INCORPORATING PERCEPTUAL PRIOR TO IMPROVE MODEL'S ADVERSARIAL ROBUSTNESS

Anonymous authors
Paper under double-blind review

ABSTRACT

Deep Neural Networks trained using human-annotated data are able to achieve human-like accuracy on many computer vision tasks such as classification, object recognition and segmentation. However, they are still far from being as robust as the human visual system. In this paper, we demonstrate that even models that are trained to be robust to random perturbations do not necessarily learn robust representations. We propose to address this by imposing a perception based prior on the learned representations to ensure that perceptually similar images have similar representations. We demonstrate that, although this training method does not use adversarial samples during training, it significantly improves the networks robustness to single-step and multi-step adversarial attacks, thus validating our hypothesis that the network indeed learns more robust representations. Our proposed method provides a means of achieving adversarial robustness at no additional computational cost when compared to normal training.

1 INTRODUCTION

One of the primary goals of a classification or object recognition pipeline in applications such as autonomous navigation is to meet human performance, and eventually outperform the same. While training Deep Neural Networks to achieve this goal, similarity to human perception is enforced only by using a human-annotated dataset. While this level of supervision is sufficient to achieve human-like accuracy, this does not necessarily result in training models that are as robust as the human visual system (Geirhos et al., 2019). Unlike humans, Deep Neural Networks get easily misled by various types of deviations in the data due to factors such as noise and spatial transformations. Augmenting the training data with images subject to such deviations has led to an improvement in the generalization of these models to random perturbations. However, it is still possible to craft imperceptible noise that can easily fool these networks (Szegedy et al., 2013), leading to the belief that the model has not learned truly robust representations. In this paper we demonstrate that existing training methods fail to incorporate knowledge on perceptual similarity into the learned feature representation of the network. While we expect that the distance between features of perceptually similar images should be lesser than that between distinct images, we demonstrate that this is not always true. This shows that models trained with augmentations such as random noise may be memorizing the true labels of the randomly perturbed samples, and not actually learning to be robust. Hence these networks do not show improved robustness to adversarial samples (Szegedy et al., 2013), which are images perturbed using engineered noise, crafted with an intention to manipulate the network's behavior.

Szegedy et al. (2013) demonstrated that adversarial samples are transferable across networks of different architectures. Adversarial samples crafted using one model can also fool other models with possibly different architectures. This transferable property of adversarial samples, enables an attacker to launch a simple black-box attack (Liu et al., 2017; Papernot et al., 2017) on models deployed in the real world. These properties of adversarial samples pose a challenge for deployment of models in critical applications such as surveillance systems and autonomous driving.

The most common methods of improving the adversarial robustness of models involve training the model with adversarial samples only, or a combination of clean and adversarial samples (Goodfellow et al., 2015; Kurakin et al., 2017; Tramèr et al., 2018; Madry et al., 2018). This requires the generation of adversarial samples on the training dataset, which is computationally expensive. Ad-

versarial samples can be generated by adjusting the input image such that cross entropy loss of the predicted output with respect to the ground truth labels (or an equivalent loss) is maximized. This maximization problem can be solved using Projected Gradient Descent (PGD, Madry et al. (2018)), which involves multiple iterations of small steps in the direction of maximum gradient, and reprojection on an l_p norm ball (typically l_2 or l_∞ is used), such that the overall perturbation is below a certain threshold. The threshold on the pixel intensity of perturbations ensures that the perturbed image is perceptually similar to the original image. The state-of-the-art method of training a robust model today is PGD Adversarial training, where the network is trained using PGD adversarial samples. However, this process is prohibitively expensive and does not scale to large datasets such as ImageNet. While it is possible to generate perturbations using single step (non-iterative) methods such as Fast Gradient Sign Method (FGSM, Goodfellow et al. (2015)), it has been shown that single-step adversarial training causes models to converge to a degenerate minima, where models appear to be robust to single step attacks (Tramèr et al., 2018). This method does not improve the robustness to iterative adversarial attacks.

We propose to use the knowledge on perceptual similarity between mildly perturbed images to learn more robust features. We claim that, by imposing similarity in the learned representations of perceptually similar images, augmentations such as random noise and spatial transformations can lead to better generalization. We validate our claim by demonstrating improved performance on PGD, FGSM, DeepFool (Moosavi-Dezfooli et al., 2016) and C&W (Carlini & Wagner, 2016) attacks in white-box and black box settings. It is to be noted that, we achieve adversarial robustness without explicitly exposing the model to adversarial samples. This shows that the model cannot be over-fitting to adversarial samples, which is one of the main drawbacks of adversarial training. Secondly, the computational cost of our proposed method is similar to that of normal training, hence leading to adversarial robustness at a very low cost.

We summarize our contributions in this paper below:

- We demonstrate that the average l_2 distance between the logits of clean and mildly perturbed images is greater than that between the logits of distinct images (from the same or different classes); leading to the conclusion that the model does not necessarily learn robust representations.
- To address this issue, we propose a regularizer to enforce similarity between the logits of perceptually similar images.

The paper is organized as follows: section 2 discusses related works, section 3 demonstrates the failure of normal training methods in incorporating perceptual similarity into the learned feature representation of the network, and further proposes a training method which addresses this issue, section 4 presents experiments to validate our claim, and section 5 concludes the paper.

2 Related works

Conventionally, deep neural networks for classification are trained on human-labelled images by minimizing the average loss of the model over the training images (Vapnik & Chervonenkis, 1971). To prevent memorization and improve generalization, techniques such as regularization (e.g., dropout, weight decay) and data augmentation (e.g., random crop, horizontal flip) are widely used.

