Project Group 30:
Amir Bachir K.B., Alessandro Cortese, Gianmarco Lodi, Matteo Scarbini

UNIVERSITY OF TWENTE.

Machine Learning II - Project

Comparing Different Data Augmentation Methods by Evaluating Their Impact in Varying Dataset Settings on Image Classification

5 February 2023

Abstract

Image classification tasks employing Deep Neural Networks require a significant amount of labeled data to train the model effectively. Therefore, data scarcity represents a problem because, without sufficient data, the model may not learn to recognize patterns in the images or may overfit the training data, resulting in poor performance on unseen images. A common way to address this issue is data augmentation, which artificially increases the size and diversity of a dataset by applying various transformations to the original images. This project aims to compare the effectiveness of different data augmentation methods in improving image classification performance, studying their relationship with respect to dataset size and augmentation factor. The proposed approach consists in evaluating four state-of-the-art data augmentation techniques, *i.e.* AugMix, CutMix, MixUp and GridMask, on an image classification task employing the ResNet20 Convolutional Neural Network classifier on different downsampled versions of the CIFAR-10 dataset. The results of these experiments show that AugMix and GridMask are the most promising techniques with respect to test accuracy. Moreover, we test the robustness of the resulting augmented datasets to several kinds of image corruptions, using the CIFAR-10-C dataset. In this out-of-distribution evaluation, AugMix and MixUp appear to be the most suitable augmentation techniques to increase model robustness and help generalize to unseen corrupted data.

1 Introduction

Image classification is a task in which a model is trained to recognize and categorize objects within an image. Convolutional Neural Networks (CNNs), which are a specific class of Neural Networks, are commonly used to perform image classification tasks, due to their ability to learn complex features and patterns in images [1]. However, because of the massive number of parameters, CNNs have a tendency to overfit on small datasets [2]. A network is said to be overfitting when it properly models the training set but is unable to generalize its knowledge to reliably forecast the class of unseen data. Therefore, it follows that one of the main challenges when training CNNs for image classification is data scarcity [3]. Regrettably, not all domains are able to satisfy the increasing demands for larger datasets. The lack of sufficiently large datasets represents a challenge for fields such as medical image analysis and bioinformatics, where collecting and annotating images can be an expensive and time-consuming process [4, 5, 2].

Several methods have been proposed to reduce overfitting when training deep learning models, such as dropout [6], batch normalization [7], L1 and L2 regularization [8] and transfer learning [9]. Another technique, which has been considered essential to nearly every cutting-edge result in image classification is data augmentation [10]. This approach involves enlarging the number of images in a dataset through the use of synthetic images derived by applying transformations to the original ones.

This has found applications in several domains, yet much of the theoretical understanding behind why some techniques work better than others with respect to certain datasets remains limited, and data augmentation is considered highly domain specific [11]. Many questions remain unanswered, such as why combining such techniques has been proven to increase models performance in some cases [2], whereas it was not successful in others [12]. Moreover, there is still no consensus as to which ratio of original to final dataset size will result in the best performing model [13].

The goal of our research is to perform a comprehensive examination of the relationships between various data augmentation methods, the quantity of original images in the dataset, and the number of augmented data points. Additionally, we investigate the generalization ability of data augmentation techniques on out-of-distribution (OOD) samples, under varying configurations of the aforementioned parameters.

To accomplish this, we evaluate the accuracy of a CNN classifier on four different downsampled versions of the CIFAR-10 dataset, employing four different data augmentation techniques and three augmentation factors. We evaluate the efficacy of the augmentation methods in two ways. First, we increase the subset size up to the same size as the original one, and compare the classifier performance. This provides insight into how augmentation techniques can fill the gap caused by data scarcity, while keeping the training time constant. Second, we linearly increase the size of the subsets using fixed augmentation factors which also go beyond the original dataset size, in order to explore how much variability can be extracted from the data. Finally, each configuration is tested on the full CIFAR-10-C dataset, recording its mean accuracy with respect to every type of corruption and level of intensity.

The remainder of this paper is structured as follows: after exploring relevant literature in Section 2, we describe the experimental setup in Section 3, explaining the chosen techniques and dataset configurations. Section 4 presents the main results, which are discussed, along with their limitations and further works in Section 5. Finally, Section 6 concludes and an Appendix showing the complete tables of results can be found on the last pages.

2 Related work

Recently, the field of deep learning has seen a surge in interest regarding image augmentation, resulting in a rapid increase in the number of papers published on the subject in the last few years [14]. Data augmentation is, in fact, critical to improve the performance of data-hungry Deep Learning models, such as Deep Convolutional Neural Networks, by increasing the size of training datasets and using existing data more effectively [13, 15].

Several techniques have been employed in a number of Computer Vision tasks, such as object detection, image classification and image segmentation, often obtaining remarkable results in boosting model performance [16, 17]. According to [13], these can be split into two main groups: basic image manipulations (such as rotation, flipping, cropping, shear mapping, and color space manipulations) and deep learning approaches (based on generative models, such as GANs [18]).

Many state-of-the-art image classifiers are accompanied by manually designed data augmentation methods encompassing transformations of the first group [19, 20, 21]. However, while implementing these image manipulations is relatively straightforward, care must be taken to maintain the correct label associations (e.g., in the case of the MNIST dataset, flipping can alter the class of “6” to “9” and vice versa). On the other hand, generative models have emerged more recently but have been shown to obtain equal or greater performance than basic transformations in a wide range of Computer Vision tasks [22, 23, 24].

Naveed *et al.* (2021) add to these two groups a relatively new augmentation strategy, which mixes and deletes different image regions, showing promising results in terms of overfitting reduction, generalization against data variations and robustness against corruptions [25]. Among the techniques analysed by them, we focus our attention on four state-of-the-art approaches: CutMix [26], AugMix [27], MixUp [28] and GridMask [29].

Furthermore, recent research has focused on learning data augmentation strategies directly from the data itself, in order avoid manually designing data augmentation strategies [30, 31]. For instance, Smart Augmentation employs a network that creates new data by combining two or more samples from the same class [32].

With respect to future directions of research, Yang *et al.* (2022) highlight the number of generated data as an interesting point to be further explored. Moreover, they emphasize that the increase in the amount of training data is not exactly proportional to the increase in the classifier performance, which could be partly due to the fact that the diversity of the data does not increase much despite the expansion in the number of augmented data [33]. For this reason, we aim to investigate how much data should be generated using these techniques. Finally, Shorten and Khoshgoftaar (2019) affirm that it would be interesting to examine which size of the expanded data is needed to achieve equal or better performance compared to the full training set. [13].

