Image Augmentations for GAN Training
– Further Research

Nir Shaul
nir.shaul@campus.technion.ac.il

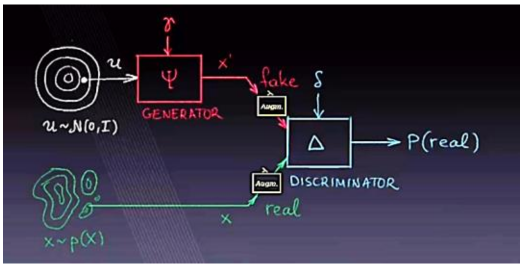
Eldad Ohayon
eldadoh@campus.technion.ac.il

Deep Learning on
Computational Accelerators Course
(CS236781)
Sep. 2020

Abstract

Data augmentations have been widely studied to improve the accuracy and robustness of classifiers. However, the potential of image augmentation in improving GAN models for image synthesis has not been thoroughly investigated in previous studies. In this work, we followed the systematically study of the "Image Augmentations for GAN Training" writers, and recover some of their results, while we implemented more new augmentation techniques for GAN training in a variety of settings. We validate the claim that augmentation for real and fake images before Discriminator phase, does improve the learning while. We also validate the claim that augmentations that result in spatial changes improve the GAN performance more than those that induce mostly visual changes.

We followed the systematically analysis method for evaluating the results using FID measurement. Our analysis was built from combining many pieces of codes, and using new code for augmentation implementation, and for experiments and analysis.



Intro

Data Augmentation has played an important role in deep representation learning. Especially in the field of Generative adversarial networks (GAN) which are tend to be unstable and hard to train. As we learn in the course there are some rules of thumb for good learning, but our scope of the course did not focus on this aspect of data augmentation in that field. We learn from the proposed paper of Augmentations for GAN Training, that this issue has known for long time as a method for improving GAN results, buy this issue has never investigates systematically as this paper did, and we follow as will be explain next.

The method of Data Augmentation, is known for improving the results while there are 2 explanation for that: one is during enriching the data and the ability of the DNN for learning features that are more unique. The other reason for using data augmentation is for enlarging the amount of training set. We will focus in this paper on the first one: we shall enrich our data without enlarging the training set or the training time (epochs) at all to focus on the importance of enriching the data with different aspects and new methods of data augmentation for GAN's.

One of the main results the paper found is that spatial augmentations were more effective than visual ones. This is not a surprise because we know from our experience in the course that spatial specific learning might end with over-fitting, and as we learn in the course there are many methods to deal with it and enlarging the receptive field. We did surprised by the fact that visual changes did not result as effective. We thought that maybe the visual augmentation that Zhengli Zhao et al. used, were wrong and maybe other methods might get improvements. In our first brain storming we mentioned the following new augmentation:

Augmentation	Operation type
Gamma Correction (>1, <1)	Visual
Image rotation	Spatial
Horizontal stretching	Spatial
Vertical stretching	Spatial
Noise, sharpening, blurring	Visual

Table 1 - new augmentation proposed for GAN learning

Methods

In this paper we follow the methods that was proposed in the paper, in order to validate the paper results (as much as possible while they did not provide any code), and in order to generate new methods as further research which are investigated in the same analysis tools and methods.

The Original approach was using experimental method to evaluate the effect of different types of augmentation on GAN learning. They conduct extensive experiments to assess the efficacy and robustness for different augmentations.
They focus on Vannila BigGAN architecture and learned over the data set of CIFAR10. The evaluation metric was Fréchet inception distance (FID), which is a very common metric for the evaluation of GANs.

They applied the augmentation in 2 ways:

1. Only on the Real images.
2. On real and fake images.

Moreover, they use parametric value named lambda for the definition of the strength of the augmentation that was applied. Lambda was a value in the range between 0.1 to 1, while 0.1 is for weak augmentation, and 1 is the strongest.

