CS 229: Milestone
Learning Adversarially Robust and Rich Image Transformations for Object Classification

Matthew Tan (mratan), Kimberly Te (kimte), and Nicholas Lai (nicklai)
Category: Computer Vision

December 14, 2019

Abstract

Adversarial attacks pose major safety and ethical dangers in object classification-based systems, such as facial recognition and autonomous driving. Our project aims to learn adversarially robust image transformations as a means of defense against these attacks for the image classification tasks. We analyzed the use of lossy compression techniques to "clean" adversarial images prior to feeding them to the object classification systems. Specifically, we implemented and evaluated black-box compression techniques, namely JPEG compression, Gaussian smoothing, K-Means, total variance minimization (TVM), and Vector-Quantized Variational Autoencoders (VQ-VAEs). These defenses were tested against various white-box adversarial attacks (FGSM, PGD, CarliniWagnerL2Attack, and DeepFool). on the MNIST and CIFAR-10 datasets. We used industry standard models for MNIST and CIFAR-10 as our baseline models. Our results showed that our defenses were capable of cleaning adversarial noise from the images to improve accuracy, suggesting potential against adversarial attacks. TVM and JPEG had the highest accuracies, where TVM was able to achieve at least 70% accuracy on both MNIST and CIFAR-10 despite the presence of strong adversarial attacks.

Introduction

With the increasing adoption of machine learning and deep learning systems in safety critical applications, [15], incentives to abuse these systems have also increased, where adversarial attacks can cause these systems to misbehave. This poses ethical concerns and safety risks on their applications in real-world systems, such as healthcare, sensors, autonomous driving, and facial recognition [5]. Despite major advancements of deep-learning in object classification, state-of-the-art algorithms are incredibly susceptible to adversarial perturbations [2] [7] [20]. These adversarial perturbations cause algorithms to output highly confident erroneous predictions and undermine the effectiveness of neural network models (See Figure 1). Therefore, there is a need to build defenses against adversarial attacks and to develop more adversarially robust models. Understanding how to defend against these adversarial attacks is paramount to building safe, ethical and widespread systems.

Therefore, the purpose of our project was to investigate possible defenses towards developing adversarially robust systems for object classification [1]. Specifically, we focused on applying input image transformations as defenses to "clean" the adversarial perturbations or noise using lossy compression and neural-network based techniques. Notably, we investigated the Vector Quantized Variational Autoencoders as a potential novel defense for adversarial attacks. Finally, we then examined these adversarial image transformation methods against state-of-the-art adversarial attacks on the models.

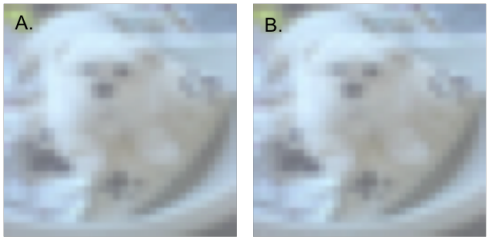


Figure 1: CIFAR-10 example of FGSM attack and resulting incorrect prediction. Attacks can seem inperceptible to the human eye.

Related Work

Adversarial examples are defined as inputs specifically designed by an adversary to make a model predict erroneously [1]. For this project, we follow the formal definition, where adversarial examples are inputs x with small perturbations η such that a prediction on $x + \eta$ outputs high probabilities for a class is different from the class predicted on x [6].

These adversarial attacks can be generally classified into two threat models: black-box attacks and white-box attacks. In the black-box threat model, the attacker does not have access to the model when designing the inputs. On the other hand, in the white box model, the attacker has full access to the model and can exploit this information in customizing their inputs. For this project work, we focus on the white-box threat model due to risk transferability to black-box threat models. In particular, [4] has shown that adversarial examples transfer between models. Thus an adversarial example that works on one model may work for another different model. Because of the non-differentiability of some of defense models, for fair-

[1] By adversarially robust, we mean that the model can classify correctly in the presence of adversarial perturbations of $l_\infty < \eta$ for some small η

ness, we treat all defenses as white-box and do not provide it to the attacking model.

