EasyChair Preprint

№ 3948

Data Generation Using Gene Expression Generator

Zakarya Farou, Noureddine Mouhoub and Tomáš Horváth

EasyChair preprints are intended for rapid dissemination of research results and are integrated with the rest of EasyChair.

July 25, 2020

Data Generation Using Gene Expression Generator

Zakarya Farou[1][0000−0003−3996−2656], Noureddine Mouhoub[2], and Tomáš Horváth[1,3][0000−0002−9438−840X]

[1] Department of Data Science and Engineering, Telekom Innovation Laboratories, Faculty of Informatics, ELTE - Eötvös Loránd University Pázmány Péter sétány 1/C, H-1117, Budapest, Hungary
{zakaryafarou,tomas.horvath}@inf.elte.hu
[2] Science and Technology Campus, Bordeaux Computer Science Laboratory - LaBRI, University of Bordeaux, 351 Cours de la Liberátion, F-33400, Talence, France
noureddine.mouhoub@labri.fr
[3] Institute of Computer Science, Pavol Jozef Šafárik University Jesenná 5, 040 01 Košice, Slovakia
http://t-labs.elte.hu/

Abstract. Generative adversarial networks (GANs) could be used efficiently for image and video generation, where labeled training data are available in bulk. In general, building a good machine learning model requires a reasonable amount of labeled training data. However, there are areas such that the biomedical field where the creation of such a data set is time-consuming and requires expert knowledge. Our goal is to use data augmentation techniques as an alternative to data collection to improve data classification. We propose the use of a modified version of GAN named Gene Expression Generator (GEG) to augment data samples at hand. The proposed approach was used to generate synthetic data for binary biomedical data sets to trains existing supervised machine learning approaches. Experimental results showed that using GEG for data augmentation with a modified version of leave one out cross-validation increased the performance of classification accuracy.

Keywords: Data generation · Generative adversarial networks · Gene expression data · Cancer classification

1 Introduction

The automated diagnosis of oncology diseases such as colon cancer is extremely complex and requires careful attention. Gene expression profiling is widely adopted to learn and analyze the conditions of cells and their response to diverse states, which is helpful in the pathogenesis of diseases. Gene expression microarrays can be used for diagnostic purposes as well as for insights into biology. *Gene Expression Data* (GED) is a high dimensional data having a high number of features that indicate gene levels, however, with only a very few records. Usually,

we have a satisfying amount of measurements for each tumor sample, where each measurement belongs to a particular gene.

Getting accurate results while training a classifier becomes difficult due to the lack of available, variant, and meaningful data. Thus, adding more samples is recommended to train the classifier. Augmenting the original data, i.e. generating supplementary samples from the existing ones, for the imbalanced class would steer clear of overfitting situation and improve the classification process.

1.1 Motivation

Building a satisfying Machine Learning (ML) model usually requires a large number of samples. In the biomedical domain, collecting such data is very costly and time-consuming: Experts should store, examine and annotate the recorded data (which can be either an image, information or clinical tests) to obtain a clean, meaningful and useful data set. Recently, Deep Neural Networks (DNNs) [9] has brought many improvements in ML, mainly where massive datasets are available. The popularity of DNNs led to their use also in cases where only a small number of samples are available, and simpler ML techniques, requiring less computational effort, would perform considerably well or even better than DNNs. However, in the case of very few data samples, training of any type of ML models with reasonable performance is difficult. For this reason, we will show that even the performance of simple classification techniques can be improved by providing the appropriate augmentation technique.

In the classification of cell dysplasia (cancer data), the sensitivity of the data should be maintained. This means that when we generate new instances using data augmentation techniques we need to be sure that the generated data is close to the original data not only in terms of values but also in terms of the semantics of the data.

Generative Adversarial Networks (GANs) [6], established over the last few years, have attracted the attention of researchers (see Fig. 1). Several variants of GAN have been proposed for generating high-quality synthetic data, where they have been used for data augmentation in the case where traditional data augmentation methods do not yield good results. Thus, the feasibility of using GANs as a data augmentation technique to enhance classifier performance in GED classification is worth exploring. According to our knowledge, the use of GAN in connection with cancer classification by gene expression has not been done so far.

