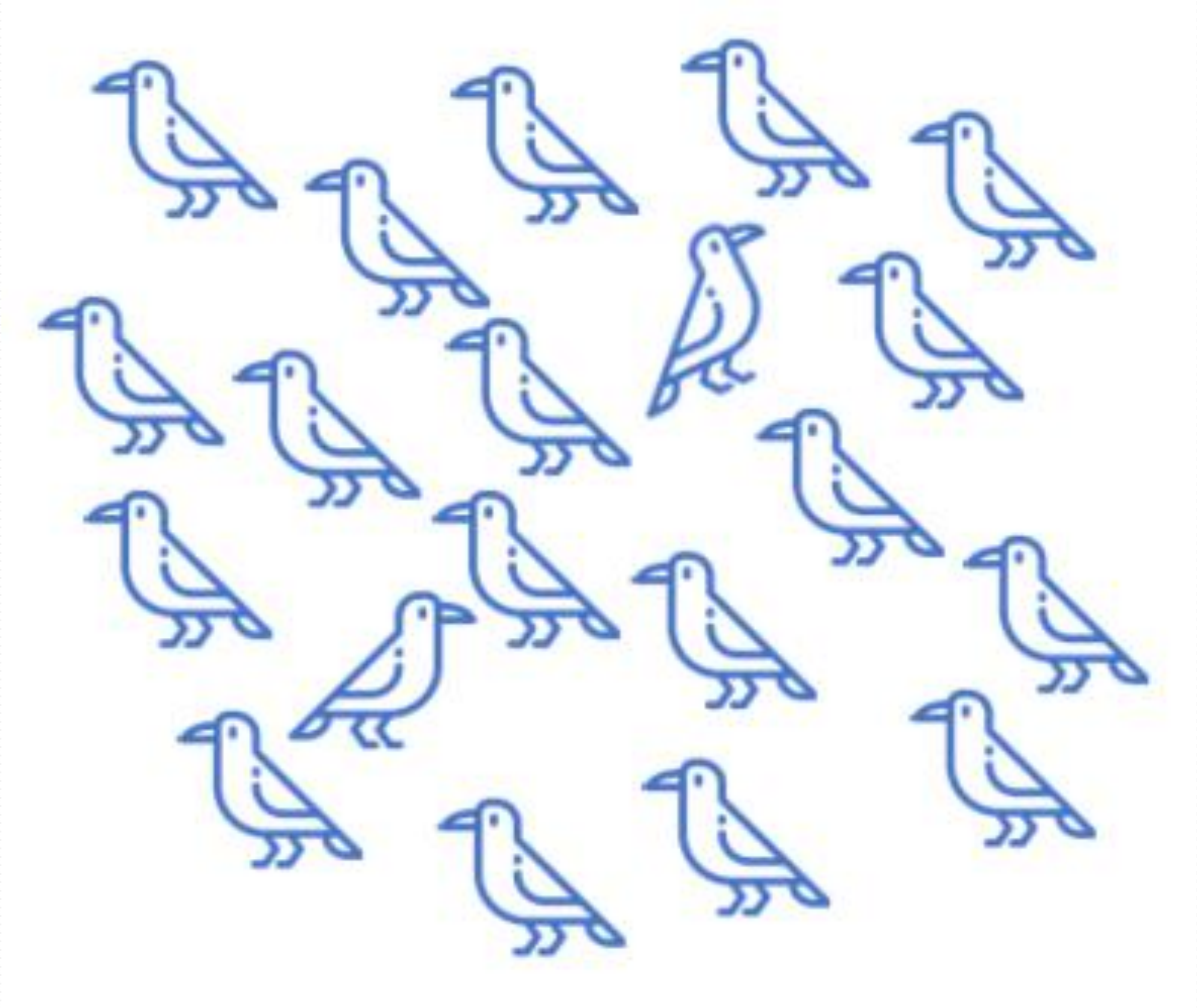


GAN for Anomaly Detection

Kae-Jy Chou

2021, 05, 04

01 Introduction to Anomaly Detection

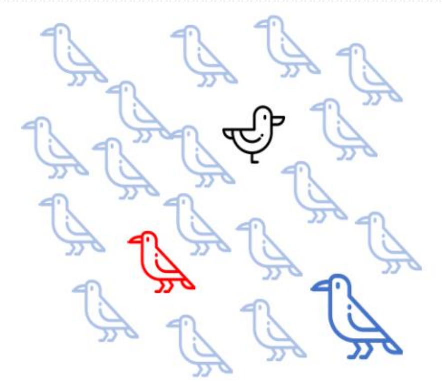
02 AutoEncoder for Anomaly Detection

03 F-AnoGAN for Anomaly Detection

04 Exercise

What is Anomaly Detection

- *Anomaly* detection (or *outlier* detection or *novelty* detection) is the identification of data (an observation) that does not fit to the distribution of normal data, i.e. does not conform to normal appearance, semantic content, quality, or expected behavior.



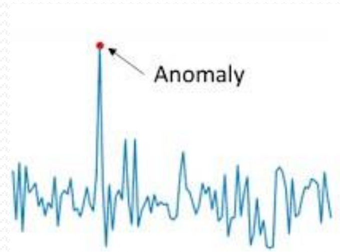
Basic Assumption:
normal data instance, in some sense, belongs to **large and dense clusters**, while anomalies belongs to **small or spare clusters**

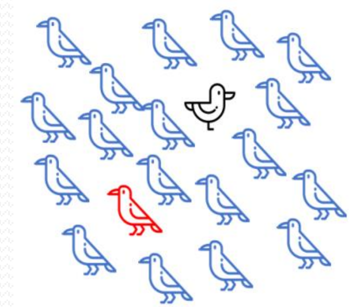
Anomalous data may be useless (caused by measurement error) which will be discarded, or extremely informative which is the objective of most anomaly detection.

Anomaly Type

Point Anomaly

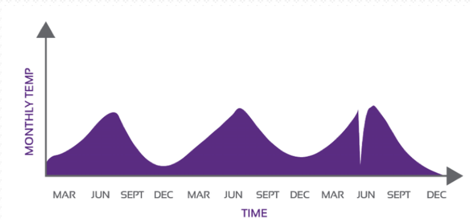
An individual data instance is considered as anomalous with respect to the rest of data

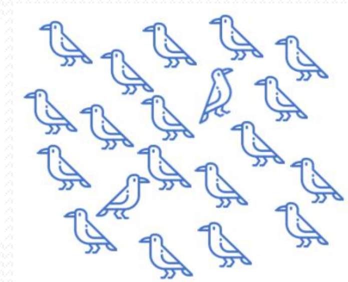




Contextual Anomaly

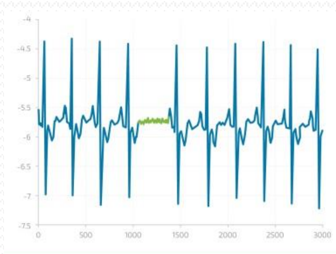
a data instance that is anomalous in a specific context such as time, space.