While recent works have shown success in creating robust classifiers and defenses on simple datasets [22] [6], currently, there is no defense that is both generalizable and effective against all adversarial examples. Recent work has developed a variety of techniques to mitigate the impact of these attacks. Broadly, we can classify these techniques into: obfuscating gradients [2], manifold projection using GANs / autoencoders [10] [9], and other methods (K-Nearest Neighbors, convex optimization) [19] [18] [23]. However, recent work has shown neural network models that rely on obfuscating gradients can be circumvented by approximating the gradients [2]. Moreover, several of these other works [6] [23] propose methods with theoretical guarantees against certain types of attacks, however none have been shown to scale effectively to larger datasets.

Our use JPEG compression, traditional image transformations, and quantized encoders as adversarial defense.

Relevant to our work, Shin et al. [17] shows that JPEG compression is a simple yet powerful defense on the Imagenet dataset against Fast Gradient Sign Method (FGSM) [7] and Iterative FGSM (I-FGSm) [11]. Our work extends this by further exploring JPEG compression against new state-of-the-art attack methods such as Projected Gradient Descent (PGD) [13] and Deep Fool Attack [14].

Guo et al. found that traditional transformations to input images could act as potential adversarial defenses, such as cropping, image quilting and total variance minimization (TVM) [8]. Our work further explores the TVM approach on additional datasets and new attack methods (PGD), and utilizes other image transformations, such as K-Means and a Gaussian blur filter.

Vector Quantized-Variational AutoEncoders (VQ-VAE) [21] is a variant of Variational Autoencoders (VAE). It learns two models, an encoder and a decoder that uses a discrete latent space. Previous work has shown the potential of VAE for adversarial defense [12] with limited results. We hypothesize that the quantized nature of VQ-VAEs can serve as a "lossy" compression mechanism similar to JPEG. From our current knowledge, our work is a novel potential approach at using VQ-VAE for adversarial defense.

Dataset and Features

We utilized two datasets, MNIST and CIFAR-10, due to their versatile application for object classification (See Figure 2). MNIST is a dataset of 60,000 greyscale handwritten digits from one to nine, each $1 \times 28 \times 28$ pixels, split into 50,000 training examples and 10,000 test examples. CIFAR-10 is a dataset of 60,000 RGB images, each $3 \times 32 \times 32$ pixels, with ten possible classes of object. The dataset is divided into 50,000 training and 10,000 test images for each class.

To minimize the computational cost of evaluating the effectiveness of our defensive transformations, we tested on a subset of 1,000 randomly sampled examples. [2]

[2]We attempted to running the full dataset for one model, but it yielded similar results and this took more than 24 hours, where computational resources were limited.

Figure 2: Examples of VQ-VAE reconstruction on MNIST (first row), CIFAR-10 (second row) without any adversarial perturbations.

.

Since we used convolutional neural networks as the basis for our image models, we extract features from the data itself rather than imposing features as would be done in supervised learning. Furthermore, in many of the studied attack methods, they rely on these same extracted features (or gradients) to adversarially transform images. As such, the features that we used for each run vary depending on the dataset and initialization parameters we use.

Methods

Models

For our classifier models, we used two models for the two datasets. First, for the MNIST dataset, we trained a 2 Conv + 2 FC network for 10 iterations with a batch size of 64 and learning rate of 0.01. We also used an SGD optimizer with a momentum of 0.5. The images are also normalized before being fed into the model. [3] This achieved an accuracy of 99% on the test set. This served as our baseline accuracy for the MNIST dataset.

