Image Anomaly Detection using GAN – AnoGAN and GANomaly

Jihye Ye
Department of Statistics, Ewha Womans University

Abstract

Image anomaly detection differs from the general image classification task in that the data of anomaly cases are very insufficient compared to normal images. In this study, the structures of AnoGAN and GANomaly, which are models that use GAN for image outlier detection, are compared and applied to two data. Data for detecting clouded images among satellite images and manufacturing defects data were used. After applying each model to the two images, the performance is compared, and the results of model learning are interpreted by visualizing how the image was processed. As a result, GANomaly was superior to AnoGAN, but this may differ depending on each data characteristics and model settings.

1 Introduction

In computer vision, the field of image classification has already developed beyond human capabilities. Most of them are supervised learning methods, so sufficient train data for all classes must be given to the models to classify test data. Therefore, if there are not enough data for any class, it cannot be learned effectively. Many cases in the field of anomaly detection belong to here, and it is also difficult to collect enough data because anomaly cases are not observed frequently. For example, there is not enough data to catch weapons of airplane terrorists because it is very rare. Also, it is difficult to gather data on rare diseases that very few people have in the world have. However, image anomaly detection is a very important issue, as failure to detect this rare case will cause great damage. In addition, image anomaly detection is used in the manufacturing field to check the quality of products, catch criminals or detect fallen people through security CCTVs.

Recently, many studies have been conducted to solve the data imbalance problem of image anomaly detection. Largely, there are methods of reproducing fake images like normal images using GAN, one class classification method that allows normal features to be gathered at one point, and feature matching method that learns the probability distribution of features. Among them, we would like to understand the model structure of GAN-based reproducing methods, AnoGAN and GANomaly, which were first studied, to find out how GAN was applied to image anomaly detection, and to apply it to new data.

2 Structure of Generative Adversarial Networks (GAN)

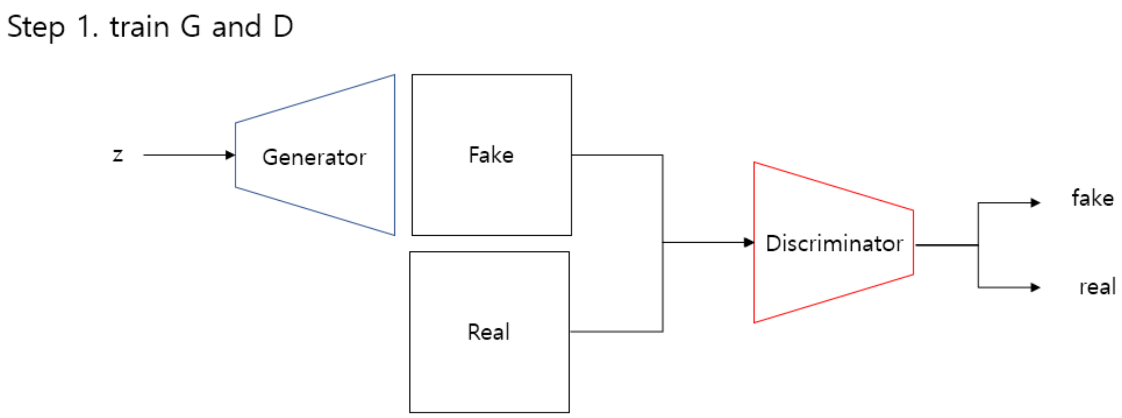
GAN was first proposed by Goodfellow et al. [1] to create realistic images. It is a structure that improves the performance of the generator by adversarial learning the generator that generates images and the discriminator that classifies real and fake images. The generator G maps 1d vector z, the noise generated in the latent space, to 2d images x located in the real image space and learns the distribution of the real image, $p_{data}(x)$. Discriminator D is a classifier that maps 2d image as a single scalar value learning whether the image is real or fake. D and G are repeatedly learned from each other and optimize the following objective functions.

$$\min_{G} \max_{D} V(D, G) = E_{x \sim p_{data}(x)}[logD(x)] + E_{x \sim p_z(z)}[\log\left(1 - D\left(G(z)\right)\right)]$$

DCGAN applies CNN structures to GAN mainly used for image generation, and Radford et al. suggests which element is best used to apply CNNs to GANs. [2] It uses convolution layers and batch-normalization instead of GAN's fully connected layers. ReLU is used as the activation function of the generator, but LeakyReLU is recommended as the discriminator. In addition, when down sampling in the generator, stride is used instead of maxpooling. Both AnoGAN and GANomaly used in this study are based on DCGAN.