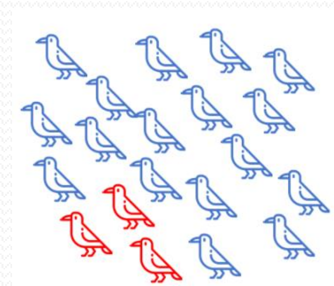




Group Anomaly

a subset of data instances anomalous as a whole w.r.t. the other data instances.





Data Type

Structured data

Data format can be expressed as rows and columns that each row indicates an individual data and columns represent the features of each data.

Features

Data

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S

- ✓ System log
- ✓ Sensory data
- ✓ Time-series
- ✓ CSV
- ✓ Excel file
- ✓ DataBase

General approach: Use traditional machine learning method, such decision trees, clustering algorithm, support vector machine,..

Unstructured data

Data format comes in its diversity of forms and can not be expressed as rows and columns

- ✓ Audio
- ✓ Language
- ✓ Text
- ✓ Image
- ✓ Video
- ✓ 3-D object

General approach: Use deep learning method or hybrid methods which incorporates traditional machine learning

Characteristic of Anomalous Data

Rare

- In real-world applications, anomaly rarely happens, especially in a reliable system or process.

Data imbalance $\frac{Anomalous\ data}{Normal\ data} < 1\%$

Unknown

- Since most anomalies are unknown beforehand, we can not use supervised learning methods.

Supervised Learning

Variability

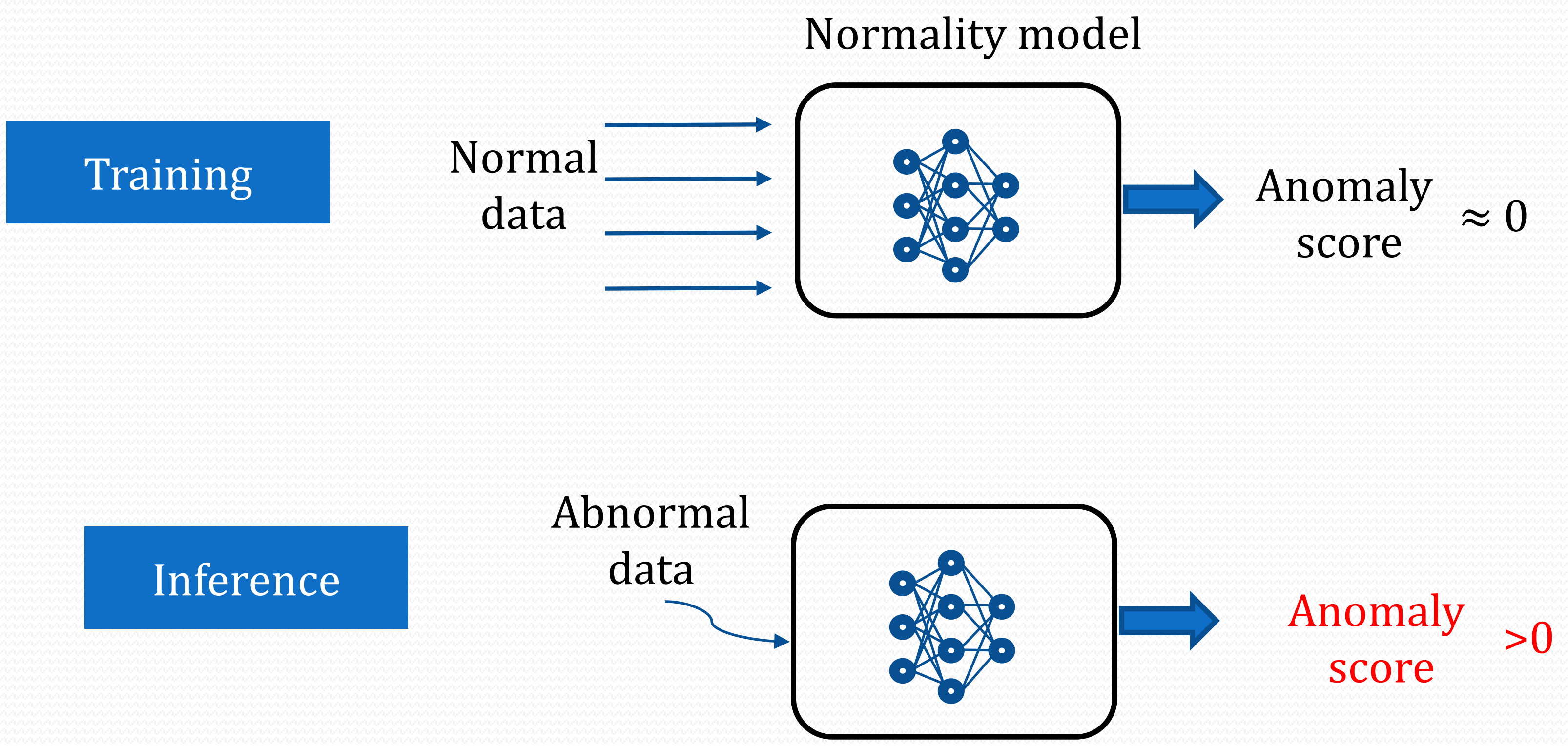
- Variability within normal data can be very large
- Anomalies can be diverse too

In anomaly detection for structured data, preprocessing, normalization, feature selection, noise reduction are employed to reduce the variability and improve detectability.

General Approaches-1

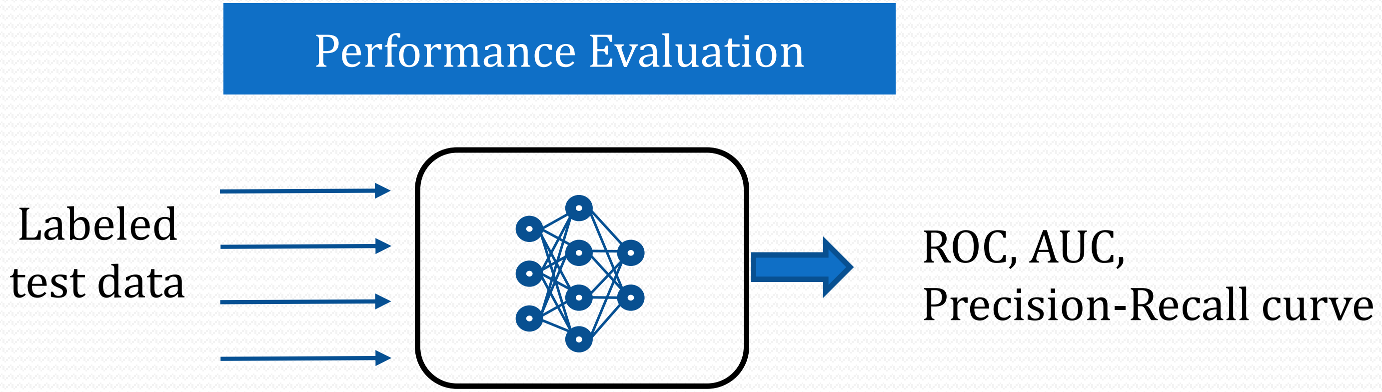
- What should we do if we don’t know unknown anomaly beforehand?