For the CIFAR dataset, we utilized a pre-trained DenseNet based model and ran it for 150 iterations with a learning rate of 0.1 and batch size of 128. We then use a SGD optimizer with momentum 0.9 and weight decay 5e-4. Our implementation also incorporates various data augmentation methods such as random cropping, random flipping. Finally the images are normalized before being fed into the model. [4] This achieved a baseline accuracy of 85% in the CIFAR-10 dataset. The accuracy we achieve are consistent with state-of-the-art models for these datasets.

Defense Pipeline

Our defense pipelline consists of running our baseline model on the original images, adversarial images, and transformed images. Our methodology was implemented

[3]We use the implementation from Pytorch examples `https://github.com/pytorch/examples/tree/master/mnist`

[4]We use implementation from `https://github.com/kuangliu/pytorch-cifar`

in Pytorch [5] [6]. A flowchart of our work process implementation can be seen in Figure 3.

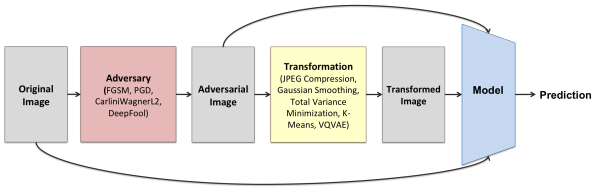


Figure 3: Our project model pipeline process. After transforming original image input into adversarial examples, we utilized our five defenses to "clean" the adversarial attributes to try to restore the original accuracy. As given above, we produced accuracies for the original images, adversarial images, and defended images.

We examined five different lossy transformation techniques, including a neural-network based VQ-VAE:

1. **JPEG Compression**: is a standard lossy image compression technique based on discrete cosine transform. The quality of the compression is controlled as a hyperparameter with 100% being lossless compression. We use JPEG compression to transform our adversarial images into a cleaned variant.
2. **Gaussian Smoothing**: widely used technique in computer graphics for image smoothing. It acts as a low-pass filter that can smooth out high frequency noises. The quality of this transformation is controlled by the the parameters of the Gaussian filter used (Σ).
3. **K-Means compression:** lossy image compression algorithm that uses K-Means to clusters colors, quantize the colors, and compress the image. The number of possible clusters (and colors) is controlled as a hyperparameter. Clusters were randomly initialized in this implementation.
4. **Total Variance Minimization (TVM):** a technique that reconstructs an image using a randomly selected subset of the image pixels with minimal and localized perturbations. Total variance of the fine-scale image is measured to remove excessive, namely adversarial, perturbations. We apply pixel dropout by sampling the pixels using a Bernoulli distribution and use the Bregman method to minimize total variation based on [8].
5. **Vector-Quantized Variational Autoencoder (VQ-VAE):** is a variant of variational autoencoders that uses discrete latent variables. We use the a forward pass through the encoder and decoder as a lossy compression mechanism. In some of our methods, we train the VQ-VAE with Gaussian noise on the inputs.

We then tested and evaluated these defenses over a range of hyperparameters. Our final accuracy results were based on the hyperparameters given below:

[5]Code available at a private Repository: `https://github.com/mratan1/cs229_final_proj`

[6]Please email `mratan@stanford.edu` for access

Defense	Parameters
JPEG Comp.	Quality: 90%
Gauss. Smooth.	σ_x : 0 σ_y : 0.3
K-Means	centroids (MNIST): 16 centroids (CIFAR-10): 50
Total Var. Min.	dropout rate: 0.5 weight: 0.03
VQ-VAE	hidden size: 256 k: 512 batch size: 128 epochs: 100 $\alpha : 2e-4$ β: 1 noise: [0, 0.25, 0.75]

To evaluate the robustness of our image transformations, we tested them against four state-of-the-art adversarial attacks. Each of these attacks uses some function to transform a normal input into an adversarial one. [7] These are:

1. **Fast Gradient Sign Method (FGSM):** adds noise in the same direction of the cost function gradient from [7]
   $x + \epsilon sgn(\nabla_x L(\theta, x, y))$;
2. **Projected Gradient Descent (PGD):** finds perturbations focused on gradients that maximize loss from [13]
   $x^{t+1} = \pi_{x+S}(x^t + \alpha sgn(\nabla_x L(\theta, x, y)))$;
3. **CarliniWagnerL2Attack:** applies an L_2 penalty from [3]
   $\min ||\frac{1}{2}(\tanh(w)+1) - 1||_2^2 + cf(\frac{1}{2}(\tanh(w)+1))$;
4. **DeepFool Attack:** projects onto decision boundaries from [14]
   $\text{argmin}_r ||r||_2$ such that $\exists k \ : \ w_k^T(x_0 + r) + b_k \ \geq \ w_{\hat{k}(x_0)}^T(x_0 + r) + b_{\hat{k}(x_0)}$.

Based on earlier works, We use $\max \epsilon = 0.3$, which has previously allowed for the most powerful attacks.

Examples of these attacks are shown in Figure 1 for FGSM (See Figure 1 for original and Figure 4for attacks). Qualitatively, FGSM, PGD, and CarliniWagner2 had minimal perturbations to the visible eye. Meanwhile, DeepFool generated noticeably different with the least realistic outputs.

For VQ-VAE, we first pre-train the model on the reconstruction task. The weights to the model are then frozen and the encoder / decoder is used as a transformation for the adversarial image.

Results and Discussion

After applying our defenses, we evaluated their model accuracies against the original and adversarial accuracies on the test dataset for MNIST and CIFAR. We summarize our results in Table 6. For each cell, we describe the accuracy we obtained on a given combination of attack and defense with MNIST's accuracy on the top, and CIFAR's accuracy on the same combination on the bottom.

[7]We use the Foolbox [16] implementation for our work

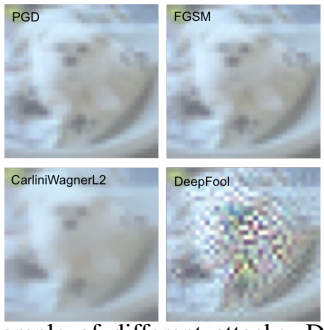


Figure 4: Example of different attacks. DeepFool produced the most noticeable attack.

All the adversarial attacks were effective against the undefended models, achieving a degradation of 0.00% accuracy on both datasets (Column 1 in Figure 6). Among the attack methods, DeepFool was the most resilient against our defenses, in which our defenses had the lowest improvements in accuracy. All defenses had improvements less than 10% accuracy. Interestingly, visual inspection showed that DeepFool also produced the least realistic images. It could be possible that it was more difficult for the compression techniques to remove such large perturbations and distortions. On the other hand, PGD was the easiest attack to overcome. Qualitatively, the images produced are the most realistic. In many cases, the adversarial perturbations were not visible. Since they were not as visually dominant, the compression algorithms could have better compressed and removed these perturbations.

Despite substantial differences between original and adversarial examples after transformation, all defenses improved the accuracy of the classifiers (See 6). They appeared to minimize perturbations and preserved useful features. Among the methods, TVM and JPEG performed the best followed by K-Means. JPEG had the highest accuracies for most of the attacks on MNIST, achieving a 99% accuracy for FGSM, PGD, and CarliniWagnerL2. Meanwhile, TVM had the highest accuracies on CIFAR, achieving over 70% on FGSM, PGD, and CarliniWagner. Excluding DeepFool, TVM model achieved 70% accuracy against attacks on both MNIST / CIFAR, which is less than a 30% difference from the original model. This shows that TVM and JPEG are both strong simple defenses. K-Means was also able to improve accuracies exceeding 50% for both MNIST. More complex defenses such as Gaussian Smoothing and VQ-VAE did not perform nearly as well.

Both from these results and qualitative inspections, it seems that the traditional transformations were able to effectively compress and remove noise from adversarial perturbations, like FGSM, PGD, and CarliniWagnerL2. Since their pertubations appeared small, and TVM, JPEG, and K-means compress to smaller subspaces this likely cleaned the noise. However, DeepFool had larger variance, which could make it more difficult to recover a similar output to the original.