3 Structure of AnoGAN

AnoGAN was proposed by Thomas Schlegl in 2017, [3] the first model to use GAN for anomaly detection. Trained with only normal images, Generators are based on the idea that they will not be able to produce abnormal images well, and it is difficult to properly place abnormal images in the latent space of normal images. Therefore, after training the basic structure of GAN with only normal images, it maps test images to the learned latent space, and generate images with the generator, defining how poor their performance about abnormal images is as loss.



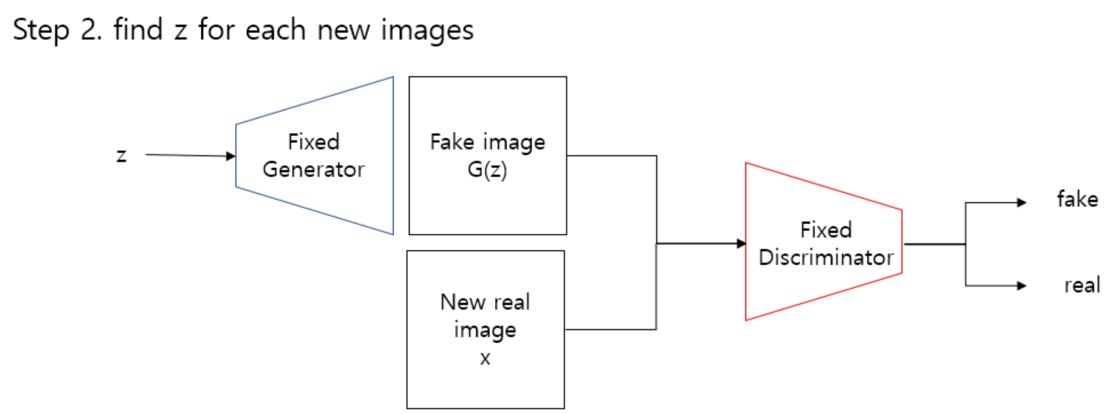


Figure 1 Structure of AnoGAN. This figure is a model learning sequence, learning G and D in Step 1, and z in Step 2.

AnoGAN can be divided into two stages. In the first step, a model with the same structure as DCGAN is trained only with normal images. All structures and loss functions of the model are the same as DCGAN, only using normal data. In the second step, new test images are mapped to the trained latent space. It is a process of finding the most suitable latent variable z for the new images. In other words, after fixing all the parameters of G and D of the DCGAN learned earlier, only z generated randomly is updated. For this purpose, two losses are defined.

$$\text{residual loss : } L_R(z_\gamma) = \sum |x - G(z_\gamma)|$$

$$\text{discrimination loss : } L_D(z_\gamma) = \sum |f(x) - f(G(z_\gamma))|$$

Residual loss evaluates how visually similar the new image and the generated image. For this loss to be small, the new image must first be well mapped into the latent space to find the appropriate z_γ, and an image must be created similar to the original image through G. In the case of an abnormal image, it is difficult to find z_γ, and the image generated through this also has characteristics of a normal image, so the loss is inevitably increased.

Discrimination loss uses feature matching idea for the purpose of mapping new images well into latent space. This is to extract feature representation of x and $G(z_\gamma)$ properly by utilizing the previously trained discriminator as a feature extractor.

$$L(z_\gamma) = (1 - \lambda) \cdot L_R(z_\gamma) + \lambda \cdot L_D(z_\gamma)$$

$$A(x) = (1 - \lambda) \cdot R(x) + \lambda \cdot D(x)$$

The weighted sum of these losses is defined as the final loss of the second stage, and each time a new image is entered, a latent variable z corresponding to that image is learned, and this loss value in the last stage is utilized as an anomaly score. In addition, we can understand which part was judged as abnormal through residual image $x_R = |x - G(x_\Gamma)|$ which means the difference between the generated image and the actual image.

