Chapter 11
GAN-Based Novel Approach for Data Augmentation with Improved Disease Classification

Debangshu Bhattacharya, Subhashis Banerjee, Shubham Bhattacharya, B. Uma Shankar and Sushmita Mitra

1 Introduction

In medical domain, due to scarcity of balanced a need for methods to improve upon the performance of Deep Convolutional Neural Network (DCNN) models [3]. While Transfer Learning and Domain Adaptations are possible solutions [8], there are no pre-trained networks on a large annotated dataset of medical data. This makes us look for data augmentation as a solution [23]. Classical augmentations, using transformations like translation, rotation, scaling, flipping and shearing, have been successful on a large number of datasets. However, in some medical datasets (like the chest X-ray images) some of those transformations, like rotation and flipping, should not be applied to preserve the properties of the annotated data, while the other remaining transformations lead to image duplicity and causes the model to overfit. Ever since Generative Adversarial Networks (GAN) was introduced by Goodfellow et al. [9], it has been extensively studied and used in various fields [12, 18, 30]. The generator of this network can generate the distributions of the original data,

D. Bhattacharya
Chennai Mathematical Institute, Chennai, India
e-mail: dbangshu16@gmail.com

S. Bhattacharya
Heritage Institute of Technology, Kolkata, India
e-mail: sb.skyfall@gmail.com

S. Banerjee · B. Uma Shankar (✉) · S. Mitra
Machine Intelligence Unit, Indian Statistical Institute, Kolkata, India
e-mail: uma@isical.ac.in

S. Banerjee
e-mail: mail.sb88@gmail.com

S. Mitra
e-mail: sushmita@isical.ac.in

O. P. Verma et al. (eds.), *Advancement of Machine Intelligence in Interactive Medical Image Analysis*, Algorithms for Intelligent Systems,
https://doi.org/10.1007/978-981-15-1100-4_11

and this method of synthesizing artificial data indistinguishable from original data has been used to good effect in various fields, including data augmentation [1, 4, 7, 14, 15, 24, 26–29]. GANs have already achieved state-of-the-art performance in image generation tasks on various domains. Recently it is also being applied in the medical domain. There have been several studies recently on using GAN to augment medical datasets [6, 22, 27] or using it for segmentation tasks [21]. These augmentations are 'offline' augmentations increasing the dataset size as a method to improve performance. We perform an 'online' augmentation technique keeping the dataset size as constant. For every mini-batch of images, we keep a percentage of original images and the rest are replaced by GAN-synthesized images. This 'keep-prob' parameter was tuned and we used the value 0.7 (i.e. the probability of retaining 70% data and replacing 30% data by online augmentation).

In the present work, the contribution can be summarized as follows: (1) We propose a CNN model, which has optimized hyperparameters to achieve better accuracy for disease classification using X-ray images, where data from the three classes are very small. (2) A schema is proposed for using Deep Convolutional Generative Adversarial Network (DCGAN) model effectively. This approach of online augmenting using DCGAN model-based generated images helped in regularization that minimizes the overfitting and improves the accuracy on the test dataset.

In this chapter, we proposed a novel approach of using the generated images for augmentation of the data based on GAN model to improve the performance of the proposed CNN for disease classification using X-ray images. The rest of the chapter is as follows. Section 2 present the details of the data used, whereas Sect. 3 provides the proposed approach, describing the considered GAN model and CNN model. Section 4 gives the complete details of the experiments and results obtained with analysis. At the end, we provide the conclusions in Sect. 5, with some details about the future work.

2 Chest X-ray Classification

The dataset used is a part of the NIH chest X-ray dataset provided at website[1] [25]. The data used contains images of three classes: Infiltration, Atelectasis and No Findings. The example images are provided in Fig. 1. Our dataset contains 4,215 images of patients having Atelectasis; 9,547 images of patients having Infiltration and 13,762 images of patients having Normal condition. We are splitting the data into training (23,524 images), validation (2,000 images) and testing (2,000 images) sets with each of validation and test data having 500, 500 and 1000 images from Atelectasis, Infiltration and No Findings (NF), respectively. All the images are resized to size 64×64.