3 Methodology

To conduct our study, we employ CIFAR-10, a benchmark dataset which is widely used in Computer Vision tasks [34]. The dataset consists of 10 classes containing animals and transportation means, each one having 6000 color images, totaling 60,000 images. The images' sizes are all 32×32 pixels.

As our focus is on the comparison of augmentation techniques, and not on pushing the state-of-the-art accuracy, the chosen classifier is a ResNet20 model [35]. The main goal of this architecture is to simplify the network's learning of complex patterns in images by introducing a series of "residual connections", which avoid the so called "curse of depth" typical of traditional Deep Neural Networks [36].

When processing an image, ResNet transfers the original picture to the next layer in addition to the one which has been transformed by the convolutions. This makes it simpler for the network's final layers to comprehend patterns in the image, as they can see both the original and the altered version of it.

We follow the training regime implemented by He *et al.* (2016), subtracting the per-pixel mean to the inputs, using a mini-batch size of 128 and SGD as optimizer [35]. The initial learning rate is 0.1, which gets divided by 10 at respectively epoch 91 and 137. The training ends at epoch 182. These values correspond respectively to 32k, 48k and 64k iterations when training with the original non-augmented dataset. However, we decided not to apply the light data augmentation used in the paper, in order to isolate the effect of the specific augmentations under study. No augmented images are used for validation nor testing.

3.1 Training, Validation and Test

The first step of the analysis consists in splitting the dataset in training, validation and test. The original CIFAR-10 test set consists of 1000 images per class, and it kept equal across all the experiments, independently of the training set size. In order to investigate the relationship between the training set size and the augmentation methods performance, all the experiments are repeated with four downsampled training sets. I More specifically, the original training set, composed of 5000 images per class, is downsampled into four different configurations, each one corresponding to a percentage of original images to keep, namely 20%, 40%, 60% and 80%. Subsequently, 10% of the training set images are used for validation, as described in [35].

3.2 Augmentation Techniques and Augmentation Factors

The original training set is not subject to any augmentation procedure, but the accuracy on such dataset is recorded nonetheless as a baseline reference. This procedure is repeated for the four dataset configurations, registering for each the corresponding baseline accuracy without augmentation.

For each configuration, four augmentation techniques are separately applied, with the resulting augmented images added to the corresponding dataset. In particular, these are:

- **AugMix**: this technique is characterized by the utilization and combination of several simple augmentation methods, such as translation, rotation, posterization and equalization. These operations are stochastically sampled and layered to create a wide range of augmented images. The detailed process consists in creating a series of augmentation chains, each of them constructed by composing from one to three randomly selected augmentation methods. Furthermore, also the severity of some operations, such as the rotation angle, are uniformly sampled for each application. The final image is a mix of the resulting image from each chain [27].

- **MixUp**: this technique consists of creating a linear combination of random pairs of training images. This means that the labels of the augmented images are proportional to the degree of prevalence of the samples that have been mixed to compose the final image [28]. Even though mixing images together by averaging their pixel values seems counterintuitive, and results in “odd” images to human observers, several studies demonstrated that effective augmentation strategy can be achieved using this method [37, 38].

- **CutMix**: this technique creates augmented datapoints by cutting and pasting patches among training images. The ground truth labels of the resulting images are mixed proportionally to the areas of the patches. A possible advantage with respect to the relatively unnatural datapoints created by MixUp is that the augmented images generated by CutMix are more locally natural [26], so that the model is trained to recognize objects from partial views.

- **GridMask**: this technique randomly removes parts of the image, encouraging the model to learn more comprehensive features and to consider the whole image, without focusing on just a portion of it [39]. The augmented images are created by randomly eliminating a set of spatially uniformly distributed squares and the removed pixel sets are disconnected from each other. These “dropping regions” can be controlled in density and size, thus providing a statistically lower probability of failure with respect to other methods relying on information removal, such as CutOut [29].

Each augmentation technique is repeated according to four augmentation factors, namely 100%, 200%, 300%, plus one that adds to the downsampled training set the necessary number of augmented images to reach the size of the original dataset (*i.e.* 4500 training images per class). The first three factors represent the increase in size of the dataset when the augmented images are added. For example, with the first factor, each of the original images is used to create just an augmented datapoint, corresponding to an increase of 100% in the dataset size, whereas using the second factor, two different augmented images are constructed for every original one, meaning that the dataset grows by 200%.

A separate explanation is useful to clarify the purpose of the last factor, which varies according to the dataset to be augmented: with the maximum level of downsampling (*i.e.* Configuration 1, having 900 training images and 100 validation images, or 20% of the original CIFAR-10 dataset), the augmentation factor that is needed to reach the original configuration is 400%, whereas with Configuration 4, the necessary augmentation factor is just 25%.

The idea is to compare the test accuracies obtained with this “tailored” augmentation factors with the baseline relative to the original training set. This particular comparison is made keeping the total number of training images fixed. Indeed, as the number of original images decreases due to downsampling, the number of synthetic datapoints employed to reach 4500 training images increases. In this way we can address the first goal of our paper, namely understand how much a dataset can be downsampled and then augmented back to its original dimension without affecting the classifier performance.

3.3 Out-Of-Distribution Testing

In order to address the second research question of this paper, we employ the CIFAR-10-C dataset [40], whose images are the corrupted versions of the ones in the CIFAR-10 test set. In particular, the images in this dataset are modified with 19 different image perturbations, such as noise, blur, and weather changes, each one having 5 levels of intensity. Multiplying the number of corruptions for the number of intensities results in 95 possible test sets, each containing 10,000 images; therefore the total number of samples of this new dataset is 950,000. For every experiment, we compute an average of the accuracy obtained on this whole OOD test set, which represents a measure of how well the model can handle and generalize to unseen corrupted images.

3.4 Workflow

The overall workflow is summarized in Figure 1. The total number of experiments amounted to 69[1], resulting in as many in-distribution and out-of-distribution accuracy scores.

