The Effectiveness of Generative Models for Data Augmentation in Medical Applications

Hava Chaptoukaev

MSc Data Science and Artificial Intelligence
Université Côte d'Azur

Abstract. The objective of this project is to explore and compare multiple solutions for the problem of data augmentation in the medical domain. The generation of artificial training data for machine learning tasks can be very useful in situations where researchers face imbalanced data sets or insufficient number of observations, and previous work has already shown the effectiveness of simple augmentation techniques such as cropping, and rotating images. In this work we focus on data augmentation using generative models and experiment with autoencoders, variational autoencoders and generative adversarial networks.

Keywords: data augmentation · autoencoder · variational autoencoder · generative adversarial network

1 Introduction

Effective training of deep neural networks requires large amounts of data. In fact, over the last decades neural networks have reached unprecedented performance levels on a wide range of tasks, when given sufficient data. The effectiveness of those models, and particularly image classification models, is largely reliant on the amount and the diversity of data available during training, and the more data an algorithm has access to, the more effective it can be. Unfortunately, real world image classification data sets are often not uniformly distributed between different classes, and specific complex tasks can make it hard to get a sufficiently large number of observations to train those models. This is especially true in the medical domain where imbalanced data sets are common, and patients' data privacy policies make it hard to gather subsequent data sets. As a consequence, medical image analysis is often constrained by the availability of labelled training data. To overcome these problems, techniques have been developed over the years. Data augmentation methods are part of them, and aim to alleviate the data availability problems by using existing data more effectively. In this work, we explore the problem of data augmentation for medical images classification and evaluate different techniques based on generative models such as autoencoders, variational autoencoders (VAEs) and generative adversarial networks (GANs). We test each augmentation method in turn on the MedMNIST Classification Decathlon [1] data sets, and evaluate the impact of data augmentation on the final classification scores. Finally, we discuss the effectiveness of each generative models for this task.

2 Related Work

The field of data augmentation is not new, and in fact, various techniques have been applied to specific problems in the past. In this project we are interested in the use of generative models for this task. In previous work in this field, Babei et al. [2] proposed an approach based on an autoencoder capable of deriving meaningful features from high-dimensional data sets while doing data augmentation at the same time for the purpose of anomaly detection. Along these lines, Jorge et al. [3] conducted an experiment to asses whether the use of VAEs to produce new artificial data could be helpful for training various supervised models. In more recent work, GAN-based approaches have been proposed for data augmentation. GAN networks have been introduced by Goodfellow et al. [4] as a powerful tool to artificially generate realistic images. The underlying idea is to simultaneously train two models: a generative model G that captures the data distribution, and a discriminative model D that estimates the probability that a sample came from the training data rather than G. Radford et al. [5] extended this idea and proposed Deep Convolutional GANs (DCGAN) as a way to build good image representations by using parts of the generator and discriminator of a GAN as feature extractors for supervised tasks. Arjovski et al. [6] later brought optimization improvements and reduced some of the failures of the GAN learning process by introducing the Wasserstein GANs. Among GAN applications for data augmentation we can cite Frid-Adar et al. [7], who used GANs to generate synthetic medical images to improve the performances of a CNN trained for liver lesion classification. Similarly, Tanaka et al. [8] also proposed the use of GANs as a way to generate artificial training data for supervised learning tasks, and demonstrated that decision trees classifiers showed good results using augmented data. Another interesting type of networks used for data augmentation are conditional GANs in which class information is fed to the generator. For instance, Odena et al. [9] proposed the Auxiliary Classifier GAN (ACGAN), a variant of GANs employing labels to train the discriminator to perform classification in addition to discriminating between real and fake data, which in turn allows the generator to learn representative class samples. Along these lines, Antoniou et al. [10] proposed the Data Augmentation GAN (DAGAN) framework to increase the performance of vanilla classifiers. In this framework the generator is essentially an autoencoder that learns a shared family of transformations to generate a synthetic image. The DAGAN discriminator distinguishes between a pair of images from the same class, which incentivizes the decoder to learn transformations which do not change the class. Similarly, Miriani et al. [11] proposed the Balancing GAN (BAGAN) network as an augmentation tool to restore balance in imbalanced data sets. Once again, in this network the generator in the GAN uses elements of an autoencoder to learn the distribution of the overall dataset. Then, class conditionning is applied in the latent space to drive the generation process towards a target class.

