# Exploiting the Potential of Datasets: A Data-Centric Approach for Model Robustness

**Yiqi Zhong, Lei Wu, Xianming Liu*, Junjun Jiang**

Advanced Imaging and Intelligent Analysis Lab
School of Computer Science and Technology
Harbin Institute of Technology, Harbin, China
{21s003117, 21s003031}@stu.hit.edu.cn, {csxm, jiangjunjun}@hit.edu.cn

## Abstract

Robustness of deep neural networks (DNNs) to malicious perturbations is a hot topic in trustworthy AI. Existing techniques obtain robust models given fixed datasets, either by modifying model structures, or by optimizing the process of inference or training. While significant improvements have been made, the possibility of constructing a high-quality dataset for model robustness remain unexplored. Follow the campaign of data-centric AI launched by Andrew Ng, we propose a novel algorithm for dataset enhancement that works well for many existing DNN models to improve robustness. Transferable adversarial examples and 14 kinds of common corruptions are included in our optimized dataset. In the data-centric robust learning competition hosted by Alibaba Group and Tsinghua University, our algorithm came third out of more than 3000 competitors in the first stage while we ranked fourth in the second stage.

## 1. Introduction

Deep learning has set off a revolution in artificial intelligence research and has made remarkable achievements in many fields such as medical diagnosis, autonomous driving, large-scale decision making, etc. However, it's been proved that DNNs are vulnerable to adversarial examples (Szegedy et al. 2013), which are clean samples with imperceptible perturbations that cause a model to make mistakes, posing a serious threat to AI security. For adversarial defense, existing work either modify the model structures themselves, or optimize the process of inference or training, among which adversarial training (Madry et al. 2017; Zhang et al. 2019) proves to be the most effective strategy.

Some works point out that DNNs are also susceptible to common corruptions that widely exist in real-world application scenarios (Zhang et al. 2020; Hendrycks and Dietterich 2019). These corruptions stem from geometric variations of cameras caused by rotation and translation, or some environmental factors like rain, snow, noises, etc. The techniques towards robustness to common corruptions are mainly focus on data augmentation (Hendrycks et al. 2020) and auxiliary training (Zheng et al. 2016; Zhang et al. 2020).

Numerous works have tried to improve robustness of DNN models. However, none of them considered engineering the original dataset to make it more suitable for training robust models, leaving the potential of datasets unexplored. The encouraging result of data-centric AI competition launched by Andrew Ng (url) tells us that it's time to move from model-centric approach to data-centric approach and design reliable, effective, systematic data to furthur stimulate the potential of deep learning.

In this paper, we propose a data-centric algorithm for dataset enhancement to train robust models. Based on the following two conclusions: 1) a DNN model achieves optimal performance when its testing set follow the same data distribution as its training set; 2) maliciously perturbed data and benign data come from different distributions (Song et al. 2017; Samangouei, Kabkab, and Chellappa 2018), we believe that the optimized training set should also contain adversarial examples and corrupted samples drawn from the corresponding distribution. Therefore, the basic framework of our algorithm is quit simple — we add adversarial perturbations and common corruptions to some randomly selected samples in the original training set. We make use of transferable adversarial examples and as many as 14 kinds of corruptions to further improve the effectiveness. Note that this is different from the techniques exploring extra data which usually result in larger datasets, we keep the number of training samples unchanged. Our algorithm is proposed for participating in the data-centric robust learning competition hosted by Alibaba Group and Tsinghua University (url), in which we beat over 3,000 participants and won the 3rd and 4th place in stage one and stage two respectively. In summary, our main contributions are as follows:

- We propose a simple but effective algorithm to improve deep model's robustness from the perspective of data. This brand new and promising data-centric view largely enrich the research community.
- We study the robustness of DNN models in a challenging but practical setting — adversarial examples and common corruptions both exist in the testing phase while most of the previous work consider them separately.
- The competition and experimental results demonstrated the effectiveness of our algorithm, indicating that the data-centric strategy is feasible for model robustness.

[*] Corresponding author

## 2. Related Works

### Model Robustness

Robustness of DNNs to the perturbations on model inputs is of great concern in trustworthy AI. There are two kinds of perturbations studied in the literature, one is adversarial perturbation (Szegedy et al. 2013) which is a small perturbation that can drastically change the network output while being quasi-imperceptible to humans, and another is common corruption such as rain, snow, Gaussian noise, etc.

