SinGAN Data Augmentation Approach in Extremely Imbalanced Dataset for Small Images

Aiden Zhang, Zelin Li, Yihao Li
`zz2699@columbia.edu, zl2848@columbia.edu, yl4326@columbia.edu`

Abstract

For data augmentation work with GAN, most of the structures and implementations need image datasets with relatively large sample size in the minority class. As an unconditional generative model from a single image, SinGAN can be well applied in the scenario with extremely limited-size minority classes. We explored SinGAN's architectural parameters setting specific to data augmentation tasks, and then quantitatively evaluated SinGAN's performance as a data augmentation engine from its quality and diversity perspective in various image classification tasks.

1 Introduction

In classification task, the imbalanced dataset will significantly harm the model accuracy. To solve this issue, a common and effective approach is to re-balance the dataset by augmenting the minority classes with data augmentation methods. There are several known approaches, e.g. the geometric transformation built in most of the existed deep learning libraries, Google's AutoAugment[1], GANs, etc.

Under the main idea of "generating similar output as input", GAN is naturally a good candidate in the data augmentation and was already used in generating synthetic medical images, whose result showed that data augmentation with GAN improved performance[2]. However, in a extremely imbalanced dataset, there might not be enough minority-class images to train a GAN. BAGAN[3] overcame this problem by including all available training images in the adversarial training period and condition on the latent space to let the generating process generate a target class, which improved both the quality and diversity of synthetic data compared to the original GAN. But its performance is still not satisfying when the data is extremely imbalanced.

SinGAN is an unconditional model and is able to provide meaningful results for general natural images in different applications like paint-to-image, harmonization, etc[4]. What makes SinGAN unique is that it takes individual picture as input and generates imitations of this specific picture. This feature makes us believe that SinGAN may has the potential to be a data augmenting engine for extreme minority class in the image classification task.

To examine it's performance, we did the following exploration: First, SinGAN performs well in creating high-similarity imitations. However, for data augmentation, it's necessary to keep the diversity of generated data. We tested the ability of SinGAN keeping the diversity of images generated while applying internal distributions captured under different structural hyperparameter settings. Second and what's more important, we evaluated the classification results when using SinGAN as the augmentation engine, and made a comparison with other mainstream data augmentation approaches like traditional geometric transforming and state-of-the-art GAN data augmentation technique under extremely imbalanced (97.5%, 99%, and 99.5% drop from balanced class) classes.

34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

2 Methods

To create comparable results with previous works, we selected the MNIST[7] (28×28), CIFAR10[8] (32×32) and GTSRB[9] (resized to 56×56) datasets and the ResNet18 model used in previous works[2]. We evaluated the images generated with respect to both quality and diversity. For illustration, we select BAGAN[3], which is a state-of-the-art GAN data augmentation approach, and imgaug[10], a geometric transforming image augmentation library as the benchmark and tested them with SinGAN under the same framework.

2.1 Hyperparameter Tuning

The pyramidal structure's scale (the number of GAN scale) is automatically defined as

$$\text{scale} = \lceil \log_{\text{scale factor}} \left(\frac{\text{min scale size}}{\text{original image width}} \right) \rceil + 1 \tag{1}$$

and the number of convolution layers are 5 for each GAN. We added an additional integer hyper-parameters at the end of this formula in order to have a more flexible control to the number of scale(num_scale). Meanwhile we controlled the number of convolution layers (num_layers) in each generator/discriminator. We visualized the result of SinGAN's pyramidal architecture under different settings.

In the structural hyperparameter tuning part, we do not have a quantitative criterion which can simultaneously measure how the imitation keeps the structure of input image and how it maintains the diversity of the target population. Thus we generated a series of fake images from different hyperparameter settings and tested their quality and diversity separately. We manually selected the optimal setting by the criterion that if the resulting model generally keeps the class information. Then follows the following pipeline:

1. Randomly select a class, drop samples in this class to create a extremely imbalanced dataset;
2. Train SinGAN models on each images in the downsampled class with all the hyperparameter settings individually.
3. Creating augmented datasets with balance restored through different hyperparameter settings;
4. Train ResNet18 models all the restored balanced datasets
5. Evaluate the performance of the ResNet18 models in the test set.