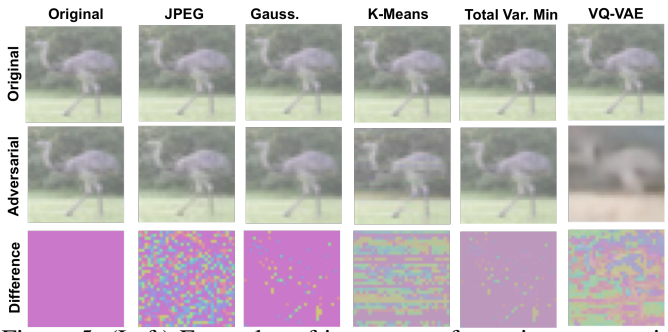
Between MNIST and CIFAR datasets, we found that the defenses performed much better in the MINST dataset. This could possibly be due to the relative complexity of the CIFAR dataset. MNIST was smaller in dimensions, had a smaller range in colors, and had more similar qualtitative content. For some adversarial attacks with visible changes, the new output images could appear as different digits.

Comparing the model predictions between the defenses, the confusion matrices showed that simpler methods, like JPEG, tended to mispredict a class for specific respective classes (eg. bird for cat). However, VQ-VAE was more uniformly dispersed across the classes, possibly due to its blurring effects. Confusion matrices for JPEG and VQ-VAE are shown (See Figure 7).

Regarding the hyperparameters, we also performed basic ablations studies on some of the models to understand their sensitivity to hyperparameters. In particular, adjusting the quality of the JPEG compression did not change the accuracy significantly. On the other hand, the number of clusters of K-means was sensitive to tuning. For both MNIST and CIFAR-10, a low number of centroid clusters (eg. n=4 out of 28) resulted in the resulting image being too compressed and losing rich features that the classifier requires. On the other hand, a very large number of centroid clusters (eg. n=27 out of 28) resulted in minimal compression of adversarial noise and therefore low accuracy.

Applying a neural-network based approach, VQ-VAE was also investigated as a possible new defense. However, we found that it yielded the lowest accuracies across the different defense transformations (See columns 6-8 in 6). While it had one of the highest accuracies on CIFAR for Deep Fool, this was below 10%. It was not effective out of the box.

While adding noise to the VQ-VAE training was effective in improving the results for the MNIST dataset, but it did not improve the CIFAR-10 dataset. In particular, we added Gaussian noise with standard deviations [0, 0.25, 0.75] to the input image during VQ-VAE training for the reconstruction task. This resulted in much more blurred reconstructions and a higher reconstruction loss, but increased object classification accuracy for MNIST (See 5). Increasing the amount of noise however was not effective at cleaning out noise for the CIFAR-10 dataset. Furthermore, increasing the noise past 1.0 did not result in improved accuracy on either MNIST or CIFAR and only made reconstruction worse. We also attempted to tune the hyper-parameters for VQ-VAE by performing a coarse grid search of the latent space and hidden size but this did not result in an improved accuracy. We find that the resulting "cleaned" image, seen in Figure 5 in the VQ-VAE adversarial example, was significantly more blurry than the training images. We hypothesize that this distributional shift, which was more apparent in the CIFAR dataset, is the likely cause of the poor performance of the resulting model. While retraining the model with the blurred images will likely increase its accuracy, this is outside the scope of this project. Furthermore, the defense becomes attack-specific which is not the goal of the defensive transformations. On the other hand, one limi-



