Defendr: A Robust Model for Image Classification

Apoorva Dornadula
Stanford University
apoorvad@stanford.edu

Ashwini Pokle
Stanford University
ashwini1@stanford.edu

Akhila Yerukola
Stanford University
akhilay@stanford.edu

1. Abstract

Current image classification models are susceptible to being fooled by adversarially perturbed images, or adversarial examples. In this paper, we present Defendr, a robust image classification model that is able to predict the correct class of adversarial examples as well as unperturbed images with high accuracy. Defendr takes an image as input, which is then pre-processed and de-noised (potentially adversarial noise is removed). Then the image is passed through a ResNet-101 [5] and fully connected layer to output final class predictions. Different loss functions are used to encourage the model to represent an image and its adversarial counterpart in a similar manner. Images are zero-padded and resized during inference time before being passed into the model. Defendr is trained on a subset of the ImageNet dataset using 201 classes, with 70 images in each class. There are 10854 training images, 1206 validation images and 2010 test images. From our experiments, we found that the model with a Guided Denoiser [10] performed the best against adversarial and unperturbed images, achieving an accuracy of 86.60% (on a fully adversarial test set) and 92.4%, respectively. Among the ensembled models we used, the one using the Guided Denoiser, Adversarial Logit Pairing [7], and Clean Logit Pairing [7] performed the best. We were also able to visualize how the activation of the correct label class is shifted off the object when recognized incorrectly.

2. Introduction

Deep learning is being used for various computer vision tasks today. One of the more popular applications is the use of computer vision for autonomous vehicles (AV). AV aim to understand their environment by interpreting images and scenes around them. One of the biggest concerns surrounding the deployment of these autonomous vehicles is whether they are robust to adversarial or tampered sensory inputs. A scenerio where this would be life threatening is if a street sign was modified in such a way that it was either ignored or misunderstood by the autonomous vehicle. There is past work that has studied adversarial alterations of road signs [2] which is a step forward to making AV safer. The methods outlined in this paper can be applied to AV vision systems to increase its robustness to adversarial examples.

In this paper we introduce Defendr, an image detection classifier robust to adversarial examples. The input to Defendr is an image, either unperturbed or adversarial. The image gets processed by a Guided Denoiser [10], passed through a partially pretrained ResNet [5], and finally through a fully connected layer. Loss functions introduced in [7] are used to train our model. Preprocessing techniques such as those described in [24] are also used. We ensemble model ablations to improve performance. The output of our model is a class prediction for each input image.

We find that the predicted outputs from Defendr are more robust to adversarial examples, without any drop in accuracy when tested against original unperturbed images.

In Section 3, we discuss work related to object detection, adversarial attack generation, and adversarial attack defenses. In Section 4, we outline the methods used in Defendr. In Section 5, we discuss the dataset we are using to train and evaluate our model, as well as preprocessing steps. Section 6 discusses experiments we ran on Defendr and an analysis of our model. We conclude in Section 7 and 8, providing tracks for future work that can be done in this space.

3. Related Work

3.1. Object Recognition

Ever since the ImageNet [19] dataset has been released, there have been many high performing object recognition architectures proposed such as [8, 21, 18], to name a few. These works use convolutional models among other techniques to exact features and predict object classes. However, these architectures do not account for adversarially crafted inputs that are generated with the purpose of fooling the object detector.

3.2. Adversarial Attacks

Adversarial attacks involve using examples which have a small difference from the clean images to cause the classifier to predict class labels incorrectly. Let x be

the clean image. With a sufficiently small perturbation magnitude ϵ we obtain x^* such that $y_{x^*} \neq y_x$.

Fast Gradient Sign Method (FGSM): Goodfellow et al. [3] proposed a simple adversarial attack which finds the adversarial perturbation that yields the highest increase of the linear cost function under l∞-norm. The update equation is:

$$x^{adv} = x + \epsilon.sign(\bigtriangledown_x L(x, y^{true}; \theta))$$

where ϵ controls the magnitude of the adversarial perturbation.
FGSM computes the gradient once, unlike the L-BFGS[22] and thus is more efficient. It uses the true label $y = y_{true}$ while computing the gradients. Multiple variations of the FGSM attack have been proposed which try to combat the 'label leaking' effect in the adversarially generated images. A better approach to prevent label leaking is by utilizing the model predicted class y_x instead of the ground truth label y_{true}.