[1] https://nihcc.app.box.com/v/ChestXray-NIHCC/.

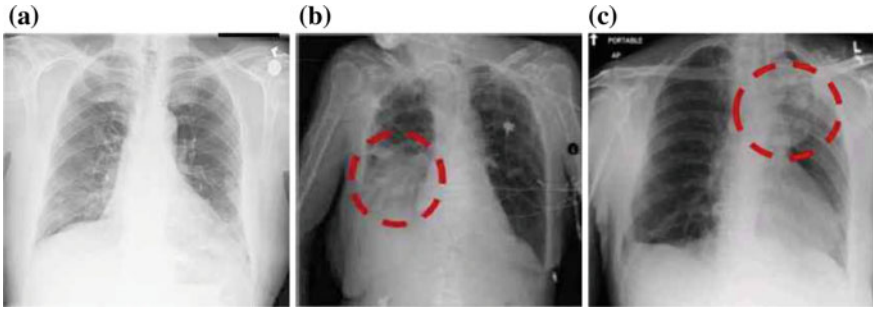


Fig. 1 Visual examples of ordinal images: **a** No Finding, **b** Infiltration and **c** Atelectasis

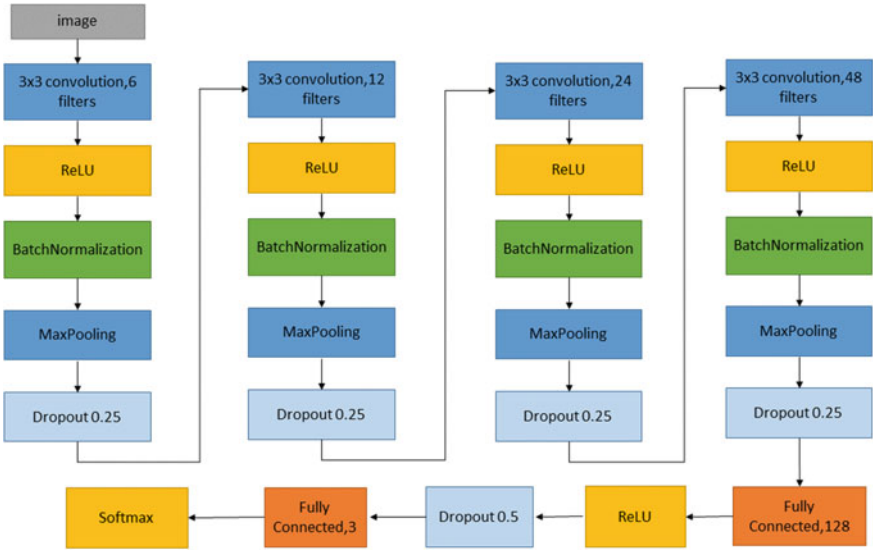


Fig. 2 The proposed model

2.1 CNN Architecture

The network architecture, as shown in Fig. 2, has four convolutional blocks followed by two fully connected blocks. Each convolutional block has a convolutional layer with ReLU activation [5] function followed by Batch Normalization [11], Max Pooling [10] and Dropout layer [5]. Each convolutional layer has a 3×3 kernel with padding ('same'), whereas each maxpooling layer has pool size of 2×2 and has a stride of 2. The kernel of the convolution layer is initialized (by 'he_normal' initialization). The final layer of our model is a softmax layer for classifying images into 3 classes.

Training: The images were normalized to the range 0–1. The loss function used was categorical cross-entropy. Adam optimizer was used with a learning rate of 0.001 [13]. Mini-batches of size 32 were used to train the model for 100 epochs. In an imbalanced dataset like ours, where most examples are from a particular class, the model tends to overfit to a particular class. To combat this, we used the class weights-parameter (weighted loss function) [19], which optimizes the model to handle misclassifications of minority class during backpropagation.