4 Structure of GANomaly

GANomaly, was proposed by Samet Akkay [4] in 2018, based on the model structure of AnoGAN, but doesn't need to learn latent variable for each image. Instead starting with random generation of latent variable z, this model starts with the process of compressing the actual image. So after learning only the normal image as input and fixing the model, new test images simply pass the model and calculate anomaly score.

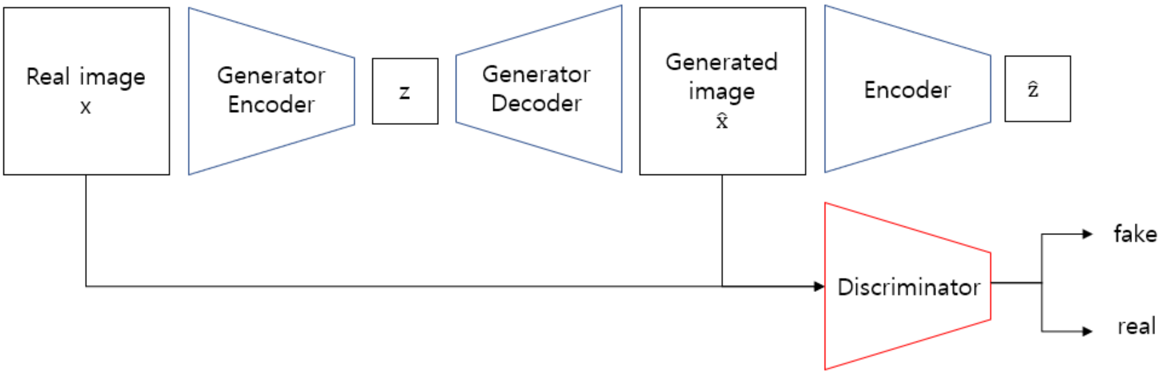


Figure 2 Structure of GANomaly. At training stage, three encoders and one decoder are learned, and for the new image, anomaly score is calculated through the already trained model.

GANomaly can be largely divided into three subnetworks. They include a generator that generates an image similar to an input, an encoder that compresses it again, and a discriminator that classifies real and fake images. Generators consist of encoders and decoders again, so they will be referred to as G_E and G_D. Overall, it consists of three encoder structures and one decoder structure.

First, the generator aims to compress the input image x into the z of the best representation 1d vector, thereby generating $\hat{x} = G_D(z) = G(x)$ similar to the original image. It is a section for learning contextual information of normal images. Therefore, the loss to be minimized in this section is defined as contextual loss, and the equation is as follows.

$$\text{contextual loss}: L_{con} = E_{x \sim p_x} \|x - G(x)\|_1$$

Second, Encoder compresses the generated image back to $\hat{z}$ again. It is the same as the structure of the G_E, but has different parameters, through which it learns the feature representation of the normal images. That is, the generator learns the reproduction of a normal image, and encoder learns the feature compression of the normal image. Therefore, the loss in this process is defined as encoder loss and is as follows.

$$\text{encoder loss}: L_{enc} = E_{x \sim p_x} \|G_E(x) - E(G(x))\|_2$$

To put it simply, it minimizes the distance between z and $\hat{z}$. This loss is used as an anomaly score after learning because the anomaly images are difficult to compress into a proper z. Also the generated image $\hat{x}$ through the encoder doesn't have the anomaly information because encoder has only context information of the normal image. Therefore, $\hat{z}$, which is compressed from $\hat{x}$ again, inevitably differs greatly from z.

The last discriminator is the same as the discriminator in AnoGAN, and the loss function is the same. It is for adversarial learning and extracts the features of each image through a function f corresponding to the output of the intermediate layer of D. The loss function is as follows. The weighted sum of the three losses is defined as the loss of the entire model.

$$\text{adversarial loss : } L_{adv} = E_{x \sim p_x} \left\| f(x) - E_{x \sim p_x} f(G(x)) \right\|_2$$

$$L = w_{adv} L_{adv} + w_{con} L_{con} + w_{enc} L_{enc}$$

GANomaly has a pretty complex structure, but in fact it has only added encoder to AnoGAN. To replace the process of learning the latent variable z of each image, it added a process of compressing the input image to z from the time we learn the normal image. Also, apply the process to the reproduced image to utilize the difference as an anomaly score. To intuitively understand anomaly score, min-max scaling is performed using the score of the entire test data to have a value between 0 and 1.