A central idea in anomaly detection is to learn a model of normality from normal data in an unsupervised manner, so that anomalies become detectable through deviations from the model (anomaly score > 0).



General Approaches– About Data

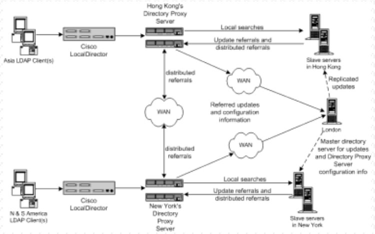
- Even though it is called unsupervised learning, we still need expert to label data to normal data and anomalous data, which is much easier than annotating the data.
- In unsupervised learning, we still need some labeled data for the performance evaluation. The labeled data is not for the training. They are used for the inference or test stage that we can measure the model performance..
- If a small number of labeled anomalous data with a limited variation is available during training, they can be used as a “validation” set to fine-tune the model’s hyperparameters.



Applications

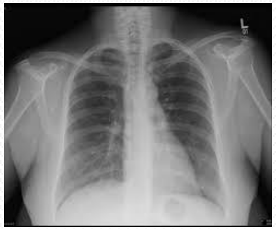
Structured data

- Intrusion in networking & cyberspace
- Fraud detection in finance, insurance, heath care
- Monitoring of infrastructure (IaSS, PaSS, SaSS)
- Industrial fault and damage detection
- …



Unstructured data

- Acoustic novelty detection
- Medical diagnosis
- Industrial inspection
- Science discovery in chemistry, genetics, physics, astronomy
- …



01 Introduction to Anomaly Detection

02 AutoEncoder for Anomaly Detection

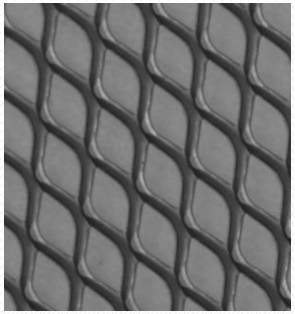
03 F-AnoGAN for Anomaly Detection

04 Exercise

Anomalous Image Type

Image Level

(semantics anomaly)



Classification: anomaly appear in different object class.

Identification of anomalous image.

Pixel Level

(sensory anomaly)

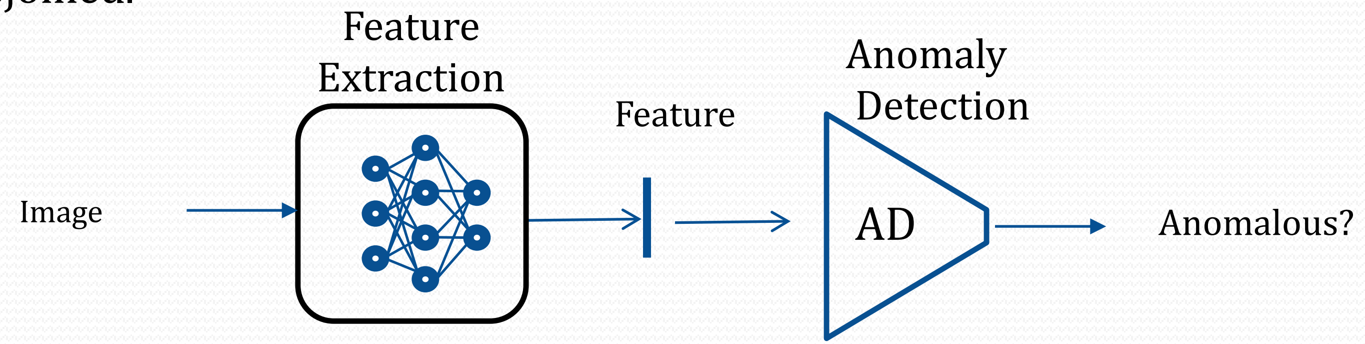
Segmentation: anomaly appear in subtle deviation of the image.

Localization of anomalies on the pixel level

In computer vision, one needs to distinguish between two variants of this task. First, there is the classification scenario, where novel samples appear as entirely different object classes that should be predicted as outliers. Second, there is a scenario where anomalies manifest themselves in subtle deviations from otherwise known structures and a segmentation of these deviations is desired.

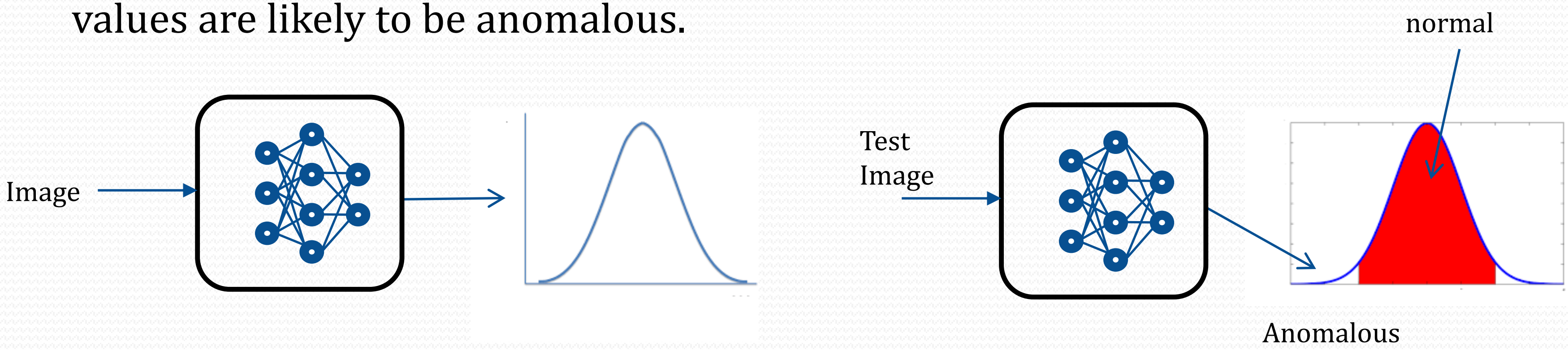
Three Main Types of Method

1) **Feature-Extracted based** (Distance based) method: Map image to features space and detect anomalies based on distance. The mapping can be pre-trained models or a designed model for the purpose. Feature extraction and anomaly detection are disjoined.



2) **Probability based method**: assume anomalies occur in low probability region of the normal images.

2.1 Establish the probability density function (PDF) of normal image
2.2 Evaluate the test images by PDF and those with low probability density values are likely to be anomalous.