3 Generating Synthetic Chest X-rays

We use a Deep Convolutional Generative Adversarial Network (DCGAN) [20] as a generative model. Generative Adversarial Networks (GANs) comprises two networks, Generator G and Discriminator D, which compete against one another in a minimax approach and improve each other's performance. The discriminator aims to identify if the data fed to it is fake or real, whereas the generator aims to generate data which the discriminator fails to identify as fake. At the end of training, the generator can produce data which is indistinguishable from real data, successfully recreating the original data distribution. DCGANs use deep convolutional neural networks for both the generator and discriminator architecture (Fig. 3).

The discriminator takes as input a chest X-ray sample x and it outputs $D(x)$, the probability of the sample being real. The generator takes as input a 100-dimensional noise vector $z \in [0, 1]$ and maps $G(z)$ to the image I_g (generated image).

The loss function used in this adversarial training is given below:

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim p_{data}(x)} \left[\log D(x)\right] + \mathbb{E}_{z \sim p_z(z)} \left[1 - \log D(x)\right] \quad (1)$$

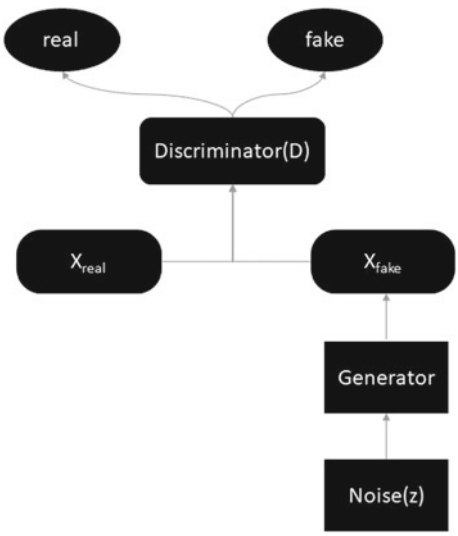


Fig. 3 DCGAN architecture

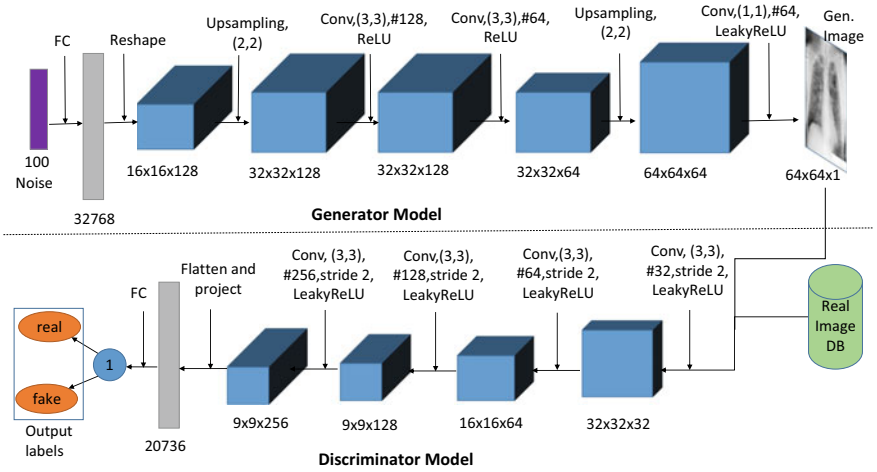


Fig. 4 Generative and discriminative model

where p_{data} is the probability distribution of the original data and p_z is the prior defined on the noise variable [9].

The Discriminator tries to maximize the loss value $V(D, G)$ whereas the generator tries to minimize it. To train the discriminator, the parameters of the generator model are made untrainable, i.e. they cannot be modified by backpropagation and the generator acts as a feedforward network. Similarly for training the Generator, the Discriminator is made untrainable.

Generator Architecture: The Generator takes as input of a 100-dimensional noise vector drawn from normal distribution and outputs a chest X-ray image of shape $64 \times 64 \times 1$ as shown in Fig. 4. There is one fully connected layer, two upsampling layers and three convolutional layers. Each convolutional layer has kernel size of 3×3 with ('same') padding. The inner two convolutional layers are followed by Batch Normalization and ReLU Activation layers. The final convolutional layer is followed by a 'tanh' activation layer.