3 Experiment

3.1 Proposal and Evaluation Method

To evaluate the utility and efficiency of generative models as a data augmentation tool we set up a simple experiment. We compare the performances of a vanilla image classifier on the original data sets we have with the data sets augmented by several different generative models in turn. Inspired from the MedMNIST decathlon paper [1], we implement A ResNet-18 with a simple early-stopping strategy on validation set as a classification method. In this model, the input channel is 1 for grey-scale data sets, and 3 for triple-channel data sets. The input resolution is 28. The model is trained for 100 epochs, using a cross entropy loss and a stochastic gradient descent optimizer with a batch size of 128 and an initial learning rate of 1×10^{-3}. For simplicity, Area under ROC curve (AUC) and Accuracy (ACC) are used for the evaluation of the classifier performance on each data set. AUC is a threshold-free metric to evaluate continuous prediction scores, and ACC is the ratio of number of correct predictions to the total number of input samples. While ACC works well if there are equal number of samples belonging to each class, AUC is less sensitive to class imbalance.

3.2 Augmentation Methods

We select 3 simple models as augmentation methods and discuss their objectives and architectures in this section. Given, the small resolution of the images of the MedMNIST data sets, we choose to work with basic architectures and fully connected layers.

Deep Autoencoder Autoencoders are a type of neural networks used to learn a representation for a set of data. The aim of an autoencoder is to find the function mapping an input x to itself. This objective is known as *reconstruction*, and an autoencoder accomplishes this through the following process: first an encoder learns the data representation z from the input features x in lower-dimension space by extracting the most relevant features of the data. Let h_e be the hidden layers of the encoder, and f the encoder function, then z is given by:

$$z = f(h_e(x)) \tag{1}$$

Then a decoder learns to reconstruct the original data based on the learned representation by the encoder. Let h_d be the hidden layers of the decoder, and g the decoder function, then the data reconstruction $\hat{x}$ of x is given by:

$$\hat{x} = g(h_d(z)) \tag{2}$$

To optimize our autoencoder to reconstruct data, we minimize the following reconstruction loss,

$$\frac{1}{N}\sum_{i=1}^{N} ||\hat{x_i} - x_i||^2 \tag{3}$$

The reconstruction loss is this case is actually the mean-squared error loss. We implement a simple autoencoder with 4 hidden fully connected layers for both its encoder and decoder components. The model is trained for 300 epochs, using an Adam optimizer with a batch size of 128 and an initial learning rate of 1×10^{-3}.

Conditional VAE VAEs, introduced by Kingma and Welling [12], aim to model the underlying probability distribution of a data set so that it could sample new data from that distribution. Let X be the data we want to model, z a latent variable, and $P(X)$ the probability distribution of the data. The objective of the VAE is to find $P(X)$, which using the law of probability can be written as follows,

$$P(X) = \int P(X|z)P(z)dz \tag{4}$$

where $P(z)$ is the probability distribution of latent variable, and $P(X|z)$ the distribution of generating data given latent variable. $P(z)$ being unknown, the idea is to infer $P(z)$ using $P(z|X)$. Although, $P(z|X)$ being also unknown, a simpler distribution Q that is easy to evaluate, e.g. Gaussian, is used to model it and minimize the difference between those two distribution using KL divergence metric. The structure of a VAE is similar to the structure of an autoencoder. That is, $Q(z|X)$ is the encoder net, z is the encoded representation of the data, and $P(X|z)$ is the decoder net. The objective function of a VAE, known as the variational lower bound, is given by:

$$E[\log P(X|z)] - D_{KL}[Q(z|X)||P(z)] \tag{5}$$

In the case of data augmentation, it can be useful to have corresponding labels for new generated data, and therefore we direct our choice towards a conditional version of a VAE. Sohn et al. [13] introduced the conditional VAEs (cVAE) to model latent variables and data, both conditioned to some random variables. While vanilla VAEs allow no control over the data generation process, the cVAEs can be used to generate specific data. Let c be the condition, which in our case of are image labels, the cVAE encoder is now conditioned to two variables X and c and is given by $Q(z|X, c)$. Similarly, the decoder becomes $P(X|z, c)$. Finally, the variational lower bound becomes:

$$E[\log P(X|z, c)] - D_{KL}[Q(z|X, c)||P(z|c)] \tag{6}$$

We implement a simple cVAE with 2 hidden fully connected layers for both its encoder and decoder components. The model is trained for 300 epochs, using an Adam optimizer with a batch size of 128.