Projected Gradient Descent (PGD): The paper [11] claims that PGD acts as universal "first order adversary" i.e the strongest attack utilizing the local first order information about the network. PGD is essentially a multi-step variant of FGSM. The update equation is:

$$x_{t+1} = \Pi_{x+S}\Big(x_t + \alpha.sign(\bigtriangledown_x L(x, y; \theta))\Big)$$

The paper talks about a concrete guarantee that an adversarially robust model must satisfy. For each data point x, the set of allowed perturbations is $S \in R^d$. With regard to image classification, S is chosen to capture perceptual similarity between images. Since we allow the adversary to perturb the images before feeding into the model, it gives rise to the following saddle point problem:

$$min_\theta \;\; \rho(\theta) \text{ where } \rho(\theta) = E_{(x,y)\ D}\Big[max_{\delta \in S} L\big(\theta, x+\delta, y\big)\Big]$$

where D is the data distribution over the pairs of examples $x \in R^d$ and corresponding labels $y \in [k]$.
Gradient descent directions for saddle point problem:

- Compute the gradient of the loss function at a maximizer of the inner problem.
- This corresponds to replacing the input points by their corresponding adversarial perturbations and normally training the network on the perturbed input.
- This is a valid descent direction for the saddle point problem by Daskin's theorem

3.3. Adversarial Defenses

Adversarial Training:

Adversarial training is one of the most commonly used defense techniques against adversarial attacks [3, 9, 23]. The aim of adversarial training is to make the model more robust since the training data is augmented with adversarially perturbed data. It improves classification accuracy of the target model on adversarial images [3, 9, 22, 23]. However, adversarial training is more time consuming than training only on clean images since the adversarial images generations needs extra computation, also it takes more epochs to fit the adversarial images. Due to this, harder attacks are not used in adversarial training.

Preprocessing:

Preprocessing based techniques are often preferred since they process the perturbed images with certain transformations to remove the adversarial noise. Gu and Rigazio [4] first propose the use of denoising auto-encoders as a defense. Osadchy et al. [13] apply a set of filters to remove the adversarial noise by using filters such as median filer and Guassian low-pass filter. Das et al. [1] preprocess images with JPEG compression to reduce the effect of adversarial noises.

Gradient Masking Effect:

Another class of adversarial defenses are based on gradient masking effect [15, 16, 23]. These defenses apply regularizers or smooth labels to make the model more robust. Gu and Rigazio [4] propose a deep contrastive network which uses a layer-wise loss term to achieve model robustness. Papernot et al. [17] adapts knowledge distillation [6] to adversarial defense, and use the output of another model as soft labels while training the target model. . Nayebi and Surya [12] use saturating networks for robustness to adversarial noises. The loss function encourages the activations to be in their saturating regime. This approach of gradient masking is useful to combat white-box adversarial examples making it harder to construct them. However they suffer from black-box attacks generated on other models.

The previous work outlined in this section improve specific parts of the image classification pipeline (preprocessing, loss functions, inference, etc). Defendr uses recently proposed architectures that improve the model's robustness to adversarial images along the entire pipeline.

4. Methods

Defendr uses many techniques to be robust against adversarial examples. We start with introducing randomness in the dataset by including adversarial examples. We also implement novel and high performing architectures during training as well as during inference.

4.1. Dataset Alterations for Adversarial Training

Adversarial training involves augmenting the training dataset with adversarial examples and training a model on this augmented dataset. This method was was used by GoodFellow et. al. [3] and Kurankin et. al. [9] to improve robustness of model. We augment our data set with adversarial examples generated by FGSM and PGD based attacks of varying levels of perturbation ϵ for adversarial training.

4.2. Training

4.2.1 Guided Denoiser

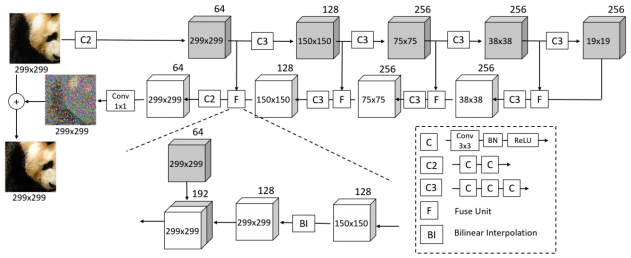

Figure 1. DUNET Architecture [10]

We follow the general approach of denoising adversarial examples before sending them to the target model as proposed in [10]. The paper has two interesting ideas with regard to denoising adversarial images as shown in Figure 1:

- To address the amplification of the small residual perturbation to the top layers of the model, remaining from the standard denoiser which uses pixel-level reconstruction loss function, the paper proposes using the reconstruction loss function as the difference between top level outputs of the target model induced by original and adversarial examples. The denoiser trained by this loss function is called "high-level representation guided denoiser" (HGD)
- A convolutional version of Denoising autoencoder (DAE) called the DUNET (DAE with a U-net) is used to better represent the fine-scale information necessary for reconstructing high resolution images by using additional lateral connection from the encoder layers to the corresponding decoder layers as shown in Figure 2.

4.2.2 Adversarial Logit Pairing

We use adversarial logit pairing as proposed in [7]. Adversarial logit pairing (ALP) adds an extra regularization term to the objective function to match the logits of a clean image x and its corresponding adversarial image $\tilde{x}$. This guides the model towards better internal representation of the data.

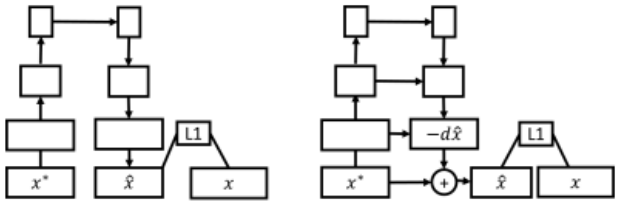

Figure 2. Diagrams of DAE (left) and DUNET (right) [10]

Adversarial logit pairing minimizes the following objective loss function

$$J(M,\theta) + \lambda \frac{1}{m} \sum_{i=1}^{m} L(f(x(i);\theta) f(\tilde{x}^{(i)},\theta))$$

We use another loss function proposed in [7] called Clean Logit Pairing (CLP). In clean logit pairing, the logits of unperterbed images are encouraged to be similar following the below loss function:

$$J(M,\theta) + \lambda \frac{2}{m} \sum_{i=1}^{m/2} L(f(x(i);\theta) f(x(i+\frac{m}{2}),\theta))$$

The success of clean logit pairing suggests that the model predicts logits of small magnitude to prevent it from becoming overconfident. To encourage smaller logits, we use another loss term (also introduced in the original paper) called clean logit squeezing which penalizes the l2-norm of the logits.

4.3. Inference

We used randomization at inference time to mitigate the adversarial effects as proposed in [24]. The paper proposes two randomization operations at inference time as shown in Figure 3:

- Random resizing: Resizes the input images to a random size
- Random padding: Pads zeros around the input images in a random manner

The paper shows that utilizing this proposed randomization method along with an adversarially trained model performs better than a model which has just adversarial training alone.
The proposed method has several advantages:

- Randomization at inference time makes the model more robust to single-step and iterative attacks (both black box and white box attacks)

- No additional training is required
- Nearly no run time increase is observed due to the randomization layers

The general intuition behind this approach is that low-level image transformations, e.g., resizing, padding, compression, etc, may probably destroy the specific structure of adversarial perturbations, thus making it a good defense. Along with that, during white box iterative attacks, the test image undergoes random transformations and thus the attacker does not know the specific transformation when generating adversarial noise.

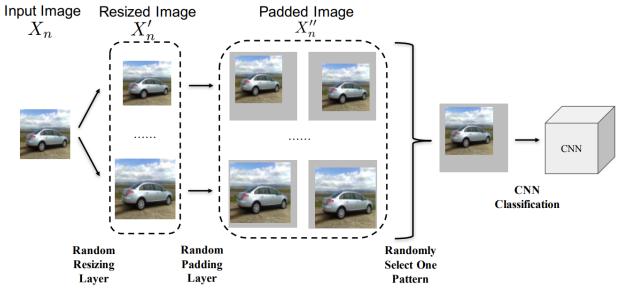

Figure 3. Randomization-based defense mechanism [cite]

4.4. Model Ensemble

We experiment with ensembling different models to help increase the robustness of our image classifier. For each model included in the ensemble, we gather its class predictions for each image. Then, we normalize the scores such that the row sum of each model's prediction matrix is 1. We must do this normalization to ensure equal contribution of all models to the final score prediction. Each model's normalized score predictions are summed to get class predictions for each image (in the batch). Finally, we predict the class of the image to be the index of the highest scoring class. This is one approach to ensembling. Many other ensembling methods can be tried which we mention in Section 8.