The research community has made great efforts to improve model robustness. To defense against adversarial examples, one of the most effective strategies is adversarial training — train a model in an adversarial fashion that continuously generating adversarial examples and then minimizing the loss on these samples (Madry et al. 2017; Zhang et al. 2019). In addition, There are also some works that pursue robust model structures by leveraging ensemble strategies (Lu et al. 2021), NAS (Hosseini, Yang, and Xie 2021), or some well-designed modules for denoising (Xie et al. 2019a), purifying (Shi, Holtz, and Mishne 2020), and malicious sample rejection (Cohen, Sapiro, and Giryes 2020). For common corruptions, existing works mainly focus on optimizing the learning strategies (Hendrycks et al. 2019; Zheng et al. 2016) or data augmentation (DeVries and Taylor 2017). Note that data augmentation can be seen as a data-centric algorithm, but it often result in a larger training set while our algorithm does not.

The above strategies, while effective, are based on fixed datasets or extra training samples. From a complementary perspective, in this work, we show that it's possible to effectively improve model robustness simply by improving existing datasets while without increasing the amount of data.

### Data-Centric AI

Data-centric AI stems from a competition launched by Andrew Ng (url). Different from previous competitions that pursue high-performance models with fixed dataset, this competition fix the models and pursue a high-quality dataset by fixing incorrect labels, applying data augmentations, etc. In this work, we follow a stricter requirement — improve a dataset without increasing the number of samples. For model robustness, previous work focus almost exclusively on models, and now it's the time to exploit the potential of datasets.

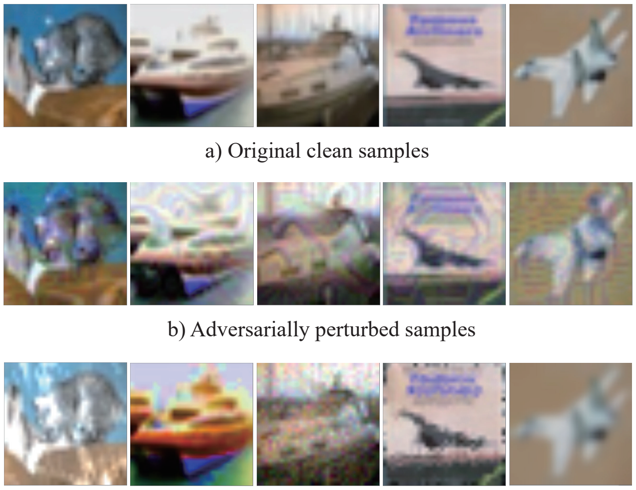
## 3. Proposed Method

In this section, we present our data-centric algorithm for model robustness. We first formalize our optimization goal.

### Problem Formulation

Given a training set $\mathcal{D} = \{(x_1, y_1), \cdots, (x_n, y_n)\}$ consisting of $n$ image-label pairs and a DNN-based classifier $f(\theta; x) : \mathbb{R}^d \to \mathbb{R}^k$ with parameter $\theta$, a standard scheme to train model $f$ is empirical risk minimization (ERM). Let $J(\theta; x, y)$ be the loss function of $\theta$ with input $x$ and one-hot label $y$. Usually, $J$ can be KL divergence, i.e.,

$$J(\theta; x, y) = \mathrm{KL}(f(\theta; x) \| y).$$



a) Original clean samples

b) Adversarially perturbed samples

c) Corrupted samples

Figure 1: Comparison of our generated perturbed samples and the original clean samples.

The objective of ERM on training set $\mathcal{D}$ can be formulated as follows:

$$ERM(\mathcal{D}) = \arg\min_{\theta \in \Theta} \mathbb{E}_{(x,y)\sim\mathcal{D}}[J(\theta; x, y)],$$

where $\Theta$ is the parameter space. Suppose there is a robustness metric $R(f, \theta)$ and a larger value indicates a more robust classifier $f$ with parameter $\theta$, our goal is to develop a data-centric algorithm $\mathcal{A}$ such that $R\left(f, ERM(\mathcal{A}(\mathcal{D}))\right)$ can be maximized and $|\mathcal{A}(\mathcal{D})|$ is not greater than $|\mathcal{D}|$, where $\mathcal{A}(\mathcal{D})$ is the enhanced dataset used to train robust models. We will give the definition of $R(f, \theta)$ in Sec. 4.

