EDFAR: Exploring Data Augmentation for Adversarial Robustness

Sayali Jathar
New York University
snj4459@nyu.edu

Shantanu Kumar
New York University
sk9698@nyu.edu

Mahati Madhira VSK
New York University
mm12032@nyu.edu

Abstract

Adversarial training is prone to robust overfitting[1], a situation where the model's ability to perform well on test data that has been perturbed or modified in some way starts to deteriorate as the model is trained. In this paper, we want to compare the performance of our Deep Learning model against adversarial attacks with and without heuristic-driven Data Augmentation and Data Corruption techniques to test the hypothesis "Does Data Augmentation and Corruption have an effect in improving Adversarial Robustness against common perturbations?" Firstly, we trained our baseline model on the CIFAR-10 dataset and reached an accuracy of 80.26% which degraded to 31.53% when an FGSM adversarial attack was performed on it with eps=0.05. Next, we gathered augmented data using various heuristic-driven Data augmentation techniques like CutOut, MixUp, CutMix, AugMix and RandAug and retrained the model over it. The accuracy of the model was then tested by performing an FGSM attack whose results are listed in Tables 1 and 2. The GitHub repository link to this paper is: https://github.com/n0vay/EDFAR

Keywords: *Data Augmentation, Adversarial training, Adversarial attack, Comparative study, Classification*

1 Introduction

Adversarial robustness refers to the ability of a machine learning model to maintain good performance when it is presented with inputs that have been intentionally modified to be difficult or confusing for the model. These modifications, known as adversarial examples, are designed to trick the model into making mistakes or giving incorrect predictions. Adversarial robustness is important because it allows a model to continue to perform well on real-world data, which may contain noise, errors, or other types of variations.

Adversarial training is a method for improving the robustness of machine learning models by intentionally introducing adversarial examples into the training data. During adversarial training, the model is presented with both normal training examples and adversarial examples, and it is trained to classify both correctly. This helps the model to learn to recognize and resist adversarial examples, and can improve its overall robustness and performance on real-world data. Data augmentation is a technique used to artificially increase the size of a dataset by generating additional, synthetic data samples. The goal of data augmentation is to improve the generalization of machine learning models by providing them with more diverse training data. This can help to prevent overfitting and improve the model's ability to perform well on unseen data. Data Corruption is a technique of introducing common perturbations into a dataset to test the ability of a model to handle these perturbations in a controlled manner. We introduced 6 different corruptions into CIAFR-10 dataset and trained our model over these corrupted datasets to test its robustness against adversarial attacks. Models with high adversarial robustness should be able to maintain good performance on these examples, while

models with low adversarial robustness may be easily fooled and give incorrect predictions. In particular, it has been shown that the addition of imperceptible deviations to the input, called adversarial perturbations, can cause neural networks to make incorrect predictions with high confidence. In this paper, We wanted to answer the question:

"Does Data Augmentation and Corruption have any effect on improving Adversarial Robustness? If it does, how much improvement in performance can different techniques bring?

We propose to combine augmented data obtained from different data augmentation techniques with original data to compare the improvement in performance that these techniques bring to our model. Overall, we made the following contributions:

- We show that using heuristic-based data augmentation techniques, such as Cutout, CutMix, MixUp, AugMix and RandAug can improve the adversarial robustness of deep learning models.

- We show the performance of our model against FGSM attack when trained with CIFAR-10-C which is corrupted dataset that has been perturbed using 6 different techniques: Snow, Frost, Impulse noise, Gaussian noise, JPEG compression, GlassBlur.

- Contrary to the findings of Rice et al., Wu et al., and Gowal et al., [1]who all attempted to use data augmentation techniques without success, we were able to achieve robust accuracies by using any of the aforementioned data augmentation techniques (Cutout, CutMix, and MixUp, AugMix and RandAug). We found that CutMix and MixUp were the most effective methods, resulting in robust accuracies of 26.82% and 22.61% on the CIFAR-10 dataset when using perturbations of size eps = 0.20, an improvement of 50% and 40% over the baseline accuracy of 13.73%.