Yet, several studies have revealed the inherent weakness of deep neural networks, to perform well on data belonging to distributions slightly different from that of the training data (Ben-David et al., 2010). Szegedy et al. (2013) showed that imperceptible, crafted noise called adversarial perturbation can fool models with very high success rate. Ever since the vulnerability of deep neural networks became apparent, there have been several attempts to improve their robustness. One of the most widely used techniques is adversarial training (Goodfellow et al., 2015), where training data is augmented with adversarial samples during training. The state-of-the-art adversarial training method proposed by Madry et al. (2018) uses a min-max formulation to find the strongest first-order adversary (PGD adversary), and minimizes the loss of the model over these adversarial samples. This method is computationally very expensive and yet achieves only $\approx 46\%$ accuracy on PGD adversarial samples for CIFAR-10 (Krizhevsky et al.) dataset. Despite all these attempts, Engstrom et al. (2019) showed that these models can still be fooled by adversarial translation and rotation.

Kannan et al. (2018) claimed to achieve SOTA adversarial robustness by adding a regularizer to the loss function, to minimize the distance between logits of clean and the corresponding adversarial samples. This approach requires the generation of adversarial samples and hence is computationally expensive. Contrary to this, we propose to improve the model's robustness by reducing the distance between the logits of perceptually similar images.

Recently, there have been works related to learning better feature representations without adversarial training to improve the generalization and robustness of deep neural networks. Guo et al. (2018) proposed sparse DNNs which are obtained by pruning dense DNNs, as a way to improve adversarial robustness without adversarial training. Input $mixup$ (Zhang et al., 2018) augments the training data with virtual training examples, generated using convex combinations of the actual training data and their corresponding labels. Verma et al. (2019) improve upon input $mixup$ by generating convex combination of input, or latent representations from any random hidden layer of the network. Though input $mixup$ works by interpolating the raw inputs, it does not take into consideration the perceptuality of the combined inputs. We demonstrate better robustness when compared to these methods.

3 Incorporating perceptual prior

In this section, we demonstrate that models trained using normal training regime does not incorporate perceptual prior into the learned feature representation. Further, to address this issue, we propose a regularizer to enforce similarity between the logits of perceptually similar images.

3.1 Feature representation of the network trained using normal training method

Consider a neural network C, that maps an input image x to class scores y, and let $y_{pred} = argmax(y)$ be the prediction of the network. Let y_{GT} represent the ground truth label of the image, x. The parameters of the neural network are learned using the loss function, J. Let m be the size of the training mini-batch, B. Ideally, perceptually similar images should have similar feature representation, and one would expect this prior knowledge to be incorporated into the learned feature representation of the network. Let, d_{per} be the average l_2 distance between the logits of clean images and their corresponding noisy images in a mini-batch. Noisy image is generated such that it is perceptually similar to its corresponding clean image. d_{intra} be the average l_2 distance between the logits of images of same class, in a mini-batch. d_{inter} be the average l_2 distance between the logits of images of different class, in a mini-batch, B.

During normal training of a network, cross-entropy loss would cause d_{intra} to become less than d_{inter}. Further, it is implicitly assumed that $d_{per} < d_{intra}$, i.e., average l_2 distance between the logits of perceptually similar images (clean and its corresponding noisy image) is lesser than that of average l_2 distance between the logits of images of the same class. This assumption is based on the observation that networks are robust to random perturbations ($||\delta_r||_p \leq \xi$, typically $p = 2$ *or* ∞ is used). In section 4.1, we empirically show that this implicit assumption ($d_{per} < d_{intra}$) is not true for the model trained using the existing normal training method. We show this by obtaining plots of d_{per} versus training iteration, and d_{intra} versus training iteration, for the model trained using normal training method (refer row-1 of Fig. 1). From the obtained plots, we observe that for the entire training duration $d_{per} > d_{intra}$ (compare column-1 and column-2 plot in row-1 of Fig. 1). Further, during PGD adversarial training method (Madry et al., 2018) (models trained using this method are robust to both iterative and non-iterative attacks) we observe $d_{per} \leq d_{intra}$ (compare column-1 and column-2 plot in row-2 of Fig. 1). This demonstrates that the perceptual prior is naturally incorporated into a model that is trained to be robust, although there is no explicit regularizer to enforce the same.

3.2 Proposed Regularizer

In the previous subsection 3.1, we demonstrated that existing normal training method fails to incorporate perceptual prior into the learned feature representation of the network. Further, we showed that this prior knowledge is incorporated into the learned feature representation of the network trained using adversarial training method. Based on this observation, we propose a normal training