Beside the classification accuracy, the diversity of the synthetic data was measured by the structural similarity index(SSIM), which evaluates the similarity between the two images. For each generates class, we selected 1000 randomly selected pairs drawn from the chosen class samples, calculate the SSIM and use the average as the result

We repeated this process 3 times for each downsampled classes, and evaluated on the average of results.

2.2 Cross-approaches Comparison

To evaluate the quality of data augmentation, we applied the following approach derived from the BAGAN paper[3]: With non-model based approach with geometric transforming. We applied random cropping, and random affine transformations like scaling, rotating and shearing with a limited angle. Since the classification of the handwriting of digits is related to the orientation, we did not apply flipping in the process. The detailed parameters used are described in Appendix. For BAGAN configuration, we used it's default configuration from its official release in GitHub. The evaluation pipeline is as following:

1. Randomly select a class and remove a large portion of this class to create a extremely imbalanced dataset
2. Train SinGAN and BAGAN models on the data left in the imbalanced class

3. Creating a augmented datasets with balance restored through different data augmentation methods
4. Train ResNet18 models with the imbalanced dataset(from 1) and restored balanced datasets from three approaches(from 3)
5. Evaluate the performance of the different models in the test set

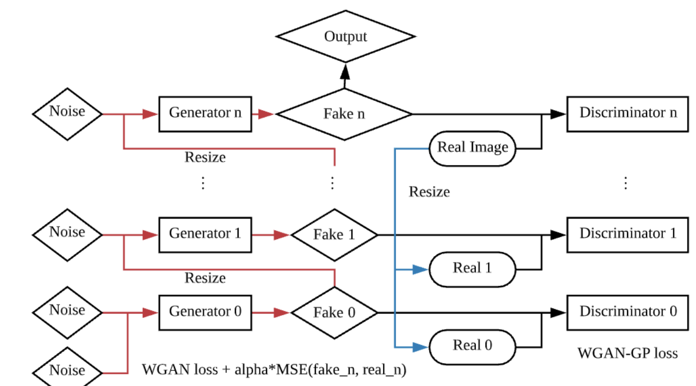
Similarly, we computed SSIM for synthetic data generated to test its diversity and compare it with the result from benchmark methods. The method is same as the one mentioned before

We repeated this process 3 times for different chosen downsampled class and evaluate on the averaged result.

For each test with same dataset, test label, and drop ratio, the data sample we use was kept the same for imgaug, BAGAN, and SinGAN

3 Architecture Implementation

SinGAN, from essence, is a multi-scale Wasserstein GAN. Each individual scale in SinGAN takes in different scale of downsampled image of original real image, and trains a Wasserstein GAN with additional gradient penalty (WGAN-GP[5]) on a pure-convolutional discriminator and additional scaled least square loss (the idea of LSGAN[6]) on a pure-convolutional generator. The fake image is generated from two parts: 1. the fake result from lower scale, 2. a random generated noise. Between adjacent scales, the difference of image size are connected by rezising and zero-paddings.



Note: Red arrows stand for the generating procedure
Blue arrows stand for pre-training operation

Figure 1: The Architecture of SinGAN

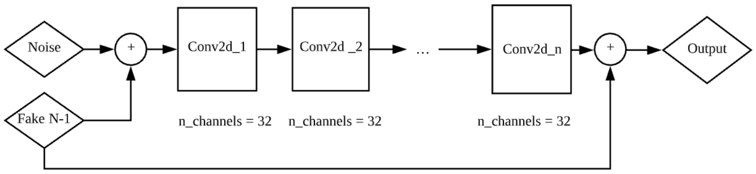


Figure 2: The architecture of SinGAN generator used

We made a TensorFlow-Keras reproduction of the original code and a PyTorch version as well (GitHub repo: https://github.com/codeconomics/DLProjectSinGAN). During the test, the TensorFlow-Keras version was significantly slower(took about 5x time as much as the PyTorch version). For the efficiency and to keep the model and data format unified, we used the PyTorch implementation in all of the following experiments.