5. Dataset and Features

We used the ImageNet [19] dataset to train our robust image detection model.
We used a subset of ImageNet consisting of 14070 color images in 201 classes, with 70 images per class. There are 10854 training images, 1206 validation images and 2010 test images. All images are reshaped to 299x299. The mean across all three channels is subtracted and then normalized before being fed into the classifier. The red, green, and blue channels are normalized to have mean 0.485, 0.456, 0.406 and standard deviation 0.229, 0.224, 0.225, respectively. These values match the normalization method that Pytorch used to pretrain its ResNet models.

Figure 4. Original Image from ImageNet

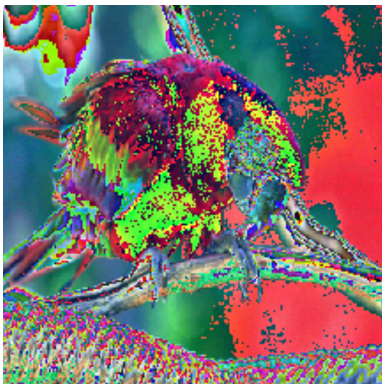

Figure 5. Image in 4 resized to 299 x 299 and normalized.

Adversarial inputs for all images corresponding to the dataset were generated. We used the Clever Hans Python library(https://github.com/tensorflow/cleverhans) [14] to generate adversarial examples to train on. The adversarial images were generated using the Fast Gradient Sign Method (FGSM) [3] with an attack step size $\epsilon = 0.06$, $\epsilon = 0.12$ and $\epsilon = 0.3$, Projected Gradient Descent Attack (PGD) [11] with an attack step size $\epsilon = 0.06$, $\epsilon = 0.12$ and $\epsilon = 0.3$. The mean across all three channels is subtracted and then normalized for all the adversarial images generated before being fed into the classifier as explained before.

Note: When performing adversarial training, the data set is augmented with the corresponding perturbed images, thus the training, validation and testing data sets are doubled.

6. Experiments

6.1. Defendr Performance

We implemented CLP [7], ALP [7] and DUNet [10] in Defendr. The performance of each of these individual

Figure 6. Adversarial image generated using FGSM with $\epsilon = 0.12$

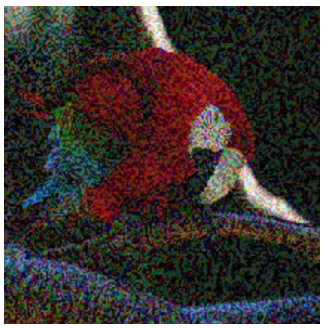

Figure 7. Adversarial image generated using FGSM with $\epsilon = 0.3$

Figure 8. Adversarial image generated using PGD with $\epsilon = 0.12$

Figure 9. Adversarial image generated using PGD with $\epsilon = 0.3$

models have been summarized in Table 1. We obtain the best performance with DUNet architecture which achieves 86.6% and 72.5 % accuracy against FGSM attacks with ϵ = 0.12 and 0.3 respectively. It achieves an accuracy of 85.8% and 79.0% on PGD attacks. The DUNet performs better than ALP because it takes into account the difference in representations between the non-adversarial and adversarial image at multiple locations in the model, rather than just 1, like ALP does.

6.2. Implementation Details

We used PyTorch as the main deep learning library. We used the Adam optimizer with learning rate of 0.001 to train all the models. The models were trained for 15-20 epochs using a batch size of 128 (for baseline, ALP, CLP) or 4 (DUNet). We used pretrained Resnet-101 model. As our images were 299X299 instead of 224X224, we increased the dimensions of the last average pooling layer from 7 to 10, and retrained this layer and the fully-connected layer. This retrained Resnet-101 model was then reused in all the experiments.

6.3. Class Visualizations

Figure 10 shows the original non perturbed image of a 'parrot'. This image was correctly classified by Defendr. We utilized the Gradient-weighted Class Activation Mapping (Grad-CAM) [20] to visualize the activation heat map to obtain intuition about the contextual cues used by the classifer to make its prediction. Grad-CAM uses the gradients of any target concept, flowing into the final convolutional layer to produce a coarse localization map highlighting the important regions in the image for predicting the concept. Using Grad-CAM we applied the heat map on the image shown in Figure 10 to obtain Figure 11. Figure 11 shows us that Defendr accurately predicts its class label by primarily looking at the parrot's face.