2 Related Work

2.1 Adversarial Training as a Defense

Adversarial training, a method developed by Madry et al. in 2018[9], involves using adversarially modified examples in the training data to improve the robustness of deep neural networks. It is considered to be one of the most effective methods for training robust models. There have been many variations of adversarial training developed over time, including modifications to the attack procedure, loss function, and model architecture. Some examples of these variations include incorporating momentum in the attack procedure, using a logit pairing loss function, and adding feature denoising to the model architecture. Recently, the work of Rice et al. in 2020[8] explored the issue of robust overfitting and found that improvements similar to those obtained using TRADES could be achieved more easily using classical adversarial training with early stopping. This study also revealed that early stopping was competitive with other regularization techniques and demonstrated that data augmentation techniques beyond the usual method of random padding and cropping were ineffective on the CIFAR-10 dataset. In addition, Gowal et al. in 2020 [9]investigated how different hyperparameters, such as network size and model weight averaging, affected robustness and were able to achieve significant improvements over the state-of-the-art. However, their study did not include a thorough investigation of data augmentation techniques and reached the same conclusion as Rice et al., that data augmentation techniques beyond random padding and cropping do not improve robustness.[1]

2.2 Heuristics-driven Data Augmentation

Data augmentation has been shown to reduce the generalization error of standard (non-robust) training, particularly for image classification tasks. Heuristic-driven data augmentation is a method for improving the performance and robustness of machine learning models by artificially increasing the size and diversity of the training dataset. This is typically done by applying specific, pre-defined transformations to the training data, such as cropping, scaling, rotating, or adding noise. The goal of heuristic-driven data augmentation is to expose the model to a wider range of variations and variations that may be present in the real-world data, in order to improve its generalization performance and robustness. More advanced techniques such as Cutout (DeVries and Taylor, 2017)[10], CutMix (Yun et al., 2019)[11], and MixUp (Zhang et al.,2018a)[12] have also been shown to be highly ef-

fective. However, it is surprising that these techniques have not been found to be effective when training adversarially robust networks.

3 Methodology

3.1 Model

The ResNet-18 model consists of 18 layers, including 13 convolutional layers and 5 max-pooling layers. It also includes skip connections, which allow the model to bypass intermediate layers and directly combine the outputs of earlier layers with the outputs of later layers. This helps to improve the model's ability to learn complex functions and reduce the risk of vanishing gradients.

3.2 Dataset

The CIFAR-10 dataset is a widely used dataset for image classification tasks. It consists of 60,000 32x32 color training images and 10,000 test images, with 10 classes (airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck). The images are evenly divided among the 10 classes, with 6,000 images per class in the training set and 1,000 images per class in the test set.

3.3 Adversarial Training

Adversarial training is a method for improving the robustness of machine learning models by intentionally introducing adversarial examples into the training data. Adversarial examples are inputs that have been modified in some way to be difficult or confusing for the model, and they are designed to trick the model into making mistakes or giving incorrect predictions. During adversarial training, the model is presented with both normal training examples and adversarial examples, and it is trained to classify both correctly. This helps the model to learn to recognize and resist adversarial examples and can improve its overall robustness and performance on real-world data. Adversarial training can be implemented in a variety of ways, and there are many different methods for generating adversarial examples. Some common approaches include adding small, imperceptible perturbations to the input data, or using optimization techniques to find the input modifications that are most likely to cause the model to make mistakes. Adversarial training is an active area of research in machine learning, and there are many techniques have been developed to improve the effectiveness of this approach.

3.4 Adversarial Robustness

Adversarial robustness refers to the ability of a machine learning model to maintain good performance when it is presented with inputs that have been intentionally modified to be difficult or confusing for the model. These modifications, known as adversarial examples, are designed to trick the model into making mistakes or giving incorrect predictions. Adversarial robustness is important because it allows a model to continue to perform well on real-world data, which may contain noise, errors, or other types of variations.

In general, adversarial robustness is a measure of the model's ability to resist being fooled or confused by adversarial examples. This can be evaluated by testing the model on a set of adversarial examples and measuring its performance. Models with high adversarial robustness should be able to maintain good performance on these examples, while models with low adversarial robustness may be easily fooled and give incorrect predictions. Adversarial robustness is an active area of research in machine learning, and there are many techniques that have been developed to improve the robustness of machine learning models.

3.5 Data Augmentation

Data augmentation is a technique used to artificially increase the size of a dataset by generating additional, synthetic data samples. The goal of data augmentation is to improve the generalization of machine learning models by providing them with more diverse training data. This can help to prevent overfitting and improve the model's ability to perform well on unseen data.

