EE-411: Fundamentals of inference and learning **2021-2022**

Paper 4: Augmix

Lecturer: Florent Krzakala

Students: Seyed Khashayar Najafi, Soroush Mehdi, Gojko Cutura

Code: https://github.com/gojkoc54/augmix

***Abstract*—In this paper a data processing technique that is simple to implement, adds limited computational overhead, and helps models withstand unforeseen corruptions has been proposed. This technique increases the robustness of the image classifiers towards test data with different distribution to train data. This is done by performing different augmentations on input image data and mixing them up for training the models, hence the name AUGMIX. In this report, we reproduce some of the tables and figures form the original paper and add other structures to them. Also we take a critical view towards hyper-parameters and their effect on the final trained model.**

I. Introduction

Overconfident predictions is the main problem with modern modern deep neural networks. They achieve high accuracy when train and test data have the same distribution, but in practical cases it rarely happens that test data has identical distribution as training data. And When the train and test distributions are mismatched, accuracy can plummet. Small corruptions to the data distribution are enough to subvert existing classifiers, and techniques to improve corruption robustness remain few in number.

Data augmentation can greatly improve generalization performance. For image data, random left-right flipping, cropping, CutOut, MixUp and CutMix can be used for augmentation. However, Mixing augmentations allows us to generate diverse transformations, which are important for inducing robustness, as a common failure mode of deep models in the arena of corruption robustness is the memorization of fixed augmentations.

In the original paper CIFAR-10 and CIFAR-100 were used to train the classifiers. They have respectively images of 10 and 100 classes of objects. However, CIFAR-100 takes a great amount of time to train so we only used CIFAR-10. To test the robustness of models, images from CIFAR-10 were corrupted manually and created train set of CIFAR-10-C.

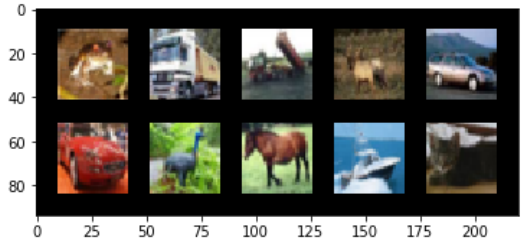
II. Models and Methods

A. Data Augmentation

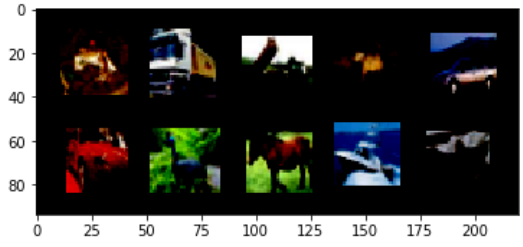
As mentioned, data augmentation is a technique which uses different transformations and can improve generalization performance effectively. In these techniques, images are transformed with known transforms as Cropping, Rotating, Polarize, Cutting, etc. The transformed picture using different algorithms can get mixed to provide a better training set for the learning network. In order to reach this purpose, 13 different transformations (Auto-contrast, Equalize, Posterize, Rotate, Solarize, Shearing in x and y axis, Translating vertically and horizontally , Changing color, Changing contrast, Changing brightness and Changing sharpness) are implemented in this work.

B. AUGMIX

AUGMIX is a data augmentation technique which improves models' robustness and uncertainty estimates and slots in easily, to existing training pipelines. In AUGMIX, we send images through different augmentations in parallel branches and sum them up then add the augmented images with the original image with certain weights. The number of augmentations in series for each branch corresponds to AugMix's depth and the number of branches corresponds to its width. In mixing branches, weights are random variables from Drichlet distribution. After we mixed all paths together, we will add the original picture with augmented picture with a weight coming from Beta distribution. To better comprehend the method, a diagram of AugMix technique and an example of AUGMIX technique on CIFAR-10 images can be seen in Fig.1 and Fig.2



