Filtering Adversarial Noise with Double Quantization

MaungMaung AprilPyone*, Yuma Kinoshita* and Hitoshi Kiya*
* Tokyo Metropolitan University, Asahigaoka, Hino-shi, Tokyo, 191-0065, Japan
E-mail: {april-pyone-maung-maung@ed.,kinoshita-yuma@ed.,kiya@}tmu.ac.jp

***Abstract*—Despite deep learning being powerful to solve challenging problems, it is vulnerable towards adversarial examples. To defend these adversarial blind spots in the deep learning, researchers have proposed various approaches. However, conventional adversarial training can reduce the accuracy significantly. In this paper, we propose a method to incorporate quantized images in both training and testing to maintain identical accuracy for both normal and adversarial examples. Specifically, the proposed method utilizes dithering during training and dithering and linear quantization as a mean of adversarial filtering during testing. We evaluated the proposed method with a well-known strong first-order adversary and also conducted experiments in different bit depths. The results suggest that the proposed method achieves 87.14% and 85.28% accuracy for 2-bit and 1-bit dithered models for both normal and adversarial tests on the noise level of 8. In addition, due to having identical accuracy for both adversarial and normal tests, the proposed method can detect adversarial examples if the original test dataset is known. The code for the experiments is released on https://github.com/fugokidi/one-bit-quantization.**

I. Introduction

Deep Learning has been proven as a powerful tool to solve difficult problems in computer vision as well as other fields [1]. The outstanding success has enabled deep learning models to be deployed in security-critical applications such as face recognition, biometric authentication, autonomous cars, spam filters, malware detection systems, etc. These security-sensitive applications demand deep learning to be reliable regardless of its remarkable performance. Therefore, reliability in the deep learning is quintessential in these security-important applications.

Nevertheless, machine learning in general suffers from two types of attacks: model inversion attacks and adversarial attacks. In this work, we focus on adversarial attacks. Researchers have already discovered that neural networks are vulnerable to adversarial examples [2], [3], [4]. Imperceptible adversarial perturbation causes neural networks to misclassify with high confidence or force to classify a targeted label. In computer vision, it is a clear threat. An example of adversarial sample is depicted in Fig. 1 in which the network classifies "dog" as "horse" with 100% confidence. One recent work shows that adversarial example can be photographed with smartphone and the taken picture can still fool the neural network [5]. This is potentially dangerous especially for autonomous cars. An attacker can potentially paint or use stickers to cause the accidents [6]. Therefore, deep learning has got a significant amount of attention and a lot of effort has been put towards adversarial robustness.

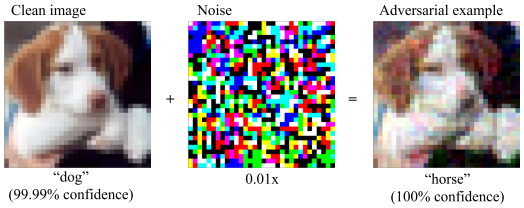

Fig. 1. Example of an adversarial example.

Numerous ways of attacks and defenses have been proposed towards adversarial examples such as [7], [8], [9]. We have also hypothesized in our previous work that learnable image encryption has certain degree of adversarial robustness [10]. However, the models trained to be robust against adversarial examples drop the accuracy significantly. To the best of our knowledge, there is no robust model that has the same accuracy as the normal trained model. Recently, Miyazato et al. introduced to use bit depth variation to improve adversarial robustness while maintaining good accuracy [11]. But, their work has been tested only on an easy adversary. Maintaining accuracy and getting adversarial robustness is a growing concern and on-going research in the deep learning community.

In this paper, we propose a method to achieve adversarial robustness in a specific scenario where the rightful user prepares the test images and adversarial noise is added to the test images before classification. Usually, this is the situation where the model is deployed in the cloud server and it is illustrated in Fig. 2. The main idea of the proposed method is to use two types of quantization: quantization with dithering and linear quantization. Specifically, the network is trained with quantized images and the test quantized images are quantized again to remove the adversarial noise during testing. We make the following contributions in this paper.

- We propose a mechanism to train the network with dithered images and to use linear quantization as an adversarial filter over dithered images during testing.
- We evaluated the proposed method on different bit depths with or without dithering against adversarial examples.
- Additionally, since our method produces the same accuracy even under attacks, inferrence to detect adversarial examples can be made if the original test dataset is known.