The method of FID for evaluating the quality of GAN were new to us as it never been discussed in the course. While we learn about this method, we found that it is very difficult and not trivial at all to evaluate

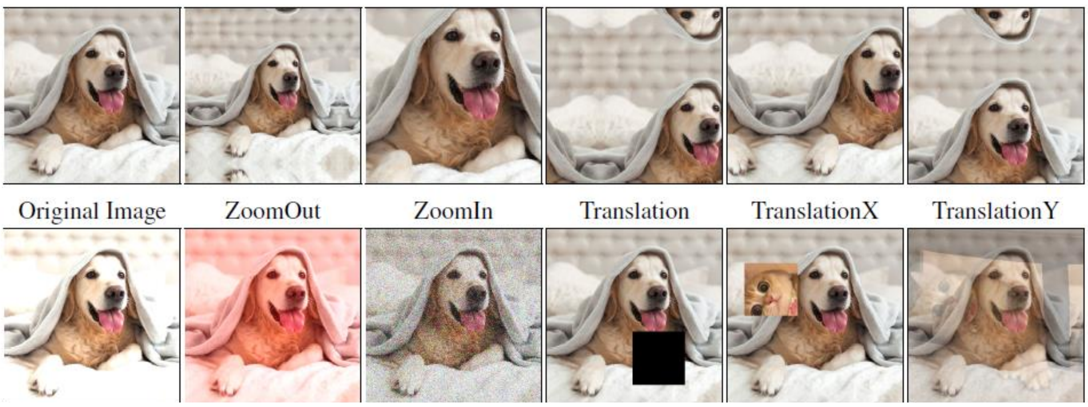


Figure 1- augmentation method applied in the original paper

generator quality. This way of FID metric, make use of other good feature extraction net (inception net), that operates on the generated samples. Than the FID score is calculated using the following equation: $d^2 = \|\mu_1 - \mu_2\|^2 + Tr\left(\boldsymbol{C_1} + \boldsymbol{C_2} - 2\sqrt{\boldsymbol{C_1 C_2}}\right)$.

While μ_i refer to the feature-wise mean of the real and generated images, e.g. 2,048 element vectors where each element is the mean feature observed across the images, $\boldsymbol{C_i}$ are the covariance matrix for the real and generated feature vectors.

As they found that adding augmentation for the fake and the real images, were much more effective, we decided to focus on this method also.

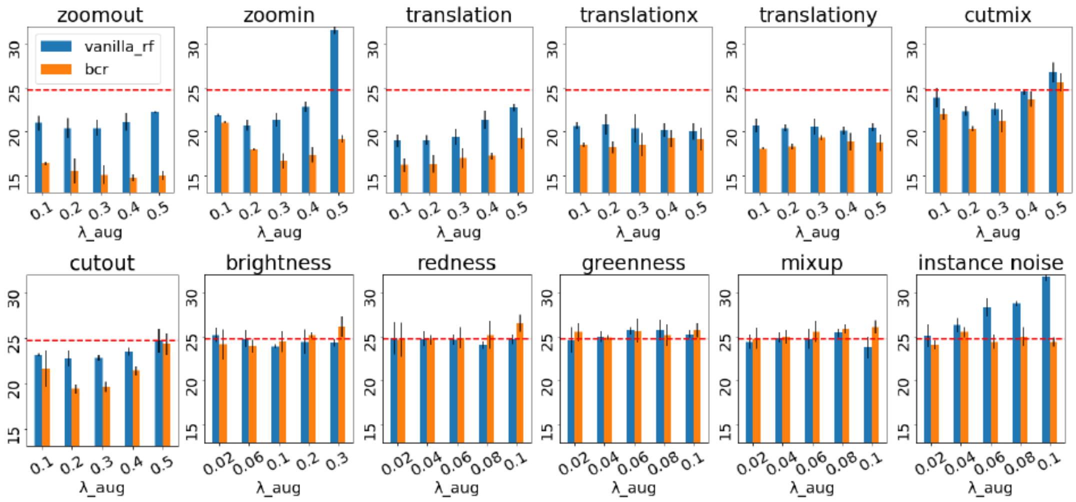


Figure 2 - results of augmentation applied in the original paper

Implementation and experiments

As explained before, the original paper does not include any code, so in order to reproduce some of the results and continue from this, we had to combine it from different pieces of code. Our first decision was to stick with CIFAR10 data set which is very basic, small and common datat set for that purpose. Second we found some basic GAN network we could fully understand and could apply our extension easily. We decided to focus on the basic DCGAN of Ksuryateja[i]. This architecture is one of the popular and successful network designs for GAN. It mainly composes of convolution layers without maxooling or fully connected layers. It uses convolutional stride and transposed convolution for the downsampling and the upsampling. Architecture guidelines for stable Deep Convolutional GANs as mentioned by oumith Chintala.