Discriminator Architecture: The Discriminator takes a typical $64 \times 64 \times 1$ chest X-ray image as input and outputs if it is real or fake. The network comprises four stride convolutional layers followed by a fully connected layer of 1 neuron having a sigmoid activation. Each convolutional layer has a kernel size of 3×3 with stride of 2 and ('same') padding. Each convolution layer is followed by Batch Normalization, LeakyReLU activation and dropout layer.

Training: We trained the DCGAN model for each of the three classes individually. The images were normalized to the range -1 to 1 and then fed into the DCGAN model. We used Adam Optimizer with a learning rate of 0.0002 and momentum of 0.5. We used mini-batches of size 32 and trained for 50, 60 and 70 epochs for

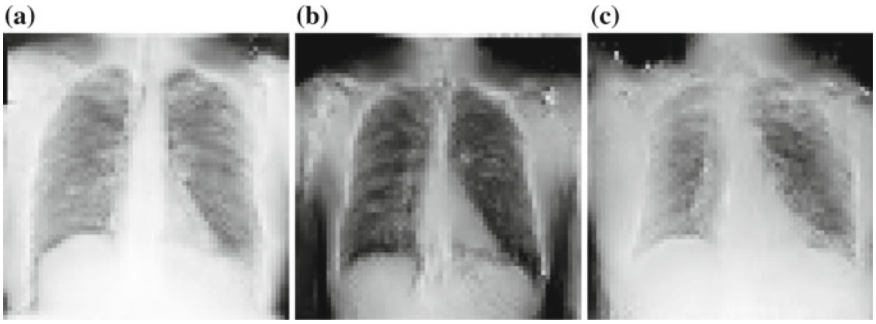


Fig. 5 Visual examples of generated images: **a** No Finding, **b** Infiltration and **c** Atelectasis

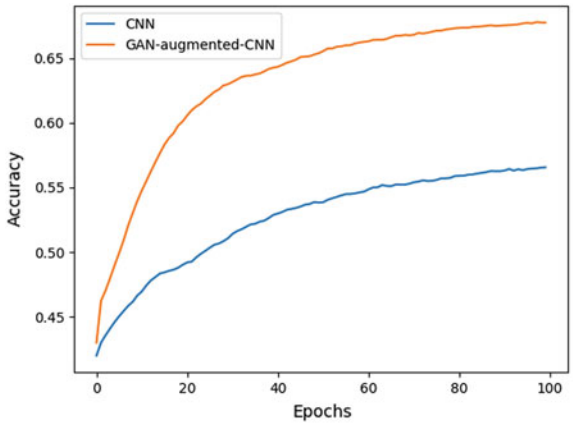
classes Atelectasis, Infiltration and NF, respectively, to avoid the overfitting as the data size of three clashes are not same. Figure 5 is an example of image generated by the generator for each of the classes.

4 Experiments and Results

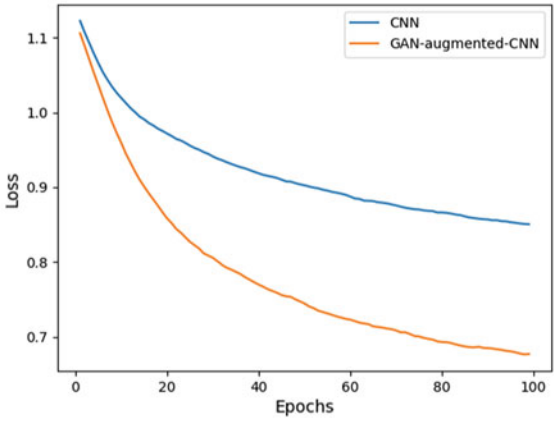
We employed the CNN architecture described in Sect. 2.1, used for training the classifier. Once the generator model of the DCGAN network was trained, we used those models to perform real-time augmentation of the dataset by replacing a part of the data with generated images of the respective classes. We performed this using mini-batches of size 32. Then we used the exact same training procedure as that of CNN training was followed. We compare the performances of the CNN on original dataset and on our online-augmented dataset. The details of online data augmentation algorithm are given in Algorithm 1.

