DD2424 Group 118: The mechanisms, powers and limitations of some Data Augmentation techniques

Anton Stråhle **Jan Alexandersson** **Fredrika Lundahl**

Abstract

To obtain good results in deep learning the quality and quantity of the data is crucial. A smart and cheap way of increasing and improving the available data is data augmentation. In this project we have tried different augmentation techniques such as Mixup, Fourier transforms and more ordinary manipulations such as rotations and brightness adjustments on top of a decently performing CNN as well as some experiments on ResNet50. We have tried to see under which circumstances the augmentations have effect and how big that effect is. The data set used consists of about 25 000 colored images of 200 bird species, mostly from the United States. The data has further been divided into subsets of different sizes and classification difficulty. We have mainly worked with subsets with little data, as most of the available papers survey the performance on massive data sets, but often one does not have that situation. Our results show that the Fourier transform does not seem to have a positive effect on the accuracy. We also see that basic data augmentations techniques such as flips and rotations seem to increase the performance in all examined circumstances, whilst Mixup and Random Erasing which manipulates the data seem to require a more fine tuned implementation.

1 Introduction

One of the major drawbacks of supervised learning is the need of immense amounts of labeled data. Obtaining the quantities needed for optimal results can be both costly and extremely time consuming, if even possible at all, which may be the case with medical data where only a limited number of known and active cases may exist. (Shorten & Khoshgoftaar, 2019).

Data augmentation provides a partial solution to this issue when it comes to image classification. After being briefly introduced to some data augmentation techniques during the lectures we wanted to explore this topic further and investigate the payoff (or the lack thereof) from some data augmentation techniques as well as getting an understanding of when it is, and when it is not, worth the effort. We are interested in interpretability and have therefore used this project in an expereimental way to find the how, when and whys.

Our main focus has been to try out Mixup, Random erasing and Fourier transformation in practice, but we have also chosen to apply some more standard augmentations such as rotations simple adjustments to have something to compare with.

To clarify, our project does not aim to obtain the highest possible testing accuracy but rather aims to show the impact of data augmentations on the accuracy.

For our experiments we created a basic CNN architecture as well as implemented a fully trained version of ResNet50 in order to also examine the effects on data augmentations techniques in the case of transfer learning.

To create different settings we divided the original data into different subsets with different sizes and characteristics, some random and some more targeted, such as birds with bright colors. We have used

rather small data sets because of limitations in time and computing resources, but also because that is the situation many deal with. Most papers display the performance on huge data sets and we would like to investigate if the performances remains when we decrease the size and worsen the quality of the network.

Our results show that basic geometric augmentations such as rotations give a constant positive effect, for Mixup and Random Erasing one must spend time tuning the implementation since both techniques serve as regularizers which can lead to complications with certain network architectures, as in one of our cases.

2 Related Work

2.1 Basic Data Augmentation

A survey on Image Data Augmentation for Deep Learning by Shorten & Khoshgoftaar(2019) is a profound survey paper providing guidelines for different data augmentations and applications. The power of data augmentations is a red line throughout the text but the authors also emphasize the need of choosing the right augmentations for the specific dataset, and points out that in situations with very little data, augmentations may even lead to further overfitting, which is the direct opposite of its purpose. This is something we would like to investigate further, as being aware of problems may be as important as advantages.

2.2 Mixup

The concept of Mixup was introduced by Zhang et al(2018) and was published as a conference paper at ICLR 2018. Mixup is really fascinating because of its' creativeness and ability to improve performance while being very simple. As a motivation to the augmentation the authors mention that networks often pay too much attention to contradictory cases and tend to memorize instead of paying attention to the general features. Mixup aims to confuse the network enough for it to focus more on the general and essential parts and has proven to be successful on for example CIFAR10 and CIFAR100.

It creates virtual training samples by combining two images by

$$\begin{aligned} \tilde{x} &= \lambda x_i + (1-\lambda)x_j, && \text{where } x_i, x_j \text{ are input vectors} \\ \tilde{y} &= \lambda y_i + (1-\lambda)y_j, && \text{where } y_i, y_j \text{ are one-hot endoced labels} \end{aligned}$$

where (x_i, y_i) and (x_j, y_j) are two randomly drawn samples from the training data, and λ is a probability, that is $\lambda \in [0, 1]$. Some examples where Mixup where performed is shown in Figure 2. Usually, λ is randomly drawn from a Beta(α, α) distribution, for each pair of images which are to be combined. Examples and further details are described under Methods.