1.2 Contributions

As collecting a large number of medical data is expensive and hard to acquire due to some privacy constraints, data generation is an alternative to data collection, notably when synthetic data can be generated from a small number of existing samples. Our contributions in this paper are the following :

– Proposal of a modified version of GAN named *Gene Expression Generator* (GEG). GEG learns the GED distribution and tries to synthesize new samples that are consistent with the original data. GEG's discriminator uses the Wasserstein distance to reflect the similarity between original and synthetic data distributions providing more stability when training. GEG also uses data restriction, i.e. the discriminator is fed only with data belonging to a single class, which helps us to avoid unnecessary steps leading to labeled generated data after the GEG training is completed. Although we only used GEG for binary classification, it would work for multi-class classification problems as well;
– Introduction of a modified version of leave one out cross-validation (LOOCV) by not merging the synthetic data with the original data but instead using the generated instances only for training the classifiers. By using the generated data as an extension of the samples for training, testing the performance of the model with the original data only, and improving the results pays more attention to data sensitivity. The generated data helps the model to understand the data better while did not interfere with the original data.

2 Related work

GAN, introduced by Goodfellow et al. [6], gained more interest very fast and researchers started to explore its capacity in a wide variety of applications [27]. The most successful application of GAN is computer vision including image translation [8], image super-resolution [10], image synthesis [28], video generation [25], face aging [2], 3D object generation [23] or detection of small objects [29]. Another application domain of GAN concerns natural language processing [5], speech processing [13], and also for text generation [26].

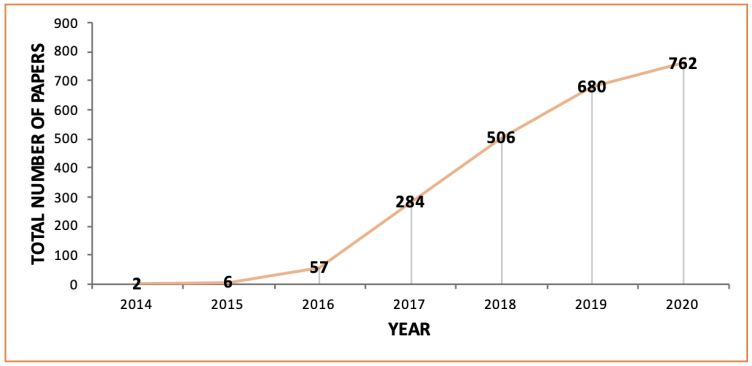


Fig. 1: Cumulative number of paper publications/journals related to GANs per year since its introduction in 2014

Many variant of GAN has been proposed during the last years, for instance, a conditional version of generative adversarial nets named CGAN [18] is extending GAN by conditioning the generator and discriminator with extra information. Deep Convolutional GAN (DCGAN) [20] has certain architectural constraints and is powerful while dealing with unsupervised learning problems. Another variant, called GRAN [7] was used to generate images with recurrent adversarial networks. MGAN [12] is a Markovian GAN, a technique for training generative neural networks for efficient texture synthesis. GAN-CLS [21] demonstrated the capability to generate plausible images of birds and flowers just by using textual descriptions. An alternative of GAN called VIGAN [22], an extended version of CycleGAN[30], handled imputations of missing data of MNIST. A distributed adversarial network called DAN was proposed in [11] where the adversarial training relies on the entire sample as a unit and not its sample points. Compared to researchers where they usually use thousands of images, Marchesi et al. [16] investigated the possibility of applying GAN to produce high-quality megapixel images where a restricted amount of data was utilized. Lu et al. [15] suggested a new approach named Bi-GAN that uses two generators dedicated only for generating synthetic data from Gaussian noise, two evaluators, and a discriminator for data classification. Later on, Wang et al. [27] suggested working with a set of generators against a single discriminator. Last, but not least, Ian Goodfellow, introduced a new a robust and simple to train method called latent adversarial generator (LAG) [3] that generates high-resolution images using latent spaces.