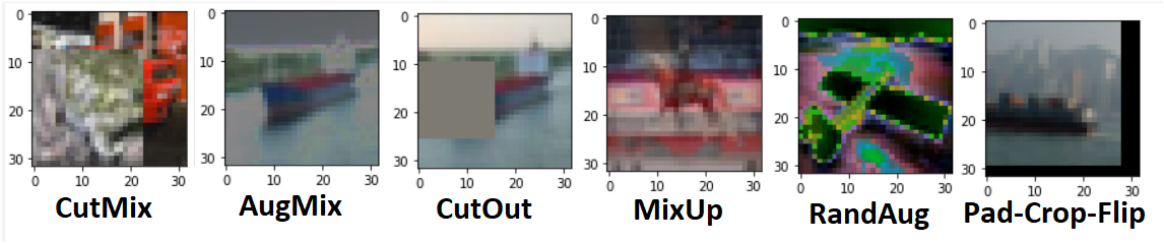


Figure 1
Data Augmentations

There are many ways to generate synthetic data samples for data augmentation. Some common techniques include applying random transformations to existing data samples, such as cropping, scaling, rotating, or adding noise. Other methods involve synthesizing new data samples based on the underlying distribution of the original data, such as using generative models or interpolating between existing data points.

Data augmentation can be especially useful for training machine learning models in situations where the amount of available training data is limited. It can also be useful for improving the robustness of models by exposing them to a wider range of variations and variations that may be present in the real-world data.

3.5.1 CutOut

Cutout is a data augmentation technique that involves randomly masking out a portion of an image during training. To apply cutout to an image, a rectangular region is selected at a random location within the image and the pixels within that region are replaced with a constant value, such as black or the mean pixel value of the image. This creates a ”cutout” or ”hole” in the image that the model must learn to ignore during training.The idea behind cutout is to create additional variability in the training data by forcing the model to learn to recognize objects and patterns in different contexts, which can improve its generalization performance.

3.5.2 MixUp

Mixup is a data augmentation technique that involves generating synthetic training examples by linearly interpolating pairs of examples from the training set. Given a pair of training examples (x1, y1) and (x2, y2), Mixup generates a new synthetic example (x’, y’) by sampling a convex combination of the two original examples, according to a coefficient alpha drawn from a Beta distribution:

$$x' = \text{alpha} * x1 + (1 - \text{alpha}) * x2$$

$$y' = \text{alpha} * y1 + (1 - \text{alpha}) * y2$$

The goal of Mixup is to encourage the model to learn a smoother, more generalized decision boundary, as opposed to one that is overly sensitive to individual training examples. This can help to reduce overfitting and improve the generalization performance of the model.

3.5.3 CutMix

CutMix is a data augmentation technique that involves combining two or more images to create a new image. This can be done by randomly selecting a rectangular region from each image and combining them to create the new image. The resulting image will have elements from both of the original images, and can be used to train machine learning models.

One benefit of using CutMix as a data augmentation technique is that it can help to improve the generalization performance of the model by providing it with additional examples that are constructed from a combination of existing examples. This can be particularly useful in situations where the training dataset is limited, as it allows the model to learn from a larger number of examples without having to actually collect additional data.

To use CutMix as a data augmentation technique, you would typically first select a pair of images to combine and then randomly select a rectangular region from each image. You would then combine the two regions to create the new image, and use this new image as part of the training process for your machine learning model. You can repeat this process multiple times to generate a larger dataset for training.

3.5.4 AugMix

AugMix is a data augmentation technique that involves generating new data samples by applying a combination of multiple augmentation techniques to a base image. The idea behind AugMix is to create a more diverse set of augmented images, which can improve the generalization performance of machine learning models.

To generate an AugMix image, the base image is first transformed using a set of randomly chosen augmentation techniques, such as color jitter, affine transforms, and image cropping. These transformed images are then combined using a random mixup ratio to create the final AugMix image.

The use of multiple augmentation techniques and the mixup ratio allows AugMix to create a wide range of augmented images, which can help to improve the robustness of machine learning models. AugMix has been shown to be effective at improving the performance of image classification and object detection models, and has been used in a number of state-of-the-art approaches for these tasks.

3.5.5 RandomAugment

It involves randomly applying a set of predefined image transformations to a dataset in order to increase the diversity and size of the dataset. These transformations include operations such as cropping, padding, rotating, flipping, and applying color and brightness transformations.