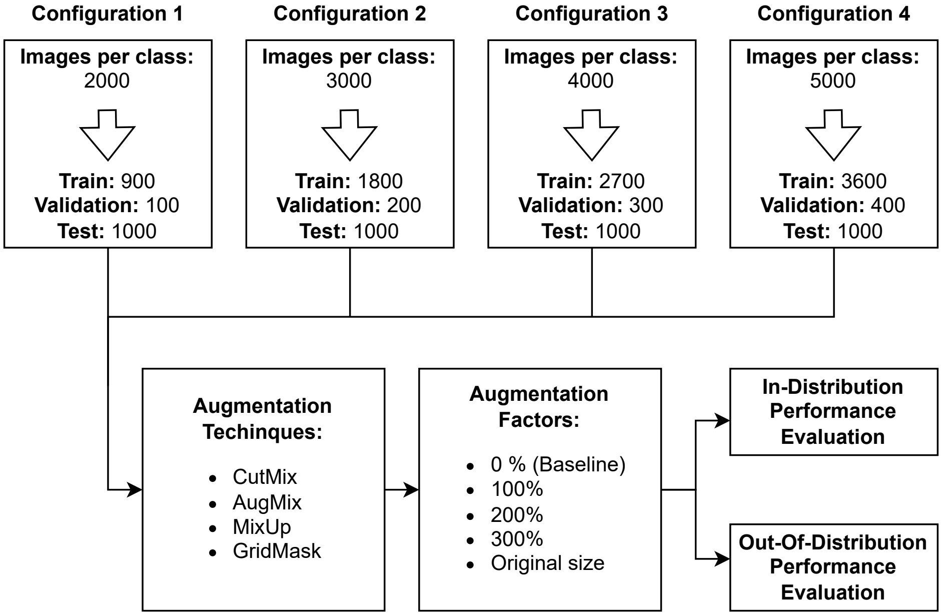

Figure 1: Workflow: each configuration of downsampled subsets considered is augmented with augmentation techniques and augmentation factors, then evaluated in and out of distribution.

4 Results

4.1 Comparison among different dataset sizes

As it is possible to see in Table 1 (see Appendix), the accuracy on the test set obtained without any kind of data augmentation on the complete dataset is 84.21%. This represents a much lower value with respect to the accuracy score of 91.25% obtained by He *et al.* (2016) using the same dataset and classifier. Since we mimicked their training regime, except for the light data augmentation policy, we know that such difference is due to the online (on the fly) random flipping and translations that they employed during training.

Figure 2 shows a comparison of the performance obtained by each kind of augmentation. For each technique, the performance value is plotted with respect to the sample size and the augmentation factor. Unsurprisingly, the accuracy grows with the number of samples of non-augmented configuration used in training, but such increase is not proportional. The average increase in accuracy by going from one subset to the bigger one slows down at an approximately 50% rate. Moreover, AugMix and GridMask tend to increase the model performance with any of the augmentation factors, even for the smaller size configurations. On the other hand, an interesting pattern can be found using CutMix and MixUp. Using such techniques, the 100% augmentation factor performs worse than the respective baseline, irrespective of the configuration (and thus, of the dataset size), as demonstrated by the negative slope between the baseline and first one or two augmentation factors. Another interesting observation is that MixUp is the only model which never reaches the baseline accuracy of the original model. However, the other models

[1] This number is obtained by multiplying 4 configurations by 4 augmentation techniques and by 4 augmentation factors, then adding 5 experiments corresponding to each baseline.

trained with the other augmentation techniques are generally able to reach such accuracy only employing Configuration 4 and high augmentation factors (200 % and 300%).