Conditional GAN The GAN framework [4] establish a min-max adversarial game between two neural networks: a generative model G that captures the data distribution, and a discriminative model D. The discriminator model, $D(x)$, is a neural network that estimates the probability that a point x in data space is

a sample from the data distribution that we are trying to model, rather than a sample from our generative model. Concurrently, the generator uses a function $G(z)$ that maps samples z from the prior $p(z)$ to the data space. $G(z)$ is trained to maximally confuse the discriminator into believing that samples it generates come from the data distribution. In fact, GANs are rooted in game theory, and their objective is to find the Nash Equilibrium between a D and G. The solution to this game can be expressed as:

$$\min_G \max_D E_{x\sim p_{data}}[\log D(x)] + E_{z\sim p_z}[\log(1 - D(G(z))] \tag{7}$$

Once again, for the task of data augmentation, we direct our choice towards a conditional version of a GAN. Mirza and Osindero [14] introduced conditional GANs (cGAN) to control and guide the generator. In the cGAN framework, the discriminator's evaluation is done not only on the similarity between fake data and original data but also on the correspondence of the fake data image to its input label. We implement a simple version of a cGAN with 3 hidden fully connected layers for both the generator and the discriminator. The model is trained for 1000 epochs, using an Adam optimizer with a batch size of 128 and an initial learning rate of 1×10^{-4}.

3.3 Data Sets

The experiments of this project are made on 8 of the 10 open medical data sets of the MedMNIST decathlon [1] collection. The MedMNIST data is standardized and pre-processed into the same format and same small size of 28×28 pixels, covers diverse tasks, and is split in training-validation-test subsets. In addition, MedMNIST covers the primary data modalities in medical image analysis. To reflect the size of data sets available in medical imaging community, we artificially restrict our access to data to small subsets of the MedMNIST data sets. Cho et al. [15] empirically answered the question of how many images are necessary for training in medical image analysis. The authors evaluated the accuracy of a CNN with GoogLeNet architecture in classifying individual axial CT images of 6 different body regions. With 200 training images per label, accuracies of 88-98% were achieved on test sets of 6000 images. Inspired by [15], we select random subsets of our original data sets in order to keep approximately 150 images per class, and preserve a train-validation ratio of 9:1. We then generate 150 additional images per label to exceed the previous threshold of 200 images per class. Table 1 represents an overview of our data base. In addition, the constraint of small data sets will also allow to highlight the effectiveness of the various tested augmentation methods.

4 Results

4.1 Experiment Results

For each data set, the training set is augmented using the 3 augmentation methods in turn. The ResNet-18 is trained on original data and augmented data, and

Table 1. An overview of MedMNIST data subsets.

Name	Data Modality	# Classes	#Training	#Validation	#Test
PathMNIST	Pathology	9	1,350	150	7,180
DermaMNIST	Dermatoscope	7	1,050	117	2,005
OCTMNIST	OCT	4	600	67	1,000
PneumoniaMNIST	Chest X-ray	2	300	34	624
BreastMNIST	Breast Ultrasound	2	546	78	156
OrganMNIST_Axial	Abdominal CT	11	1,650	184	8,889
OrganMNIST_Coronal	Abdominal CT	11	1,650	184	8,268
OrganMNIST_Sagittal	Abdominal CT	11	1,650	184	8,829

Table 2. Overall performance of MedMNIST in metrics of AUC and ACC.