One of the main benefits of using RandAugment is that it can significantly improve the generalization performance of deep learning models, especially when combined with other regularization techniques such as dropout and weight decay.

3.6 Data Corruption

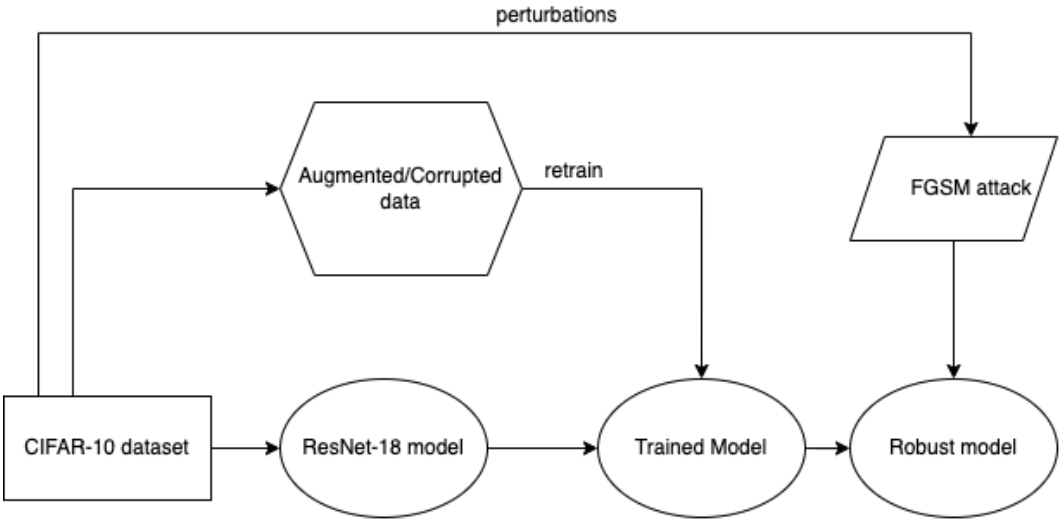
Figure 2
Different types of Corruptions

- **Gaussian noise** : Gaussian noise data corruption is typically done by adding a noise signal with a mean of zero and a standard deviation that is proportional to the signal being corrupted. The noise signal is added to each sample in the dataset, resulting in corrupted data that has the same size and shape as the original data, but with added noise.
- **Impulse noise** : Impulse noise perturbation is a type of data perturbation that involves the introduction of noise to the images in the dataset, or by introducing missing values or other changes to the data. Testing the robustness of a model trained on the CIFAR-10 dataset to impulse noise perturbation can be useful in evaluating its ability to handle real-world errors and changes in the data.
- **Glass blur** : Glass blur perturbation on the CIFAR-10 dataset involves adding a blur effect to the images in the dataset, simulating the effect of looking through a distorted or blurry

lens. This type of data perturbation can be challenging for models trained on the CIFAR-10 dataset to handle, as it can significantly impact the accuracy and performance of the model.

- **Snow** : Snow perturbation on the CIFAR-10 dataset involves adding noise or "snow" to the images in the dataset. This can be achieved by adding random pixel values to the images, or by applying a noise filter to the images.
- **Frost** : Frost perturbation on the CIFAR-10 dataset involves the introduction of small, randomized perturbations to the data. These perturbations are designed to mimic the effects of frost or other types of noise on an image.
- **JPEG Compression** : JPEG compression reduces the size of an image by eliminating certain details and reducing the number of colors used to represent the image. This can result in lossy compression, where some of the original information in the image is lost.

3.7 Process Flow



(a)

Figure 3
Process flow for our project

4 Results

4.1 Data Corruption

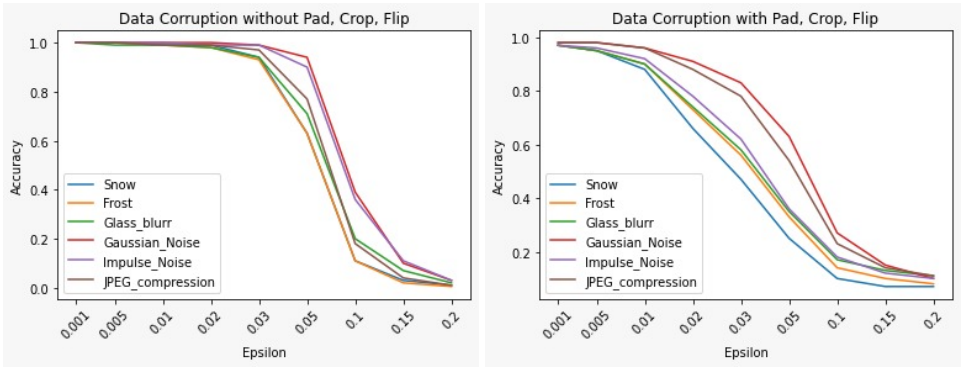
When implementing Data corruption on CIFAR-10 dataset without PCF (Fig.4(a)), we observed that the model maintains very high accuracy till eps=0.03 and then, there is a very sharp descent in the accuracy till eps=0.1 from 90% to 15%. Similar behaviour can be observed with PCF but the drop in accuracy starts much earlier and there is a steady decline in accuracy.

