387*, 2017.

[23] S. Lukas, R. Jonas, B. Matthias, and B. Wieland. Towards the first adversarialy robust neural network model on mnist. *arXiv preprint arXiv:1805.09190*, 2018.

[24] Y. Luo, X. Boix, G. Roig, T. Poggio, and Q. Zhao. Foveation-based mechanisms alleviate adversarial examples. *arXiv preprint arXiv:1511.06292*, 2015.

[25] X. Ma, B. Li, Y. Wang, S. M. Erfani, S. Wijewickrema, M. E. Houle, G. Schoenebeck, D. Song, and J. Bailey. Characterizing adversarial subspaces using local intrinsic dimensionality. *arXiv preprint arXiv:1801.02613*, 2018.

[26] L. v. d. Maaten and G. Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[27] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

[28] D. Meng and H. Chen. Magnet: a two-pronged defense against adversarial examples. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 135–147. ACM, 2017.

[29] F. Milletari, N. Navab, and S.-A. Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *3D Vision (3DV), 2016 Fourth International Conference on*, pages 565–571. IEEE, 2016.

[30] P. Nicolas, F. Fartash, C. Nicholas, G. Ian, F. Reuben, K. Alexey, X. Cihang, S. Yash, B. Tom, R. Aurko, M. Alexander, B. Vahid, H. Karen, Z. Zhishuai, J. Yi-Lin, L. Zhi, S. Ryan, G. Abhibhav, U. Jonathan, G. Willi, D. Yinpeng, B. David, H. Paul, R. Jonas, and R. Long. Technical report on the cleverhans v2.1.0 adversarial examples library. *arXiv preprint arXiv:1610.00768*, 2018.

[31] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*, pages 506–519. ACM, 2017.

[32] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami. Distillation as a defense to adversarial perturbations against deep neural networks. *arXiv preprint arXiv:1511.04508*, 2015.

[33] J. Rauber, W. Brendel, and M. Bethge. Foolbox: A python toolbox to benchmark the robustness of machine learning models. *arXiv preprint arXiv:1707.04131*, 2017.

[34] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[35] A. Rosenberg and J. Hirschberg. V-measure: A conditional entropy-based external cluster evaluation measure. In *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL)*, 2007.

[36] P. J. Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.

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

[38] L. Schmidt, S. Santurkar, D. Tsipras, K. Talwar, and A. Madry. Adversarially robust generalization requires more data. *arXiv preprint arXiv:1804.11285*, 2018.

[39] U. Shaham, J. Garritano, Y. Yamada, E. Weinberger, A. Cloninger, X. Cheng, K. Stanton, and Y. Kluger. Defending against adversarial images using basis functions transformations. *arXiv preprint arXiv:1803.10840*, 2018.

[40] Y. Sharma and P.-Y. Chen. Breaking the madry defense model with l_1-based adversarial examples. *arXiv preprint arXiv:1710.10733*, 2017.

[41] Y. Song, T. Kim, S. Nowozin, S. Ermon, and N. Kushman. Pixeldefend: Leveraging generative models to understand and defend against adversarial examples. *arXiv preprint arXiv:1710.10766*, 2017.

[42] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. *arXiv preprint arXiv:1602.07261v2*, 2017.

[43] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

[44] S. A. Taghanaki, A. Das, and G. Hamarneh. Vulnerability analysis of chest x-ray image classification against adversarial attacks. *arXiv preprint arXiv:1807.02905*, 2018.

[45] T. Tanay and L. Griffin. A boundary tilting persepective on the phenomenon of adversarial examples. *arXiv preprint arXiv:1608.07690*, 2016.

[46] A. Toshev and C. Szegedy. Deeppose: Human pose estimation via deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1653–1660, 2014.

[47] F. Tramèr, A. Kurakin, N. Papernot, I. Goodfellow, D. Boneh, and P. McDaniel. Ensemble adversarial training: Attacks and defenses. *arXiv preprint arXiv:1705.07204*, 2017.

[48] D. Tsipras, S. Santurkar, L. Engstrom, A. Turner, and A. Madry. There is no free lunch in adversarial robustness (but there are unexpected benefits). *arXiv preprint arXiv:1805.12152*, 2018.

[49] J. Uesato, B. O'Donoghue, A. v. d. Oord, and P. Kohli. Adversarial risk and the dangers of evaluating against weak attacks. *arXiv preprint arXiv:1802.05666*, 2018.

[50] N. X. Vinh, J. Epps, and J. Bailey. Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *Journal of Machine Learning Research*, 11(Oct):2837–2854, 2010.

[51] X. Wang, Y. Peng, L. Lu, Z. Lu, M. Bagheri, and R. M. Summers. Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 3462–3471. IEEE, 2017.

[52] Y. Wu, H. Wang, B. Zhang, and K.-L. Du. Using radial basis function networks for function approximation and classification. *ISRN Applied Mathematics*, 2012.

[53] C. Xie, J. Wang, Z. Zhang, Z. Ren, and A. Yuille. Mitigating adversarial effects through randomization. *arXiv preprint arXiv:1711.01991*, 2017.

[54] C. Xie, J. Wang, Z. Zhang, Y. Zhou, L. Xie, and A. Yuille. Adversarial examples for semantic segmentation and object detection. *arXiv preprint arXiv:1703.08603*, 2017.

[55] W. Xu, D. Evans, and Y. Qi. Feature squeezing: Detecting adversarial examples in deep neural networks. *arXiv preprint arXiv:1704.01155*, 2017.

[56] X. Yuan, P. He, Q. Zhu, R. R. Bhat, and X. Li. Adversarial examples: Attacks and defenses for deep learning. *arXiv preprint arXiv:1712.07107*, 2018.

[57] V. Zantedeschi, M.-I. Nicolae, and A. Rawat. Efficient defenses against adversarial attacks. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pages 39–49. ACM, 2017.