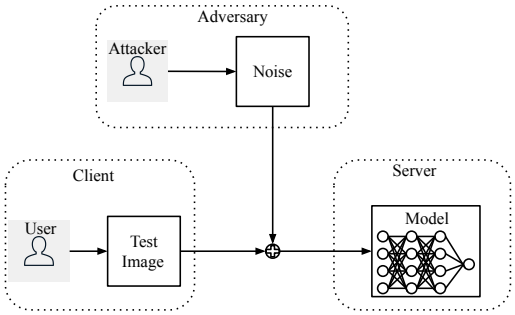

Fig. 2. Attack scenario that the proposed method targets to defend against adversarial examples.

II. Related Work

A. Adversarial Attacks

Adversarial attacks are attacks towards neural networks where an adversary can make the neural network misclassify with high confidence or force the network to classify a targeted label. In the context of image classification, the attacks are carried out by inputting carefully designed perturbed images to the neural network. These well-crafted perturbed images are so called adversarial examples. Usually, adversarial examples are generated by using optimization techniques to maximize the loss of the objective. Based on the knowledge available to an attacker, the attacks can be catagorized into two groups: white-box and black-box. White-box attacks have direct access to the model and black-box ones do not have.

The attacks with white-box settings include Fast Gradient Sign Method (FGSM), Projected Gradient Descent (PGD), etc. FGSM is a l_∞-bounded attack with the goal of misclassification poposed by Goodfellow at el al. [12]. It is computaionally efficient and an adversarial example x' can be generated as

$$x' = x + \epsilon \, sign(\nabla_x J(\theta, x, y)), \tag{1}$$

where x is the original image, ϵ is the allowable perturbation size and $\nabla_x J(\theta, x, y)$ is the gradient of the loss with respect to the input image. FGSM is a single step approach and the extension of FGSM is Basic Iterative Method (BIM) [13]. It is a straightforward way of applying FGSM multiple times iteratively. The adversarial example on $(t+1)^{\text{th}}$ iteration with BIM is

$$x'_{t+1} = clip_{x,\epsilon}(x'_t + \alpha \, sign(\nabla_x J(\theta, x, y))), \tag{2}$$

where $clip_{x,\epsilon}(X)$ is to clip $X_{i,j}$ to be in the range of $[x_{i,j} - \epsilon, x_{i,j} + \epsilon]$ and α is the step size. Madry et al. pointed out that BIM is equivalent to Projected Gradient Descent (PGD) [8]. The difference between BIM and PGD is that PGD projects the perturbation back onto the l_∞-norm ball (i.e., clipping perturbation in the range of $[-\epsilon, +\epsilon]$) in each step [7].

There are also black-box attacks where an attacker only knows inputs and ouputs of the model. The adversarial examples are built by using a surrogate model instead of a real one. Such black-box attacks were proposed in [6], [14].

B. Adversarial Defenses

The intuitive way of defending the adversarial examples is including adversarial examples in the training process. It is known as adversarial training. Early work [12] suggested to use adversarial objective function during the training that works as an effective regularizer. However, such a trained model still cannot resist stronger adversaries such as PGD. One of the other state-of-the-art methods is PGD training [8]. The basic idea is to train the network with PGD perturbed input. Nevertheless, PGD training is computationally expensive and drops the accuracy significantly. From the MIT MadryLab CIFAR10 Challenge leaderboard, the accuracy of the adversarially trained model against 20-step PGD is 47.04% [15].

To improve adversarial robustness while maintaining the accuracy, Miyazato et al. proposed to use quantized images that maximizes the loss during the training process [11]. Their method is basically to use quantized images solely in the training process and was tested on FGSM only. FGSM is fast, but not a reliable adversary. Our experiments proved that the models trained with even severely quantized images (i.e., 1 to 2-bit quantization) cannot resist PGD attack without reinforcing our proposed method (i.e., linear quantization on dithered adversarial examples). Therefore, the method proposed by Miyazato et al. [11] is not robust against PGD attack. In contrast, our method requires to use dithered images in both training and testing. The linear quantization in the proposed method removes adversarial noise completely.

There are other defense mechanisms such as defensive distillation where two models are used to train the network [16]. The first model is trained by using hard labels and the second one is trained to predict the probabilities of the first model. Nonetheless, it was reported that defensive distillation is not robust against adversarial examples [17].