### Our Data-centric Algorithm

For a DNN model, to achieve optimal performance in the testing phase, the testing set should follow the same data distribution as our training set. However, numerous works have shown that maliciously perturbed data and benign data come from different distributions, which result in poor performance of DNNs under attack (Song et al. 2017; Samangouei, Kabkab, and Chellappa 2018). This inspires us that our training set should also contain perturbed samples drawn from the corresponding distribution.

In summary, our data-centric algorithm $\mathcal{A}$ can be described as follows. We randomly split the original training set $\mathcal{D}$ into three parts $\mathcal{D}_1$, $\mathcal{D}_2$, and $\mathcal{D}_3$ in a ratio of $\alpha_1 : \alpha_2 : \alpha_3$. For $\mathcal{D}_1$, we keep the original samples unchanged; for $\mathcal{D}_2$, we adversarially perturb each sample; for $\mathcal{D}_3$, we apply a random kind of common corruption to each sample. Finally, $\cup_{i=1}^{3} \mathcal{D}_i$ is our optimized training set.

Next, we introduce our technical details about adversarial perturbations and common corruptions.

**Adversarial perturbation**. A successful data-centric algorithm should be a plug and play approach that works well for any model $f$. Therefore, the generated adversarial perturbations should be transferable, which means they can fool any unknown models that perform the same task. In general, the process of generating adversarial perturbations can be formulated as follows[1]:

$$\delta_{i+1} = \text{Proj}_{\epsilon}\left(\delta_i + \alpha \cdot \text{sign}(\nabla_{\delta_i} J(\theta; x + \delta_i, y))\right),$$

where $\delta_0$ is a randomly initialized perturbation and $\text{Proj}_{\epsilon}(\cdot)$ projects the current $\delta_i$ into the $l_p$ norm ball with radius $\epsilon$. There are many effective techniques in the literature to enhance the transferability of adversarial examples, e.g.,

- *Momentum update* (Dong et al. 2018), which integrates a momentum term into the calculation of gradients to stabilize update directions and avoid poor local optima:

$$\begin{aligned} g_i &= \mu \cdot g_{i-1} + \frac{\nabla_{\delta_i} J(\theta; x + \delta_i, y)}{\|\nabla_{\delta_i} J(\theta; x + \delta_i, y)\|_1}, \\ \delta_{i+1} &= \text{Proj}_{\epsilon}(\delta_i + \alpha \cdot \text{sign}(g_i)). \end{aligned} \tag{1}$$

- *Gradient smoothing* (Dong et al. 2019), which applies Gaussian smoothing to the gradients to weaken their correlation with a particular model:

$$\begin{aligned} \delta_{i+1} = \text{Proj}_{\epsilon}(\delta_i + \alpha \cdot \text{sign}( \\ W * \nabla_{\delta_i} J(\theta; x + \delta_i, y))), \end{aligned} \tag{2}$$

  where $W$ is the Gaussian kernel.

- *Input diversification* (Xie et al. 2019b), which applies random transformations to the input images at each iteration that can be seen as a special kind of data augmentation to avoid overfitting:

$$\begin{aligned} \delta_{i+1} = \text{Proj}_{\epsilon}(\delta_i + \alpha \cdot \text{sign}( \\ \nabla_{\delta_i} J(\theta; T(x + \delta_i, p), y))), \end{aligned} \tag{3}$$

  where $T(x + \delta_i, p)$ is the randomly transformed sample.

- *Logit loss* (Zhao, Liu, and Larson 2021), which can avoid the vanishing gradient problem caused by cross entropy loss:

$$\begin{aligned} J(\theta; x, y) &= -z^t, \\ t &= \arg\max_i y^i. \end{aligned} \tag{4}$$

  where $z$ is the output of the logit layer.

- *Model ensemble* (Dong et al. 2018), which fuses the logits of multiple models together to get the final output:

$$Z_{ensemble} = \frac{1}{|F|} \sum_{f \in F} z_f \tag{5}$$

  where $F$ is the model set, $z_f$ is the logit of model $f$ and $|F|$ is the cardinality of $F$.