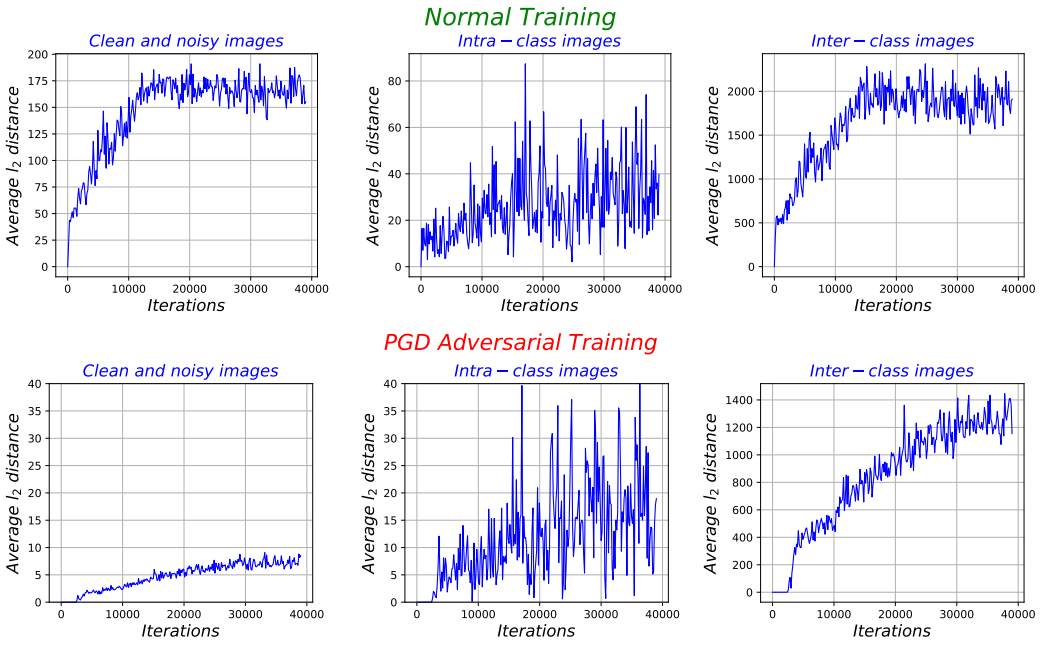


Figure 1: Column-1: Average l_2 distance between the logits of clean and its corresponding noisy images (d_{per}) versus iterations. Column-2: Average l_2 distance between the logits of images belonging to the same class (d_{intra}) in a training mini-batch versus iterations. Column-3: Average l_2 distance between the logits for images belonging to different classes (d_{inter}) in a training mini-batch versus iterations. Rows represent the training method. Row-1: Normal training. Row-2: PGD adversarial training.

method with a regularizer to enforce similarity between the logits of perceptually similar images. Eq. (1) represents the proposed total training loss, where f_{clean} and f_{noisy} represents the logits of clean and noisy images. The first term in the Eq. (1 represents the cross-entropy loss on the mini-batch containing both clean images and their corresponding noisy images, and the second term represents the proposed regularizer. The hyper-parameter λ decides the weightage given to the regularizer term. The proposed regularizer causes training loss to increase, when the distance between the logits of clean and their corresponding noisy images increases. In section 4, we empirically show that incorporating perceptual prior using the proposed regularizer results in significant improvement in the model's robustness against adversarial attacks.

$$\mathcal{L}oss = \frac{1}{2m} \sum_{s=\{clean,noisy\}} \left(\sum_{i=1}^{m} J(C(x_s^i;\theta), y_{true}^i) \right) + \lambda \sum_{j=1}^{m} \left\| f_{clean}^j - f_{noisy}^j \right\|_2^2 \tag{1}$$

Generating perceptually similar images: In this work, we generate perceptually similar image pairs by adding small random perturbation ($||\delta_r|| \leq \xi$) to the clean image using Eq. (2), where, ξ is the l_∞ norm of the random perturbation and $\mathcal{N}$ represents normal distribution. We observed that random perturbations with l_∞ norm constraint are more effective than random perturbations with l_2 norm constraint.

$$x_{noisy} = x + \delta_r; \quad where,\ \delta_r = \xi.sign(\mathcal{N}(0^d, I^d)) \tag{2}$$

The proposed regularizer encourage models to learn similar feature representation for perceptually similar images. This nature of the proposed regularizer, enable us to extend the framework to defend against other types of adversarial attacks such as spatial transformation attack (Engstrom et al., 2017). In section 4.3, we show that by generating perceptually similar image pairs using spatial transformation such as translation and rotation, results in significant improvement in the robustness of the model against transformation based attack only. Further, we observe improvement in the model's robustness against perturbation and transformation based attack if both types of perceptually similar pairs are included while training.

4 EXPERIMENTS

In this section, we provide empirical results for the observations made in section 3. Further, we show the performance of models trained on MNIST (LeCun), Fashion-MNIST (Xiao et al., 2017), and CIFAR-10 (Krizhevsky et al.) datasets using the proposed method, against adversarial attacks. Results for Fashion-MNIST dataset are shown in the appendix. We use MNIST-Network shown in table 6 for MNIST dataset, and WideResNet-28-10 (WRN-28-10) (Zagoruyko & Komodakis, 2016) for CIFAR-10 dataset. These models are trained using SGD with momentum, and step-policy is used for learning rate scheduling. For all datasets, images are normalized to be in [0,1] range. We ran our experiments on Nvidia TITAN X GPU.

We compare the proposed method with normal training, FGSM adversarial training (Goodfellow et al., 2015), Adversarial Logits Pairing (ALP) (Kannan et al., 2018), and PGD adversarial training (Madry et al., 2018) methods. Further, in order to show the effectiveness of the proposed regularizer term, we show results on three ablation experiments. (i) **Ablation-1:** train on mini-batchs containing both clean and their noisy images, with no regularizer (i.e., λ=0), (ii) **Ablation-2:** proposed training method, but noisy image is generated by adding perturbation with l_2 norm constraint, and **Ablation-3:** Ablation-2 with no regularizer (i.e., λ=0). We show results for perturbation and transformation based attacks. For l_∞ norm bounded perturbation based attacks, we use FGSM, I-FGSM and PGD attacks, and for unbounded perturbation based attacks DeepFool and C&W attacks are used. For transformation based attacks, we use the grid-search method proposed by Engstrom et al. (2017). For perturbation based attacks, we follow (Madry et al., 2018) for the attack perturbation strength (ϵ) and attack parameters.