For both variations, Gaussian Noise Corruption was best in performance while Frost Corruption performed worst without PCF and Snow performed worst with PCF.

4.2 Data Augmentation

We observed that model with PCF has higher accuracies exceeding one without PCF by 8%. Re-training the model with Augmented data also increased the performance of the model by 3-4%.

For PCF, an average accuracy of 90% was achieved while accuracy for non-PCF model was around 83%.



(a) (b)

Figure 4
(a)Data Corruption without PCF (b) Data Corruption with PCF

Corruption	Accuracy	Epsilon								
		0.001	0.005	0.01	0.02	0.03	0.05	0.1	0.15	0.2
Snow PCF	0.961	0.97	0.95	0.88	0.66	0.47	0.25	0.10	0.07	0.07
Frost PCF	0.94	0.97	0.95	0.90	0.73	0.56	0.33	0.14	0.10	0.08
Glass blurr PCF	0.95	0.97	0.95	0.90	0.74	0.58	0.35	0.17	0.13	0.11
Gaussian Noise PCF	0.95	0.98	0.98	0.96	0.91	0.83	0.63	0.27	0.15	0.10
Impulse Noise PCF	0.98	0.97	0.96	0.92	0.78	0.62	0.36	0.18	0.12	0.10
JPEG compression PCF	0.97	0.98	0.98	0.96	0.88	0.78	0.54	0.23	0.14	0.11

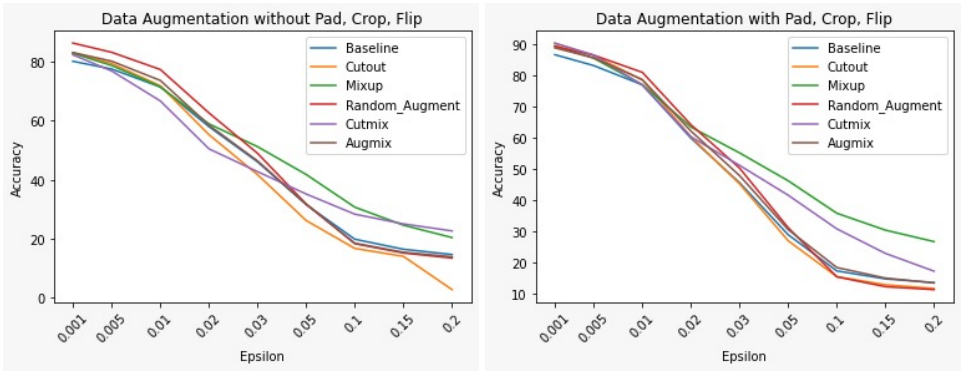
Table 1
Comparison of test accuracies for various Data Corruption techniques with PCF

After performing adversarial attack on the model, for PCF we saw that the MixUp augmentation was performing best with an accuracy of 35% which is 48% better than the baseline model's accuracy for eps=0.1. Similar results can be seen for eps=0.1 and eps=0.2. CutMix PCF was the second best performer with an accuracy of 30.9% for eps=0.1.

CutMix performs best for model without PCF as well for eps=0.2 with an accuracy of 22.6% while baseline has an accuracy of 14.6% which is 37% better. While MixUp performed better for lower eps values with an accuracy gain of 30% over baseline.