2.3 Random Erasing

Random erasing was first introduced by Zhong et al (2017). It is another photometric augmentation which puts a random sized (within some range) rectangle over some randomly chosen part of the image with some probability, see Figure 3. This has lead to "consistent improvements", for example on CIFAR10 and CIFAR100. A mentioned advantage, aside from that it prevents overfitting, is that it makes the network more robust to occlusion, which often happens when the picture is taken in lively environments.

3 Data

In this project we have worked with a dataset consisting of images of different species of birds, the Bird Species Dataset published on Kaggle. Each image has the format $224 \times 224 \times 3$ and the images are cropped such that the bird covers at least 50% of the pixels. Most pictures include the whole body of the bird. There are a total of 200 species and the training data consist of 27503 images. The data is not balanced but does however contain at least 100 training images for each species. Both the validation set and the test set consists of 5 images of each species. It should also be said that around 80% of the images are of male birds and 20% of female birds which, by the nature of birds, may look entirely different, which has sometimes caused some trouble when the data set has been used.

We will not always work with the full dataset but instead use the following categories:

- All birds (200 species)
- Bright colored birds (84 species)
- Dull colored birds (41 species)

The dull and bright colored species have been determined by inspection. For each category we have made subsets with a random selection of 15 birds and a small and medium versions, the sizes giving the number of training images per bird species, not including augmented images. We have used 5 for small and 50 for medium, giving 6 subsets in total, 7 with the complete data.

4 Methods

To get a deeper understanding of the impacts different kinds of data augmentations we have chosen four different categories: *Basic Data Augmentations* which consists of transformations such as rotations and flips, *Mixup*, which blends images and labels for different images, *Random Erasing*, which erases a random part of an image and lastly *Fourier Transforms*, which we consider as an interesting experiment.

4.1 Basic Data Augmentations

The idea behind Data Augmentation is to increase the relevant training data using the available data. This is obtained by manipulating the image such that it appears to be a new image with new valuable information to the network. For this project we have chosen some of the most common and simple Data Augmentation techniques to see what difference simple changes can make (or not), and to be able to compare them with more sophisticated methods.

Figure 1 demonstrates some simple augmentations, where we have four versions of the same picture where rotation, flipping, brightness and shearing (a kind of stretching) have been applied. Other common techniques are location shifts and zooming, but since the pictures are cropped such that at least 50% of the pictures are covered by the bird, applying any of those techniques often leads to the bird being beheaded or loosing a major body part, which might not be a relevant image in this dataset as most images appear to be professional photos with the bird centered and having its whole body in the picture.

Figure 1: Rotation, shearing, flipping and brightness adjustments applied to a picture

4.2 Mixup

As mentioned under Related Work, Mixup creates virtual training example by combining two images by

$$\begin{aligned} \tilde{x} &= \lambda x_i + (1-\lambda)x_j, && \text{where } x_i, x_j \text{ are input vectors} \\ \tilde{y} &= \lambda y_i + (1-\lambda)y_j, && \text{where } y_i, y_j \text{ are one-hot endoced labels} \end{aligned}$$

where $\lambda \sim \text{Beta}(\alpha, \alpha)$. This distribution seems like a reasonable choice since it has the right support(0 to 1) and the Beta-distribution is among the most natural distributions to consider when working with probabilities. An example of Mixup performed on images can be seen in Figure 2. The Beta(α, α)-distribution is also symmetric around 0.5, which may be a desirable property, however this should not

matter since we combine our randomly drawn examples with weights λ and $1 - \lambda$ and it should not matter if, for example $\lambda = 0.2$ or $\lambda = 0.8$ since it would yield the same two weights, but in different order, but since our examples are randomly drawn the order of the weight should not have an impact. The value of α is usually chosen in $[0.1, 0.4]$ and too high values of α leads to underfitting.

The above labeling is what we call fractional labeling. We will also consider performing Mixup on the input vectors of the training images but letting $\tilde{y}$ keep the label of the example with the highest weight, which we call majority labeling. That is,

$$\tilde{y} = I_{\{\lambda \leq 0.5\}} y_i + (1 - I_{\{\lambda \leq 0.5\}}) y_j,$$

Figure 2: Three examples of Mixup performed om images of different bird species.