(MNIST, CIFAR)	No Defense	JPEG Comp.	Gauss. Smooth	K Means	Total Var. Min	VQVAE 0.00	VQVAE 0.25	VQVAE 0.75
No Attack	0.99, 0.85	0.99, 0.85	0.99, 0.85	0.99, 0.85	0.99, 0.85	0.99, 0.85	0.99, 0.85	0.99, 0.85
FGSM	0.00, 0.00	**0.99**, 0.65	0.52, 0.24	0.701, 0.545	0.707, **0.727**	0.13, 0.28	0.2, 0.26	0.71, 0.23
PGD	0.00, 0.00	**0.99**, 0.73	0.67, 0.28	0.844, 0.612	0.895, **0.748**	0.14, 0.28	0.22, 0.27	0.81, 0.23
Carlini Wagner 2	0.00, 0.00	**0.99**, 0.67	0.7, 0.26	0.82, 0.555	0.906, **0.736**	0.15, 0.28	0.47, 0.27	0.86, 0.23
Deep Fool	0.00, 0.00	0.02, 0.05	0.01, 0.11	0.023, 0.057	0.02, **0.062**	0.03, 0.23	**0.03**, 0.2	0.02, **0.062**

Figure 5: (Left) Examples of image transformations comparing the original, FGSM attack, and their noise differences. Noise on the undefended example appears small to the visible eye.
Figure 6: (Right) Predictions on the test datasets for (MNIST, CIFAR) against the undefended model, JPEG compression, Gaussian Smoothing, K-Means, TVM, and VQ-VAE. Bolded accuracies are the highest accuracy for that given attack and dataset. JPEG had most highest accuracies on MNIST while TVM had the highest accuracies on CIFAR.

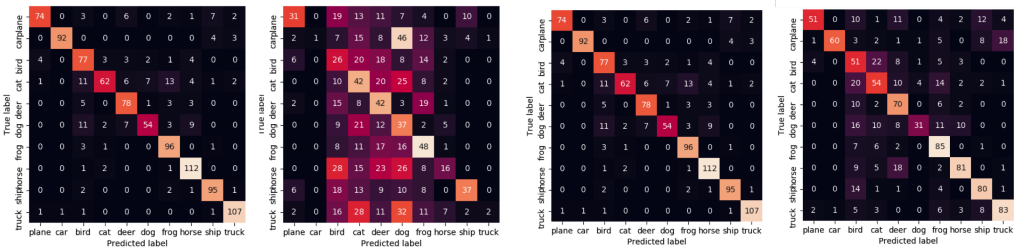


Figure 7: (Left to right): Predicted accuracy for VQ-VAE, accuracy with adversarial attacks for VQ-VAE, accuracy with adversarial attacks for JPEG, and predicted accuracy for JPEG. We see that overall VQVAE has a much more noticeable incorrect labeling than does JPEG, and that this is distributed across labels.

tation of neural network based defenses is that an adversary can use the white box model on the overall (defense + classifier) model to optimize an attack. To the best of our knowledge there is no known solution to this problem yet and it remains an open area of research. This blurring issue was not observed in the misclassifications of the classifier following the JPEG transformation. Upon visual inspection, we found that these misclassifications often did not look realistic or had boundary values (0 or 255) in various areas of the image, for VQ-VAE example in 5. Upon visual inspection, many of these images contained salt-and-pepper-like noise which introduced artifacts to the image that are not captured in the training set of our classifier, and therefore is erroneously classified.

Conclusion and Future Work

In conclusion, with the growing proliferation of object classification models in the world and the increasing capability of these adversarial attacks to deceive state-of-the-art classification models, the importance of adversarial defense is growing. In our project, we explored the use of various traditional and deep-learning based image transformation techniques for adversarial image defense on state-of-the-art attacks. Regarding novelty, we explored a new potential defense (VQ-VAE) and evaluated defenses on various datasets and attacks. Overall, all five defense transformations improved accuracy. We found that the TVM and JPEG image transformations were the most effective image cleaning methods. The simplicity of these models and their model-agnostic nature as well as their non-differentiability make them powerful and simple first line of defense against adversarial attacks.