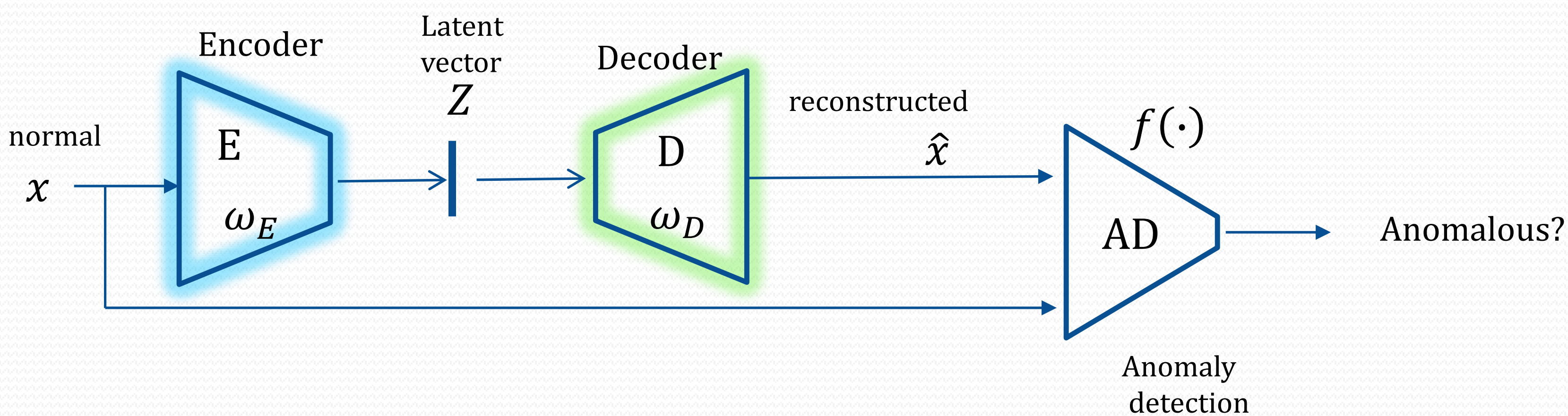
Reconstruction-Based Anomaly Detection

3) **Reconstruction-based** methods learn a model that is optimized to well-reconstruct normal data instances, thereby aiming to detect anomalies by failing to accurately reconstruct them under the learned model.

Encoder and Decoder can be trained separately(f-AnoGAN) or jointly trained (AE).

1. $dim(Z) \ll dim(x)$
2. $f(\hat{x}) \cong f(x)$; $f\big(D(E(x))\big) \cong f(x)$

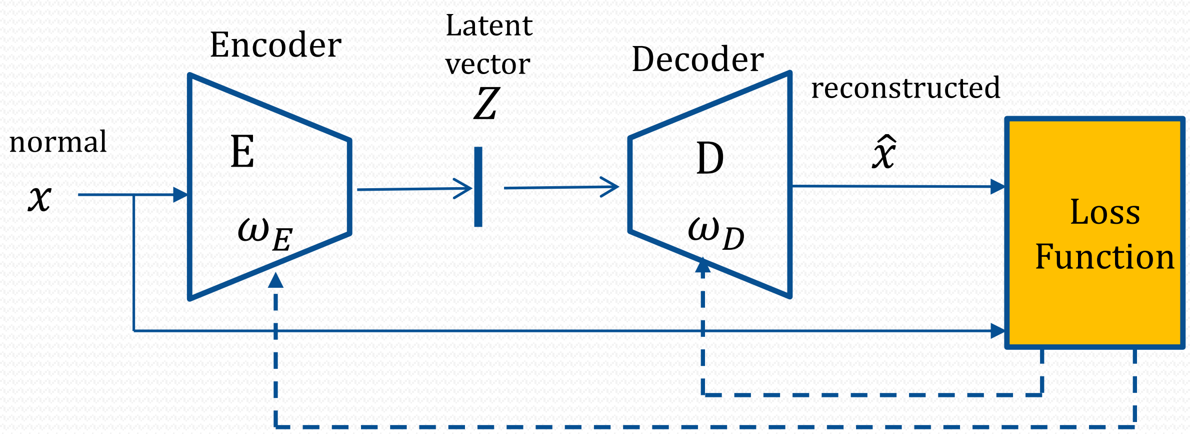
When $f(\cdot)$ is an identity function, $\hat{x} \cong x$; $D\big(E(x)\big) \approx x$



Unsupervised Learning

AutoEncoder in AD

- These models try to reconstruct their inputs in the presence of certain constraints such as a bottleneck and thereby manage to capture the essence of high-dimensional data (e.g., images) in a lower dimensional space.
- It is assumed that anomalies in the test image deviate from the training data manifold and the model is unable to reproduce them. As a result, large reconstruction errors indicate defects.



Loss Function in AutoEncoder

- L1-distance
- L2-distance
- SSIM (Structural Similarity)
- Perceptual loss
- Other image quality metrics

Drawback of l^p-Distance

- l^p-distance (i.e. l^1 or l^2)loss functions compare single pixel value and make unrealistic assumption that neighboring pixel values are mutually independent
- It yields high residuals in locations where the reconstruction is only slightly inaccurate, e.g., due to small localization imprecisions of edges (misalignment)
- It fails to detect structural differences between the input and reconstructed images when the respective pixels' color values are roughly consistent.

$$L_1(x,\hat{x}) = \sum_{r=0}^{h-1}\sum_{c=0}^{w-1} \|x(r,c) - \hat{x}(r,c)\|^1$$

$$L_2(x,\hat{x}) = \sum_{r=0}^{h-1}\sum_{c=0}^{w-1} \|x(r,c) - \hat{x}(r,c)\|^2$$

h,*w* denotes the number of height and width of the input image

Structural Similarity (SSIM)

- The SSIM index defines a distance measure between two K*K image patches $\boldsymbol{p}$ and $\boldsymbol{q}$, taking into account their similarity in luminance $l(\boldsymbol{p},\boldsymbol{q})$, contrast $c(\boldsymbol{p},\boldsymbol{q})$, and structure $s(\boldsymbol{p},\boldsymbol{q})$.