Reviewing GAN and its application domains revealed that most of the researchers are focusing on image/video data generation, while there are still some unexplored areas such as biomedical field, which is a delicate, sensitive and costly area (in terms of data acquisition), virus mutations (such as COVID-19), text generation, and also genetic data generation. Recently, there were some attempts to use GAN for genetic data [17]. Due to the minority of researchers operating on GAN for gene data we accepted the challenge to examine GAN beside GED.

3 Proposed gene expression generator

Generative adversarial networks (GANs) are deep neural networks based on game theory [6], they are considered as smart and creative machines. As described in Fig. 2 with red boxes, GAN has two principal components, a generator G and a discriminator D where G and D are neural networks. The output of G i.e. x' where $x' = G(z)$ is directly linked to the input of D beside original data x. G produces synthetic instances x' starting from random Gaussian noise z in such a way that x' and x are consistent. Whereas D checks how close synthetic data is comparable to the original ones and returns probability for each created instance between 0 and 1. In basic GAN, a cross-entropy loss is used to estimate the error between predicted / actual label from D output and the actual labels.

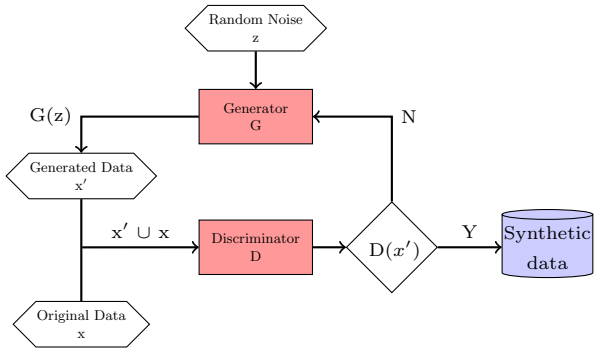


Fig. 2: Process of creating synthetic data using GAN

3.1 GEG architecture

Gene expression generator i.e. GEG is a variant of GAN that tries to replicate gene expression probability distribution. GEG uses original samples of a single class as input to generate more artificial instances referring to the same class label. Wasserstein distance WD is used as a loss function to reflect the similarity between the distribution of original and synthetic data generated by GEG. When a discriminator uses the Wasserstein loss function, it does not try to classify instances by giving them a class label (i.e. synthetic or real instance), but it attaches a number for each instance in such a way that high values are used for original data and low values to the synthetic data. However, in simple GANs, the discriminator gives a probability p for each instance, where $p \in [0..1]$ and based on a threshold (0.5, for example) it can classify instances either as artificial (negative) or original (positive) instances. A GAN may use a unique loss function θ for both G and D. Therefore, one of them (G or D) should use $-\theta$ (same loss function differing only in sign). In our case, GEG uses different loss functions, a generator loss $-\mathbf{D}(\mathbf{x}')$ where G tries to maximize this function, in other words it tries to maximize discriminator's output for its negative instances, and a discriminator loss $\mathbf{D}(\mathbf{x}') - \mathbf{D}(\mathbf{x})$ where D tries to maximize WD defined as the difference between $D(x)$ and $D(x')$, in other words it tries to maximize the difference between its output on positive instances and its output on negative instances [4].

The training data of D belongs to two groups, original data denoted by x and synthetic data denoted by x' which is generated by G. During the training of D, x and x' are used as positive and negative instances respectively. It is important to note that during the training of D, G is semi-suspended, i.e. its weights are kept constant, but at the same time G remains to generate new instances to feed D with more training data. The stopping criterion is when the

[4] x denotes original (positive) instances, x' denotes synthetic (negative) instances, z is a random Gaussian noise. $D(x)$ and $D(x')$ are discriminator's outputs for original and synthetic instances respectively, and $G(z)$ is the generator's output

critic can not distinguish between original and synthetic samples. In other words, the output of the critic for negative samples is as high as for positive samples, which makes WD very close to 0. After the training of GEG ends, we will have a set of unlabeled synthetic data. For all data belonging to that set, a class label is automatically added that's due to the way GEG is trained for (class-based data restriction). That makes the process of identifying class label for synthetic data much easier by avoiding extra steps, that is classifying unlabeled instances with a semi-supervised or a supervised approach.