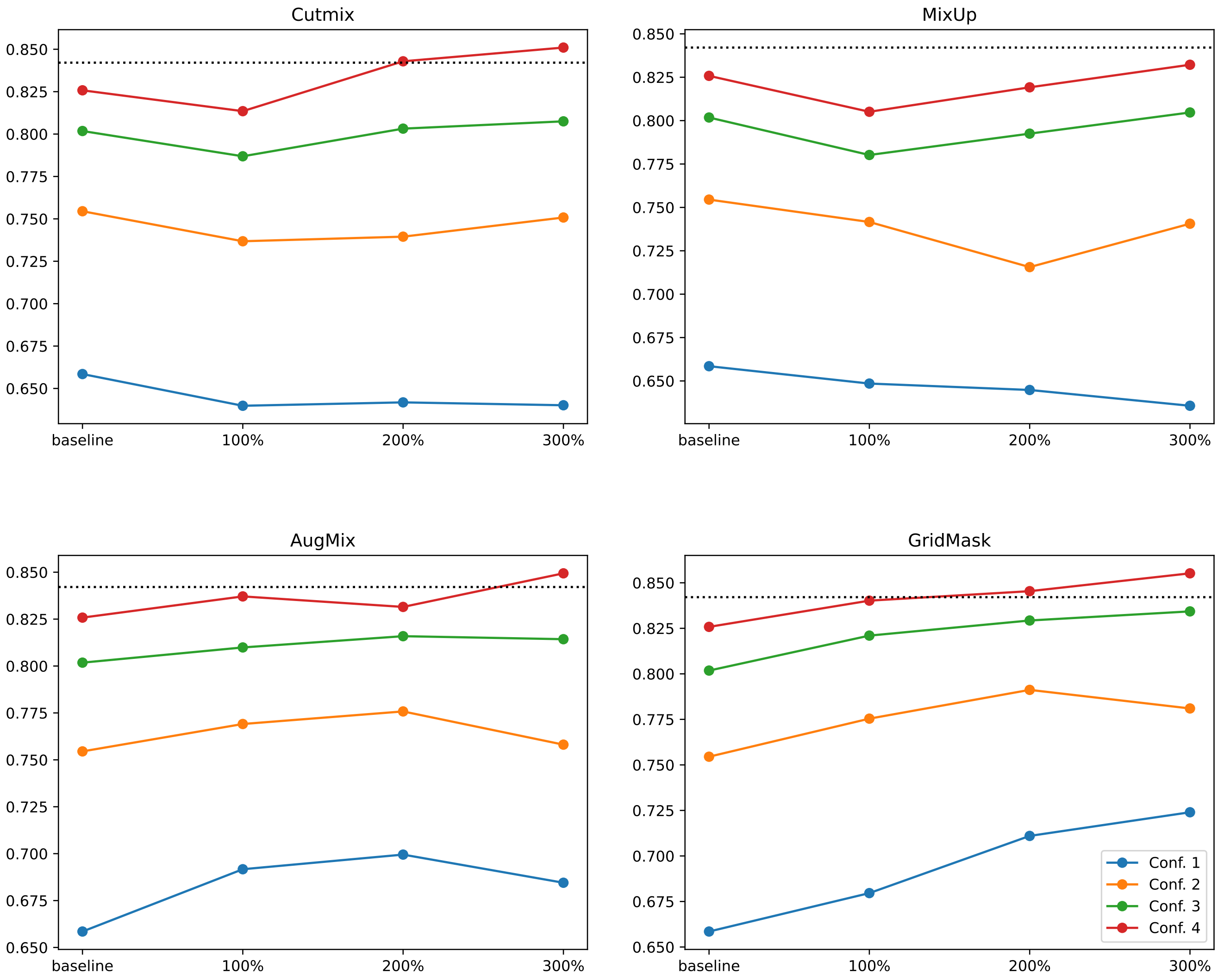

Figure 2: Test accuracy values as the augmentation factor increases, with respect to each augmentation technique and each configuration. The black dotted line represents the test accuracy for the original dataset.

The graph in Figure 3 shows the comparison of the performance of the model with respect to the different dataset configurations when augmented with adapting factors, in order to reach the original dataset size (4500 train images per class) and compare test accuracy scores with the non-augmented original one (black dotted line). It can be easily seen that none of the configurations reach the test accuracy of 0.8421 obtained with the original dataset. Nonetheless, some trends clearly show up. AugMix and GridMask (blue shade bars) perform better in each configuration, with the latter achieving the highest test accuracy of 0.8330 (see Appendix for the complete result tables) when testing on Configuration 4 and 25% of augmented images, which is the configuration with the closest number of clean samples to the original dataset. From here, it is also made clear that generally the performance increases as the size of the configurations grows and, consequently, as the amount of augmented images decreases. For instance, when adding 3600 CutMix augmented images per class to Configuration 1, which originally contains only

900 clean images, the test accuracy drops to 0.6722, despite having the same number of images as the original dataset.

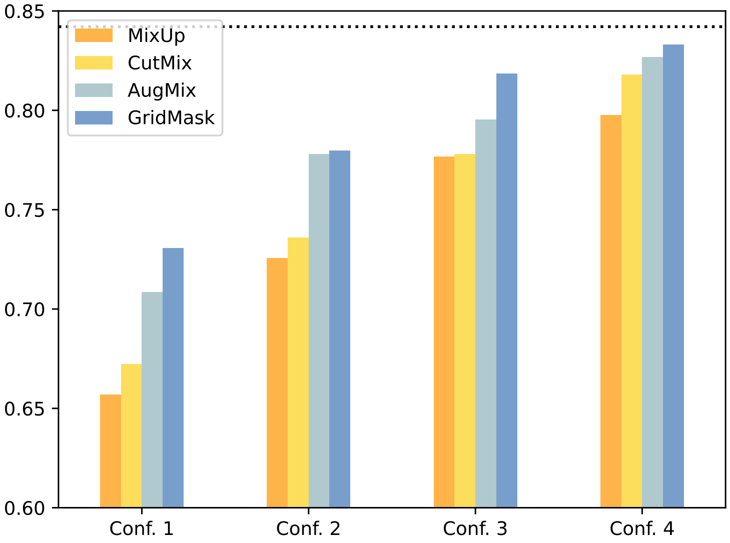

Figure 3: Comparison of test accuracy values for each configuration and each augmentation technique, reaching the original dataset size. The black dotted line represents the test accuracy for the original dataset.

4.2 Robustness test

An extensive analysis of the OOD performance on each specific CIFAR-10-C corruption and each configuration is out of the scope of this paper, simply due to the large amount of experiments. However, such analysis is performed for one single fixed configuration, namely Configuration 4, and augmentation factor (200%) and repeated across the four augmentation techniques. These specific settings were chosen because they contain the highest mean OOD test accuracy of all the experiments, *i.e.* 0.6176, obtained using AugMix. Figure 4 and Table 6 (in Appendix) show the OOD test results for every corruption of the various techniques, compared to the values obtained by the Configuration 4 baseline.

It is possible to notice that AugMix increases robustness with respect to the baseline on every corruption. This is in line with the results obtained by Hendrycks *et al.* in the paper that first introduces such technique [27]. Moreover, it appears to work especially well on blur corruptions, such as Gaussian Blur (with which it records the highest accuracy increase with respect to the baseline, *i.e.* +0.1996), Zoom Blur (+0.1843), Motion Blur (+0.1787) and Defocus Blur (+0.1781). A quite substantial improvement to the baseline is also reached on Contrast corruption, recording an increase of 0.1942.

MixUp also demonstrates to help raising robustness to every transformation under scrutiny. Its results are generally slightly inferior to those obtained with AugMix, except for Frost corruption, where it scores marginally better (+0.1148).

As far as GridMask is concerned, it moderately improves robustness with regard to the baseline on every corruption, except for Gaussian Noise (−0.0103) and Glass Blur (−0.0226).

Finally, CutMix is the only technique which leads to a lower mean OOD accuracy than the one obtained by the baseline. In general, the worst decreases are recorded on Noise corruptions, especially on Gaussian Noise (−0.1138), Shot Noise (−0.1057) and Speckle Noise (−0.0910).