III. Proposed Method

The architecture of the proposed method is shown in Fig. 3. The proposed method requires to use quantized images in training and testing. There are two types of quantization: quantization with dithering and linear quantization. Dithering radomizes the quantization error and enhances visual quality especially on low bit depth quantization. Therefore, models trained with dithered images result in better accuracy. Linear quantization is applied in the proposed method to filter adversarial noise.

As shown in Fig. 3, dithered images are used to train the network. For testing phase, test images are also dithered. In adversarial settings, the perturbed dithered images are linearly quantized. This linear quantization removes the adversarial noise resulting the exact same accuracy as testing with clean images.

IV. Experimental Results

A. Setup

We used CIFAR10 [18] dataset with batch size of 128. Floyd-Steinberg algorithm [19] was employed to dither images in the proposed method. All the images in the dataset for

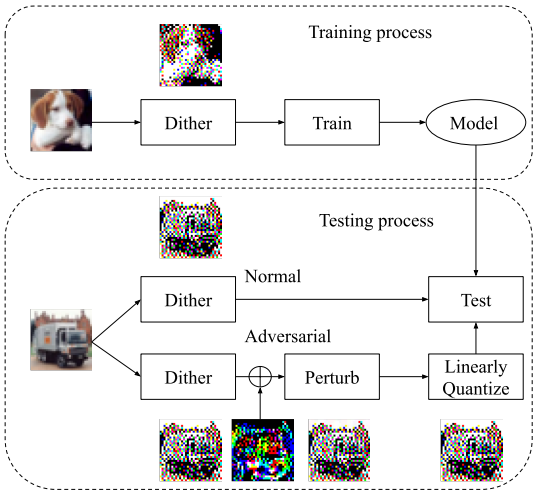

Fig. 3. Training and test phase of the proposed method.

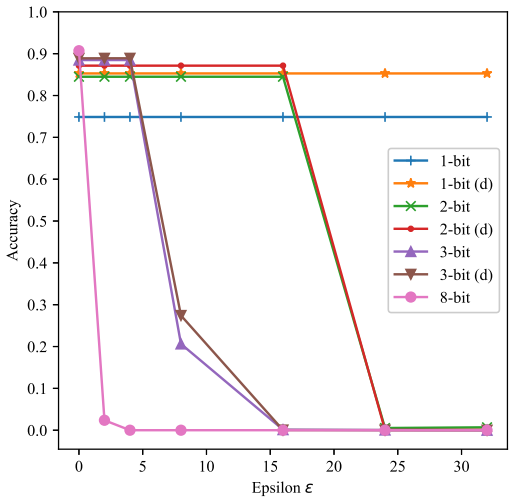

Fig. 4. Accuracy vs epsilon (1-bit dithered model).

all cases were in the range of $[0, 1]$ with live augumentation (random cropping with $padding = 4$ and random horizontal flip). However, there was no prior normalization.

For the network, we deployed deep residual network, specifically ResNet20 [20] on PyTorch platform. The network was trained for 160 epochs with stochastic gradient descent optimization with the initial learning rate of 0.1. The step learning rate scheduler was used with the parameters ($lr\_steps = 40, gamma = 0.1$). The $weight\_decay$ and $momentum$ were configured with 0.0001 and 0.9 respectively.

The settings used in adversarial testing are adapted from Madry Lab CIFAR10 challenge [15]. PGD attack with perturbation size $\epsilon = 8/255$ and step size $\alpha = 2/255$ was used for 20 iterations to evaluate the robustness of the trained models.

B. Experiments

We trained the network with quantized images in different bit depths with or without dithering resulting a total of 15 models. The models were trained in each bit depth (i.e., 1 to 7-bit) with and without dithering. The non-quantized 8-bit images are also used to train a 8-bit model.

To observe the performance of each model, we evaluated each model with test images in different bit depths that are quantized with or without dithering. For example, the first model trained with 1-bit quantized images was tested with the data quantized in 1-bit with and without dithering, 2-bit with and without dithering, and so on, till 8-bit. The results of testing the models with clean images are summarized in Table I. Intuitively, when train and test data are in the same bit depth, the accuracy is maximized as shown in each row of Table I highlighted with bold type font. The accuracy of more than 90% is maintained on 5, 6 and 7-bit models. However, the accuracy gradually drops towards 1-bit trained model. The results confirm that dithering helps improve accuracy from the experiments.