For future work, we intend to extend this project in several ways. In particular, we intend to (1) explore the effectiveness of the models on larger datasets such as Imagenet / CELEB-A, (2) explore combinations of various methods using model ensembling (eg. total variance minimization and GANs), (3) exploring other newer methods of defense such as randomly initialized models and bayesian neural networks and convex optimization, (4) training VQ-VAE with different kinds of noises (eg. saltand-pepper, gaussian, uniform). One other potential area to explore would be to increase robustness against structured and targted perturbations, e.g. graffiti on walls or paint chipping on signs.

Contributions

All project aspects were distributed among members. Matthew set-up infrastructure, implemented and tested many of the models/algorithms, devised experiments and visualizations, and contributed to write-up. Kimberly provided background research and analysis, implemented and tested models/algorithms, ran experiments and created visualizations, and contributed to write-up. Nick implemented model, provided code support, and contributed

to write-up.

References

[1] Andrew Kurakin, I. G., and Bengio, S. 2019. On evaluating adversarial robustness. *arXiv preprint arxiv:1902.06705v2*.

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

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

[4] Demontis, A.; Melis, M.; Pintor, M.; Jagielski, M.; Biggio, B.; Oprea, A.; Nita-Rotaru, C.; and Roli, F. 2019. Why do adversarial attacks transfer? explaining transferability of evasion and poisoning attacks. In *28th {USENIX} Security Symposium ({USENIX} Security 19)*, 321–338.

[5] Finlayson, S. G.; Bowers, J. D.; Ito, J.; Zittrain, J. L.; Beam, A. L.; and Kohane, I. S. 2019. Adversarial attacks on medical machine learning. *Science* 363(6433):1287–1289.

[6] Gal, Y., and Smith, L. 2018. Sufficient conditions for idealised models to have no adversarial examples: a theoretical and empirical study with bayesian neural networks. *arXiv preprint arXiv:1806.00667*.

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

[8] Guo, C.; Rana, M.; Cisse, M.; and Van Der Maaten, L. 2017. Countering adversarial images using input transformations. *arXiv preprint arXiv:1711.00117*.

[9] Jia, X.; Wei, X.; Cao, X.; and Foroosh, H. 2019. Comdefend: An efficient image compression model to defend adversarial examples. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 6084–6092.

[10] Kingma, D. P., and Welling, M. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

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

[12] Luo, Y., and Pfister, H. 2018. Adversarial defense of image classification using a variational auto-encoder. *arXiv preprint arXiv:1812.02891*.

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

[14] Moosavi-Dezfooli, S.-M.; Fawzi, A.; and Frossard, P. 2016. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2574–2582.

[15] Mozaffari-Kermani, M.; Sur-Kolay, S.; Raghunathan, A.; and Jha, N. K. 2014. Systematic poisoning attacks on and defenses for machine learning in healthcare. *IEEE journal of biomedical and health informatics* 19(6):1893–1905.

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

[17] Shin, R., and Song, D. 2017. Jpeg-resistant adversarial images. In *NIPS 2017 Workshop on Machine Learning and Computer Security*.

[18] Sitawarin, C., and Wagner, D. 2019a. Defending against adversarial examples with k-nearest neighbor. *arXiv preprint arXiv:1906.09525*.

[19] Sitawarin, C., and Wagner, D. 2019b. On the robustness of deep k-nearest neighbors. *arXiv preprint arXiv:1903.08333*.

[20] Su, J.; Vargas, D. V.; and Sakurai, K. 2019. One pixel attack for fooling deep neural networks. *IEEE Transactions on Evolutionary Computation*.

[21] van den Oord, A.; Vinyals, O.; et al. 2017. Neural discrete representation learning. In *Advances in Neural Information Processing Systems*, 6306–6315.

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

[23] Wong, E.; Schmidt, F.; Metzen, J. H.; and Kolter, J. Z. 2018. Scaling provable adversarial defenses. In *Advances in Neural Information Processing Systems*, 8400–8409.