Data set/Augmentation Method	None		AE		cVAE		cGAN	
	AUC	ACC	AUC	ACC	AUC	ACC	AUC	ACC
PathMNIST	**0.936**	**0.740**	0.924	0.708	0.887	0.614	–	–
DermaMNIST	**0.843**	**0.698**	0.832	0.682	0.820	0.686	–	–
OCTMNIST	0.791	0.501	**0.833**	**0.564**	0.813	0.543	0.803	0.557
PneumoniaMNIST	0.942	0.845	**0.949**	0.842	0.942	**0.859**	0.948	0.856
BreastMNIST	0.872	0.807	0.885	0.833	**0.901**	**0.864**	0.850	0.796
OrganMNIST_Axial	0.971	0.781	0.978	**0.819**	**0.981**	0.810	0.972	0.783
OrganMNIST_Coronal	0.976	0.824	0.982	0.844	0.982	0.842	**0.979**	**0.845**
OrganMNIST_Sagittal	**0.940**	0.641	0.939	**0.659**	0.928	0.644	0.931	0.652

the best test accuracy in 100 epochs is recorded. The results of the experiment are reported in Table 2. For several data sets, autoencoder-based augmentation, VAE-based and GAN-based augmentation perform better than no augmentation. In particular on BreastMNIST data, the cVAE augmented training performed 86.4% ACC against 80.7% without augmentation. Similarly on axial OrganMNIST data, the Autoencoder augmented training performed 81.9% ACC against 78.1% without augmentation. Inversely, the tested augmentation methods did not improve the classifier's performance on PathMNIST and DermaMNIST data.

4.2 Analysis

The PathMNIST and DermaMNIST data sets are respectively pathology and dermatoscope images. Both data sets are color images representing complex patterns with a high level of details, which explains why our simple linear models have not been able to efficiently learn and reconstruct these images (See Appendix A). Unsurprisingly the performance of the classifier on these augmented data sets decreases, due to the low quality of generated images. On the other hand, images from the OrganMNIST and the PneumoniaMNIST data sets, respectively CT scans and X-rays, have high contrast in colors and represent easily distinguishable shapes, making it easier for simple models to learn

their representation (See Appendix A).Similarly, the BreastMNIST ultrasound images, although blurry, represent distinguishable shapes allowing the augmentation models to reconstruct them easily. Overall, as expected the cVAE yields solid fake images as the model samples from a learned latent space. In the case of the deep autoencoder, the new samples are not *generated* from scratch but are actually reconstructions of existing images, the task seems less complex and requires less data to train, and it thus makes sense the autoencoder-based augmentations perform better. Although, in the case of the cGAN, convergence of the losses is very hard to achieve, and the model generates very noisy images.

5 Conclusion

We implemented basic generative models for the task of data augmentation. On simple images with distinguishable features, autoencoder-based and VAE-based models were able to generate fake images quite similar to the real ones, which in turn brought improvements to our baseline classifier performances. It is worth noting that adding convolutional layers to our models could bring good improvements to the quality of the generated images. Overall, the improvement brought by data augmentation is not huge. This can be explained by the fact that generated data is by definition similar to the existing data, meaning no new examples were really added to the original data sets. In conclusion, it appears generative models can indeed be useful for the task of data augmentation, however the problem of generating data from small data sets may be as hard as the task the data is generated for, especially in medical applications. In addition, created data might pick up the patterns existing in original data sets, replicating original biases and imbalances.

References

1. Yang J., Shi R., Ni B., 2020. MedMNIST Classification Decathlon: A Lightweight AutoML Benchmark for Medical Image Analysis. *arXiv preprint*, arXiv:2010.14925.
2. Babaei K., Chen Z., Maul T., 2019. Data Augmentation by AutoEncoders for Unsupervised Anomaly Detection. *arXiv preprint*, arXiv:1912.13384.
3. Jorge J., Vieco J., Paredes R., Sánchez J. A., Benedí J. M., 2018. Empirical Evaluation of Variational Autoencoders for Data Augmentation. in *VISIGRAPP* (pp. 96-104).
4. Goodfellow I. J., Pouget-Abadie J., Mirza M., Xu B., Warde-Farley D., Ozair S., Bengio Y., 2014. Generative Adversarial Nets, *arXiv preprint*, arXiv:1406.2661.
5. Radford A., Metz L., Chintala S., 2015. Unsupervised representation learning with deep convolutional generative adversarial networks, *arXiv preprint*, arXiv:1511.06434.
6. Arjovsky M., Chintala S., Bottou L., 2017. Wasserstein GAN, in *International conference on machine learning* (pp. 214-223).PMLR.
7. Frid-Adar M., Diamant I., Klang E., Amitai M., Goldberger J., Greenspan H. , 2018. GAN-based Synthetic Medical Image Augmentation for increased CNN Performance in Liver Lesion Classification, in *Neurocomputing*, 321, 321-331.