4.1 FEATURE REPRESENTATION OF THE NETWORK TRAINED USING NORMAL TRAINING METHOD

In this subsection, we present relevant experiments to show that existing normal training method fail to incorporate perceptual prior into the learned feature representation of the network. We train MNIST-Network on MNIST dataset using normal training method. During training, we compute the feature distance metrics d_{per}, d_{intra} and d_{inter}. Row-1 of Fig. 1 shows the obtained feature distance metric plots: (i) d_{per} versus iterations (column-1), (ii) d_{intra} versus iterations (column-2), and (iii) d_{inter} versus iterations (column-3). By comparing plots of column-1 and column-2 in row-1, it can be observed that for the entire training duration $d_{per} > d_{intra}$ i.e., average l_2 distance between the logits of perceptually similar images is greater than the average l_2 distance between the logits of images of same class. Row-2 of Fig. 1, shows the obtained feature distance metric plots for the model trained using PGD adversarial training method (Madry et al., 2018). It can be observed that during training $d_{per} \leq d_{intra}$. Note that, models trained using PGD adversarial training method are robust against perturbation based adversarial attacks.

4.2 PERFORMANCE AGAINST PERTURBATION ATTACKS IN WHITE-BOX AND BLACK-BOX SETTINGS

We train MNIST-Network and WRN-28-10 on MNIST and CIFAR-10 datasets respectively using the proposed training method. We set the hyper-parameters (λ, ξ) to (10, 0.3) and (50, 8.0/255) for MNIST and CIFAR-10 datasets respectively. Further, we train these models using normal training, FGSM adversarial training, Adversarial Logits Pairing (ALP) and PGD adversarial training methods. We follow the training procedure described in the respective papers. Table 1 and 2, shows the performance of models trained using different methods, against l_∞ norm bounded perturbation attacks in white-box setting (complete knowledge of the deployed model is available for generating adversarial attack). It can be observed that there is a significant improvement in the robustness of models trained using the proposed training method, for both non-iterative (FGSM) and iterative attacks (I-FGSM and PGD). Further, it can be observed that models trained using FGSM adversarial training method are not robust to iterative adversarial attacks. Last column of table 1 and 2 shows the training time per epoch for different training methods, it can be observed that the proposed training method is faster than FGSM and PGD adversarial training methods. The slight increase in the training time of the proposed training method when compared to normal training method, is due to the inclusion of noisy samples to the training mini-batch. In appendix A.2, we show the performance of these models against unbounded perturbation attacks (DeepFool and C&W attacks).

Table 1: White-Box attack: Classification accuracy (%) of models trained on MNIST dataset using different training methods. For all attacks ϵ=0.3 is used and for PGD attack ϵ_{step} is set to 0.01. For the proposed method, mean and standard-deviation of accuracy over three runs are reported. For Ablation-2 and Ablation-3, we generate noisy image by adding perturbation with l_2 norm constraint equals to 4.

Training method	**Attack Method**					**Training time**
	Clean	*FGSM*	*I-FGSM* $steps = 40$	*PGD* $steps = 40$	*PGD* $steps = 100$	**per epoch (sec.)**
Normal training	99.27	13.84	0.39	0.03	0.00	6
FGSM adversarial training	99.45	93.02	30.90	10.02	2.94	13
PGD adversarial training	99.28	96.77	95.11	95.70	94.33	212
ALP	99.22	97.17	96.07	96.48	95.49	290
Proposed	99.10 $\pm$0.03	93.89 $\pm$0.165	89.96 $\pm$0.055	89.29 $\pm$0.082	83.48 $\pm$0.340	10
Ablation1 (Proposed, λ=0)	99.32	33.74	7.43	2.54	0.12	10
Ablation2 (l_2 *noise*, λ=10)	99.35	92.24	82.18	65.33	31.96	10
Ablation3 (l_2 *noise*, λ=0)	99.29	20.58	1.29	0.20	0.00	10

Table 2: White-Box attack: Classification accuracy (%) of models trained on CIFAR-10 dataset using different training methods. For all attacks ϵ=8/255 is used and for PGD attack ϵ_{step} is set to 2/255. For the proposed method, mean and standard-deviation of accuracy over three runs are reported. For Ablation-2 and Ablation-3, we generate noisy image by adding perturbation with l_2 norm equals to 0.2.

Training method	**Attack Method**					**Training time**
	Clean	*FGSM*	*I-FGSM* $steps = 7$	*PGD* $steps = 7$	*PGD* $steps = 20$	**per epoch (sec.)**
Normal training	94.75	28.16	0.07	0.03	0.00	194
FGSM adversarial training	94.04	98.54	0.31	0.09	0.00	419
PGD adversarial training	85.70	53.96	48.65	47.30	43.09	1524
ALP	85.11	57.46	54.28	53.75	51.07	2140
Proposed	85.81 $\pm$1.371	45.70 $\pm$1.557	38.22 $\pm$0.973	35.84 $\pm$0.777	28.50 $\pm$0.500	299
Ablation1 (Proposed, λ=0)	94.90	31.24	2.08	0.69	0.04	299
Ablation2 (l_2 *noise*, λ=50)	93.47	28.89	0.04	0.01	0.00	299
Ablation3 (l_2 *noise*, λ=0)	94.95	6.83	4.84	4.38	3.32	299

Table 3 and 4, shows the performance of models trained using PGD adversarial training method and the proposed training method against FGSM attack in black-box setting (partial or no knowledge of the deployed model is available for generating adversarial attack). It can be observed that there is no significant drop in the performance of models against adversarial attacks. Note that, source model (normally trained) is used for generating adversarial samples, and these generated samples are fed to the target model.

4.3 Performance against spatial transformation attacks

In this section, we extend the proposed training method to defend against spatial transformation attacks. Engstrom et al. (2017) demonstrated that neural networks are susceptible to spatial transformations of input image i.e., models can be fooled by rotating and translating the input image. Here, the extent of translation and rotation is constrained. Further, Engstrom et al. (2017) proposed a grid search method to find the parameters of the transformation matrix (translation along x-axis tx, translation along y-axis ty and rotation θ). We show that incorporating perceptually similar images that are generated by spatial transformations (e.g., rotation and translation), during the proposed training method results in a model that is robust to spatial transformation attacks. Further,

Table 3: Black-Box attack: Classification accuracy (%) of models trained on MNIST dataset using the proposed and PGD adversarial training methods, against FGSM attack with ϵ=0.3. A normally trained model (source model) is used for generating adversarial samples, and these samples are fed to the target model. M represents MNIST-Network and subscript denotes the training method.