Figure 10. Original Non-perturbed Image

6.4. Error Analysis

Top 10 classes with highest & least prediction accuracy: To better understand the performance of Defendr, we first obtain the top 10 classes with highest prediction accuracy

Single Architecture Performance (Adv. Ex.: FGSM with $\epsilon = 0.12$)				
		No Adv. Ex.	50% Adv. Ex.	100% Adv. Ex.
Baseline	No Adv. Train	90.70	78.46	66.22
	Adv. Train	90.40	**80.70**	70.00
CLP	No Adv. Train	90.85	78.66	66.47
	Adv. Train	-	-	-
ALP	No Adv. Train	-	-	-
	Adv. Train	90.75	80.65	70.55
DUNET	No Adv. Train	-	-	-
	Adv. Train	**92.4**	-	**86.6**
Randomization	No Adv. Train	85.92	75.55	65.82
	Adv. Train	85.47	76.69	68.01

Table 1. Assessing Single Architecture Performance. The right column shows model accuracy when tested on a unperturbed test set. The middle column shows model accuracy when half the test set contains adversarial examples. The left column contains exclusively adversarial examples.

Attack Effectiveness (Tested 100% Adv. Ex.)				
	FGSM ($\epsilon = 0.12$)	FGSM ($\epsilon = 0.3$)	PGD ($\epsilon = 0.12$)	PGD ($\epsilon = 0.3$)
Baseline	70.00	47.81	80.00	63.88
ALP	70.55	48.91	80.95	62.64
DUNET	**86.6**	**72.5**	**85.8**	**79.0**

Table 2. Assessing attack performance

Model Ensemble Accuracy	
	Accuracy
ALP + CLP	66.22
DUNET + CLP	77.71
DUNET + ALP	77.71
DUNET + ALP + CLP	84.86

Table 3. Ablative analysis of the models

Model Robustness Tested 100% Adv. Ex.		
	FGSM	PGD
Baseline FGSM (0.12)	71	80
Baseline PGD (0.12)	69.35	81
ALP FGSM (0.12)	69.55	79.40
DUNET FGSM(0.12)	86.6	82.9
DUNET FGSM (0.3)	72.5	68.0
DUNET PGD (0.12)	70.9	85.8
DUNET PGD (0.3)	49.7	79.0

Table 4. Model Robustness against other attack. Numbers in brackets indicate values of perturbation.

(Table 5) and the top 10 classed with least prediction accuracy (Table 6).

Model Robustness:
We tested model against attacks that were different from

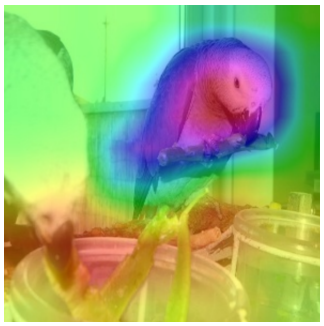

Figure 11. Class activations produced by Defendr on 10

the ones it was trained on. The results have been summarized in Table 4. We observe that models trained on PGD are more robust compared to the models trained on FGSM. We also observe that FGSM is a strong attack and causes drastic decrease in performance of the model at high perturbations. We observed that accuracy of DUNet trained on PGD attacks reduced to 49.7% from 75.9% on increasing perturbation of FGSM attack images from 0.12 to 0.3.

We also observe that for a given attack class, changing perturbation causes change in accuracy of the model. For example, DUNet trained on FGSM with perturbation 0.3 achieved accuracy of 66.2% against attacks from FGSM with perturbation 0.12, which is a drop of 5%. This indicates that model attunes to the perturbation level of the

dataset. However, this is not observed in the case of models trained on PGD images. This further indicates that models trained with PGD images tend to be more robust than those trained on FGSM images.

Confusion matrix:

Figure 15 shows the confusion matrix of the predictions from Defendr where the y-axis represents the predicted labels of the clean images, and the x-axis represents the predicted class labels of the corresponding perturbed images. The diagonal of the matrix is the correct predictions. There are no visible patterns except the fact that there are some noisy clusters.

For example: there is a noisy square cluster between the classes 150 - 200. Most of the classes between 150 and 200 represent some breed of a dog. Hence as the classifier is bound to get confused between the different breeds of the dogs, there exists a cluster there.

Gradient Class Activation Maps:

We visualized the Grad-CAM[20] - greyscale and heat maps for some of the incorrectly classified adversarially generated images.