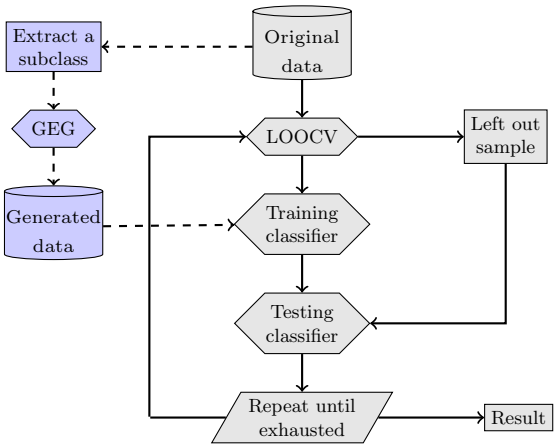


Fig. 3: Proposed LOOCV training diagram

3.2 Proposed LOOCV training diagram

Before starting the classification process, it is essential to pre-process the data. As GED features (genes) have different ranges, it might influence the classification accuracy which will result in a miss classification of the data. To avoid such a situation, the normalization step is incorporated. Normalization is the process of scaling the values of numerical columns into a unique range such as [0, 1] which will guarantee the allocation of balanced weights for each feature. Consequently, normalization minimizes training error, by this means demonstrating the classification issue's accuracy. The gene-level of each feature is normalized by Eq. 1:

$$new_{value} = \frac{actual_{value} - val_{min}}{val_{max} - val_{min}} \times (upper - lower) + lower \tag{1}$$

Where val_{max} and val_{min} are the maximal and minimal original value of a given gene respectively. *upper* is the upper bound (which is 1), while *lower* is the lower bound (which is 0). new_{value} is the normalized expression level. After normalization, all genes will be included between [0, 1].

Since we are dealing with very small and sensitive data, we use leave one out cross-validation (LOOCV) which is suitable for performance validation of models. LOOCV is a cross-validation technique where N data samples are sliced into two sets, a training set trn_{ds} containing $N-1$ samples and a test set tst_{ds} containing a single sample. A classifier is trained with trn_{ds}, and the constructed model is tested with tst_{ds} by adopting any performance measurement (such as accuracy). The procedure is repeated N times so that all instances are used once in trn_{ds} and once in tst_{ds}, but nevermore in both sets simultaneously. In addition to the traditional LOOCV training process described in Fig. 3 with the gray color, we suggest to improve training classifiers by adding more instances to trn_{ds} (purple color in Fig. 3), these instances are generated using GEG to further learning methods and improve the classification performance only on original data. It means that our models are only tested with the original GED.

4 Experiments, results and discussion

4.1 Datasets description

We used publicly available GED[5] of colon cancer tissues and breast cancer tissues [4]. These tissues are either healthy (normal histological structure tissues) or belong to one of the subtypes of cell dysplasia (cancerous tissues), for a better comprehension of gene datasets see [14]. As described in Table 1, the datasets had expression levels of 6500 and 24,481 genes measured from 62, and 99 patients respectively, both data sets are binary classification data sets. During experiments, we adopted a refined (filtred) datasets instead of original datasets as an alternative to features selection (the refined datasets contains only important genes that have a high impact factor for GED classification)

Table 1: Description of datasets used for the experiments

Dataset	Nbr of original genes	Nbr of used genes	Classes distribution
Colon cancer	6500	2000	40/22
Breast cancer	24481	4997	44/34

– **Colon cancer dataset** is composed of 40 different types of dysplasia colon tumors and 22 normal cell and histological structure of tissue samples from an Affymetrix oligonucleotide array that has 6500 genes. A clustering algorithm revealed wide consistent patterns that suggest a high degree of organization underlying gene expression in these tissues [1]. The filtered dataset has 2000 gene expressions (instead of the original 6500 genes).