8. Tanaka F. H. K. D. S., Aranha C., 2019. Data Augmentation Using GANs, *arXiv preprint*, arXiv:1904.09135.
9. Odena A., Olah C., Shlens J., 2017. Conditional image synthesis with auxiliary classifier gans, in *International conference on machine learning* (pp. 2642-2651). PMLR.
10. Antoniou A., Storkey A., Edwards H., 2017. Data augmentation generative adversarial networks, *arXiv preprint*, arXiv:1711.04340.
11. Mariani G., Scheidegger F., Istrate R., Bekas C., Malossi C., 2018. BAGAN: Data Augmentation with Balancing GAN, *arXiv preprint*, arXiv:1803.09655.
12. Kingma D. P., Welling M., 2013. Auto-Encoding Variational Bayes, *arXiv preprint*
13. Sohn K., Lee H., Yan X., 2015. Learning structured output representation using deep conditional generative models, in *Advances in neural information processing systems*
14. Mirza M., Osindero S., 2014. Conditional Generative Adversarial Nets, *arXiv preprint*
15. Cho J., Lee K., Shin E., Choy G., Do S., 2015. How much data is needed to train a medical image deep learning system to achieve necessary high accuracy?, *arXiv preprint*, arXiv preprint arXiv:1511.06348.

Appendix A cVAE Generated images

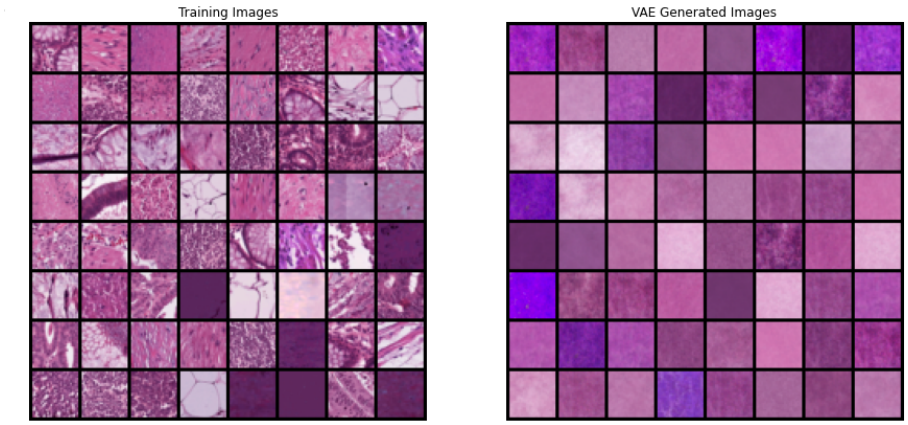

Fig. 1. cVAE generated PathMNIST images.

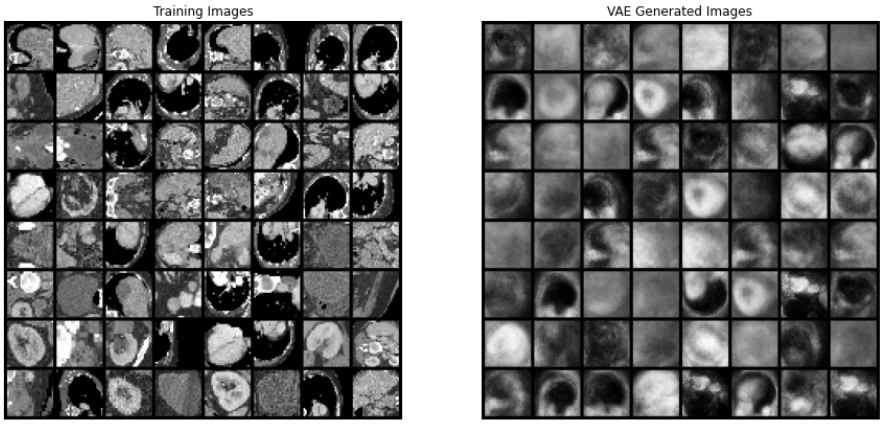

Fig. 2. cVAE generated axial OrganMNIST images.