We will state that this GAN is not state of the art, and was invented in 2014. The original paper used more advanced type of DCGAN and therefore its baseline is different from ours especially in terms of FID score. It is not problem for us because our method is examine the effectiveness of new augmentation and lambda values in comparison to our new baseline which is achieved without any augmentation.

As mentioned before, while writing this paper, Pytorch has no any built-in FID score function, so we had to use those of mseitzer[ii]. In this method of FID score calculation the inception model being used is

InceptionV3, while it based on the tensorflow implementation with some updates. As mentioned in source, the Pytorch FID value might be quite different from those of Tensorflow, but it is very small difference that not relevant to our scope. In the original paper they did not mentioned what type of FID implementation they used.

In our experiment we implemented the following Augmentation functions:

- TranslationX: We sample $a \sim U(-\lambda_{aug}, \lambda_{aug})$, and shift the image horizontally by $|\alpha|H$ in the direction of $sign(a)$ with reflection padding.
- TranslationY: We sample $a \sim U(-\lambda_{aug}, \lambda_{aug})$, and shift the image vertically by $|\alpha|W$ in the direction of $sign(a)$ with reflection padding.
- ColorNoise: We sample $n \sim N(0, \lambda_{aug})$, and added n to the each and every color channel, different noise per channel.

In our experiment we conducted, we scanned the λ_{aug}values and see how it affected the values of FID scores, compare to the baseline value of 112.

In order to isolate the effect of this augmentation we keep the data set size, and the learning time equal – of 23 epochs. This number was chosen as time which the net without any augmentation achieved its best result. We do not claim that this is the best operating point in augmentation cases but this issue is left for future work.

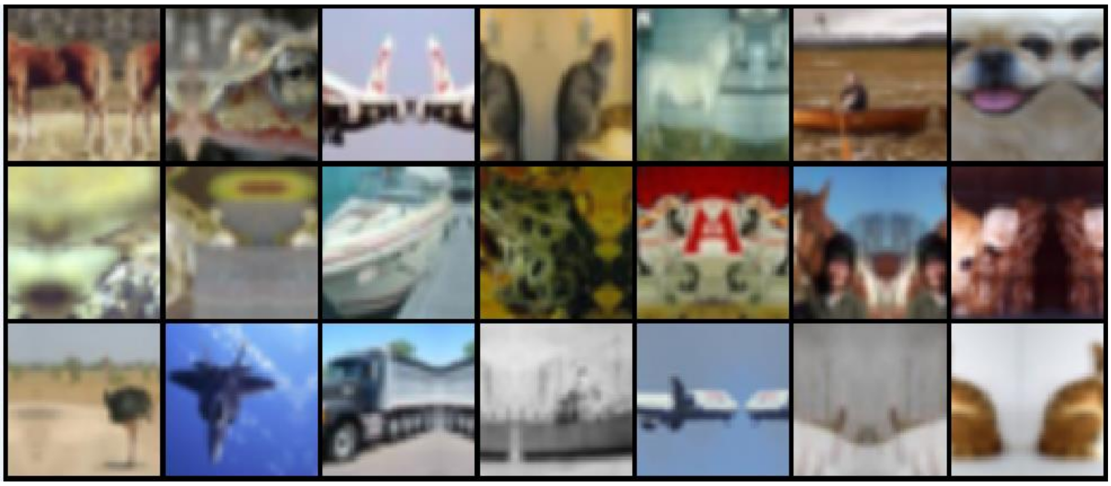


Figure 3 - real data augmented with Horizontal Translation (TranslationX) in values of lamda = 1

Results and discussion

Our main result for validating the original paper results is the effect of horizontal translation on FID score. As explain before, the original paper has different baseline for the fid score (which is better due to the use of SOA[1] architecture). But the results are the same: as can be seen from figure 4, this kind of spatial regularization that operated on both real and fake data, improved the FID score results until the strength of ~0.5. As can be expected, too big values of lambda (→ 1) are too strong and not helps for enriching the learning with valued data.