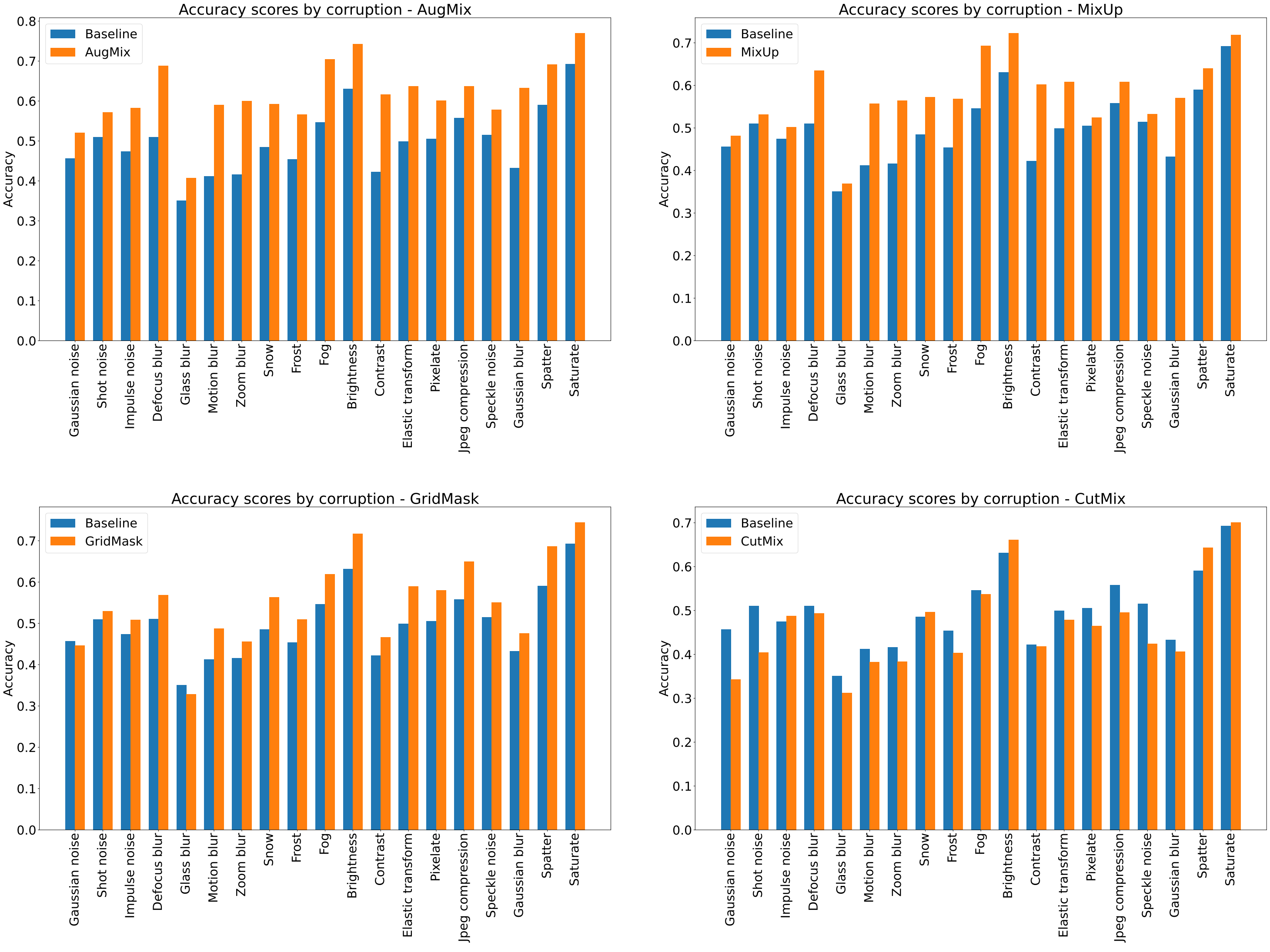

Figure 4: Test accuracy scores by corruption on CIFAR-10-C of Configuration 4 without augmentation and with every technique with augmentation factor 200%

5 Discussion

5.1 Results interpretation

From the results presented in Figure 2 the two categories of augmentation appear to follow similar patterns. CutMix and MixUp, which start from two samples and combine both the images and the labels, obtain inferior accuracy (sometimes worse than the corresponding configuration's baseline) with respect to AugMix and GridMask, which use only on a single image at a time. We assume this is a consequence of the way CutMix and MixUp create the augmented images: both methods change the content of the image when mixing samples from different classes. In fact, the visualization of augmented images produced using CutMix revealed that many of them were difficult to interpret due to the random placement of image fragments. In numerous instances, important features of an image were obscured by the addition of irrelevant elements, hindering the ability to accurately perform the classification task. On the other hand, the use of AugMix and GridMask resulted in new images that preserved relevant characteristics, enabling the model to effectively capture essential patterns and improve classification performance. It is assumed that the trend observed in the data analysis may also be attributed to the small image resolution of the CIFAR-10 dataset (32 x 32 pixel). Further studies should investigate if this behavior changes with the use of larger image sizes.

With regard to the comparison in Figure 3, which is performed between datasets with the same number of images, it emerges that augmented datasets cannot reach the same variability provided by a completely original dataset. This is due to the fact that augmented images are only partly different than the images that generated them, or they are even lacking some information (as in the case of GridMask). For this reason, in most cases, augmenting the training set to reach the original size is not sufficient to fill the gap caused by data scarcity. A possible solution, as done in our previous experiment settings, could be using a greater augmentation factor, bringing the size of the training set beyond that of the original one, although at the expenses of computational time.

It is important to notice that these results refer to a benchmark dataset, and that domain knowledge is crucial to apply the right data augmentation technique. Based on our results from the OOD experiments, AugMix could prove extremely useful in a number of applications, such as in the field of medical imaging, where the availability of annotated data is often limited, due to the rareness of certain diseases and the requirements of medical experts. Additionaly, such data itself is highly sensitive [41, 15].

Moreover, medical imaging data can be highly variable, with different imaging modalities, scanning protocols, and patient populations. In this case, AugMix augmentation can help to increase robustness to such variations, which can be especially beneficial for tasks like lesion detection and segmentation, where small alterations in the images can lead to large shifts in the model output [42]. Furthermore, data augmentation can mitigate the problem of class imbalance, which can be found, for example, in the presence of medical images of rare pathologies [43].

5.2 Limitations

A possible limitation to our study is that the results were not cross-validated. Since the performances of Deep Neural Networks are always characterized by some degree of variability, repeating the experiments and averaging the results could definitely increase their statistical significance, as seen in [44, 35].

Another area of possible improvement, that would help to generalize our results, would be performing the same experiments on a different dataset in terms of image size (*e.g.* ImageNet [45]).

Then, in order to obtain a more considerable and significant distribution of the performance of each augmentation technique, more settings should be taken into account. This would be particularly interesting with regard to higher augmentation factors, since techniques as AugMix and GridMask tend to increase the model performance with any of the augmentation factors. For these two techniques, the critical factor after which the augmentation ceases to be beneficial still remains to be determined.

Finally, a different approach to the problem could also be taken. As a matter of fact, the augmentation techniques we applied were done in an offline fashion, as a preprocessing step which increased the dataset size by specific factors. One possible further development could be to compare these performances with online augmentation at training time, thus making available to the model different augmented images at each epoch. This can save memory on disk, as the generated images are immediately discarded, but slows down training due to the additional computational overhead [13].