Following (Zhao, Liu, and Larson 2021), we combine (1) $\sim$ (5) together to generate highly transferable adversarial perturbations. The iterative formulas are as follows:

$$\begin{aligned} \delta_i &= T(x + \delta_i, p) - x, \\ J(\theta_F; x + \delta_i, y) &= -Z^t_{ensemble}, \\ g_i &= \mu \cdot g_{i-1} + \frac{\nabla_{\delta_i} J(\theta_F; x + \delta_i, y)}{\|\nabla_{\delta_i} J(\theta_F; x + \delta_i, y\|_1}, \\ \delta_{i+1} &= \text{Proj}_{\epsilon}(\delta_i + \alpha \cdot \text{sign}(W * g_i)). \end{aligned} \tag{6}$$

To further obtain more diverse perturbation patterns, we generate both $l_2$-bounded and $l_\infty$-bounded perturbations for each sample. Next, we add them together to get the final adversarial perturbation. In Fig. 1b, we show some instances of our generated adversarial examples.

**Common corruption**. Using the imgaug library (Jung et al. 2020), we implement 14 kinds of common corruptions including rain, snow, frost, Gaussian blur, Gaussian noise, elastic transformation, etc. For each sample in $\mathcal{D}_3$, we randomly select one of the 14 methods to corrupt it. In Fig. 1c, we show some instances of our corrupted samples.

# 4. Experiments

In this section, we will empirically demonstrate the effectiveness of our designed data-centric algorithm $\mathcal{A}$, which is proposed for participating in the data-centric robust learning competition hosted by Alibaba Group and Tsinghua University as one of the series of AI Security Challengers Program (url). We first introduce this competition as well as the robustness metric to evaluate an algorithm.

## Data-Centric Robust Learning Competition

In this competition, we need to optimize the CIFAR-10 dataset using our data-centric algorithm $\mathcal{A}$. Based on this dataset, we are able to train some robust models. To evaluate the robustness of these models, the competition constructed a private testing set $\mathcal{P} = \{\mathcal{P}_{ori}, \mathcal{P}_{adv}, \mathcal{P}_{cor}\}$ based on CIFAR-10. $\mathcal{P}$ consists of three subdatasets which contain clean samples, adversarial examples and corrupted samples respectively[2]. For a particular model $f$ with parameter $\theta$, its robustness $R(f, \theta)$ is defined as the classification rate on $\mathcal{P}$, which can be formulated as follows[3]:

$$R(f, \theta) = \frac{1}{|\mathcal{P}|} \left( \sum_{\mathcal{P}_i \in \mathcal{P}} \frac{1}{|\mathcal{P}_i|} \sum_{(x_i, y_i) \in \mathcal{P}_i} \mathbf{1}(f(\theta, x_i) = y_i) \right). \tag{7}$$

This competition consists of two stages. In each stage, we are given several DNN models and we need to train these models on our optimized dataset. The trained models are subsequently submitted to the competition platform. Finally, our score is calculated as the mean of $R$ for each model based on Eq. (7). In the first stage, the models to be trained are ResNet50 (He et al. 2016a) and DenseNet121 (Huang et al. 2017). The score of the baseline dataset is 75.23. We came third out of 3691 participants with a score of 98.96. In the second stage, the models to be trained are WideResNet (Zagoruyko and Komodakis 2016) and PreactResNet18 (He et al. 2016b). The models are evaluated on a different private testing set and the baseline is 63.31. We came fourth out of 50 participants with a score of 85.19.

## Comparison With the Baseline

Exploiting data-centric AI for model robustness is a brand new idea with no competitive method in the literature.

[1] Note that this is the formulation of $l_\infty$-bounded PGD attack. For $l_2$-bounded PGD attack, we should replace $\text{sign}(\cdot)$ with $\text{norm}(\cdot)$, where $\text{norm}(v) = \frac{v}{\|v\|_2}$.

[2] The clean samples in $\mathcal{P}$ are not necessary selected from the original CIFAR-10 dataset.

[3] Here, the label $y$ and the output of $f$ are all scalars representing the class indexs, which are different from the definitions in Seq. 3.

Table 1: Performance comparison of models trained on original CIFAR-10 and our optimized CIFAR-10. $ACC(\cdot)$ stands for the classification rate on some subdataset and $R(f, \theta)$ stands for the robustness score of model $f$ with parameter $\theta$.