4 Result

4.1 Hyperparameter Tuning

For a more straightforward expression, we visualized the result with a random picture in GTSRB (label 33), for each number of scales, only the smallest number of layer create "acceptable" results and the largest number of layer create "unacceptable result" is shown. For $num\_scale = 4$, $num\_layer = 5$ is somewhat controversial, thus we also showed $num\_layer = 6$ result.

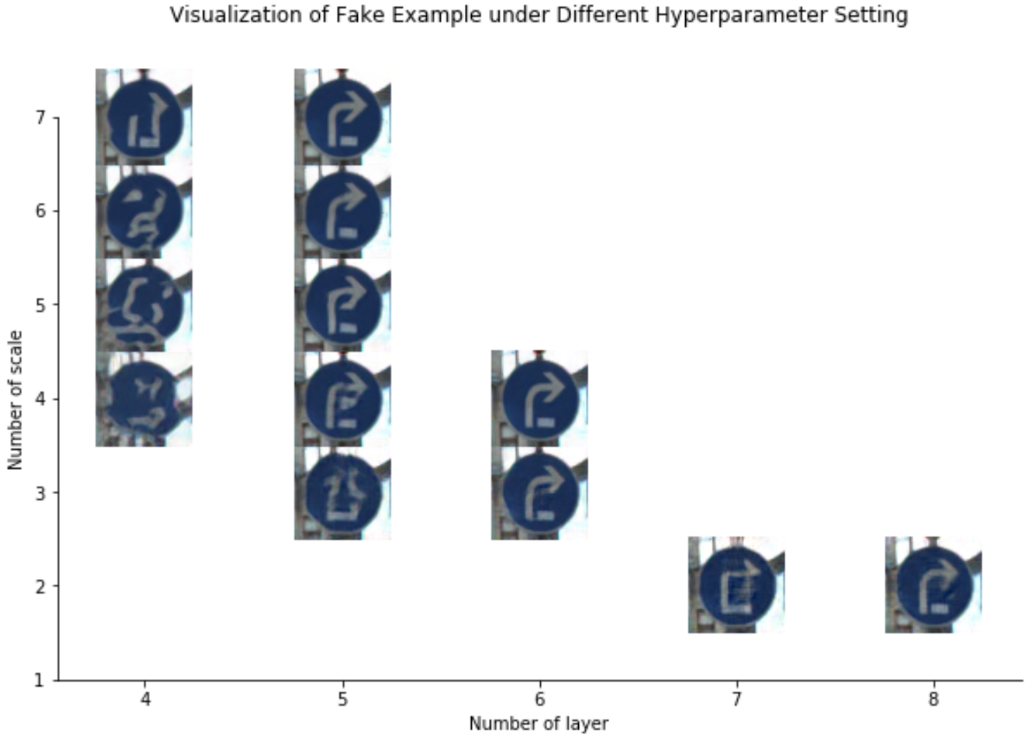


Figure 3: The visualization of tuning

From the illustration, the official release default $num\_layer = 5$ can be consider as the minimal number of layer possible keeping the quality of fake image generated. In the quantitative measuring, we listed the following 4 best-performing settings with following results:

Table 1: Hyperparameter tuning result

Hyperparameter	accuracy	SSIM
num_layer = 5, num_scale = 5	0.7095	0.3802
num_layer = 5, num_scale = 4	0.6714	0.3463
num_layer = 6, num_scale = 3	0.5667	0.3762
num_layer = 6, num_scale = 4	0.7190	0.4304

From the visualization along with the quantitative measurement shown above, we concluded that the official default hyperparameter $num\_layer = 5$ is one of the best performing settings. From our experiments, when the number of scale is set lower with higher num_layer, the model produced comparable result with a higher time efficiency. However, if need to generate images with strong structural requirement e.g. human face image, the number of scales should be kept as large as possible[4]. Any $num\_layer \leq 4$, from our experiments, need at least 3 more scales and the result is not promised, thus is not recommended for small scale images. Considering that the num_scale is not controllable in official release of SinGAN, we use the officially defined num_scale in all the following experiments and only tune the num_layer. The settings used with different dataset are listed in the appendix.