$$\mathrm{A}(\hat{x}) = \|G_E(\hat{x}) - E(G(\hat{x}))\|_1$$

$$s_i' = \frac{s_i - \min(S)}{\max(S) - \min(S)} \; when \; S = \{s_i : A(\hat{x_i}), \hat{x_i} \in test\ set\ \widehat{D}\}$$

5 Experiments

5.1 Data

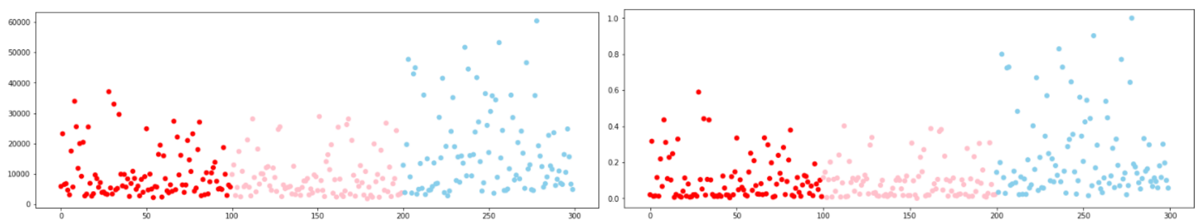
Two image data used in this study requires anomaly detection in reality. They can be downloaded from kaggle. The code can be found at https://github.com/jihye0115/2022-Anomaly-Detection-using-GAN.

Cloud and non-cloud data [5] was taken by satellite and cannot properly photograph the ground if it is clouded, so cloud data is abnormal. A total of 1,400 non-cloud data were used as train data, and 100 cloud data and 100 non-cloud data not used for training were used as test data. The image size was attempted from 256 to 64, and the performance at 64 was rather good, so it will be introduced at the results later.

Casting data [6] is mould image data used in metal processes and the metal liquid is poured into it to solidify, so the defect in the mould leads to casting defects. Since most of them look similar, fine anomalies must be captured. 2,875 normal images were used as train data, and 100 anomaly images and 100 normal images not used for training were used as test data. As for the image size, 256 to 64 were attempted, but in the case of size 256, it was not possible to learn until the end with insufficient RAM. Therefore, size 128 was selected, and if the number of channels is designated as 3, the loss of the abnormal image tends to be small, so the number of channels was set to 1. Since the color of the image is almost gray, it is rather stable to learn.

5.2 Results

Cloud and non-cloud data



(a) AnoGAN (b) GANomaly

Figure 3 The result of applying cloud data to (a) AnoGAN and (b) GANomaly. Red is the train normal image, pink is the test normal image, and sky blue is the test anomaly image.