Algorithm 1 Online Data Augmentation algorithm

```
Initialize keep_prob, batch_size;
n ← Calculate number of mini-batches.
for each mini-batch in n do
    prob_vec ← probability of each image in the mini-batch
    for each image i in mini-batch do
        if prob_vec[i] ≤ keep_prob then
            pass
        else
            y ← class of i^th entry.
             Generator generates image of class y.
             Generated image is normalized to the range 0 to 1.
             Replace original image by generated image.
        end if
    end for
end for
```



Training accuracy



Training loss

Fig. 6 Comparison between CNN and GAN-augmented-CNN

As we can see from the training graph shown in Fig. 6, the GAN-augmented CNN model is doing significantly better at classifying chest X-ray images. We are using the total classification accuracy as a metric in evaluating model performance. Accuracy is defined as

$$accuracy = \frac{\sum TP}{\text{Total number of chest X-rays}} \tag{2}$$

We compare the test set accuracies of the final trained network of the two models (CNN and GAN-augmented-CNN). The confusion matrices of both the model are shown in Fig. 7. The accuracy of the CNN model is reported as 60.3% while the accuracy of the GAN-augmented-CNN model is **65.3%**.

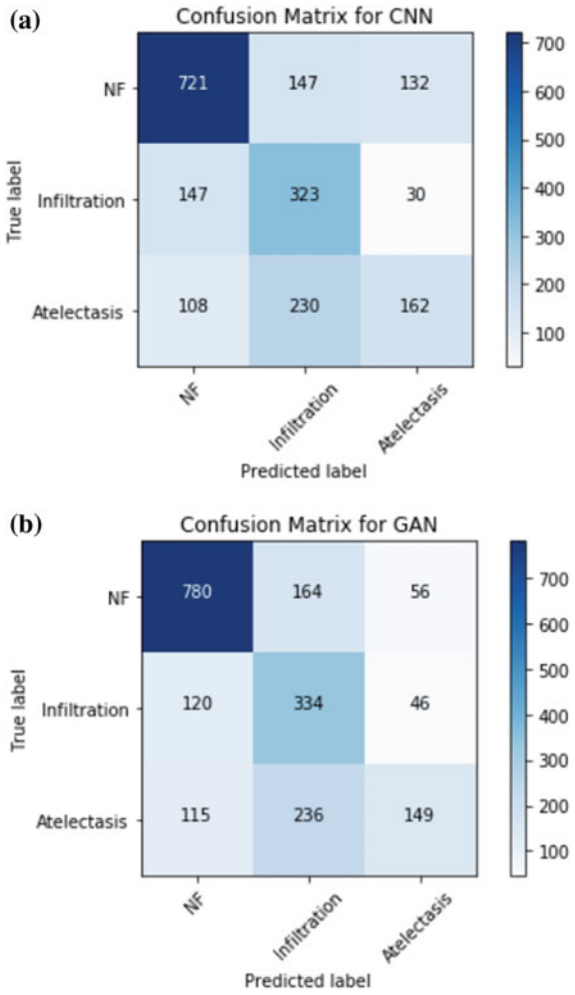


Fig. 7 Confusion matrices of **a** CNN and **b** GAN-augmented-CNN

We repeated the same experiment with three different models. The model with the best results has been shown (model-3). We used Stochastic Gradient Descent (SGD) [13] optimization for one model and had the same architecture as in Section 2.1, but with more filters in each convolution block (32, 64, 128 and 256, respectively). For the other model, we did not use class-weights update during model fitting, i.e. we did not penalize the misclassifications of the lower class data, but the remaining architecture is same as that of Sect. 2.1. We saw that for the SGD optimization, the result obtained was worse than the Adam optimization. As for the model, without using class-weights, the CNN model accuracy was on par with our model CNN performance, but it was mainly overfitting to the classes with more data (majority classes), i.e. No Finding and Infiltration. It failed to classify Atelectasis class which