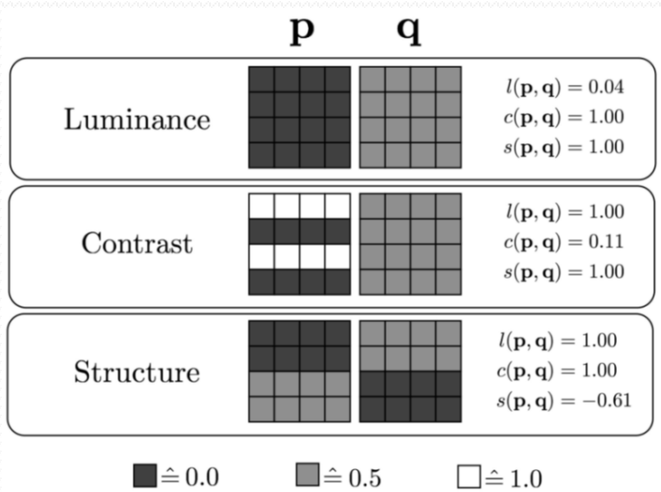
- It computes three different statistical features for image comparison and operates on local patch regions

$$SSIM(p,q) = l(p,q)^{\alpha} c(p,q)^{\beta} s(p,q)^{\gamma}$$

$$l(\boldsymbol{p},\boldsymbol{q}) = \frac{2\mu_p\mu_q + c_1}{\mu_p^2 + \mu_q^2 + c_1}$$

$$c(\boldsymbol{p},\boldsymbol{q}) = \frac{2\sigma_p\sigma_q + c_2}{\sigma_p^2 + \sigma_q^2 + c_2}$$

$$s(\boldsymbol{p},\boldsymbol{q}) = \frac{2\sigma_{pq} + c_2}{2\sigma_p\sigma_q + c_2}$$



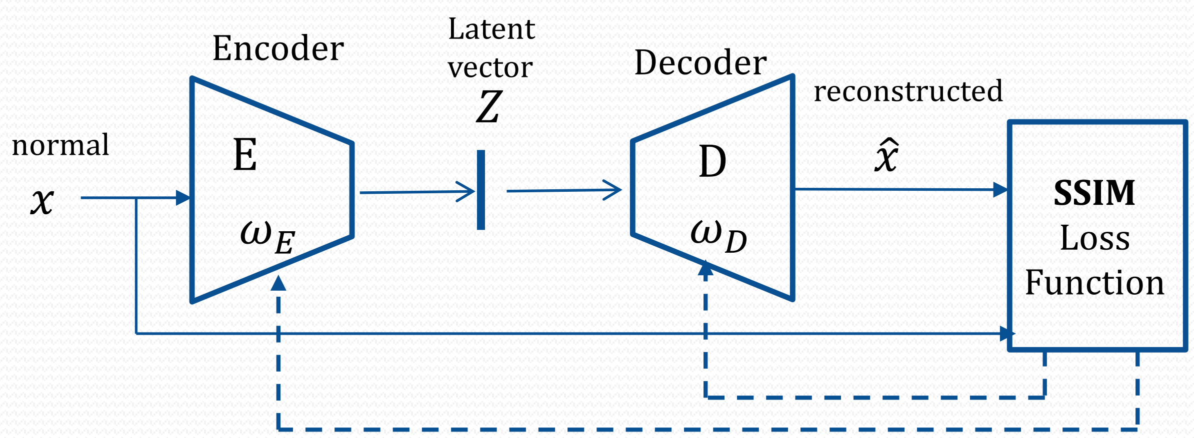
1) $SSIM(p,q) = SSIM(q,p)$
2) $SSIM(\boldsymbol{p},\boldsymbol{q}) \leq 1$
3) $SSIM(p,q) = 1$ if and only if $p == q$

✓ SSIM is less sensitive to edge alignment and gives importance to salient differences between two images.

Image quality assessment: from error visibility to structural similarity

SSIM Autoencoder

- SSIM is employed as loss function and evaluation metric to obtain residual map.
- Dimension of latent vector is 100 and increases to 500 for highly complex image.
- Encoder and Decoder use CNN network.
- Strided convolutions are used to down-sample the input feature maps in the encoder and to up-sample them in the decoder

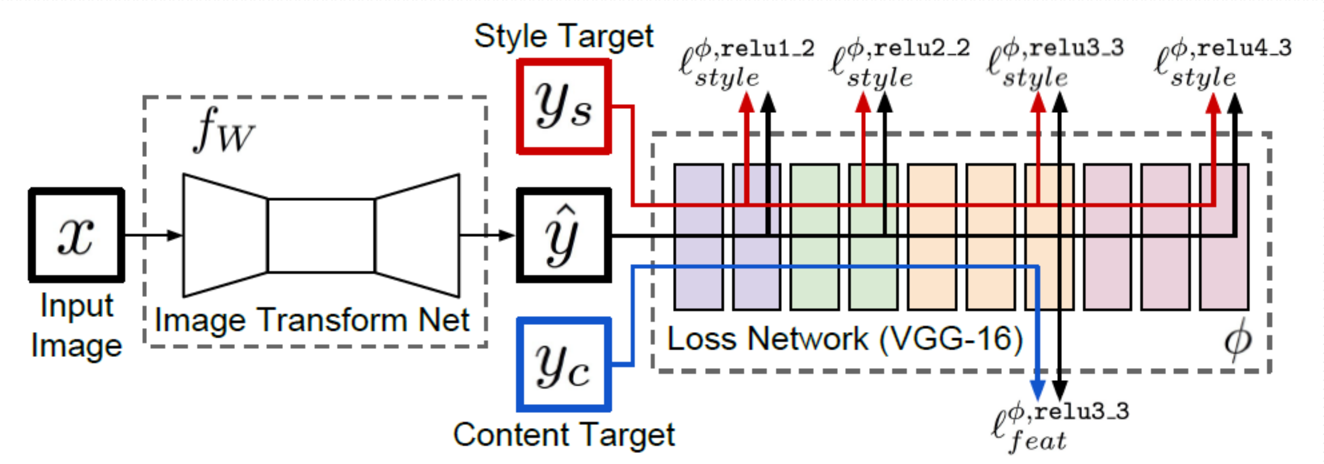


$$L_{SSIM} = 1 - SSIM$$

Improving Unsupervised Defect Segmentation by Applying Structural Similarity to Autoencoders

Perceptual Loss

- Perceptual loss functions is based on high-level features extracted from pretrained networks.
- Original designed for Style transfer and super-resolution
- Use loss network to define several loss functions.
- The loss function are pre-trained and fixed during training. For example, 16-layer VGG network pretrained on the ImageNet dataset.



Assumption: the loss network has learned to encode the perceptual and semantic information.

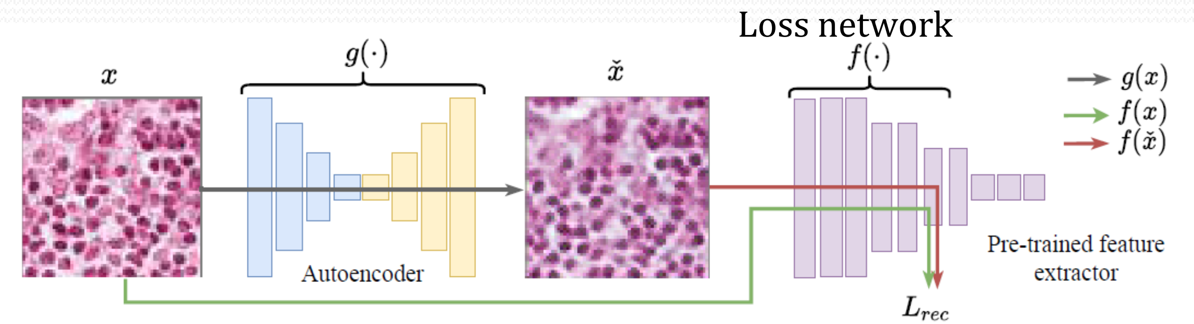
Perceptual Losses for Real-Time Style Transfer and Super-Resolution by Justin Johnson, Alexandre Alahi, Li Fei-Fei