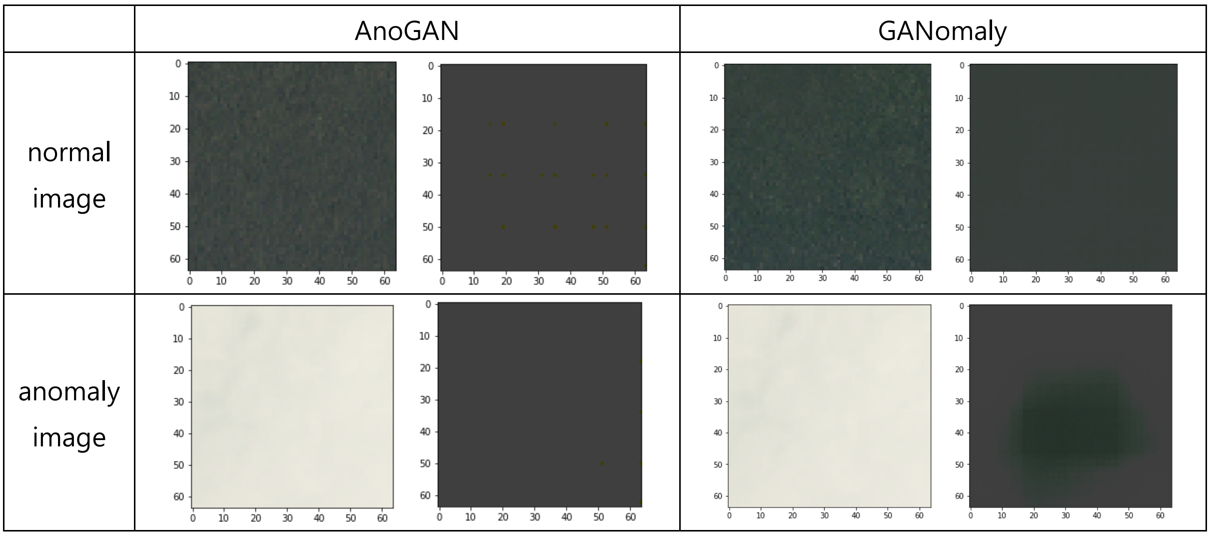
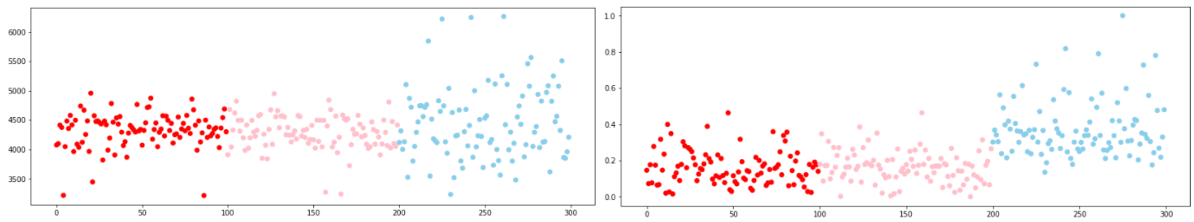


Figure 4 Within each compartment, the left is the actual image and the right is the reproduced image through model.

Although the results of the two models seem similar, GANomaly's results (b) are more concentrated in small values except for some anomaly images. By the reproduced image from the normal image and the anomaly image on each model, you cannot see a certain shape. Rather, a monochromatic image of gray was produced. Therefore, the more clouds there are, the whiter the image is, so the loss seems to have been large. However, in the case of normal images with rivers or bright paths, rather than dark mountains, i.e. images with a large distribution of bright colors are likely classified as anomaly by detecting them as clouds. Conversely, even if there are clouds, it will be classified as a normal image if gray rain clouds appear, or if the image of the dark ground only appears blurry due to light fog. Due to the characteristic of the data, it can be interpreted that colors are simply selected as a criterion for judging anomalies rather than creating images close to reality because many images of dark ground are learned from the train data.

Casting data



(a) AnoGAN (b) GANomaly

Figure 5 The result of applying casting data to (a) AnoGAN and (b) GANomaly. Red is the train normal image, pink is the test normal image, and sky blue is the test anomaly image.

It can be seen at a glance that GANomaly is much better than AnoGAN in Casting data. In AnoGAN, the losses of anomaly images are very spread, and there are many data with a lower loss than the normal image, but in GANomaly, the loss is clearly large, and if the threshold is set well, it can have an accuracy of more than 80%.

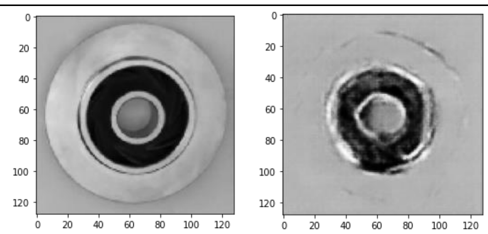
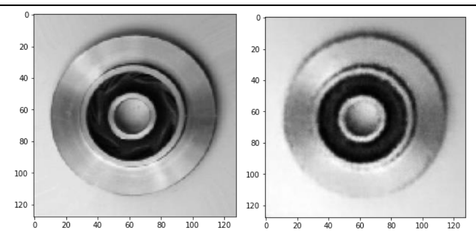
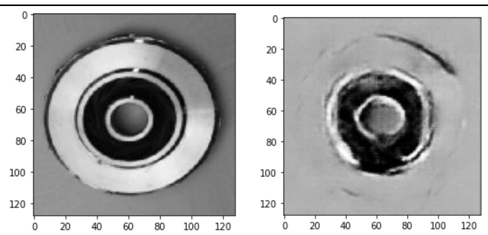
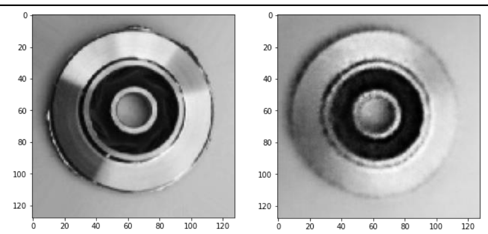
	AnoGAN	GANomaly
normal image		
anomaly image		

Figure 6 Within each compartment, the left is the actual image and the right is the reproduced image through

model.