| | **Performances of models trained on original CIFAR-10 (%)** | | | | | |
|---|---|---|---|---|---|---|
| | ResNet50 | WideResNet | PreactResNet18 | DenseNet121 | VGG16 | MobileNetV2 |
| $ACC(\mathcal{P}_{ori})$ | 98.64 | 99.28 | 98.66 | 98.63 | 98.66 | 90.91 |
| $ACC(\mathcal{P}_{adv})$ | 32.73 | 34.28 | 33.75 | 31.84 | 37.17 | 43.25 |
| $ACC(\mathcal{P}_{cor})$ | 55.12 | 61.32 | 62.78 | 56.71 | 64.17 | 56.67 |
| $R(f, \theta)$ | 62.16 | 64.96 | 65.06 | 62.39 | 66.67 | 63.61 |
| | **Performances of models trained on optimized CIFAR-10 (%)** | | | | | |
| | ResNet50 | WideResNet | PreactResNet18 | DenseNet121 | VGG16 | MobileNetV2 |
| $ACC(\mathcal{P}_{ori})$ | 92.75 | 96.30 | 93.30 | 95.04 | 94.94 | 84.74 |
| $ACC(\mathcal{P}_{adv})$ | 60.52 | 62.42 | 58.34 | 61.50 | 58.86 | 46.99 |
| $ACC(\mathcal{P}_{cor})$ | 83.36 | 89.58 | 83.93 | 85.54 | 86.93 | 72.58 |
| $R(f, \theta)$ | **78.88** | **82.77** | **78.52** | **80.69** | **80.24** | **68.10** |

Therefore, our baseline algorithm for comparison is simply the identity map $\mathcal{I}$, i.e., the original CIFAR-10 is the baseline dataset to train non-robust models. Note that the number of training samples contained in our optimized dataset should not exceed 50,000 during the competition, which is less than the total number of samples in CIFAR-10 (includes training set and testing set). As what we did in the competition, we randomly remove 10,000 samples from CIFAR-10 before our experiments.

Using the proposed algorithm $\mathcal{A}$ described in Sec. 3, we generate our optimized training set based on the original CIFAR-10. In practice, the split ratio $\alpha$ is set as $0 : 1 : 4$, which yields the best performance in the competition. This ratio indicates that there are no clean samples in our optimized training set. This is because the proportion of clean samples in the private testing set $\mathcal{P}$ is relatively small (4.55% in the second stage), and also, the perturbed samples contain some information about the distribution of clean samples. We generate transferable adversarial examples through 300 iterations. The hyper-parameters setting in the experiments is: $\epsilon = 8$, $\alpha = 2/255$ for $l_\infty$-bounded perturbations and $\epsilon = 1.0$, $\alpha = 0.025$ for $l_2$-bounded perturbations. Other hyper-parameters are consistent with those in (Zhao, Liu, and Larson 2021). To implement model ensemble, we select 5 common DNN models, including ResNet50, WideResNet, DenseNet121, VGG16 (Simonyan and Zisserman 2014) and MobileNetV2 (Sandler et al. 2018). Among them, ResNet50 and DenseNet121 have been demonstrated to be the best choices to generate transferable adversarial examples.

We trained 6 different models on both the original CIFAR-10 and our optimized CIFAR-10. The private testing sets of the competition have been released. Therefore, we simply use the testing set of the second stage to evaluate the performances of our trained models. The classification rates $ACC$ on three subdatasets and the robustness score $R(f, \theta)$ of each model are shown in Tab. 1. The models trained on our optimized dataset show significant improvements in terms of robustness, and the classification rates on both adversarial and corrupt samples increase by more than 20%, which proves the effectiveness and general applicability of our algorithm. In practice, we can trade off the performance loss on clean samples and the robustness to perturbations by adjusting the split ratio $\alpha$.

### Limitation and Future Work

Although our algorithm effectively improves the robustness of models to adversarial and corrupted samples, the magnitude of improvements are not comparable to current state-of-the-art models-centric techniques. On one hand, our study is based on a more challenging setting — both adversarial examples and corrupted samples exist in the testing set. On the other hand, when the number of samples in the training set cannot be increased, it's difficult to carry enough information about the distributions of clean samples and perturbed samples. While only limited performance gains can be obtained by optimizing the datasets alone, it will be interesting to explore whether further breakthroughs can be achieved by combining data-centric approaches with state-of-the-art model-centric approaches, which is a promising future work.

## 5. Conclusion

In this paper, we show the possibility of constructing a high-quality dataset for model robustness by presenting a novel data-centric algorithm. Our competition and experimental results demonstrate the effectiveness and general applicability of the algorithm. Data-centric AI is a promising approach in the field of model robustness, and we believe that more encouraging results can be achieved by combining it with existing state-of-the-art model-centric techniques.