4.3 Random Erasing

Random erasing is a bit similar to Mixup in the sense that it aims at confusing the network a little by manipulating the image, hopefully just enough for it to see the more general traits and overfit and memorize less.

We have chosen to let the box be different nuances of gray, so that the network does not focus on the color. Random colorization of the pixels in the box has proven to often be the most successful, however we though that in this dataset using more colors might confuse, and also gray(white-black) was easier to implement.

We have constrained the box to cover between 10 and 20 percent of the image and let the probability of a box being placed be 0.3. Figure 3 shows possible versions of a training image.

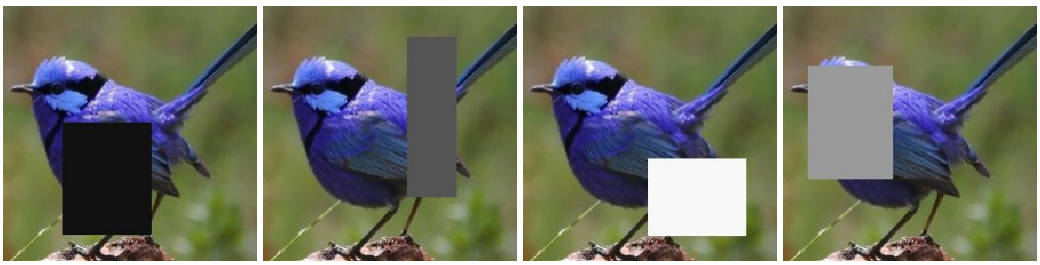

Figure 3: Four different random erasings for an image

4.4 Fourier Transformation

The 2D Fourier transform is used in for instance in image compression and gave us the idea of the possibility using it in deep learning and image classification. The 2D fast Fourier transform takes our image, with size $[224 \times 224 \times 3]$, as input and returns a complex valued matrix of the same size. From this output we can now get the amplitudes by taking the absolute value of each element in the matrix and the phase angles by computing $arctan(y, x)$, where x is the real part and y is the imaginary part. An interesting property is that the phase angels are more important than the amplitudes and contain more information necessary to recreate the image again. As an example, in Figure 4 the Fourier transform was applied to two images which yielded amplitudes and phase angles respectively. The

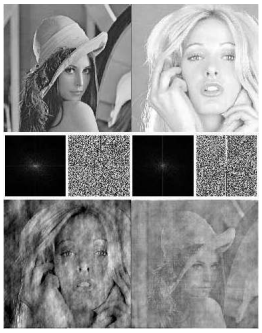

Figure 4: Example of recreation of images using amplitudes and phase angles of the Fourier transformation. Bottom left image use the amplitudes from the first image and the phase angels from the second image and bottom left image use the phase angles from the first image and the amplitudes from the second image

images were then recreated using the amplitudes from the first image and the phase angles from the second image and vise versa.

Using this as the inspiration we wanted to see if it is true that the phase angles contain more essential information about the image by applying image classification on the amplitudes and phase angels respectively to see if our hypothesis of higher accuracy on the phase angles is satisfied or not.

Initially we had no expectations that using the Fourier transform as augmentation would yield an increase the classification accuracy compared to using the raw data, however we thought that it would be an interesting experiment. In Figure 5 we can see an example of the Fourier transformation on one of the images in our dataset.

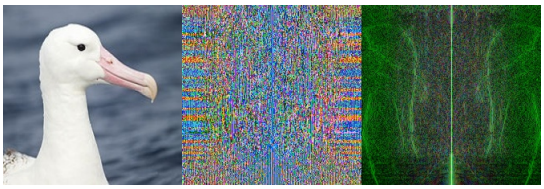

Figure 5: Left: Original image. Middle: Phase angles. Right: Amplitudes.

4.5 Underlying CNNs

In order to examine the effects of our data augmentation methods we wanted to observe them for some different networks. Due to restrictions in time and computational resources we were only able to focus on two. Below are the networks we examined.

(i) Basic CNN (about 65 % accuracy on the whole data set)

(ii) ResNet50 trained on ImageNet (about 95% accuracy on the whole data set)

For the Basic CNN the structure of the layers are shown in Figure 6.