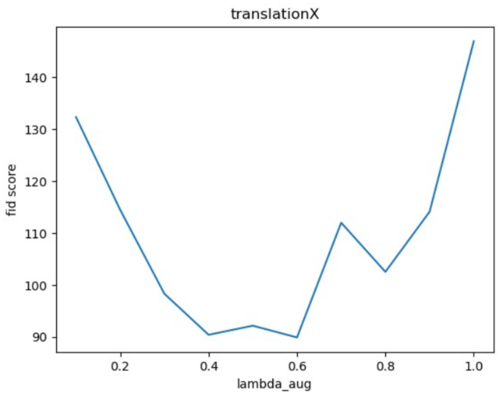


Figure 4 - Horizontal Translation augmentation effect on FID score for augmentation values from 0 to 1

As can be seen in figure 3, values of lambda→ 1, are making big shift in the horizontal axis. This might create second object (because of the mirroring), which enriching the data which most of the time include (in CIFAR10) only one object. From the other hand, it's not always creates image which is makes sense. For example, the image of plain which is very close to other plane is very wrong, also the dog with 3 eyes.

We also would like to show some visual results from our best generator. We can see in figure 5 that our results are quite good (compared to the real results of figure 3). The Generator created unique images of mammals, birds, and some vehicles, which are quite nice and makes sense visually.

In figure 6 we can see that noise in the color channels created more sharp images.

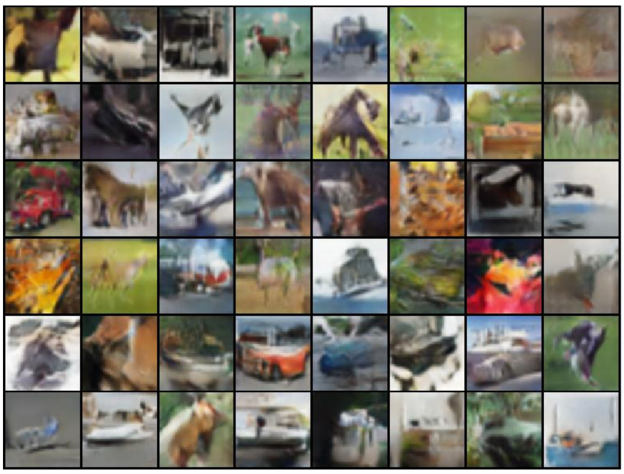


Figure 5 - resultes of generated translationX augmentation images with Lambda of 0.4 (epoch 23), fid=90.439