Table 1 Comparison of models

Model	CNN (%)	GAN-augmented-CNN (%)
Model-1 (SGD)	56.50	58.45
Model-2 (Adam and without class-weights)	60.15	64.35
Model-3 (Adam and class-weights)	60.30	65.30

had low data (minority class). However, the GAN-augmented-CNN model did well to generalize the classifications even without the class-weights parameter. This shows that our method of augmenting data online with generated images reduced overfitting to a large degree and helps in generalizing better results across classes.

Table 1 shows the test set accuracies of the three different models. The consider model (model-3) performed best among all the three models, whereas all the three models with GAN-augmented performed better than plain CNN.

5 Conclusions and Future Work

In this chapter, we have implemented a DCGAN architecture that can be used to create synthetic chest X-ray images for three classes (Infiltration, Atelectasis and No Findings). We proposed a novel schema for using online GAN model effectively. This approach of online augmentation using GAN model-based generated images helped in achieving improved accuracy on the test dataset. This has been demonstrated using three different models; particularly the model-3 has produced reasonably good results for the disease classification. This method shows substantial improvement in results for all the three models in this low and imbalanced data regime. The future work will be concentrated on improving the performance by changing GAN architecture using Conditional GAN (C-GAN) [17] and Wasserstein GAN (W-GAN) [2]. Recently C-GAN and W-GAN studies show that they converge better to Nash equilibrium [16]. Hence, the generated images produced by those architectures when employed in our schema should produce even better results. One of the basic problem of X-ray image analysis for disease classification consist of multi-label classification and associated with different diseases. Our objective will be to extend our schema to such multi-label classification problems. Another reason as to why the results of the Atelectasis class did not come out very well is because it is a condition related to many other diseases. The future work will concentrate on identifying the Atelectasis condition properly with modified GAN architectures.

Acknowledgements The authors (Debangshu Bhattacharya and Shubham Bhattacharya) would like to thank Prof. Dinabandhu Bhandari of Heritage Institute of Technology for his valuable suggestions and guidance. One of the authors, S. Banerjee, acknowledges financial support from the Visvesvaraya PhD Scheme for Electronics and IT by Ministry of Electronics & Information Technology (MeitY), Government of India.