We have chosen CNNs since they have shown to perform good, sometimes extremely good, on computer vision tasks. Batch Normalization has been a necessity in order to manage the gradients and to keep the network stable. We have further used dropout to avoid overfitting which we have anticipated to encounter, both considering the fact that we have 85 million parameters and also because we will try the network on subsets with very few training examples. Overall we have tried out some standard layer structures until reaching about 60% accuracy on the training data set.

We also chose to use ResNet50 in order to observe how the data augmentations techniques alter the performance in the case of transfer learning. In the case of ResNet50 we flattened the output and added our own predictive layer in order to accommodate to our data.

Figure 6: Layers of the Basic CNN

5 Experiments

The networks were trained for 20 epochs in each experiment we used a batch size of 100 and a learning rate of 0.01, in combination with momentum (0.9) and some decay (1e-6). We also implemented early stopping in order to cut down on the training times. The choices of hyperparameters were made by using generally recommended values and then slightly tuning them to fit our needs. As the aim of this project is not to maximize the performance of the network but rather to examine the effects of certain methods the search for good hyperparameters could be seen as very coarse.

Recall that there are the same number of species in all subsets of the data set (15), what differs is the number of training images(excluding augmentations) for each species (5 or 50).

It should be said that each combination has only been run once and thus we do not know anything about the span of variation of the results. It would have been desirable to try out some augmentations together, run more on the whole dataset and to test ResNet50 without pre-training, but that was far too time consuming for this project.

5.1 Basic Augmentation

In Table 1 the results are shown for the runs with the CNNs on all subsets(except the full data sets is which is excluded since the medium ones have performed very well already) with and without data augmentations. For ResNet50, where the Small data sets performed extremely well, we did not run anything for the Medium sized sets for that reason.

For all subsets for both our basic CNN and ResNet50 we can see a great gain in accuracy when applying data augmentations. The difference is the biggest for the basic CNN, but improvements for already good performing networks are hard earned and it is really good that such improvements can be made for an already such good performing network with such little effort.

The basic CNN had quite some problems with the RandomSmall dataset and seems to overfit, here the augmentations made the most difference. The ResNet50 did not have such problems, maybe because it is pre-trained or was better at spotting the relevant parts. The ResNet50 also performed more evenly across different data sets.

Table 1: Accuracies for Basic Augmentations on subsets

Data	Basic CNN	Basic CNN w. Aug	ResNet50	ResNet50 w. Aug
RandomSmall	0.13	0.47	0.81	0.85
RandomMedium	0.65	0.79	0.85	not tested
BrightSmall	0.40	0.53	0.79	0.84
BrightMedium	0.84	0.90	0.99	not tested
DullSmall	0.27	0.38	0.83	0.89
DullMedium	0.57	0.81	0.99	not tested

As the Random data sets perform very similarly to the Dull ones we continue only with the Bright- and Dull subsets in our further experiments.

5.2 Mixup

In order to examine the effects of Mixup we created our own data generator which used two copies of our main data generator and combines the generated images in order to create the Mixup images as

displayed in Figure 2 using the formulas presented under Methods. We run the test with $\alpha = 0.2$ since for example in Zhang et al(2019) values between 0.2 and 0.4 are recommended.

As we can see in Table 2 the basic CNN with Mixup performs about the same as without any augmentations at all, sometimes a bit better and sometimes even worse. The performance is a bit more stable on ResNet50 but has no notable positive effect.

Table 2: Accuracies for Mixup on subsets

Data	Basic CNN	Basic CNN w. Mixup	ResNet50	ResNet50 w. Mixup
BrightSmall	0.40	0.31	0.79	0.80
BrightMedium	0.65	0.73	0.99	0.97
DullSmall	0.27	0.31	0.83	0.82
DullMedium	0.57	0.57	0.99	0.99

At first sight it was a bit surprising that Mixup has not given any effect, as it has proven to be very powerful. We thought it might be because there is quite little training data in the subsets on which we have run tests, so we examined it on the whole data set and tried out other values of α, the results can be seen in Table 3 below. Note that the first row indicates the training without Mixup.