[1] State of the Art

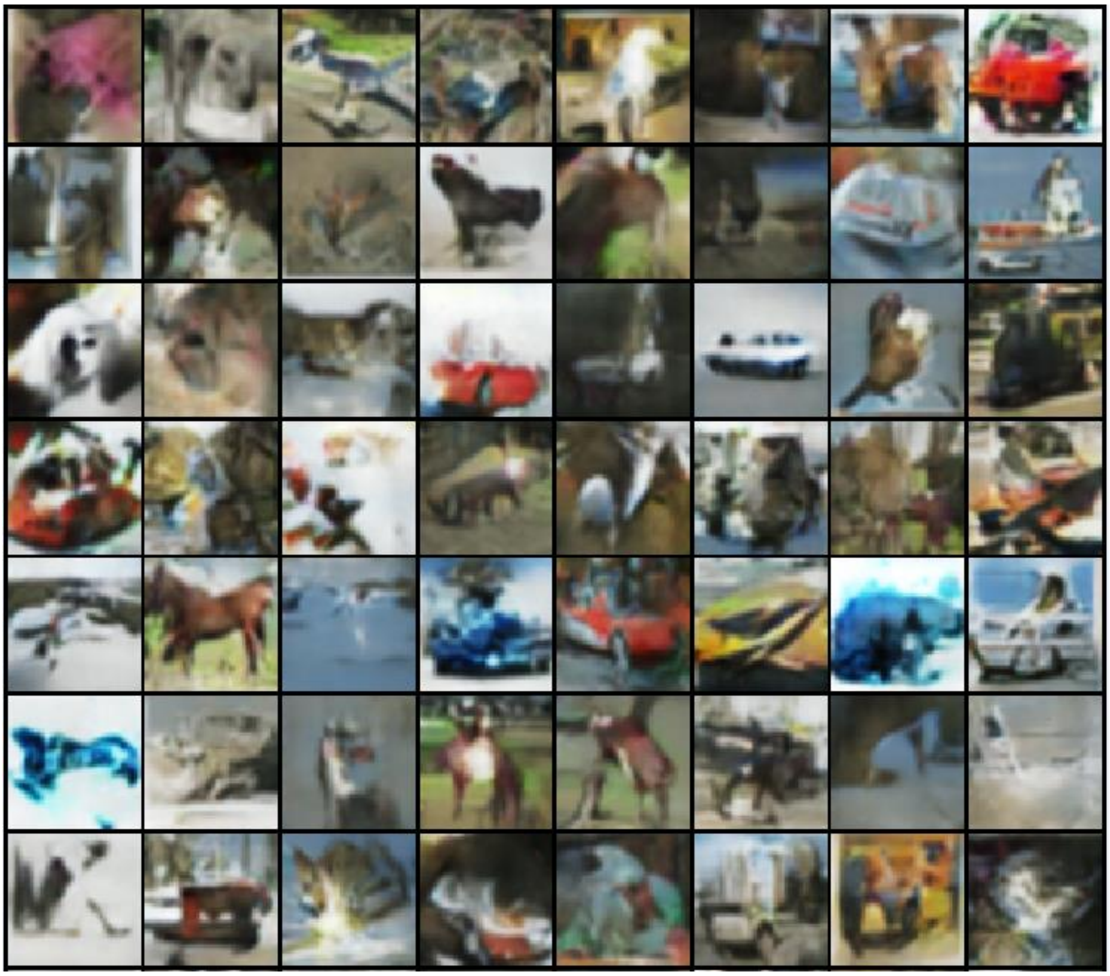


Figure 6- resultes of generated ColorNoise[2] augmentation images with Lambda of 0.7 (epoch 23)

Interesting thing we notice regarding translation Y experiments was that from lambda values bigger than 0.3, the learning was absolutely crashed. The augmentation makes the net to become not stable and ended with very bad results as can be seen in figure 7 (right).

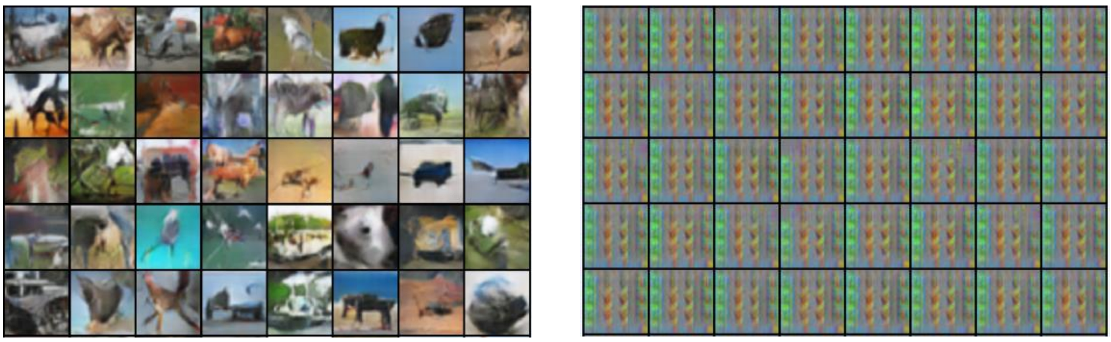


Figure7 - (left) results of translation y epoch 23 lambda 0.3 learning, (right) same setup, with lambda of 0.4

[2] Unfortunately we could not create FID results for translationY and Color noise because of some change we create in the FID code which we could not fix in time for submission of this work.

Acknowledgments

The authors would like to thank CS236781 course staff[iii], Prof. Alex Bronstein (Lecturer), Chaim Baskin (Lecturer), Aviv Rosnberg, Evgenii Zheltonozhskii and Yaniv Nemcovsky (TAs) for support on questions and time.

References

[i] https://github.com/Ksuryateja/DCGAN-CIFAR10-pytorch

[ii] https://github.com/mseitzer/pytorch-fid

[iii] https://vistalab-technion.github.io/cs236781/info/