[5] DNA microarray data: https://homes.di.unimi.it/~valentini/DATA/MICROARRAY-DATA/

– **Breast cancer dataset** consists of 99 tumor samples from breast cancer patients originally with 24,481 genes then decreased to 7650 genes in [24]. From the estrogen receptor i.e ER, we can divide cancer tumors into two subgroups. Subjects with ER^+ tumors have better survival than those with ER^- because, the patients with ER^+ can benefit from anti-estrogen, such as tamoxifen. As breast cancer is strongly correlated with ER, only genes which had a connection with ER are considered and thus approximately 5000 genes were kept. Of the 99 patients, only 78 patients were considered. Out of 78 patients 34 had a poor prognosis and thus were labeled as ER^- patients while 44 had a good prognosis which makes them ER^+ patients.

4.2 Used methods

To investigate the effect of GEG on classification accuracy, we compared the baseline results (simple classifiers without data augmentation) against classifiers using GEG as a data augmentation technique. The used classifiers are support vector machine (SVM), K-nearest neighbors (KNN), and decision trees (DT). We used supervised learning methods because cancer datasets are labeled. For the implementation of the algorithms, we used the sklearn library [19] with default hyper-parameters. SVM with a linear kernel and $C = 1$, KNN with $K = 5$, and DT with $max\_depth = 2$. It's important to note that for all performed experiments with GEG, synthetic samples are only used during the training stage as extra training data and not as test data. Details of generated samples per class are given in Table 2:

Table 2: Number of generated samples per class for each dataset using GEG.

Dataset	**Original C1/C2**	**Generated C1/C2**	**Total samples**
Colon cancer	40/22	10/28	100
Breast cancer	44/34	6/16	100

4.3 Evaluation metric

To evaluate the performance of GEG, we adopted classification accuracy as an evaluation metric. Accuracy is a good metric in our case because for each dataset, every classifier is trained with a balanced dataset (original data + generated data by a data augmentation technique). Accuracy is defined as the ratio between the total number of correctly classified instances and the total number of instances. Formally, accuracy can be computed by Eq. 2:

$$Accuracy = \frac{TP + TN}{Total} \tag{2}$$

Where: True Positive (TP) (resp. True Negative (TN)) are correctly classified positive (resp. negative) samples by the classifier, False Positive (FP)(resp. False

Negative (FN)) are incorrectly classified negative samples as positive (resp. positive samples as negative), and Total ($Total = TP + FP + FN + TN$) is the total number of samples.

In ML, we usually divide the dataset into a training set and test set. To give a more accurate estimate of a model's performance, we used cross-validation. In our experiments, we used LOOCV training described in Fig. 3 where the result of each iteration is the accuracy. It means that the comparative study between the proposed models and the baseline is performed using the following formula:

$$Model\,accuracy\ \% = 100 \times \frac{1}{N} \times \sum_{i=1}^{N} A_i \tag{3}$$

Where N is the number of samples, A_i is the calculated accuracy of iteration i, and of course we multiply by 100 to reflect model accuracy in %.

4.4 Classification results

We ran multiple tests for classification using different classifiers with different training approaches, for each case we used solely the original data (OD) without data augmentation and synthetic data generation using proposed GEG (proposed approach).

Table 3: Summary of experimental results based on classification accuracy

Dataset	Used method	SVM	KNN	DT
Colon cancer	OD	87.10	82.26	75.81
	GEG	**88.71**	**83.87**	**83.87**
Breast cancer	OD	60.26	51.28	57.69
	GEG	**75.64**	**57.69**	**73.08**

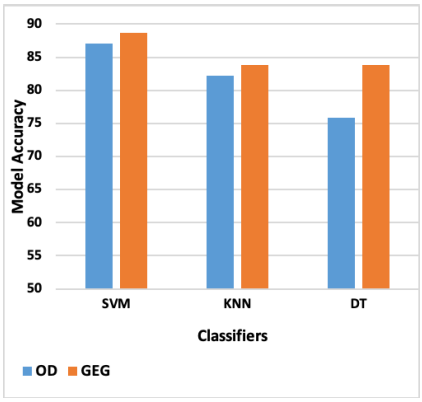
Results in bold shows that the used data augmentation i.e. GEG improved the classification accuracy compared with the baseline (original data solely).