4.2 Augmentation Qualities

With the quality evaluation method described above, we randomly chose class 2, 4, 7 for MNIST dataset, 3, 8, 9 for CIFAR10 dataset, class 13, 15, 33 for GTSRB dataset.

For a class at a time, we artificially made it imbalanced by dropping a specific amount of data with that class. For MNIST and CIFAR-10 we removed 97.5%, 99%, and 99.5% samples. The GTSRB dataset is imbalanced, so we first downsampled to 150 images for each class, and further made it imbalanced by keeping 15(90% dropped), 10(93.3% dropped), 5(96.7% dropped) images for the class chosen.

We trained a ResNet18 model on the original dataset, downsampled imbalanced dataset, and dataset restored balance with geometric transformation, BAGAN augmentation, and SinGAN augmentation. We evaluated the accuracy on the downsampled class on the test set, and averaged the results across the 3 classes chosen for each dataset. For CIFAR-10 dataset, the accuracy on the minority class is very low (close to zero) for all augmentation methods, as the result of class 3 shown in table 3.

What should be noticed is that both the classification accuracy and the SSIM score are highly dependent on the class and sample, therefore cross-experiment comparison does not give much information.

Table 2: Minority Class Accuracy for MNIST dataset

Method	97.5% drop	99% drop	99.5% drop
imbalanced data set	0.6028	0.4012	0.3380
geometric transform	0.9246	0.8503	0.5254
BAGAN aug	0.8930	0.6221	0.6252
SinGAN aug	0.8961	0.9460	0.8747
full dataset	0.9806		

The accuracy on the minor class decreased when the dropping proportion increased. All augmentation methods presented similar result when the dropping proportion is 97.5%. In more extremely imbalanced classes, with 99% or 99.5% dropped, SinGAN outperformed other methods.

Table 3: Minority Class Accuracy for CIFAR-10 dataset

Method	97.5% drop	99% drop	99.5% drop
imbalanced data set	0.002	0	0
geometric transform	0.021	0.015	0.117
BAGAN aug	0.001	0	0
SinGAN aug	0.031	0.003	0.001
full dataset	0.7026		

All data augmentation techniques performed bad in this experiment. CIFAR-10 dataset images present high variation in each sample's composition, thus both BAGAN and SinGAN can not provide informative simulated samples.

GTSRB is a dataset with generally the same composition (same traffic sign under different light environment and different angle). Since we only keep 5 to 10 samples in the minority class, the fluctuated accuracy trend may be strongly influenced by the sample we selected. However in each column, SinGAN maintained a relatively good performance.

In summary, SinGAN generally performs better than (or about the same as) BAGAN and geometric transform in a dataset with relatively static composition setting. Both SinGAN and BAGAN are not performing well with CIFAR10, which has a dynamic image composition setting.

Table 4: Minority Class Accuracy for GTSRB dataset

Method	15 left	10 left	5 left
imbalanced data set	0.8523	0.7714	0.0238
geometric transform	0.9619	0.8142	0.3857
BAGAN aug	0.8714	0.8714	0.5285
SinGAN aug	0.8380	0.8809	0.7095
full dataset	0.9761		

4.3 Augmentation Diversity

To evaluate the diversity, we calculate the SSIM on the 3 chosen classes of 1) original real images, 2) geometric transform simulated images, 3) BAGAN simulated images 4) SinGAN simulated images. The results are as followed. It should be noticed that because of the single-image processing characteristic, SinGAN's SSIM is highly dependent on the remaining samples after downsampling. As mentioned, the result illustrates the comparison in our experiments but can not be generalized to a universal measurement between experiments.

Table 5: SSIM for MNIST dataset

Method	97.5% drop	99% drop	99.5% drop
geometric transform	0.1620	0.1497	0.1530
BAGAN aug	0.5278	0.9658	0.9778
SinGAN aug	0.2844	0.3126	0.3071
Real data	0.2949	0.2993	0.3180

For the MNIST dataset, SinGAN augmentation generates simulated samples with the diversity most close to the real images in all scenario. The geometric transformation method gives the most diverse simulated samples, but the variance is higher than the real images. BAGAN augmentation generated images are less diverse comparing to real images, and the diversity dropped severely when the class is extremely imbalanced.