6 Conclusions

This paper presents a comparison of four data augmentation techniques, commonly used to address the issue of overfitting in deep learning models, which is oftentimes caused by a scarcity of data. The performance of such techniques, namely AugMix, MixUp, GridMask and CutMix were investigated in light of the size of the dataset on which they were applied and of the augmentation factor. The results indicated that GridMask is the best method in terms of enhancing the classifier's test accuracy for every augmentation factor of each configuration (with respect to the corresponding baseline). Then, also AugMix displayed promising results, followed by CutMix and MixUp. This is particularly evident in Figure 3, that refers only to the adapting augmentation factors. From this graph another major finding of the paper emerges: none of the configurations reached the test accuracy of 0.8421 obtained with the original dataset. Therefore in our experiments data augmentation was not sufficient to fill the gap caused by data scarcity when augmenting to the original dataset size, even though with greater augmentation factors, CutMix, AugMix and GridMask have been able to overcome the benchmark test accuracy.

Moreover, the paper studied the generalization capabilities of such techniques with respect to the corruptions found in the CIFAR-10-C datasets. Our specific analysis (Configuration 4 and augmentation factor 200%) showed that AugMix and MixUp increase the robustness of the classification task on every corruption. Also GridMask seemed able to improve the robustness of the model to all the corruptions, with a couple of exceptions, while CutMix was the only method that led to a lower OOD accuracy than the one obtained by the baseline. However, analysing the delta values in Table 6, AugMix appears to be the most suitable technique (among the 4 considered) to deal with corruptions, especially if belonging to the Blur family.

In view of the above, the outlook for data augmentation is extremely positive, and this is also demonstrated by the amount of research that is conducted on the topic. However, further work is needed to address some of the limitations (or research gaps?) highlighted in the previous section. These areas of improvement go from the generalization of the results by performing the experiments on a different dataset, to adopting a different approach such as the augmentation on the fly.

APPENDIX

Train Acc.	Val. Acc.	Test Acc.	Test Acc. OOD
1.0000	0.8506	0.8421	0.4965

Table 1: Accuracy values for the model trained on the original CIFAR-10 dataset

Aug. Technique	Aug. Factor	Train Acc.	Val. Acc.	Test Acc.	Test Acc. OOD
CutMix	100%	0.9595	0.6560	0.6398	0.4139
	200%	0.9551	0.6740	0.6418	0.4089
	300%	0.9659	0.6630	0.6401	0.4076
	400%	*0.9531*	*0.6670*	*0.6722*	*0.4044*
AugMix	100%	1.0000	0.6950	0.6917	0.5173
	200%	1.0000	0.7060	0.6995	0.4794
	300%	1.0000	0.6910	0.6845	0.4686
	400%	*1.0000*	*0.7100*	*0.7084*	*0.5049*
MixUp	100%	0.9404	0.6670	0.6485	0.4381
	200%	0.9427	0.6690	0.6448	0.4056
	300%	0.8924	0.6640	0.6357	0.4415
	400%	*0.9386*	*0.6550*	*0.6570*	*0.4714*
GridMask	100%	1.0000	0.6850	0.6796	0.4841
	200%	1.0000	0.7320	0.7110	0.4692
	300%	1.0000	0.7140	0.7240	0.4934
	400%	*1.0000*	*0.7380*	*0.7307*	*0.4990*
Baseline	0%	0.9961	0.6840	0.6585	0.4281

Table 2: Accuracy values for Configuration 1 (in italic the scores for the augmentation that reaches the original size).

Aug. Technique	**Aug. Factor**	**Train Acc.**	**Val. Acc.**	**Test Acc.**	**Test Acc. OOD**
CutMix	100%	0.9154	0.7390	0.7368	0.4638
	150%	*0.9506*	*0.7490*	*0.7360*	*0.4476*
	200%	0.8958	0.7485	0.7395	0.4429
	300%	0.8742	0.7595	0.7508	0.3936
AugMix	100%	1.0000	0.7765	0.7691	0.5244
	150%	*1.0000*	*0.7940*	*0.7780*	*0.5709*
	200%	1.0000	0.7880	0.7758	0.5394
	300%	1.0000	0.7605	0.7581	0.5162
MixUp	100%	0.9547	0.7530	0.7416	0.5139
	150%	*0.9443*	*0.7285*	*0.7256*	*0.4736*
	200%	0.9507	0.7360	0.7156	0.5141
	300%	0.9420	0.7395	0.7406	0.5238
GridMask	100%	1.0000	0.7750	0.7754	0.5399
	150%	*1.0000*	*0.7970*	*0.7798*	*0.5115*
	200%	1.0000	0.7835	0.7912	0.5224
	300%	1.0000	0.7915	0.7810	0.4895
Baseline	0%	1.0000	0.7460	0.7545	0.4904

Table 3: Accuracy values for Configuration 2 (in italic the scores for the augmentation that reaches the original size).

Aug. Technique	**Aug. Factor**	**Train Acc.**	**Val. Acc.**	**Test Acc.**	**Test Acc. OOD**
CutMix	*67%*	*0.9532*	*0.7920*	*0.7780*	*0.4410*
	100%	0.9576	0.7967	0.7869	0.4784
	200%	0.9192	0.8130	0.8032	0.4321
	300%	0.8504	0.8210	0.8075	0.4392
AugMix	*67%*	*1.0000*	*0.8057*	*0.7954*	*0.5169*
	100%	1.0000	0.8193	0.8099	0.5237
	200%	1.0000	0.8303	0.8159	0.5673
	300%	1.0000	0.8187	0.8143	0.5658
MixUp	*67%*	*0.9497*	*0.7757*	*0.7766*	*0.5107*
	100%	0.9426	0.7910	0.7802	0.5088
	200%	0.9435	0.7930	0.7925	0.5452
	300%	0.9242	0.8156	0.8047	0.5699
GridMask	*67%*	*1.0000*	*0.8267*	*0.8185*	*0.5623*
	100%	1.0000	0.8340	0.8210	0.5579
	200%	1.0000	0.8357	0.8293	0.5478
	300%	1.0000	0.8487	0.8343	0.5171
Baseline	0%	1.0000	0.8110	0.8018	0.5112

Table 4: Accuracy values for Configuration 3 (in italic the scores for the augmentation that reaches the original size).