As described in Fig. 4 and Table 3 that depicts the comparative results for classification accuracy, GEG could be seen as a promising data augmentation technique that enhances classification accuracy, we can summarize the results for each dataset as the following:

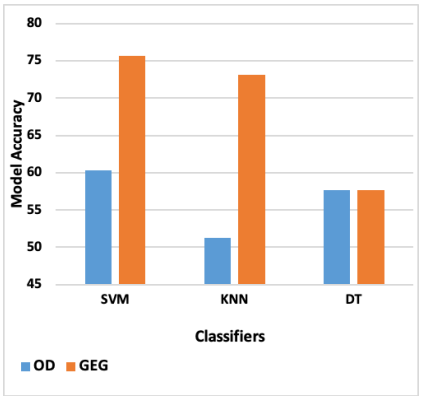
– **Colon cancer dataset:** Despite using three classifiers, GEG produced the best results in term of classification accuracy. The highest result was achieved by using GEG as a data augmentation technique and SVM as a classifier and could reach 88.71% of accuracy and 55 out 62 as a number of correctly classified instances. The results achieved by SVM are due to the linearity of the data, it means that data classes are linearly separable, this justifies the initial good results achieved without the usage of data augmentation. Therefore,

using GEG gave a slightly higher result, which means that support vectors used by SVM have been adjusted (due to the usage of GEG's synthetic data).

– **Breast cancer dataset:** For this dataset, the combination of GEG and SVM is winning again, it could improve the classification accuracy by more than 15% to reach the value 75.64% and 59 out of 78 as a number of correctly classified instances compared with the baseline using solely SVM which could only correctly classify 47 out of 78 instances. Similar to the colon cancer dataset, the improvement in results has been caused by the adjustment of support vectors, and thus the optimal hyper-plane when synthetic data generated by GEG has been used during the training process.



(a) Comparative classification accuracy on colon cancer dataset



(b) Comparative classification accuracy on breast cancer dataset

Fig. 4: Graphical representations of comparative classification accuracy on gene expression datasets

5 Conclusion

Collecting medical data for cancer detection is costly and arduous to obtain due to privacy constraints. As the available data has a disproportionate ratio between the number of available instances and the number of features and analyzing GED using solely a short amount of available samples could yield inappropriate classification outcomes, using sophisticated data augmentation approaches such as GANs with appropriate hyper-parameters could be very beneficial in such application domain. As an alternative to data collection, generating synthetic samples and expand the number of training data is suggested. However, to provide good achievements, these generated instances have to be very consistent with the original instances. The results of the classification accuracy of the duality between

GEG and simple supervised learning methods are promising. Applying GAN to more datasets and comparing it against other data augmentation techniques is yet to be done, however, at the initial stage we saw that GAN can be successfully used for producing synthetic samples that are harmonious with real samples not entirely for images, videos, and text. But also for gene expression data.

Acknowledgement

Supported by Telekom Innovation Laboratories (T-Labs), the Research and Development unit of Deutsche Telekom.

Authors would like to extend their deepest gratitude toward **Tsegaye Misikir Tashu** for his guidance, valuable advises, paper proofreading, and help in overcoming technical issues.

References

1. Alon, U., Barkai, N., Notterman, D.A., Gish, K., Ybarra, S., Mack, D., Levine, A.J.: Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. Proceedings of the National Academy of Sciences **96**(12), 6745–6750 (1999)
2. Antipov, G., Baccouche, M., Dugelay, J.L.: Face aging with conditional generative adversarial networks. In: 2017 IEEE international conference on image processing (ICIP). pp. 2089–2093. IEEE (2017)
3. Berthelot, D., Milanfar, P., Goodfellow, I.: Creating high resolution images with a latent adversarial generator. arXiv preprint arXiv:2003.02365 (2020)
4. Buza, K.: Classification of gene expression data: a hubness-aware semi-supervised approach. Computer methods and programs in biomedicine **127**, 105–113 (2016)
5. Damian, A., Piciu, L., Turlea, S., Tapus, N.: Advanced customer activity prediction based on deep hierarchic encoder-decoders. In: 2019 22nd International Conference on Control Systems and Computer Science (CSCS). pp. 403–409. IEEE (2019)
6. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Advances in neural information processing systems. pp. 2672–2680 (2014)
7. Im, D.J., Kim, C.D., Jiang, H., Memisevic, R.: Generating images with recurrent adversarial networks. arXiv preprint arXiv:1602.05110 (2016)
8. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1125–1134 (2017)
9. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems. pp. 1097–1105 (2012)
10. Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., et al.: Photo-realistic single image super-resolution using a generative adversarial network. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4681–4690 (2017)
11. Li, C., Alvarez-Melis, D., Xu, K., Jegelka, S., Sra, S.: Distributional adversarial networks. arXiv preprint arXiv:1706.09549 (2017)