Table 3: Accuracies for Mixup with different parameter settings on the whole data set

Parameter	Label	Basic CNN	ResNet50
-	-	0.64	0.95
0.1	Majority	0.48	0.98
0.1	Fractional	0.51	0.98
0.2	Majority	0.58	0.92
0.2	Fractional	0.54	0.82

For $\alpha = 0.1$ we see a strong effect of Mixup on the whole dataset for ResNet50, for $\alpha = 0.2$ however the performance is worse than for the baseline with no augmentation, therefore we do not try for bigger values. We tried to run ResNet50 with Mixup with $\alpha = 0.1$ on the BrightSmall subset, which gave a accuracy of 0.64, which is really bad. As Mixup has a regularizing effect it probably needs fine tuning on every subset it is run, it is likely that it also needs a bit more training data than the other techniques we have tried. We did not have time to try this out. This is also likely the reason as to why Mixup leads to a decrease in the accuracy of the basic CNN which already regularizes quite heavily using dropout.

5.3 Random Erasing

As we can see in Table 4, the performance of Random Erasing on the Basic CNN is worse than with no augmentation at all. Probably it regularizes too much for our Basic CNN and the network, which already has very little training data loses even more information. However, ResNet50 seems to be picking up much more details and here the regularization did well and bumped the performance quite a bit.

Table 4: Accuracies for Random Erasing on subsets

Data	BasicCNN	Basic CNN w. RE	ResNet50	ResNet50 w. RE
BrightSmall	0.4	0.34	0.79	0.89
BrightMedium	0.65	0.83	0.99	0.97
DullSmall	0.27	0.20	0.83	0.89
DullMedium	0.57	0.51	0.99	0.99

5.4 The Fourier Transform

In order to examine which components from the Fourier transform, that is the angles, the amplitudes or the combination of them, which generates the best results we examined their respective performances

using the basic CNN architecture and the whole dataset. It should be noted that the training time increases drastically when taking both the angles and the amplitudes into account as the dimensions of the input increases.

Table 5: Accuracies for Basic CNN using Fourier Transform on the whole data set

Data	Accuracy
Raw Images	0.65
Fourier Angles	0.29
Fourier Amplitudes	0.48
Fourier Angles & Amplitudes	0.47

For the basic CNN it seems as if using only the amplitudes generated the highest accuracy which seems to contradict what we initially thought, given the visuals in Figure 4, about the phase angles containing more information about the content of the image. In all, it seems as if the Fourier transform of the input data is not a valid method in order to improve the performance of a CNN, or at least not in the case of our bird data. An interesting extension could be to apply this method to different data sets with more internal variety, such as for example CIFAR100.

Due to computational constraints as well as the quite clear results from the performance on the Basic CNN we did not evaluate the effects of the Fourier transform on other data sets nor using ResNet50. A major reason for this is that the pre-trained weights of ResNet50 from ImageNet are in no way compatible with our angles and amplitudes which would imply a complete re-training using the base architectures.

6 Conclusions

In our experiments we have learned that some basic augmentations such as rotations and flips can make a large difference in the accuracy regardless of the previous performance of the network.

It is important to remember that Mixup and Random Erasing regularize, and we want different amounts of regularization depending on what kind of dataset we have, how much data we have and how much information the network is picking up. This could partially help explain why the basic CNN which implements dropout does not benefit from these two techniques as further regularization is not needed, os opposed to for ResNet where the payoff is great.

What we have implemented is *input Mixup* where we do all augmentation before training and input the images in the network after performing Mixup, however Verma et al (2019) suggest a new algorithm called *Manifold Mixup* where Mixup is performed at an intermediate layer or final layer in the network. In an intermediate layer the feature spaces are more aligned than that of the input and it is suggested that Mixup will produce better augmented data if the Mixup is performed on this layer. Therefore, this would be a natural possible future extension to this project, as we think regularization later on might be good when we have little data so that the network may learn by the non-manipulated data to begin with.

For our experiment with the Fourier transformation we learned that it was very ineffective as a possible data augmentation method and that our hypothesis that the results would be better when using the phase angles than the amplitudes was incorrect in this application.

Finally, all our experiments could use some further optimization in the parameter choices, batch size and number of epochs and also some different tests and on a different dataset, however due to the limitations in time and computational power, this was not possible.

References

Shorten & Khoshgoftaar (2019) A survey on Image Data Augmentation for Deep Learning, *Journal of Big data 6*

Verma et al (2019) Manifold Mixup: Better Representations by Interpolating Hidden States *https://arxiv.org/pdf/1806.05236.pdf*

Zhang et al (2018) Mixup: Beyond Empirical risk minimization *ICLR conference paper 2018*

Zhong et al (2017) Random Erasing Data Augmentation *https://arxiv.org/pdf/1708.04896.pdf*