Aug. Technique	Aug. Factor	Train Acc.	Val. Acc.	Test Acc.	Test Acc. OOD
CutMix	*25%*	*0.9829*	*0.8267*	*0.8180*	*0.5556*
	100%	0.9274	0.8230	0.8135	0.4821
	200%	0.8631	0.8505	**0.8429**	0.4703
	300%	0.8627	0.8720	**0.8510**	0.5081
AugMix	*25%*	*1.0000*	*0.8350*	*0.8267*	*0.5646*
	100%	1.0000	0.8495	0.8371	0.5909
	200%	1.0000	0.8455	0.8315	**0.6176**
	300%	1.0000	0.8545	**0.8494**	0.5949
MixUp	*25%*	*0.9839*	*0.8050*	*0.7975*	*0.5327*
	100%	0.9448	0.8165	0.8051	0.5239
	200%	0.9293	0.8340	0.8195	0.5793
	300%	0.9305	0.8407	0.8322	0.5354
GridMask	*25%*	*1.0000*	*0.8418*	*0.8330*	*0.5535*
	100%	1.0000	0.8495	0.8402	0.5748
	200%	1.0000	0.8630	**0.8454**	0.5513
	300%	1.0000	0.8680	**0.8552**	0.5511
Baseline	0%	1.0000	0.8380	0.8258	0.4979

Table 5: Accuracy values for Configuration 4 (in italic the scores for the augmentation that reaches the original size, in bold the best score in Test Acc. OOD and the Test Acc. values that overcome the performance on the original dataset).

Corruption	Baseline (0%)	CutMix	GridMask	MixUp	AugMix
Gaussian Noise	0.4566	0.3428 (−0.1138)	0.4463 (−0.0103)	0.4813 (+0.0247)	0.5202 (+0.0636)
Shot Noise	0.5100	0.4043 (−0.1057)	0.5291 (+0.0191)	0.5313 (+0.0213)	0.5717 (+0.0617)
Impulse Noise	0.4742	0.4877 (+0.0135)	0.5090 (+0.0348)	0.5026 (+0.0284)	0.5829 (+0.1087)
Defocus Blur	0.5103	0.4939 (−0.0164)	0.5680 (+0.0577)	0.6351 (+0.1248)	0.6884 (+0.1781)
Glass Blur	0.3506	0.3122 (−0.0384)	0.3280 (−0.0226)	0.3695 (+0.0189)	0.4080 (+0.0574)
Motion Blur	0.4122	0.3825 (−0.0297)	0.4873 (+0.0751)	0.5570 (+0.1448)	0.5909 (+0.1787)
Zoom Blur	0.4161	0.3835 (−0.0326)	0.4563 (+0.0402)	0.5641 (+0.1480)	0.6004 (+0.1843)
Snow	0.4852	0.4963 (+0.0111)	0.5629 (+0.0777)	0.5731 (+0.0879)	0.5925 (+0.1073)
Frost	0.4540	0.4031 (−0.0509)	0.5097 (+0.0557)	0.5688 (+0.1148)	0.5666 (+0.1126)
Fog	0.5464	0.5368 (−0.0096)	0.6187 (+0.0723)	0.6938 (+0.1474)	0.7047 (+0.1583)
Brightness	0.6312	0.6614 (+0.0302)	0.7165 (+0.0853)	0.7228 (+0.0916)	0.7433 (+0.1121)
Contrast	0.4224	0.4181 (−0.0043)	0.4662 (+0.0438)	0.6027 (+0.1803)	0.6166 (+0.1942)
Elastic Transform	0.4995	0.4784 (−0.0211)	0.5895 (+0.0900)	0.6080 (+0.1085)	0.6377 (+0.1382)
Pixelate	0.5055	0.4646 (−0.0409)	0.5800 (+0.0745)	0.5248 (+0.0193)	0.6008 (+0.0953)
JPEG Compression	0.5581	0.4954 (−0.0627)	0.6497 (+0.0916)	0.6085 (+0.0504)	0.6374 (+0.0793)
Speckle Noise	0.5149	0.4239 (−0.0910)	0.5503 (+0.0354)	0.5325 (+0.0176)	0.5781 (+0.0632)
Gaussian Blur	0.4330	0.4061 (−0.0269)	0.4754 (+0.0424)	0.5708 (+0.1378)	0.6326 (+0.1996)
Spatter	0.5905	0.6436 (+0.0531)	0.6864 (+0.0959)	0.6406 (+0.0501)	0.6919 (+0.1014)
Saturate	0.6926	0.7010 (+0.0084)	0.7447 (+0.0521)	0.7193 (+0.0267)	0.7699 (+0.0773)
Mean	**0.4979**	**0.4703** (−0.0276)	**0.5513** (+0.0534)	**0.5793** (+0.0814)	**0.6176** (+0.1197)

Table 6: Test accuracy values on CIFAR-10-C for Configuration 4 with augmentation factor 200% with respect to each corruption (in parentheses delta values with respect to the non-augmented Baseline).

References

[1] Keiron O'Shea and Ryan Nash. An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458*, 2015.

[2] Loris Nanni, Michelangelo Paci, Sheryl Brahnam, and Alessandra Lumini. Comparison of different image data augmentation approaches. *Journal of Imaging*, 7(12):254, 2021.

[3] Ms Aayushi Bansal, Dr Rewa Sharma, and Dr Mamta Kathuria. A systematic review on data scarcity problem in deep learning: solution and applications. *ACM Computing Surveys (CSUR)*, 54(10s):1–29, 2022.

[4] Alexander Sorokin and David Forsyth. Utility data annotation with amazon mechanical turk. In *2008 IEEE computer society conference on computer vision and pattern recognition workshops*, pages 1–8. IEEE, 2008.

[5] Pei-Yun Hsueh, Prem Melville, and Vikas Sindhwani. Data quality from crowdsourcing: a study of annotation selection criteria. In *Proceedings of the NAACL HLT 2009 workshop on active learning for natural language processing*, pages 27–35, 2009.

[6] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

[7] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. pmlr, 2015.

[8] Ismoilov Nusrat and Sung-Bong Jang. A comparison of regularization techniques in deep neural networks. *Symmetry*, 10(11):648, 2018.