Augmentation	Accuracy	Epsilon								
		0.001	0.005	0.01	0.02	0.03	0.05	0.1	0.15	0.2
Baseline PCF		86.58	83.1	76.95	59.89	45.64	29.01	17.45	14.91	13.73
Cutout PCF	90.58	90.18	86.31	78.49	60.35	45.26	27.09	15.62	13.03	11.85
Mixup PCF	90.7	89.39	85.45	76.99	63.45	55.18	46.27	35.89	30.49	26.82
Cutmix PCF	90.7	90.3	86.45	76.85	60.31	51.2	41.65	30.9	23.05	17.37
Augmix PCF	90.2	88.76	85.57	78.6	62.16	47.92	30.65	18.53	15.16	13.6
Rand Augment PCF	90.83	89.27	86.43	80.93	64.18	50.32	31.26	15.57	12.41	11.47

Table 2
Comparison of test accuracies for various Data Augmentation techniques with PCF



(a) (b)

Figure 5
(a)Data Augmentation without PCF (b) Data Augmentation with PCF

Augmentation	Accuracy	Epsilon								
		0.001	0.005	0.01	0.02	0.03	0.05	0.1	0.15	0.2
Baseline		80.11	77.48	71.4	57.95	46	31.53	19.8	16.4	14.62
Cutout	83.16	82.92	79.33	71.81	55.23	41.66	26.13	16.68	13.99	2.72
Mixup	83.25	82.81	78.67	71.44	58.89	51.16	41.73	30.66	24.56	20.36
Cutmix	84.19	82.3	76.82	66.66	50.43	42.7	35.11	28.29	25	22.61
Augmix	83.3	83.08	80.13	73.61	58.44	46.33	31.62	18.45	15.39	13.85
Random Augment	87.63	86.28	83.14	77.31	62.6	48.9	31.73	18.3	15.13	13.41

Table 3
Comparison of test accuracies for various Data Augmentation techniques without PCF

5 Future Works

There are further Data Augmentation techniques such as AutoAugment, TrivialAugmentWide, and RandomEqualize as well as Data Corruption techniques such as Shot Noise, Defocus blur, Pixelate and Elastic that we are yet to implement.

Combining Augmentation and Corruption techniques is also an unexplored possibility for improving Adversarial robustness.

We are also planning to perform other Adversarial attacks like the Projected Gradient Descent (PGD) attack and Backdoor attacks.

6 Conclusion

In this project, we made use of Data Augmentation and Corruption techniques to train models to be adversarially robust.

We have observed that combining different methods such as Pad/Crop/Flip along with other techniques such as MixUp, CutMix and Glass Blur can significantly boost the performance of the model in both normal and adversarial testing.

References

[1] Fixing Data Augmentation to Improve Adversarial Robustness Sylvestre-Alvise Rebuffi, Sven Gowal, Dan A. Calian, Florian Stimberg, Olivia Wiles, Timothy Mann

[2] Benchmarking Neural Network Robustness to Common Corruptions and Perturbations Dan Hendrycks, Thomas Dietterich

[3] Towards Deep Learning Models Resistant to Adversarial Attacks Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, Adrian Vladu

[4] https://blog.roboflow.com/what-is-cutout-augmentation-and-when-can-it-help

[5] Ground-Truth Adversarial Examples Nicholas Carlini, Guy Katz, Clark Barrett, David L. Dill

[6] Benchmarking Neural Network Robustness to common corruptions and perturbations Dan Hendrycks, Thomas Dietterich

[7] ndriushchenko, M., Croce, F., Flammarion, N., and Hein, M. Square Attack: a query-efficient black-box adversarial attack via random search. Eur. Conf. Comput. Vis., 2020.

[7] Carlini, N. and Wagner, D. Towards evaluating the robustness of neural networks. In 2017 IEEE Symposium on Security and Privacy, 2017b.

[9] adry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. Towards deep learning models resistant to adversarial attacks. Int. Conf. Learn. Represent., 2018.

[8] Rice, L., Wong, E., and Kolter, J. Z. Overfitting in adversarially robust deep learning. Int. Conf. Mach. Learn., 2020.

[9] Gowal, S., Qin, C., Uesato, J., Mann, T., and Kohli, P. Uncovering the limits of adversarial training against norm-bounded adversarial examples.

[10] DeVries, T. and Taylor, G. W. Improved regularization of convolutional neural networks with cutout.

[11] Yun, S., Han, D., Oh, S. J., Chun, S., Choe, J., and Yoo, Y. Cutmix: Regularization strategy to train strong classifiers with localizable features.

[12] Zhang, H., Cisse, M., Dauphin, Y. N., and Lopez-Paz, D. mixup: Beyond empirical risk minimization.