Table 6: SSIM for CIFAR-10 dataset

Method	97.5% drop	99% drop	99.5% drop
geometric transform	0.0578	0.0564	0.0553
BAGAN aug	0.7971	0.9044	0.9353
SinGAN aug	0.0456	0.0554	0.0668
Real data	0.0332	0.0316	0.0253

Although experiments on CIFAR-10 did not give a very informative accuracy data, we can still find that for small sized natural images with variant item structure, SinGAN's structural variety is mostly inherited from the dataset itself, while BAGAN generates almost identical images on the same input data.

Table 7: SSIM for GTSRB dataset

Method	15 left	10 left	5 left
geometric transform	0.1464	0.1381	0.1981
BAGAN aug	0.4552	0.3767	0.5064
SinGAN aug	0.2427	0.2474	0.3652
Real data	0.2276	0.1837	0.1657

In GTSRB, SinGAN data augmentation presents lower SSIM than BAGAN.

From these three tests, we can conclude that SinGAN's augmented data generally follows the diversity of the minority set, while BAGAN will generate very similar augmenting data.

5 Conclusion

SinGAN's unique architecture makes it an valid data augmentation engine for extremely imbalanced data. Based on our test results, its performance can be on par with or even better than other state-of-the-art GAN augmentation techniques when the composition of the image is generally fixed. For a more variant composition, as other GAN techniques will fail, it will not be as effective as well. By SinGAN's characteristic, the data generated will inherit the diversity of the input data even with very small minority class, which is an advantageous point when compared with some of its counterparts like BAGAN in situations with extremely small size minority class.

References

[1] Cubuk, E. D., Zoph, B., Mane, D., Vasudevan, V., & Le, Q. V. (2019). Autoaugment: Learning augmentation strategies from data. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 113-123).

[2] Frid-Adar, M., Klang, E., Amitai, M., Goldberger, J., & Greenspan, H. (2018, April). Synthetic data augmentation using GAN for improved liver lesion classification. In 2018 IEEE 15th international symposium on biomedical imaging (ISBI 2018) (pp. 289-293). IEEE.

[3] Mariani, G., Scheidegger, F., Istrate, R., Bekas, C., & Malossi, C. (2018). Bagan: Data augmentation with balancing gan. arXiv preprint arXiv:1803.09655.

[4] Shaham, T. R., Dekel, T., & Michaeli, T. (2019). Singan: Learning a generative model from a single natural image. In Proceedings of the IEEE International Conference on Computer Vision (pp. 4570-4580).

[5] Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., & Courville, A. C. (2017). Improved training of wasserstein gans. In Advances in neural information processing systems (pp. 5767-5777).

[6] Mao, X., Li, Q., Xie, H., Lau, R. Y., Wang, Z., & Paul Smolley, S. (2017). Least squares generative adversarial networks. In Proceedings of the IEEE International Conference on Computer Vision (pp. 2794-2802).

[7] LeCun, Y. & Cortes, C. (2010). MNIST handwritten digit database. , .

[8] Krizhevsky, A., & Hinton, G. (2009). Learning multiple layers of features from tiny images.

[9] J. Stallkamp, M. Schlipsing, J. Salmen, C. Igel, Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition, Neural Networks, Volume 32, 2012, Pages 323-332, ISSN 0893-6080, https://doi.org/10.1016/j.neunet.2012.02.016.

[10] Jung, Alexander B., et.al, imgaug, https://github.com/aleju/imgaug

Appendix

1. For geometric transformation data augmentation, the transformations include cropping 10% of the sides, scaling along the x and y axis from 80% to 120%, translating along x and y axis by at most 20% percent in either directions, rotating by at most 25 degrees, and shearing by at most 8 degrees
2. For SinGAN data augmentation, we used 6 layers and 3 scales for MNIST, 5 layers and 3 scales for CIFAR-10, 5 layers and 5 scales for GTSRB. We used 10 as alpha and 25 as min_scale_size(default). We trained 2000 epochs for each model.
3. The image index we selected for balanced version of GTSRB are saved in our github repo: (https://github.com/codeconomics/DLProjectSinGAN)