[9] Karl Weiss, Taghi M Khoshgoftaar, and DingDing Wang. A survey of transfer learning. *Journal of Big data*, 3(1):1–40, 2016.

[10] Alexander J Ratner, Henry Ehrenberg, Zeshan Hussain, Jared Dunnmon, and Christopher Ré. Learning to compose domain-specific transformations for data augmentation. *Advances in neural information processing systems*, 30, 2017.

[11] Georg Wimmer, Andreas Uhl, and Andreas Vecsei. Evaluation of domain specific data augmentation techniques for the classification of celiac disease using endoscopic imagery. In *2017 IEEE 19th International Workshop on Multimedia Signal Processing (MMSP)*, pages 1–6. IEEE, 2017.

[12] Marco Domenico Cirillo, David Abramian, and Anders Eklund. What is the best data augmentation for 3d brain tumor segmentation? In *2021 IEEE International Conference on Image Processing (ICIP)*, pages 36–40. IEEE, 2021.

[13] Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of big data*, 6(1):1–48, 2019.

[14] Nour Eldeen Khalifa, Mohamed Loey, and Seyedali Mirjalili. A comprehensive survey of recent trends in deep learning for digital images augmentation. *Artificial Intelligence Review*, 55(3):2351–2377, 2022.

[15] Agnieszka Mikołajczyk and Michał Grochowski. Data augmentation for improving deep learning in image classification problem. In *2018 international interdisciplinary PhD workshop (IIPhDW)*, pages 117–122. IEEE, 2018.

[16] Thorsten Hoeser and Claudia Kuenzer. Object detection and image segmentation with deep learning on earth observation data: A review-part i: Evolution and recent trends. *Remote Sensing*, 12(10):1667, 2020.

[17] Jia Shijie, Wang Ping, Jia Peiyi, and Hu Siping. Research on data augmentation for image classification based on convolution neural networks. In *2017 Chinese automation congress (CAC)*, pages 4165–4170. IEEE, 2017.

[18] Maayan Frid-Adar, Idit Diamant, Eyal Klang, Michal Amitai, Jacob Goldberger, and Hayit Greenspan. Gan-based synthetic medical image augmentation for increased cnn performance in liver lesion classification. *Neurocomputing*, 321:321–331, 2018.

[19] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[20] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

[21] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V Le. Learning transferable architectures for scalable image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8697–8710, 2018.

[22] Bin Liu, Cheng Tan, Shuqin Li, Jinrong He, and Hongyan Wang. A data augmentation method based on generative adversarial networks for grape leaf disease identification. *IEEE Access*, 8:102188–102198, 2020.

[23] Antreas Antoniou, Amos Storkey, and Harrison Edwards. Augmenting image classifiers using data augmentation generative adversarial networks. In *Artificial Neural Networks and Machine Learning–ICANN 2018: 27th International Conference on Artificial Neural Networks, Rhodes, Greece, October 4-7, 2018, Proceedings, Part III 27*, pages 594–603. Springer, 2018.

[24] Qiufeng Wu, Yiping Chen, and Jun Meng. Dcgan-based data augmentation for tomato leaf disease identification. *IEEE Access*, 8:98716–98728, 2020.

[25] Humza Naveed. Survey: Image mixing and deleting for data augmentation. *arXiv preprint arXiv:2106.07085*, 2021.

[26] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6023–6032, 2019.

[27] Dan Hendrycks, Norman Mu, Ekin D Cubuk, Barret Zoph, Justin Gilmer, and Balaji Lakshminarayanan. Augmix: A simple data processing method to improve robustness and uncertainty. *arXiv preprint arXiv:1912.02781*, 2019.

[28] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

[29] Pengguang Chen, Shu Liu, Hengshuang Zhao, and Jiaya Jia. Gridmask data augmentation. *arXiv preprint arXiv:2001.04086*, 2020.

[30] Toan Tran, Trung Pham, Gustavo Carneiro, Lyle Palmer, and Ian Reid. A bayesian data augmentation approach for learning deep models. *Advances in neural information processing systems*, 30, 2017.

[31] Justin Lo, Jillian Cardinell, Alejo Costanzo, and Dafna Sussman. Medical augmentation (med-aug) for optimal data augmentation in medical deep learning networks. *Sensors*, 21(21):7018, 2021.

[32] Joseph Lemley, Shabab Bazrafkan, and Peter Corcoran. Smart augmentation learning an optimal data augmentation strategy. *Ieee Access*, 5:5858–5869, 2017.

[33] Suorong Yang, Weikang Xiao, Mengcheng Zhang, Suhan Guo, Jian Zhao, and Furao Shen. Image data augmentation for deep learning: A survey. *arXiv preprint arXiv:2204.08610*, 2022.

[34] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[35] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[36] Yun Luo, Li-Zhen Zhu, Zi-Yu Wan, and Bao-Liang Lu. Data augmentation for enhancing eeg-based emotion recognition with deep generative models. *Journal of Neural Engineering*, 17(5):056021, 2020.

[37] Hiroshi Inoue. Data augmentation by pairing samples for images classification. *arXiv preprint arXiv:1801.02929*, 2018.

[38] Cecilia Summers and Michael J Dinneen. Improved mixed-example data augmentation. In *2019 IEEE winter conference on applications of computer vision (WACV)*, pages 1262–1270. IEEE, 2019.

[39] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 13001–13008, 2020.

[40] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*, 2019.

[41] Zeshan Hussain, Francisco Gimenez, Darvin Yi, and Daniel Rubin. Differential data augmentation techniques for medical imaging classification tasks. In *AMIA annual symposium proceedings*, volume 2017, page 979. American Medical Informatics Association, 2017.

[42] Phillip Chlap, Hang Min, Nym Vandenberg, Jason Dowling, Lois Holloway, and Annette Haworth. A review of medical image data augmentation techniques for deep learning applications. *Journal of Medical Imaging and Radiation Oncology*, 65(5):545–563, 2021.

[43] Marc D Kohli, Ronald M Summers, and J Raymond Geis. Medical image data and datasets in the era of machine learning—whitepaper from the 2016 c-mimi meeting dataset session. *Journal of digital imaging*, 30:392–399, 2017.

[44] Rupesh K Srivastava, Klaus Greff, and Jürgen Schmidhuber. Training very deep networks. *Advances in neural information processing systems*, 28, 2015.

[45] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.