Source Model	**Target Model**	
	M_{PGD}	$M_{Proposed}$
Net-A	96.07	94.95
Net-B	95.8	94.53

Table 4: Black-Box attack: Classification accuracy (%) of models trained on CIFAR-10 dataset using the proposed and PGD adversarial training methods, against FGSM attack with ϵ=8/255. A normally trained model (source model) is used for generating adversarial samples, and these samples are fed to the target model. M represents WideResNet-28-10 and subscript denotes the training method.

Source Model	**Target Model**	
	M_{PGD}	$M_{Proposed}$
VGG-11	79.24	79.24
VGG-19	83.35	81.23

Table 5: Spatial transformation attack: Performance of models trained on MNIST dataset using different training methods, against spatial transformation attack and perturbation attack i.e., PGD attack (ϵ=0.3, ϵ_{step}=0.01, $steps$=100). For spatial transformation attack, (tx, ty, θ) are constrained to ($\pm$3px, $\pm$3px, $\pm30^\circ$).

Training Method		**Perturbation attack**	**Transformation attack**
	Clean	PGD	Both rotation and translation
Normal training	99.27	0.00	21.60
FGSM adversarial training	99.45	2.94	36.04
PGD adversarial training	99.28	94.33	34.45
ALP	99.22	95.49	26.14
Proposed (only noisy samples)	99.10 $\pm$0.03	83.48 $\pm$0.34	23.25 $\pm$0.825
Proposed (only transformed samples)	98.75 $\pm$0.55	0.28 $\pm$1.35	84.14 $\pm$0.59
Proposed (both noisy and transformed samples)	98.45 $\pm$0.01	80.54 $\pm$1.36	75.99 $\pm$0.56

model's robustness improves against perturbation and spatial transformation attacks, if both, noisy and transformed images are included while training. We train MNIST-Network on MNIST dataset using the proposed training method, and during training we include (i) only noisy samples, (ii) only transformed samples, and (iii) both noisy and transformed samples. We limit $tx = ty = \pm 3$ pixels (px) and $\theta = \pm 30^\circ$ while generating spatially transformed samples. Table 5 shows the performance of models trained on MNIST dataset using different training methods, against spatial transformation attacks. It can be observed that models trained using normal, PGD adversarial training, ALP and the proposed training (only noisy samples) methods are not robust to spatial transformation attacks. Further, it can be observed that the model trained using the proposed training (only transformed samples), is robust to spatial transformation attack only. Furthermore, models trained using the proposed training (both noisy and transformed samples) shows significant improvement in their robustness against both perturbation and spatial transformation attacks. For spatial transformation attack, (tx, ty, θ) are constrained to ($\pm$3px, $\pm$3px, $\pm30^\circ$). We use grid search method proposed by Engstrom et al. (2017).

4.4 Sanity tests to detect obfuscating gradients

We obtain the following plots to detect obfuscating gradients. Models exhibiting obfuscating gradients are not robust against adversarial attacks (Athalye et al., 2018).

Plot of accuracy versus perturbation strength of PGD attack: Typically, with the increase in the

perturbation strength of an attack, the distortion in the resultant adversarial sample increases, and this causes degradation in the performance of the model. Whereas, this behavior is not observed in models exhibiting obfuscating gradients. Fig. 2 shows the plot of accuracy of the model on test set versus perturbation strength of PGD attack. It can be observed that the performance of models trained using the proposed method, degrades with increase in the perturbation strength.

Plot of loss versus perturbation strength of FGSM attack: Typically, model's loss should increase monotonically with the increase in the perturbation size of an adversarial attack, and this is not observed in models exhibiting obfuscating gradients. Fig. 3 shows the plot of average loss on test set versus perturbation strength of FGSM attack, obtained for models trained using the proposed training method. It can be observed that loss increases monotonically with increase in the perturbation strength.

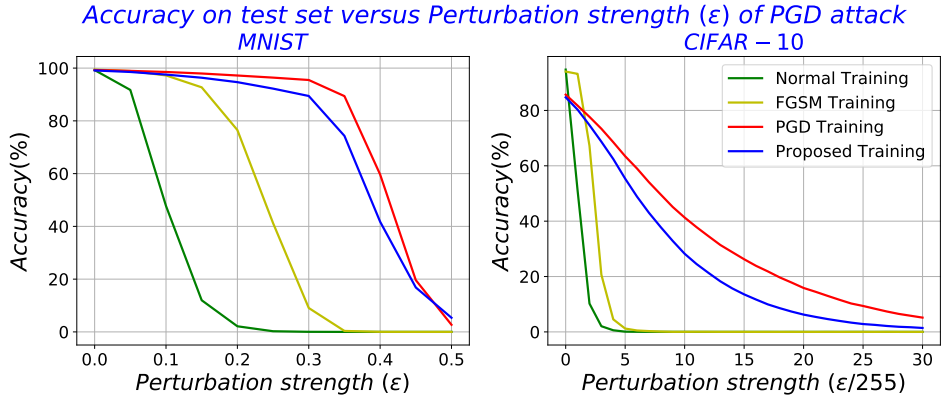


Figure 2: Plot of accuracy on test set versus perturbation strength (ϵ) of PGD attack obtained for models trained on MNIST and CIFAR-10 datasets using different training methods. PGD attack with $steps$=40 is used for MNIST, and for CIFAR-10 $steps$=7 is used. Note: Legends are the same for both the plots.