To evaluate adversarial robustness, we deployed PGD attack on the quantized test images that are quantized with and without dithering. Each model was tested with adversarial examples in different bit depths. Table II captures the results of the adversarial test for all 15 models. The results suggest that 2-bit dithered model provides the best accuracy (i.e., 87.14%) when testing against 2-bit dithered images. On the other hand, 1-bit dithered model has the accuracy of 85.28% and 2-bit model without dithering has the accuracy of 84.49%. From the results, the test images that are quantized in higher bit depth such as 3, 4, 5, 6 and 7 bit lead to poor adversarial robustness. Therefore, we can conclude that linear quantization does not work when the test images are in high bit depth.

Moreover, we also investigated how the proposed method reacts to adversarial attacks with different perturbation amount for top 6 models and 8-bit model. The noise level, ϵ values of $[0, 2, 4, 8, 16, 24, 32]$ were used to perform the adversarial tests. When the noise level is up to 16, 2-bit dithered model still keeps the accuracy of 87.14%. However, the accuracy abruptly drops when the noise level is higher than 16. The model trained with 2-bit images has the accuracy of 84.49% till the noise level of 16 and drops the accuray for higher noise levels. One bit dithered model is resilient over all tested noise levels with the accuracy of 85.28%. Similarly, one bit model is also resistant against adversarial examples with the accuracy of 74.87%. It is noteworthy that 3-bit dithered model results higher accuracy (i.e., $\approx 88.90\%$) when the noise amount is less or equal to 4. When there is no noise, in fact, the 8-bit model produces the highest accuracy. The graph of accuracy versus epsilon is plotted in Fig. 4. All in all, our benchmark noise level is 8 and 2-bit dithered gives the best accuracy on the noise level of 8.

However, when the PGD attack is carried out in 8-bit images before dithering, our method fails to defend the attack. As an example, we also tested the proposed method on 1-bit dithered model with 8-bit noise added before 1-bit dithering. The result is 4.74%. We shall improve and defend this scenario in the future work.

TABLE I
QUANTIZATION TEST

Train (d = dithered) \ Test (d = dithered)	1-bit	1-bit (d)	2-bit	2-bit (d)	3-bit	3-bit (d)	4-bit	4-bit (d)	5-bit	5-bit (d)	6-bit	6-bit (d)	7-bit	7-bit (d)	8-bit
1-bit	**0.749**	0.112	0.724	0.390	0.651	0.615	0.616	0.605	0.603	0.598	0.595	0.593	0.591	0.591	0.589
1-bit (d)	0.481	**0.853**	0.769	0.828	0.811	0.813	0.809	0.806	0.805	0.803	0.802	0.800	0.800	0.800	0.799
2-bit	0.516	0.121	**0.845**	0.528	0.841	0.809	0.817	0.817	0.807	0.806	0.801	0.800	0.798	0.797	0.797
2-bit (d)	0.454	0.410	0.781	**0.871**	0.846	0.859	0.845	0.845	0.841	0.843	0.840	0.841	0.839	0.839	0.839
3-bit	0.439	0.108	0.782	0.336	**0.885**	0.829	0.888	0.887	0.883	0.884	0.883	0.883	0.882	0.882	0.881
3-bit (d)	0.390	0.140	0.753	0.730	0.874	**0.889**	0.882	0.886	0.880	0.883	0.877	0.879	0.879	0.878	0.879
4-bit	0.359	0.172	0.692	0.282	0.862	0.720	**0.898**	0.886	0.897	0.897	0.897	0.898	0.898	0.898	0.898
4-bit (d)	0.296	0.108	0.687	0.320	0.875	0.837	0.896	**0.898**	0.898	0.899	0.898	0.899	0.897	0.897	0.897
5-bit	0.241	0.127	0.641	0.235	0.842	0.658	0.897	0.872	**0.902**	0.903	0.904	0.905	0.905	0.905	0.904
5-bit (d)	0.291	0.154	0.665	0.256	0.851	0.713	0.899	0.888	0.904	**0.903**	0.903	0.902	0.903	0.903	0.902
6-bit	0.310	0.107	0.620	0.248	0.821	0.603	0.888	0.836	0.903	0.895	**0.907**	0.906	0.907	0.907	0.908
6-bit (d)	0.294	0.149	0.667	0.246	0.846	0.687	0.896	0.877	0.903	0.903	0.904	**0.904**	0.905	0.905	0.905
7-bit	0.267	0.108	0.616	0.218	0.816	0.587	0.884	0.821	0.900	0.894	0.905	0.903	**0.907**	0.907	0.907
7-bit (d)	0.288	0.130	0.636	0.221	0.821	0.605	0.884	0.837	0.898	0.892	0.903	0.901	0.902	**0.902**	0.903
8-bit	0.278	0.117	0.619	0.265	0.815	0.567	0.880	0.816	0.901	0.890	0.906	0.905	0.908	0.907	**0.907**