(a) Original



(b) AUGMIX

Figure 1: An example of how AUGMIX changes images from CIFAR-10

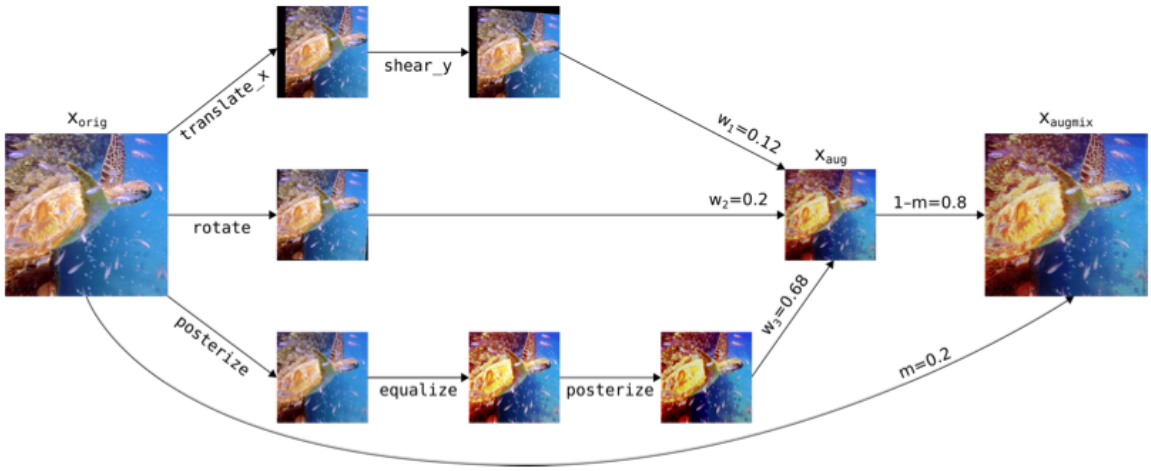


Figure 2: AUGMIX technique

C. Recreating Data from original paper

In the original paper, they used ResNeXt, Wide ResNet, AllConv and DenseNet structures for classifiers. Here we used these structures with width and depth of respectively 3 and 4 (as done originally) to recreate results. Here, however, instead of going for 90 epochs we were able to run 30 epochs. Since we were testing different configurations so the computational load was high, so in order to reduce the computation time we had to reduce the number of epochs.

To see the effect of AUGMIX on different corruptions, we ran all of the aforementioned structures and took the average error for each corruption and compared the result with standard training without AUGMIX. The result can be seen in Fig 3.

We can see that with AUGMIX the model is showing better robustness towards different types of corruption.

To go beyond what the original paper's scope, we have implemented AlexNet for CIFAR-10 data-set. It was not possible to use pytorch built-in network because of the small size of images in CIFAR-10. Putting a transform to reach minimum size of the built-in network was also a solution but it is not logical to up-size an image. Hence, we have implemented the AlexNet from scratch to be able to provide a comparison between standard training and training with AugMix on their performance with corrupted images.

D. Hyper-Parameters Ablation

In the original paper to reach the best configuration for AugMix, they have provided an ablation study over its hyper-parameters. The hyper-parameters studied in the original paper was number of epochs , mixing coefficient, chain depth and mixture width. Due to lack of information given from their study, we created a simulation framework to scrutinize AugMix's hyper-parameters. To investigate as much possible in the least time, we did not study the effect of increasing epochs which is predictable. We have investigated the effect of increasing chain depth and mixture width with different learning models.

III. Results

Our study results on the AugMix algorithm is divided into two parts. The first part is a recreation of the table number 1 in the original paper that average classification error as percentages across several architectures with and without using AugMix is reported. In our study, we also provided using AugMix with AlexNet to show how effective AugMix is even with another learning network on corrupted images. In the second part the results of our hyper-parameters ablations study is provided.

A. Recreating Data from original paper

The results of testing the augmentation technique on CIFAR-10-C with 30 epochs and with exactly the same hyper-parameters used in the paper are provided in Table.I. As it was mentioned in section 2, we tried to recreate the results of table number 1 in the original paper with smaller number of epochs. For sure, we expected that our results not to be as good as theirs. The behavioural results of networks with and without AugMix are more concerned here. The average error of the four models used by paper, is slightly higher for Standard learning and learning with

		Standard	Augmix
CIFAR-10-C	WRN	29.36	13.86
	ResNeXt	33.99	15.90
	DenseNet	31.86	15.34
	AllConv	32.26	18.74
	AlexNet	30.30	22.80
	Mean w/o AlexNet	31.87	15.96
	Mean	31.55	17.33