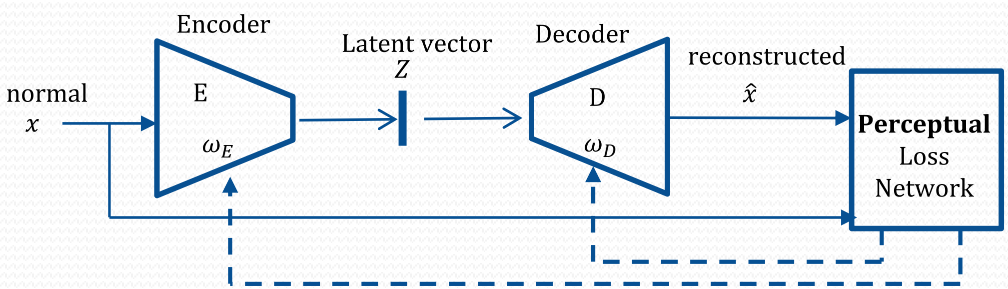
Deep Perceptual Autoencoder-1

- The main idea is not to force the network to reconstruct a realistic looking image, but to let it be flexible in understanding the content of the normal data.
- Use a loss function that measures ”pattern”-dissimilarity of the input and the output

$$L_{pl}(x,\hat{x}) = \frac{\left\|\tilde{f}(x) - \tilde{f}(\hat{x})\right\|^1}{\left\|\tilde{f}(x)\right\|^1} \qquad \tilde{f}(x) = \frac{f(x) - \mu}{\sigma}$$

The mean μ and standard deviation σ are calculated on the loss network with ImageNet dataset.

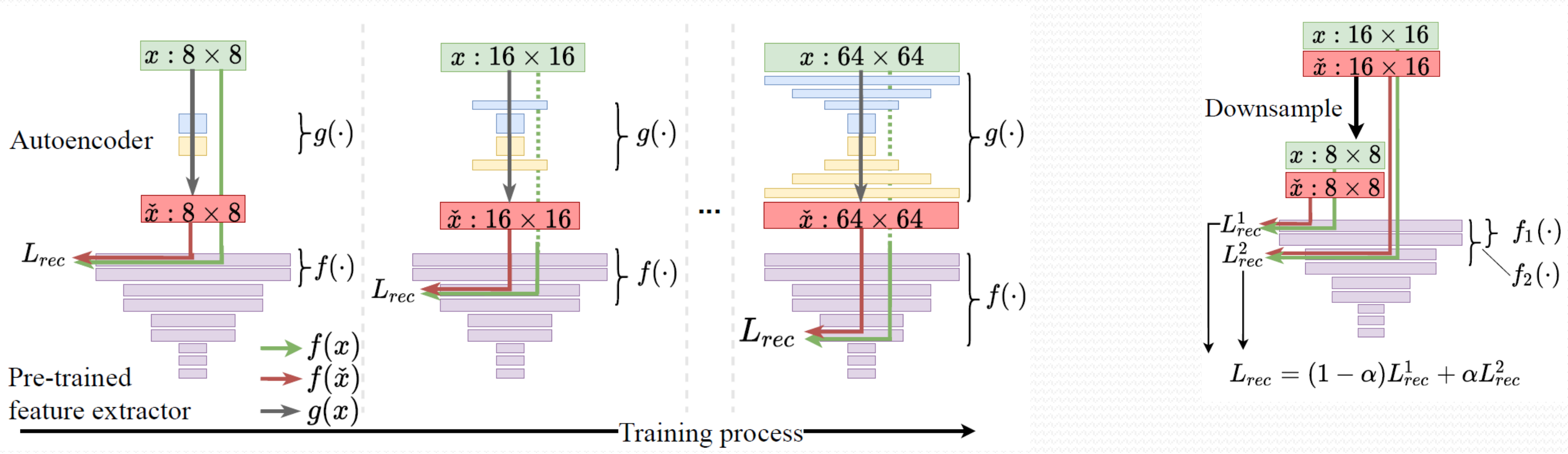




Anomaly Detection with Deep Perceptual Autoencoders

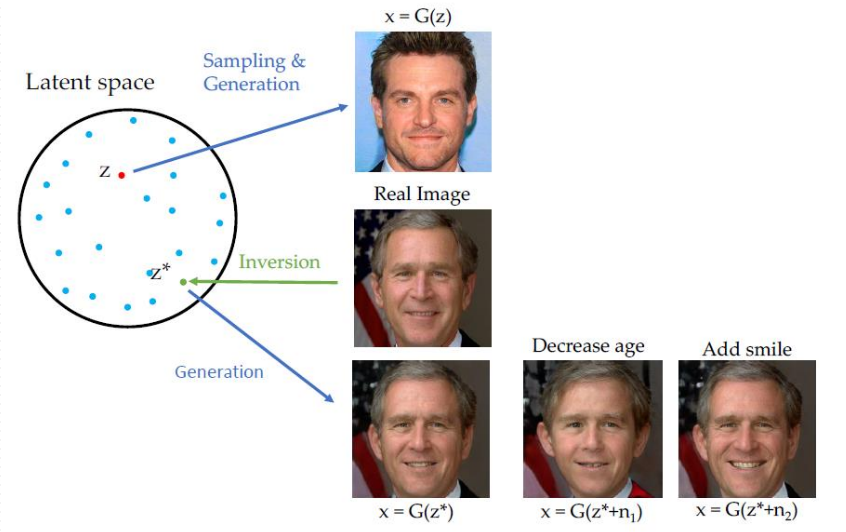
Deep Perceptual Autoencoder-2

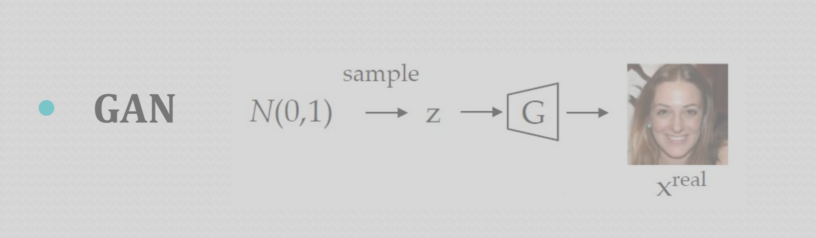
- The authors use progressive training to further improve the performance.
- Progressive Training: to synchronize addition of the new layers to the autoencoder with the gradual increase of the depth of the features entailed in the calculation of the perceptual loss
- Both the autoencoder g and the perceptual loss L_{rec} have a low "resolution" in the beginning. As the training advances, the layers are incrementally added to the autoencoder g, and the depth of the features f is increased