TABLE II
ADVERSARIAL QUANTIZATION TEST ($\epsilon = 8/255$)

Train (d = dithered) \ Adversarial Test (d = dithered)	1-bit	1-bit (d)	2-bit	2-bit (d)	3-bit	3-bit (d)	4-bit	4-bit (d)	5-bit	5-bit (d)	6-bit	6-bit (d)	7-bit	7-bit (d)	8-bit
1-bit	**0.749**	0.749	0.724	0.749	0.160	0.749	0.004	0.749	0.002	0.541	0.002	0.552	0.002	0.556	0.002
1-bit (d)	0.481	**0.853**	0.769	0.828	0.306	0.320	0.017	0.011	0.004	0.003	0.004	0.003	0.004	0.003	0.003
2-bit	0.516	0.516	**0.845**	0.845	0.271	0.271	0.005	0.005	0.001	0.001	0.001	0.001	0.001	0.001	0.001
2-bit (d)	0.454	0.410	0.781	**0.871**	0.268	0.300	0.007	0.005	0.001	0.001	0.002	0.001	0.001	0.001	0.001
3-bit	0.439	0.439	0.782	0.782	**0.206**	0.206	0.001	0.001	0.000	0.000	0.000	0.000	0.000	0.000	0.000
3-bit (d)	0.390	0.140	0.753	0.730	0.239	**0.274**	0.002	0.001	0.000	0.000	0.000	0.000	0.000	0.000	0.000
4-bit	0.359	0.359	0.692	0.692	0.166	0.166	**0.001**	0.001	0.000	0.000	0.000	0.000	0.000	0.000	0.000
4-bit (d)	0.296	0.108	0.687	0.320	0.190	0.180	0.001	**0.000**	0.000	0.000	0.000	0.000	0.000	0.000	0.000
5-bit	0.241	0.241	0.641	0.641	0.162	0.162	0.001	0.001	**0.000**	0.000	0.000	0.000	0.000	0.000	0.000
5-bit (d)	0.291	0.154	0.665	0.256	0.150	0.110	0.001	0.000	0.000	**0.000**	0.000	0.000	0.000	0.000	0.000
6-bit	0.310	0.310	0.620	0.620	0.151	0.151	0.001	0.001	0.000	0.000	**0.000**	0.000	0.000	0.000	0.000
6-bit (d)	0.294	0.149	0.667	0.246	0.147	0.102	0.001	0.000	0.000	0.000	0.000	**0.000**	0.000	0.000	0.000
7-bit	0.267	0.267	0.616	0.616	0.130	0.130	0.001	0.001	0.000	0.000	0.000	0.000	**0.000**	0.000	0.000
7-bit (d)	0.288	0.130	0.636	0.221	0.149	0.091	0.000	0.000	0.000	0.000	0.000	0.000	0.000	**0.000**	0.000
8-bit	0.278	0.117	0.619	0.240	0.125	0.262	0.001	0.269	0.000	0.205	0.000	0.209	0.000	0.207	**0.000**

V. DISCUSSION

As reported in the work [11], quantization may help improve robustness against FGSM. However, quantization in training alone cannot resist PGD attacks when the test images are 8-bit. In addition, quantization to lower bit depth reduces the accuracy. To enhance the accuracy in low bit depth images, we introduce dithering in the proposed method. An example test image in different bit depths is displayed in Fig. 6 (with dithering) and Fig. 5 (without dithering). The visual quality degrades when the bit depth is lower. However, dithering creates the illusion of color depth resulting better accuracy in the proposed method.

Why the proposed method works is that the adversarial noise generated on dithered images are removed by linear quantization. This second time of quantization process generates a strong filtering effect and the adversarial noise is completely removed. An example of dithered, perturbed and linearly quantized image is shown in Fig. 7. From the figure, the quantized image (c) is restored to the dithered image (a) after linear quantization. Therefore, the proposed method yields the exact same accuracy whether or not under attacks. However, the proposed method only works on low bit depth quantization (such as 1 or 2-bit).