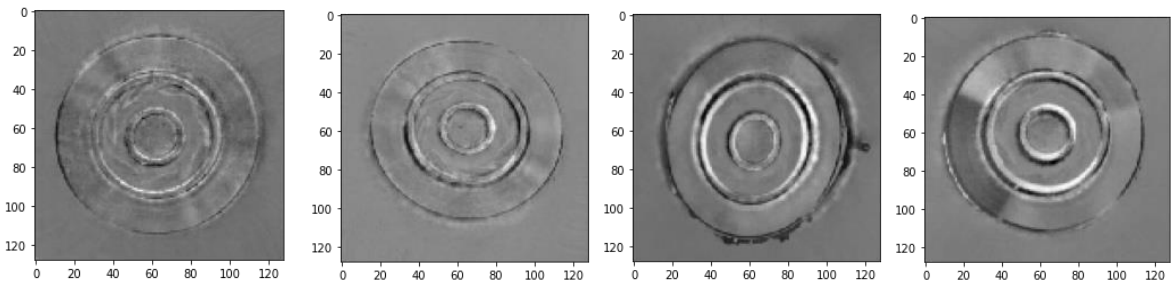


Figure 7 Residual image. The two left images are residual images of normal images, and the two right images are residual images of anomaly images.

The reproduced image shows the reasons, first of all, the reproduced image of GANomaly is much similar to the actual image compared to AnoGAN. It produces images that are so close to reality that it is difficult to distinguish with the naked eye. Looking at the residual image, the normal image has only a fine part of the outline, but the outlier image has a relatively clear outline, and uneven parts remain to make the loss larger.

6 Conclusion

I understood the structure of the two models, AnoGAN and GANomaly, and investigated how anomaly images could be detected using GAN. Based on this, an experiment was conducted on two data by modifying the referenced codes, and the performance was very different depending on the data characteristics, but in general, GANomaly performed a little better than AnoGAN. These models performed well in previous studies to learn one class with a specific form, and then classify the class with a different form as anomaly, such as MNIST or CIFAR data. But the cloud data used in this study had to learn cloudless images as normal images. Most of the casting data have similar circular shapes when viewed visually, and some fine defects must be identified, which could be detected by GANomaly by producing close-to-real images well.

Several experiments were conducted on image size, generator structure, number of channels, and weight of loss, but rich experiments could not be conducted due to GPU and RAM usage limitations. If you compare this experiment with sufficient resources and time, you will find a model with better performance even in the same model. Although the two models had limitations in learning these data, detecting anomaly in cloud and casting data is a practical topic used in real-world industries, so comparing performance by applying recent models that fit the characteristics of this data will also be a good topic for future research.

References

[1] 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. (2014)

[2] Radford, A., Metz, L., Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv:1511.06434 (2015)

[3] Schlegl, T., Seebock, P., Waldstein, S.M., Schmidt-Erfurth, U., Langs, G.: Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. arXiv:1703.05921 (2017).

[4] Samet Akcay, Amir Atapour-Abarghouei, Toby P. Breckon: GANomaly: Semi-Supervised Anomaly Detection via Adversarial Training. arXiv:1805.06725 (2018)

[5] Cloud and Non-Cloud Images (Anomaly Detection). https://www.kaggle.com/datasets/ashoksrinivas/cloud-anomaly-detection-images

[6] Casting product image data for quality inspection. https://www.kaggle.com/datasets/ravirajsinh45/real-life-industrial-dataset-of-casting-product

AnoGAN 코드 참고 : https://github.com/soomin9106/Deep-Learning/blob/main/AnoGAN/DCGAN_AnoGAN.ipynb

GANomaly 코드 참고 : https://github.com/leafinity/keras_ganomaly/blob/master/ganomaly.ipynb