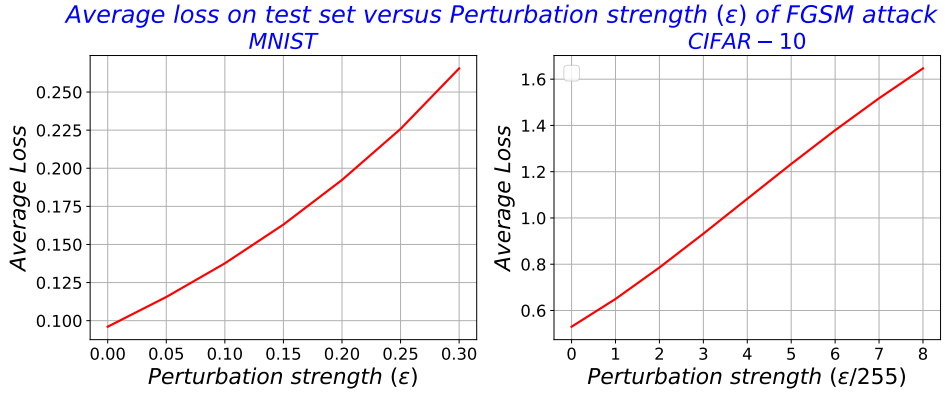


Figure 3: Plot of average loss on test set versus perturbation strength (ϵ) of FGSM attack, obtained for models on MNIST and CIFAR-10 datasets respectively using the proposed training method.

4.5 Effect of Hyper-parameters

In this subsection, we show the effect of hyper-parameter (λ) of the proposed training method. The hyper-parameter λ defines the weightage given to the regularizer term in the total training loss (Eq. 1). We train MNIST-Network on MNIST dataset, using the proposed training method with different values of λ, and after training we obtain the accuracy of the model on clean validation set and on PGD adversarial validation set (ϵ=0.3, ϵ_{step}=0.01, $steps$=100). Fig. 4 shows the effect

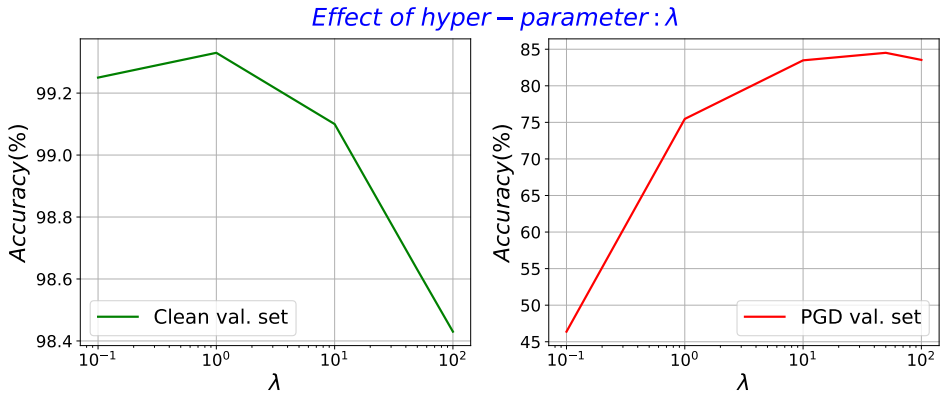


Figure 4: Effect of hyper-parameter(λ) of the proposed training method. For PGD attack, we set ϵ=0.3, ϵ_{step}=0.01 and $steps$=100. Note that x-axis is in log scale.

of varying λ from 0.1 to 100. From column-2 plot of Fig. 4, it can be observed that the model's robustness to PGD attack, initially increases with the increase in the value of λ and reaches a peak value for λ=50, and for further increase in the value of λ causes the model's robustness to decrease. At the same time, the accuracy on clean validation set, decreases with the increase in the value of λ (please refer column-1 plot of Fig. 4). Based on this trade-off between accuracy on clean and adversarial samples, we choose λ=10, to maintain good accuracy on both clean and adversarial samples.

5 Conclusion

In this work, we have demonstrated that normal training method fails to incorporate perceptual prior into the learned feature representation of the network. Further, we have proposed a training method that incorporates perceptual prior through a regularizer. The proposed regularizer causes training loss to increase when the Euclidean distance between the logits of perceptually similar images increases. The models trained using the proposed method show significant improvement in their robustness against adversarial attacks. Finally, the training complexity of the proposed method is similar to that of the normal training method.

References

Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018*, 2018.

Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine Learning*, 79(1-2): 151–175, 2010. doi: 10.1007/s10994-009-5152-4. URL https://doi.org/10.1007/s10994-009-5152-4.

Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. *arXiv preprint arXiv:1608.04644*, 2016.

Logan Engstrom, Brandon Tran, Dimitris Tsipras, Ludwig Schmidt, and Aleksander Madry. A rotation and a translation suffice: Fooling cnns with simple transformations. *arXiv preprint arXiv:1712.02779*, 2017.

Logan Engstrom, Brandon Tran, Dimitris Tsipras, Ludwig Schmidt, and Aleksander Madry. Exploring the landscape of spatial robustness. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, pp. 1802–1811, 2019. URL http://proceedings.mlr.press/v97/engstrom19a.html.

Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A. Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. In *7th International Conference on Learning Representations,*

ICLR 2019, New Orleans, LA, USA, May 6-9, 2019, 2019. URL https://openreview.net/forum?id=Bygh9j09KX.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations (ICLR)*, 2015.

Yiwen Guo, Chao Zhang, Changshui Zhang, and Yurong Chen. Sparse dnns with improved adversarial robustness. In *Advances in neural information processing systems*, pp. 242–251, 2018.

Harini Kannan, Alexey Kurakin, and Ian Goodfellow. Adversarial logit pairing. *arXiv preprint arXiv:1803.06373*, 2018.

Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-10 (canadian institute for advanced research). URL http://www.cs.toronto.edu/~kriz/cifar.html.

Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*, 2016.

Alexey Kurakin, Ian J. Goodfellow, and Samy Bengio. Adversarial machine learning at scale. In *International Conference on Learning Representations (ICLR)*, 2017.

Yann LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*.

Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. Delving into transferable adversarial examples and black-box attacks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017. URL https://openreview.net/forum?id=Sys6GJqxl.

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

Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: A simple and accurate method to fool deep neural networks. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, 2016.

Nicolas Papernot, Patrick D. McDaniel, Ian J. 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, AsiaCCS 2017, Abu Dhabi, United Arab Emirates, April 2-6, 2017*, pp. 506–519, 2017. doi: 10.1145/3052973.3053009. URL https://doi.org/10.1145/3052973.3053009.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations (ICLR)*, 2013.

Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Dan Boneh, and Patrick McDaniel. Ensemble adversarial training: Attacks and defenses. In *International Conference on Learning Representations (ICLR)*, 2018.

V. N. Vapnik and A. Y. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications*, XVI(2):264–280, 1971.

Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, David Lopez-Paz, and Yoshua Bengio. Manifold mixup: Better representations by interpolating hidden states. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, pp. 6438–6447, 2019. URL http://proceedings.mlr.press/v97/verma19a.html.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017.

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

Hongyi Zhang, Moustapha Cissé, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*, 2018. URL https://openreview.net/forum?id=r1Ddp1-Rb.

A APPENDIX

Table 6: Network used for MNIST dataset. Net-A and Net-B are used for generating black-box attacks.

MNIST-Network	**Net-A**	**Net-B**
{conv(32,5,5) + Relu}×2	Conv(64,5,5) + Relu	Dropout(0.2)
MaxPool(2,2)	Conv(64,5,5) + Relu	Conv(64,8,8) + Relu
{conv(64,5,5) + Relu}×2	Dropout(0.25)	Conv(128,6,6) + Relu
MaxPool(2,2)	FC(128) + Relu	Conv(128,5,5) + Relu
FC(512) + Relu	Dropout(0.5)	Dropout(0.5)
FC + Softmax	FC + Softmax	FC + Softmax

A.1 ADVERSARIAL ATTACKS

Following are the adversarial attacks considered in the paper.

Fast Gradient Sign Method (FGSM): Single-step adversarial attack proposed by Goodfellow et al. (2015). Adversarial image is generated by perturbing the clean image in the direction of the sign of the gradient of loss with respect to the input image. The extent of perturbation is controlled by ϵ.

$$x^* = x + \epsilon.sign\big(\nabla_x J(C(x;\theta), y_{GT})\big) \tag{3}$$

Iterative Fast Gradient Sign Method (I-FGSM): Iterative version of FGSM attack, proposed by Kurakin et al. (2016). Where, $\epsilon_{step} = \epsilon/steps$.

$$x^0 = x; \quad x^{N+1} = x^N + \epsilon_{step}.sign\big(\nabla_{x^N} J(C(x^N;\theta), y_{GT})\big) \tag{4}$$

Projected Gradient Descent (PGD): Proposed by Madry et al. (2018). First, the image is perturbed with a small random noise and then IFGSM is applied with re-projection.

DeepFool: Proposed by Moosavi-Dezfooli et al. (2016). This method finds the minimal perturbation(δ^*) that is required to fool the classifier.

$$\begin{aligned} \delta^* &= \underset{\delta}{\operatorname{argmin}} \big\|\delta\big\|_2^2 \\ \text{s.t.} \quad & C(x+\delta) \neq C(x) \end{aligned} \tag{5}$$

Carlini & Wagner Attack: Proposed by Carlini & Wagner (2016), to evaluate the model's robustness. The minimal l_∞ perturbation(δ^*), is obtained by solving the following optimization problem.

$$\delta^* = \underset{\delta}{\operatorname{argmin}} \big\|\delta\big\|_\infty + c.f(x+\delta)$$

$$\text{s.t.} \quad x + \delta \in [0,1]^n$$

where f is an objective function such that $C(x+\delta) = t$ if and only if $f(x+\delta) \leq 0, C(x) \neq t$ and the constant c is chosen as the minimum c for which $f(x+\delta^*) \leq 0$

A.2 PERFORMANCE AGAINST UNBOUNDED ATTACKS

Unbounded attacks such as DeepFool and C&W generates minimum perturbation δ that is required to fool the classifier, and robustness of the model is measured in terms of average l_2 norm of the generated perturbations. Table 7 and 8 shows the performance of models trained on MNIST and CIFAR-10 datasets using different training methods, against DeepFool and C&W attacks. For DeepFool attack we set steps=100, and for C&W attack we set iterations=1000, binary search=4. It can be observed that for models trained using normal training and FGSM adversarial training methods, perturbations with small l_2 norm is sufficient to fool the classifier. Whereas, for the models trained using the PGD and the proposed adversarial training methods, perturbations with relatively large l_2 norm is required to fool the classifier.

Table 7: Performance of models trained on MNIST dataset using different training methods, against DeepFool and C&W attack. Fooling rate (FR) defines the percentage of test set images that are misclassified by the model, post attack. Mean l_2 is the average l_2 norm of perturbations generated for test set images. For robust models, Mean l_2 should be large i.e., perturbation with large l_2 norm is required to fool the classifier. Note that for models trained using PGD method and the proposed method, perturbations with large l_2 norm is required to fool the classifier.