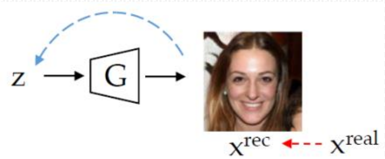
GAN Inversion-1

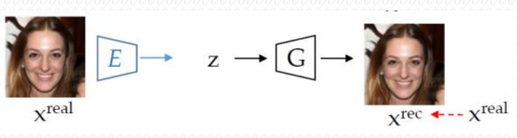
- GAN inversion aims to invert a given image back into the latent space of a pretrained GAN model, for the image to be faithfully reconstructed from the inverted code by the generator

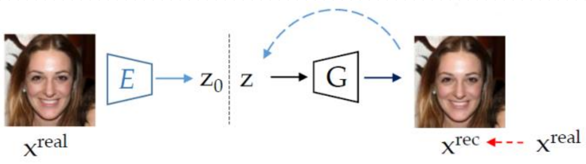




- **Optimized based**: typically reconstruct a target image by optimizing over the latent vector
- **Learned based**: Learning-based GAN inversion involves training an encoding neural network E(x; E) to map an image x to the latent code z
- **Hybrid:** an encoder network is first used to obtain an approximate embedding and then refine it with an optimization algorithm





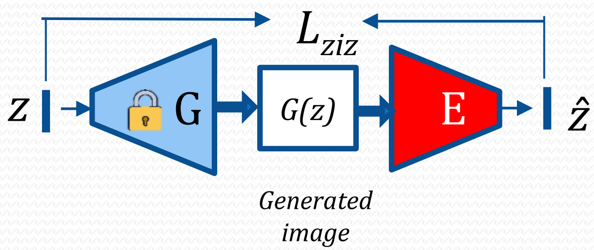


GAN Inversion-2

In the case of a normal input, mapping from image space to the latent space via the encoder and subsequent mapping from latent space back to image space via the generator should closely resemble the identity transform

- Train encoder with generated image (*ziz*)

$$L_{ziz}(x) = \frac{1}{d} \cdot \|z - G(E(z))\|^2$$

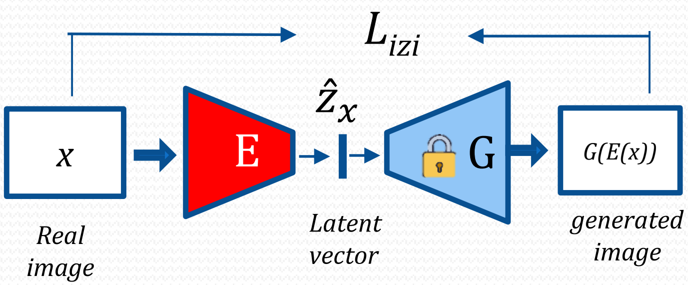


The drawback is that the encoder only "sees" generated images but never receives real input images

- Train encoder with real image (*izi*)

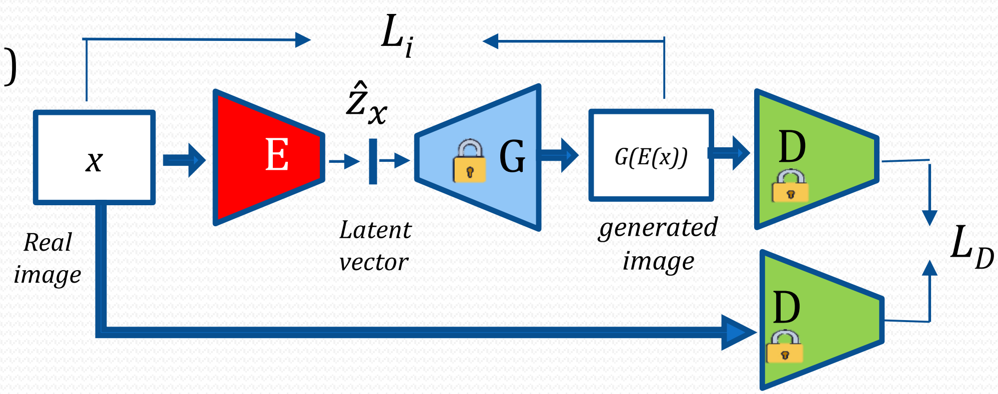
minimize the MSE residual loss of input images **x** and reconstructed images

$$L_{izi}(x) = \frac{1}{n} \cdot \|x - G(E(x))\|^2$$



- Discriminator guided *izi* encoder training (*izi$_f$*)

simultaneously guides encoder training in the image space and in the latent space



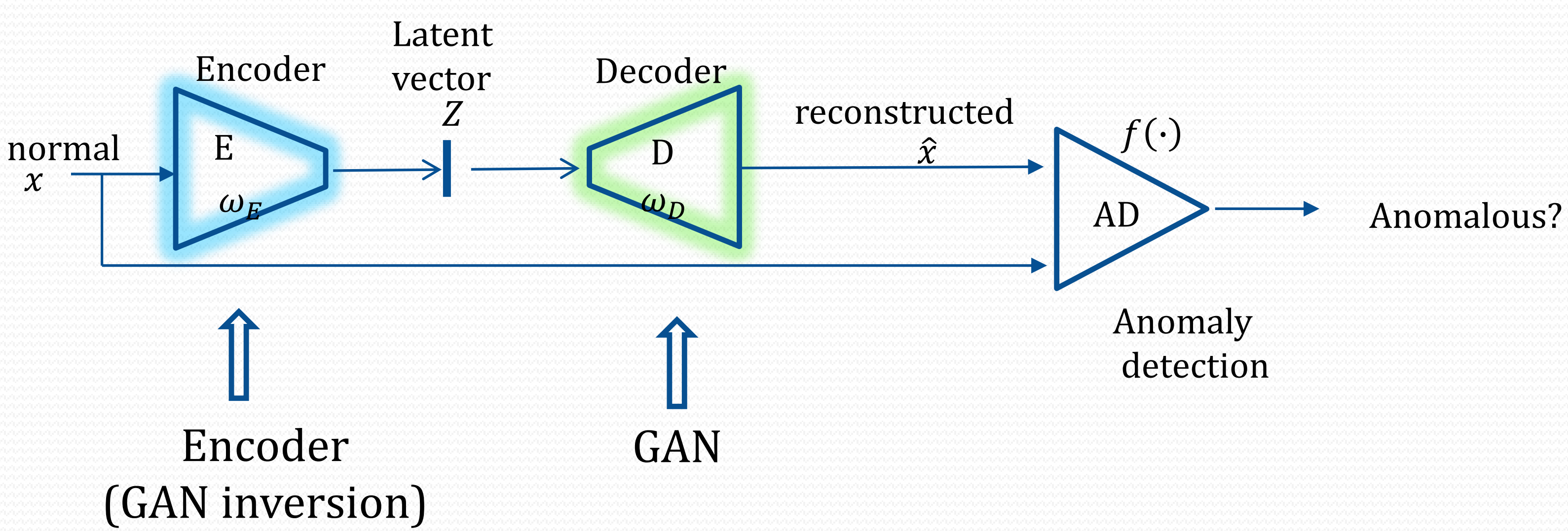
F-AnoGAN

- The generator learns the representation of the normal data

1. Select large volume of normal images.
2. Train a generative model on the normal images.
3. Do GAN inversion. Train an encoder of mapping image to latent vector
4. From encoder and generator to reconstruct the test image.
5. Compare test image and its reconstructed image to get anomaly score.