Although the proposed method cannot resist the 8-bit PGD attack, it can be used for detecting adversarial examples if the reference test dataset is known. The proposed method provides identical accuracy for both clean and adversarial examples. By

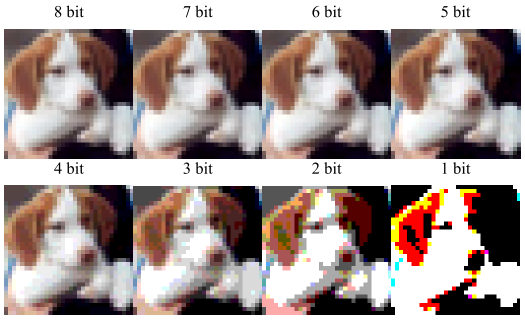

Fig. 5. Example of quantization without dithering.

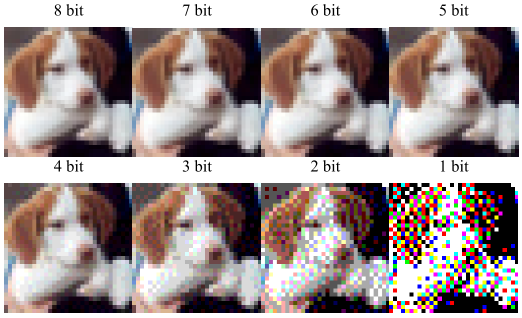

Fig. 6. Example of quantization with dithering.

comparing the accuracy of reference test dataset with the test dataset, we can detect adversarial examples.

VI. Conclusions

In this paper, we propose a method to achieve the same accuracy for both clean and adversarial examples. Specifically, the dithered images are used during training and the test dithered images are quantized again during testing. We conducted the experiments by training 15 models with different bit depths with and without dithering and evaluated each model. In the best case scenario (i.e., 2-bit dithered model), the results suggest that the proposed method achieves 87.14% accuracy on both normal and adversarial tests. In addition, the proposed method is not robust against 8-bit adversarial noise. But, the proposed method can be used to detect 8-bit adversarial examples if the clean dataset is available for references. As for future work, we shall improve accuracy and robustness to get near plain images accuracy and provable robustness against adversarial examples.

(a) Dithered (b) Perturbed (c) Quantized

Fig. 7. Example of a test image being dithered, perturbed and quantized.

References

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, p. 436, 2015.

[2] 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.

[3] B. Biggio, I. Corona, D. Maiorca, B. Nelson, N. Šrndić, P. Laskov, G. Giacinto, and F. Roli, "Evasion attacks against machine learning at test time," in *Joint European conference on machine learning and knowledge discovery in databases*. Springer, 2013, pp. 387–402.

[4] R. Jia and P. Liang, "Adversarial examples for evaluating reading comprehension systems," *arXiv preprint arXiv:1707.07328*, 2017.

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

[6] 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*. ACM, 2017, pp. 506–519.

[7] M.-I. Nicolae, M. Sinn, M. N. Tran, A. Rawat, M. Wistuba, V. Zantedeschi, I. M. Molloy, and B. Edwards, "Adversarial robustness toolbox v0. 2.2," *arXiv preprint arXiv:1807.01069*, 2018.

[8] 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.

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

[10] M. AprilPyone, W. Sirichotedumrong, and H. Kiya, "Adversarial Test on Learnable Image Encryption," *arXiv e-prints*, p. arXiv:1907.13342, Jul 2019.

[11] S. Miyazato, T. Yamasaki, and K. Aizawa, "Improving the robustness of neural networks to adversarial examples by reducing color depth of training image data," The University of Tokyo, Tech. Rep., 2019.

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

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

[14] N. Papernot, P. McDaniel, and I. Goodfellow, "Transferability in machine learning: from phenomena to black-box attacks using adversarial samples," *arXiv preprint arXiv:1605.07277*, 2016.

[15] MadryLab, "A challenge to explore adversarial robustness of neural networks on cifar10," https://github.com/MadryLab/cifar10_challenge.

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

[17] N. Carlini and D. Wagner, "Defensive distillation is not robust to adversarial examples," *arXiv preprint arXiv:1607.04311*, 2016.

[18] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," Citeseer, Tech. Rep., 2009.

[19] R. W. Floyd and L. Steinberg, "An Adaptive Algorithm for Spatial Greyscale," *Proceedings of the Society for Information Display*, vol. 17, no. 2, pp. 75–77, 1976.

[20] 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*, 2016, pp. 770–778.