Training method	**DeepFool**		**C&W**	
	Fooling Rate	Mean l_2	Fooling Rate	Mean l_2
Normal training	99.31	1.5696	100	1.4114
FGSM adv. training	99.49	3.3539	100	1.8044
PGD adv. training	95.21	12.0685	100	3.7438
Proposed	95.50	8.5979	100	2.6991

Table 8: Performance of models trained on CIFAR-10 dataset using different training methods, against DeepFool and C&W attack. Fooling rate (FR) defines the percentage of test set images that are misclassified by the model, post attack. Mean l_2 is the average l_2 norm of perturbations generated for test set images. For robust models, Mean l_2 should be large i.e., perturbation with large l_2 norm is required to fool the classifier. Note that for models trained using PGD method and the proposed method, perturbations with large l_2 norm is required to fool the classifier.

Training method	**DeepFool**		**C&W**	
	Fooling Rate	Mean l_2	Fooling Rate	Mean l_2
Normal training	96.33	0.2019	100.00	0.1218
FGSM adv. training	95.78	0.2529	100.00	0.1077
PGD adv. training	92.19	1.2284	97.80	0.6068
Proposed	90.08	2.6984	80.47	0.4539

Table 9: White-Box attack: Classification accuracy (%) of models trained on fashion-MNIST dataset using different training methods. For all attacks ϵ=0.1 is used and for PGD attack ϵ_{step} is set to 0.01.

Training method	**Attack Method**					**Training time**
	$Clean$	$FGSM$	$I\text{-}FGSM$ $steps = 40$	PGD $steps = 40$	PGD $steps = 100$	**per epoch (sec.)**
Normal training	91.6	13.37	0.66	0.0	0.0	13
FGSM adversarial training	92.5	90.58	26.04	16.09	15.67	28
PGD adversarial training	87.25	81.76	80.68	79.77	79.74	230
Proposed	89.007 $\pm$0.060	70.66 $\pm$0.072	67.073 $\pm$0.096	63.687 $\pm$0.074	63.27 $\pm$0.026	19

A.3 Results on fashion MNIST dataset

We train MNIST-Network on Fashion-MNIST dataset using the proposed training method. We set the hyper-parameters (λ, ξ) to (10, 0.1) . Further, we train these models using normal training, FGSM adversarial training and PGD adversarial training methods. We follow the training procedure described in the respective papers. Table 9 shows the performance of models trained using different methods, against l_∞ norm bounded perturbation attacks in white-box setting. It can be observed that there is significant improvement in the robustness of models trained using the proposed training method, for both non-iterative (FGSM) and iterative attacks (I-FGSM and PGD).

Table 10: Black-Box attack: Target models trained on **Fashion MNIST dataset**. The FGSM adversarial samples ($\epsilon = 0.1$) required for the attack are generated from undefended pre-trained Net A and B (refer table 6). For the target model, subscript denotes the training method and M represents MNIST-Network table (refer table 6)

Source Model	**Target Model**	
	M_{PGD}	$M_{Proposed}$
Net-A	84.5	80.64
Net-B	83.58	77.52

Figure 5: Plot of accuracy on test set versus perturbation strength (ϵ) of PGD attack obtained for models trained on Fashion-MNIST dataset using different training methods. PGD attack with $steps$=40 is used.

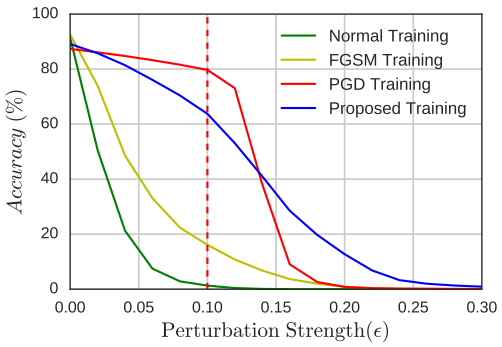
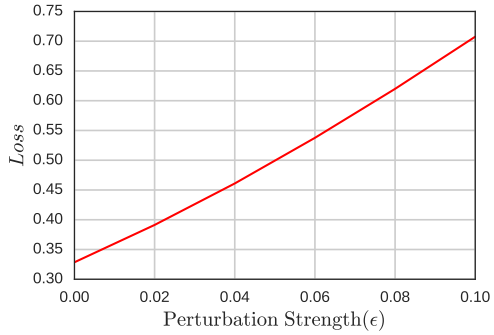


Figure 6: Plot of average loss on test set versus perturbation strength (ϵ) of FGSM attack, obtained for the model trained using the proposed training method on Fashion-MNIST dataset.



B SPATIAL TRANSFORMATION ATTACKS

Engstrom et al. (2017) investigated the robustness of the model against spatial transformations of inputs such as rotation or translation. The task of the adversary is to find (tx, ty, θ) for every input image x such that performance of the model degrades. Here,

- θ is the angle of rotation of x about the center.
- (tx, ty) denotes translation of x along x-axis and y-axis

Therefore, adversary transforms every pixel (px, py) of x to (px', py') as follows:

$$\begin{bmatrix} px' \\ py' \end{bmatrix} = \begin{bmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{bmatrix} \cdot \begin{bmatrix} px \\ py \end{bmatrix} + \begin{bmatrix} tx \\ ty \end{bmatrix}$$

An attacker can follow a variety of ways to find adversarial spatial transformations (tx, ty, θ):

1. Grid / Exhaustive Search: The adversary selects the transformation from an exhaustive, discrete grid of (tx, ty, θ) within threat model constraints for which the model performance is worst. Engstrom et al. (2017) showed this method is more effective than First Order Search and Random Search methods.
2. Worst-of-k / Random Search: The adversary chooses parameters from k arbitrarily sampled points from valid attack space that deal maximum damage to the model's performance.
3. First Order Search: The adversary starts from random parameters and moves towards the direction of gradient that maximizes the loss function. This approach is analogous to iterative attacks in ℓ_∞ space, but here the adversary optimizes in (tx, ty, θ) space rather than pixel space. However, this method is less effective when compared to the Grid Search.