12. Li, C., Wand, M.: Precomputed real-time texture synthesis with markovian generative adversarial networks. In: European conference on computer vision. pp. 702–716. Springer (2016)
13. Li, J., Monroe, W., Shi, T., Jean, S., Ritter, A., Jurafsky, D.: Adversarial learning for neural dialogue generation. In: EMNLP (2017)
14. Lin, W.J., Chen, J.J.: Class-imbalanced classifiers for high-dimensional data. Briefings in bioinformatics **14**(1), 13–26 (2013)
15. Lu, Y., Kakillioglu, B., Velipasalar, S.: Autonomously and simultaneously refining deep neural network parameters by a bi-generative adversarial network aided genetic algorithm. arXiv preprint arXiv:1809.10244 (2018)
16. Marchesi, M.: Megapixel size image creation using generative adversarial networks. arXiv preprint arXiv:1706.00082 (2017)
17. Marouf, M., Machart, P., Bansal, V., Kilian, C., Magruder, D.S., Krebs, C.F., Bonn, S.: Realistic in silico generation and augmentation of single cell rna-seq data using generative adversarial neural networks. bioRxiv p. 390153 (2018)
18. Mirza, M., Osindero, S.: Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784 (2014)
19. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research **12**, 2825–2830 (2011)
20. Radford, A., Metz, L., Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434 (2015)
21. Reed, S., Akata, Z., Yan, X., Logeswaran, L., Schiele, B., Lee, H.: Generative adversarial text to image synthesis. arXiv preprint arXiv:1605.05396 (2016)
22. Shang, C., Palmer, A., Sun, J., Chen, K.S., Lu, J., Bi, J.: Vigan: Missing view imputation with generative adversarial networks. In: 2017 IEEE International Conference on Big Data (Big Data). pp. 766–775. IEEE (2017)
23. Smith, E.J., Meger, D.: Improved adversarial systems for 3d object generation and reconstruction. In: Conference on Robot Learning. pp. 87–96 (2017)
24. Sotiriou, C., Neo, S.Y., McShane, L.M., Korn, E.L., Long, P.M., Jazaeri, A., Martiat, P., Fox, S.B., Harris, A.L., Liu, E.T.: Breast cancer classification and prognosis based on gene expression profiles from a population-based study. Proceedings of the National Academy of Sciences **100**(18), 10393–10398 (2003)
25. Vondrick, C., Pirsiavash, H., Torralba, A.: Generating videos with scene dynamics. In: Advances in neural information processing systems. pp. 613–621 (2016)
26. Wang, H., Qin, Z., Wan, T.: Text generation based on generative adversarial nets with latent variables. In: Pacific-Asia Conference on Knowledge Discovery and Data Mining. pp. 92–103. Springer (2018)
27. Wang, Z., She, Q., Ward, T.E.: Generative adversarial networks: A survey and taxonomy. arXiv preprint arXiv:1906.01529 (2019)
28. Zhang, H.: Generative Adversarial Networks for Image Synthesis. Ph.D. thesis, Rutgers The State University of New Jersey-New Brunswick and University of Medicine and Dentistry of New Jersey (2019)
29. Zhang, Y., Bai, Y., Ding, M., Ghanem, B.: Multi-task generative adversarial network for detecting small objects in the wild. International Journal of Computer Vision pp. 1–19 (2020)
30. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: Proceedings of the IEEE international conference on computer vision. pp. 2223–2232 (2017)