The leftmost image of Figure 12 shows the original clean image of the dog of a certain breed. The middle image shows the greyscale map and heat maps obtained when the original clean image is fed into the classifier. The right most image shows the greyscale map and heat maps obtained when the perturbed/adversarial image is fed into the classifier which results in an incorrect class prediction wherein the classifier predicts that the image is a dog of another breed. We notice from the middle image that the classification score is majorly determined by the face and body of the dog. When the image is slightly perturbed, it results in complete inversion of the heat map (and greyscale map). Since the class prediction for the perturbed image was a dog of another breed, we notice that the classifier does understand that both the images are indeed dogs, but however could not predict the correct breed.

Similarly in Figure 13, the left most image is of a crustacean. The middle image represents the greyscale and heat maps when the clean original image is fed into the classifier. The right most image represents the greyscale and heat maps when the perturbed image is fed into the classifier which predicts it as a fish. Here also we notice that the classifier understands that a crustacean and a fish belong to the same family.

Similarly in Figure 14, the left most image is of a parrot. The middle image represents the greyscale and heat maps when the clean original image is fed into the classifier. The right most image represents the greyscale and heat maps when the perturbed image is fed into the classifier which predicts it as a rooster. Here also we notice that the classifier understands that a parrot and a rooster belong to the same family of birds.

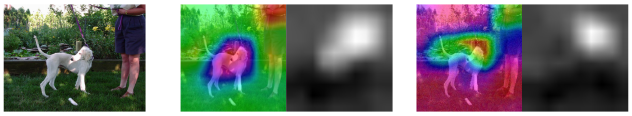

Figure 12. *Left:* dog of breed 1 image, *Middle:* class activations on original dog breed 1 image, *Right:* class activations on perturbed dog image; prediction: dog of breed 2

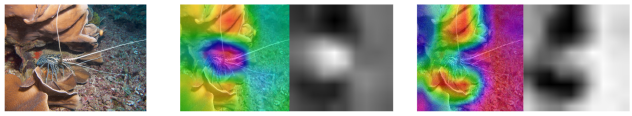

Figure 13. *Left:* crustacean image, *Middle:* class activations on original crustacean image, *Right:* class activations on perturbed crustacean image; prediction: fish

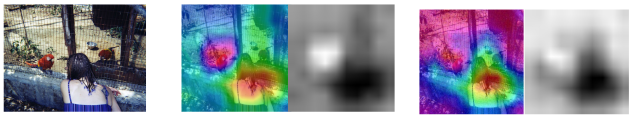

Figure 14. *Left:* parrot image, *Middle:* class activations on original parrot image, *Right:* class activations on perturbed parrot image; prediction: rooster

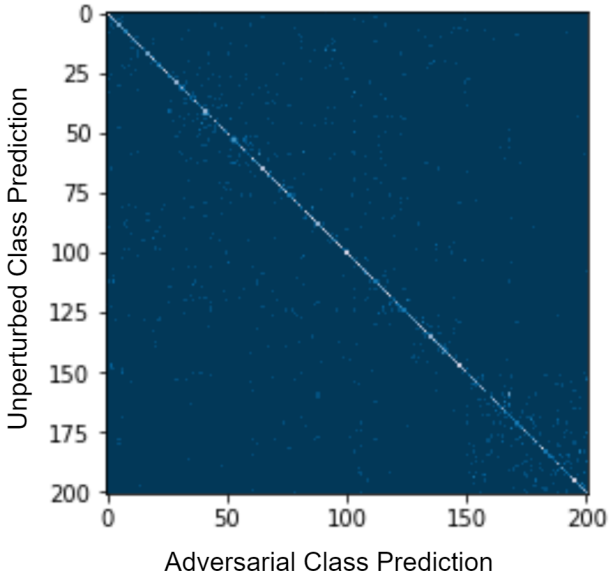

Figure 15. Confusion matrix between predictions on the unperturbed image and the adversarial image

Top 10 Best Predicted Classes		
Class Label	Class	Accuracy
83	Prairie chicken, Grouse, Fowl	100.00%
105	Koala, Kangaroo bear	100.00%
117	chambered nautilus, nautilus	100.00%
1	Goldfish, Carassius auratus	100.00%
182	Border terrier	90.91%
75	Black widow	90.91%
136	European gallinule	90.00%
90	Lorikeet	90.00%
133	Bittern	90.00%
24	Great grey owl	90.00%