## 6. Acknowledgments

This work was supported by National Natural Science Foundation of China under Grants 61922027, 6207115 and 61932022. We thank the security AI challenger program launched by Alibaba Group and Tsinghua University. We'd also like to thank Xiong Zhou, Chenyang Wang and Xingyu Hu for their kindness help during the competition.

# References

Cohen, G.; Sapiro, G.; and Giryes, R. 2020. Detecting adversarial samples using influence functions and nearest neighbors. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 14453–14462.

DeVries, T.; and Taylor, G. W. 2017. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*.

Dong, Y.; Liao, F.; Pang, T.; Su, H.; Zhu, J.; Hu, X.; and Li, J. 2018. Boosting adversarial attacks with momentum. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 9185–9193.

Dong, Y.; Pang, T.; Su, H.; and Zhu, J. 2019. Evading defenses to transferable adversarial examples by translation-invariant attacks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4312–4321.

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

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016b. Identity mappings in deep residual networks. In *European conference on computer vision*, 630–645. Springer.

Hendrycks, D.; and Dietterich, T. G. 2019. Benchmarking Neural Network Robustness to Common Corruptions and Perturbations. In *International Conference on Learning Representations*.

Hendrycks, D.; Mazeika, M.; Kadavath, S.; and Song, D. 2019. Using self-supervised learning can improve model robustness and uncertainty. *Advances in Neural Information Processing Systems*, 32.

Hendrycks, D.; Mu, N.; Cubuk, E. D.; Zoph, B.; Gilmer, J.; and Lakshminarayanan, B. 2020. AugMix: A Simple Data Processing Method to Improve Robustness and Uncertainty. In *International Conference on Learning Representations*.

Hosseini, R.; Yang, X.; and Xie, P. 2021. Dsrna: Differentiable search of robust neural architectures. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6196–6205.

Huang, G.; Liu, Z.; Van Der Maaten, L.; and Weinberger, K. Q. 2017. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4700–4708.

Jung, A. B.; Wada, K.; Crall, J.; Tanaka, S.; Graving, J.; Reinders, C.; Yadav, S.; Banerjee, J.; Vecsei, G.; Kraft, A.; Rui, Z.; Borovec, J.; Vallentin, C.; Zhydenko, S.; Pfeiffer, K.; Cook, B.; Fernández, I.; De Rainville, F.-M.; Weng, C.-H.; Ayala-Acevedo, A.; Meudec, R.; Laporte, M.; et al. 2020. imgaug. https://github.com/aleju/imgaug. Online; accessed 01-Feb-2020.

Lu, K.; Nguyen, C. M.; Xu, X.; Chari, K.; Goh, Y. J.; and Foo, C.-S. 2021. ARMOURED: Adversarially Robust MOdels using Unlabeled data by REgularizing Diversity. In *ICLR*.

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*.

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

Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; and Chen, L.-C. 2018. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4510–4520.

Shi, C.; Holtz, C.; and Mishne, G. 2020. Online adversarial purification based on self-supervised learning. In *International Conference on Learning Representations*.

Simonyan, K.; and Zisserman, A. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

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

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

Xie, C.; Wu, Y.; Maaten, L. v. d.; Yuille, A. L.; and He, K. 2019a. Feature denoising for improving adversarial robustness. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 501–509.

Xie, C.; Zhang, Z.; Zhou, Y.; Bai, S.; Wang, J.; Ren, Z.; and Yuille, A. L. 2019b. Improving transferability of adversarial examples with input diversity. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2730–2739.

Zagoruyko, S.; and Komodakis, N. 2016. Wide residual networks. *arXiv preprint arXiv:1605.07146*.

Zhang, H.; Yu, Y.; Jiao, J.; Xing, E.; El Ghaoui, L.; and Jordan, M. 2019. Theoretically principled trade-off between robustness and accuracy. In *International conference on machine learning*, 7472–7482. PMLR.

Zhang, L.; Yu, M.; Chen, T.; Shi, Z.; Bao, C.; and Ma, K. 2020. Auxiliary Training: Towards Accurate and Robust Models. In *Computer Vision and Pattern Recognition*.

Zhao, Z.; Liu, Z.; and Larson, M. 2021. On success and simplicity: A second look at transferable targeted attacks. *Advances in Neural Information Processing Systems*, 34.

Zheng, S.; Song, Y.; Leung, T.; and Goodfellow, I. 2016. Improving the robustness of deep neural networks via stability training. In *Proceedings of the ieee conference on computer vision and pattern recognition*, 4480–4488.