References

1. Ali-Gombe A, Eyad E (2019) MFC-GAN: class-imbalanced dataset classification using multiple fake class generative adversarial network. Neurocomputing. https://doi.org/10.1016/j.neucom.2019.06.043
2. Arjovsky M, Chintala S, Bottou L (2017) Wasserstein GAN. arXiv:1701.07875
3. Banerjee S, Mitra S, Sharma A, Shankar BU (2018) A CADe system for gliomas in brain MRI using convolutional neural networks. arXiv:1806.07589
4. Ben-Cohen A, Klang E, Raskin SP, Soffer S, Ben-Haim S, Konen E, Amitai MM, Greenspan H (2019) Cross-modality synthesis from CT to PET using FCN and GAN networks for improved automated lesion detection. Eng Appl Artif Intell 78:186–194
5. Dahl GE, Sainath TN, Hinton GE (2013) Improving deep neural networks for LVCSR using rectified linear units and dropout. In: 2013 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, pp 8609–8613
6. 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. CoRR arXiv:abs/1803.01229
7. Gao X, Deng F, Yue X (2019) Data augmentation in fault diagnosis based on the Wasserstein generative adversarial network with gradient penalty. Neurocomputing. https://doi.org/10.1016/j.neucom.2018.10.109
8. Goodfellow I, Bengio Y, Courville A, Bengio Y (2016) Deep learning. MIT Press, Cambridge
9. Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y (2014) Generative adversarial nets. In: Ghahramani Z, Welling M, Cortes C, Lawrence ND, Weinberger KQ (eds) Advances in neural information processing systems, vol 27, pp 2672–2680
10. He K, Zhang X, Ren S, Sun J (2015) Delving deep into rectifiers: surpassing human-level performance on imagenet classification. In: Proceedings of the IEEE international conference on computer vision, pp 1026–1034
11. Ioffe S, Szegedy C (2015) Batch normalization: accelerating deep network training by reducing internal covariate shift. arXiv:1502.03167
12. Kahng M, Thorat N, Chau DHP, Viégas FB, Wattenberg M (2018) GAN-Lab: understanding complex deep generative models using interactive visual experimentation. IEEE Trans Vis Comput Graph 25(1):310–320
13. Kingma DP, Ba J (2014) Adam: a method for stochastic optimization. arXiv:1412.6980
14. Li J, He H, Li L, Chen G (2019) A novel generative model with bounded-GAN for reliability classification of gear safety. IEEE Trans Ind Electron 66(11):8772–8781
15. Mao X, Wang S, Zheng L, Huang Q (2018) Semantic invariant cross-domain image generation with generative adversarial networks. Neurocomputing 293:55–63
16. Mescheder L (2018) On the convergence properties of GAN training. arXiv:1801.04406
17. Mirza M, Osindero S (2014) Conditional generative adversarial nets. arXiv:1411.1784
18. Oh JH, Hong JY, Baek JG (2019) Oversampling method using outlier detectable generative adversarial network. Expert Syst Appl 133:1–8
19. Pratt H, Coenen F, Broadbent DM, Harding SP, Zheng Y (2016) Convolutional neural networks for diabetic retinopathy. Procedia Comput Sci 90

20. Radford A, Metz L, Chintala S (2015) Unsupervised representation learning with deep convolutional generative adversarial networks. CoRR arXiv:abs/1511.06434
21. Ronneberger O, Fischer P, Brox T (2015) U-net: convolutional networks for biomedical image segmentation. In: International conference on medical image computing and computer-assisted intervention. Springer, pp 234–241
22. Salehinejad H, Valaee S, Dowdell T, Colak E, Barfett J (2017) Generalization of deep neural networks for chest pathology classification in X-rays using generative adversarial networks. CoRR arXiv:abs/1712.01636
23. Shen D, Wu G, Suk HI (2017) Deep learning in medical image analysis. Annu Rev Biomed Eng 19:221–248
24. Wang D, Vinson R, Holmes M, Seibel G, Bechar A, Nof S, Tao Y (2019) Early detection of tomato spotted wilt virus by hyperspectral imaging and outlier removal auxiliary classifier generative adversarial nets (OR-AC-GAN). Sci Rep 9(1):14, Article no. 4377. https://doi.org/10.1038/s41598-019-40066-y
25. Wang X, Peng Y, Lu L, Lu Z, Bagheri M, Summers RM (2017) ChestX-ray8: hospital-scale chest X-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In: 2017 IEEE conference on computer vision and pattern recognition (CVPR). IEEE, pp 3462–3471
26. Yang W, Hui C, Chen Z, Xue JH, Liao Q (2019) FV-GAN: finger vein representation using generative adversarial networks. IEEE Trans Inf Forensics Secur 14(9):2512–2524
27. Ying X, Guo H, Ma K, Wu J, Weng Z, Zheng Y (2019) X2CT-GAN: reconstructing ct from biplanar X-rays with generative adversarial networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 10,619–10,628
28. Yu X, Qu Y, Hong M (2018) Underwater-GAN: underwater image restoration via conditional generative adversarial network. In: International conference on pattern recognition. Springer, pp 66–75
29. Zhang M, Zheng Y (2019) Hair-GAN: recovering 3D hair structure from a single image using generative adversarial networks. Vis Inform. https://doi.org/10.1016/j.visinf.2019.06.001
30. Zhou D, Zheng L, Xu J, He J (2019) Misc-GAN: a multi-scale generative model for graphs. Front Big Data 2:10. https://doi.org/10.3389/fdata.2019.00003