Table 5. Top 10 Best Predicted Classes

Top 10 Worst Predicted Classes		
Class Label	Class	Accuracy
180	American pit bull terrier	0.00%
52	Thunder snake, Worm snake	0.00%
189	Lakeland terrier	0.00%
159	Rhodesian ridgeback	0.00%
36	Terrapin	0.00%
82	Ruffed grouse, partridge	0.00%
44	Alligator lizard	0.00%
58	Water snake	8.33%
150	Sea lion	10.00%
16	Bulbul	10.00%

Table 6. Top 10 Worst Predicted Classes

7. Conclusion

In this paper we presented Defendr, a robust image detection classifier for adversarial images. The proposed model involves multiple components. The data set is augmented with clean original images and their corresponding perturbed images generated using Projected Gradient Descent and Fast Gradient Sign Method (with varying levels of perturbation). The input to Defendr is an image, either perturbed or unperturbed. The image is preprocessed via a Guided denoiser framework [10] which 'denoises' the image. This preprocessed image is passed through a ResNet which has an additional loss component to additionaly combat perturbations in the images [7]. Preprocessing techniques such as those described in [24] are also used. We ensemble models to improve performance and notice that the final model ensemble with a Guided Denoiser, Adversarial Logit Pairing, and Clean Logit Pairing performs the best against adversarial images, without compromising the performance accuracy on the original clean images.

8. Future Work

There remains a lot to experiment with to continue improving our model. Tramer et. al. [23] report that with adversarial training as mentioned in the Section 4.1, the model still remains vulnerable to black box attacks. They suggest ensemble adversarial training where the training data is augmented with perturbations transferred from other static-pretrained models. This decouples generation of adversarial examples from the model being trained and improves robustness of the model. We aim to implement this approach and verify its performance.

Another interesting experiment we would want to try out is the usage of Variational Autoencoder. We assume that an original image and its adversarial counterpart will the have same latent features that can be used for more robust classification.

Finally we also want to experiment with different image preprocessing techniques to combat the perturbations. We believe that transforming images to dull outstanding features during testing and/or training will help the model's robustness. We want to observe the effects of transformations on images, such as Gaussian blurring, before passing images through our network.

9. Contributions & Acknowledgements

We would like to thank the course staff of CS231n for their immense support and encouragement throughout the course. We utilized the Clever Hans Python library in Tensorflow (https://github.com/tensorflow/cleverhans) to generate the adversarial images for the various attacks. We have shared our GitHub repository with the cs231n teaching staff.

References

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

[2] I. Evtimov, K. Eykholt, E. Fernandes, T. Kohno, B. Li, A. Prakash, A. Rahmati, and D. Song. Robust physical-world attacks on deep learning models. *arXiv preprint arXiv:1707.08945*, 1, 2017.

[3] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[4] S. Gu and L. Rigazio. Towards deep neural network architectures robust to adversarial examples. *arXiv preprint arXiv:1412.5068*, 2014.

[5] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[6] G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[7] H. Kannan, A. Kurakin, and I. Goodfellow. Adversarial logit pairing. *arXiv preprint arXiv:1803.06373*, 2018.

[8] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. 2009.

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

[10] F. Liao, M. Liang, Y. Dong, T. Pang, J. Zhu, and X. Hu. Defense against adversarial attacks using high-level representation guided denoiser. *arXiv preprint arXiv:1712.02976*, 2017.

[11] 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.

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

[13] M. Osadchy, J. Hernandez-Castro, S. Gibson, O. Dunkelman, and D. Pérez-Cabo. No bot expects the deepcaptcha! introducing immutable adversarial examples with applications to captcha. *IACR Cryptology ePrint Archive*, 2016:336, 2016.

[14] N. Papernot, N. Carlini, I. Goodfellow, R. Feinman, F. Faghri, A. Matyasko, K. Hambardzumyan, Y.-L. Juang, A. Kurakin, R. Sheatsley, et al. cleverhans v2. 0.0: an adversarial machine learning library. *arXiv preprint arXiv:1610.00768*, 2016.

[15] 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.

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

[17] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In *Security and Privacy (SP), 2016 IEEE Symposium on*, pages 582–597. IEEE, 2016.

[18] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[19] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[20] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. *See https://arxiv. org/abs/1610.02391 v3*, 7(8), 2016.

[21] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2818–2826, 2016.

[22] 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.

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

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