Table I: Recreated table

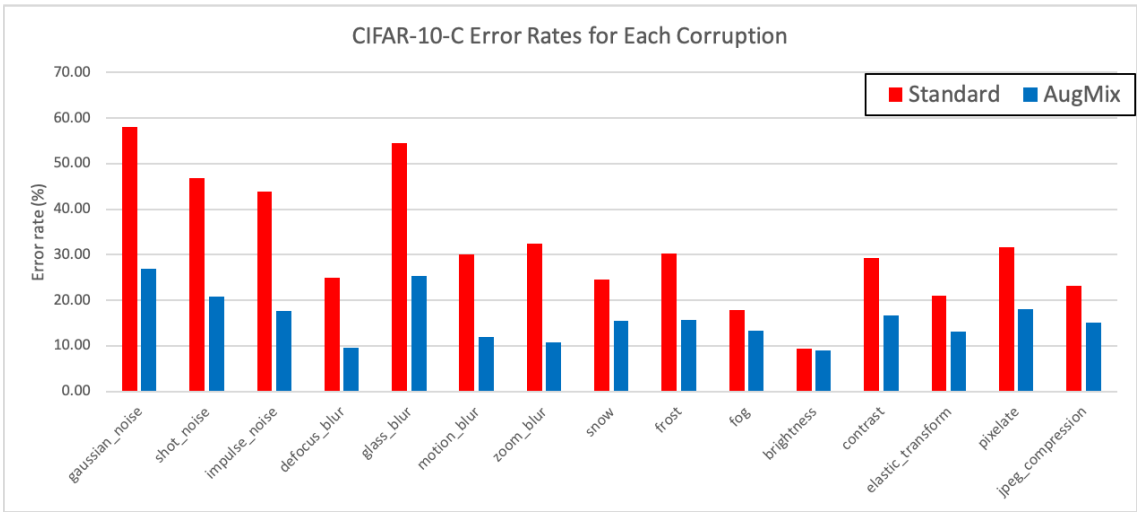


Figure 3: Mean of errors for all structures used for AUGMIX and standard training on CIFAR-10 dataset

AugMix. Comparing to their results, we have approximately 25% higher amount of errors, but the thing which we have in common, is the performance of all learning models in average got approximately 2 times better with AugMix. Moreover, we had another learning model tested. AlexNet will also have better performance about 50 % with using AugMix as its pre-processing technique for training.

We also tried to provide the average performance of all models on each corruption available in CIFAR-10-C dataset as the figure number 12 in the original paper. We see that AugMix enhances corruption robustness of all learning models across all CIFAR-10-C noise, blur, and digital corruptions. The recreated bars chart of its performance across those corruptions is depicted in Fig.3 which is quite similar as the one in the paper.

B. Hyper-Parameters Ablation

In the paper, it is mentioned that changing the hyper-parameters of AugMix does not make significant difference in the results. To validate their statement, we provide an ablation study. In this study, we have studied effects of increasing chain depth and mixture width with different learning models by fixing other parameters. Referring to the results in Table.III and Table.II., choosing mixture width of 3 and chain depth of 4, is almost the optimum choice by considering increasing chain depth and mixture width efforts. Although, changing hyper-parameters of AugMix changes the performance, It does not differ the performance significantly. Therefore, the statement "hyper-parameters are not highly sensitive and that AugMix performs reliably without careful tuning" is quite true.

	Standard	Depth=1	Depth=2	Depth=3	Depth=4
Mean Error	29.36	15.33	13.91	13.44	13.86

Table II: Error percentage for each structure with Width of 3 learnt by WideResNet

	Standard	Width=1	Width=2	Width=3	Width=4	Width=5
ResNeXt	33.99	14.90	15.26	15.90	15.92	16.07
WRN	29.36	13.92	14.04	13.86	13.65	14.06
DenseNet	31.86	15.18	15.53	15.34	15.43	15.26
AllConv	32.26	18.82	18.87	18.74	18.77	18.75
Mean Error	31.87	15.71	15.92	15.96	15.94	16.04

Table III: Error percentage for each structure with chain depth of 4

IV. Summary

In this project we tried to reproduce the results of the original paper and also try to have a critical view towards the proposed parameters and methods. In addition we tried to enhance the work of the original paper and have different view towards the AugMix and learning algorithms used.

As can be seen in the provided figures and according to our previous explanations we were fairly able to recreate the results and discuss different aspects of AugMix and its parameters.

In this project we worked closely together to achieve the goals and prepare this report. All of the works explained in this report were at first tested by each group member with low number of epochs to make sure that they work and we discussed different approaches each one of us proposed and in the end we ran the agreed methods for 30 epochs. And when we were satisfied with the results, we reported it.