Salient feature: Use GAN inversion to find the mapping of query image to latent space

Anomaly score = feature residual error + image reconstruction error



WGAN-GP Training

Step 1

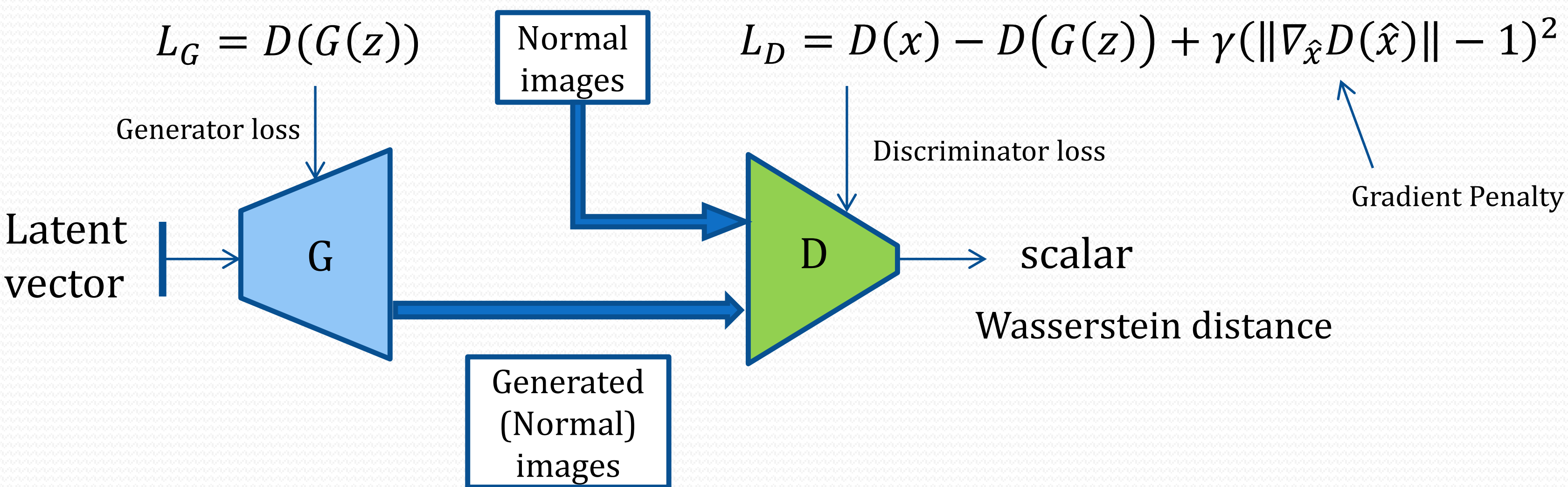
- The generator learns the representation of normal image distribution
- The discriminator outputs the Wasserstein distance between generated and real data distribution, not a measure of realness of a given image.
- Use normal image only for the WGAN-GP training
- Latent vector follows the normal distribution.
- Generator & discriminator follow the ResNet network. (In exercise, DCGAN are used instead.)

Normal image, Latent vector; $random\ \epsilon$, $0 < \epsilon < 1$

$$\hat{x} = \epsilon x + (1-\epsilon)G(z)$$

$$L_G = D(G(z))$$

$$L_D = D(x) - D\big(G(z)\big) + \gamma(\|\nabla_{\hat{x}} D(\hat{x})\| - 1)^2$$



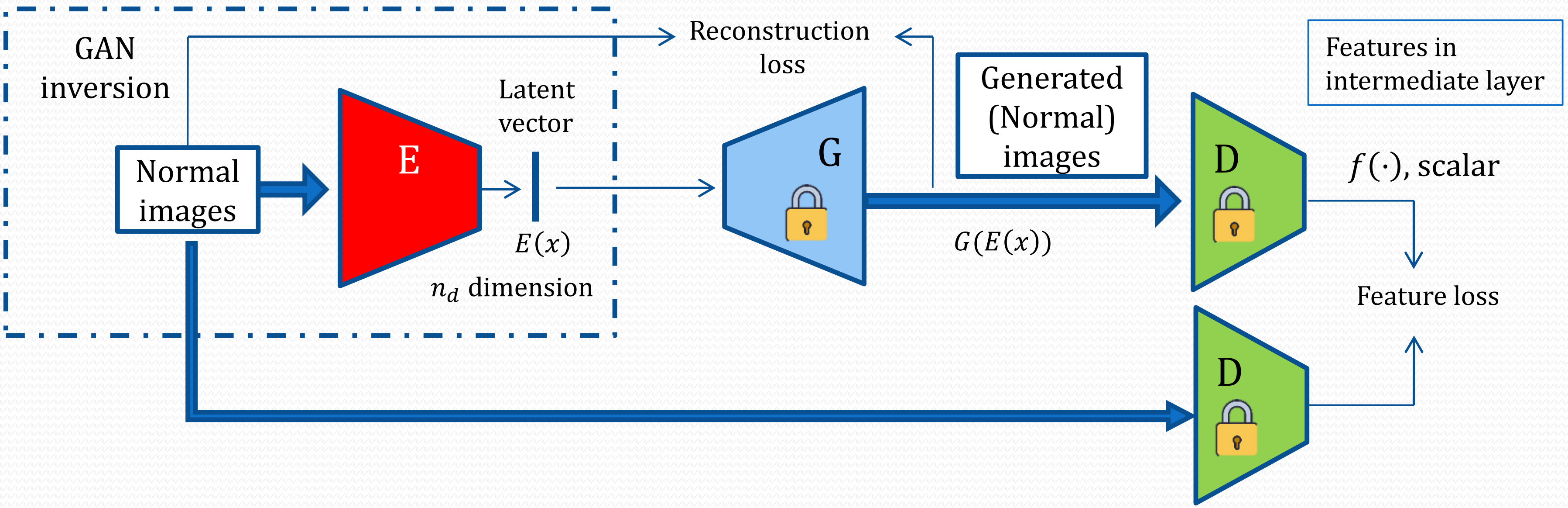
Encoder Training

Step 2

- Learn the mapping of image to a vector in latent space where the generator is trained from.
- Parameters in generator and discriminator are fixed.
- The generated image shall closely resemble to the input normal image
- The discriminator outputs the features in the intermediate layer, not Wasserstein distance nor a measure of image realness.

Encoder loss — Reconstruction loss — Feature loss

$$L_E(x) = \frac{1}{n} \cdot \|x - G(E(x))\|^2 + \frac{\kappa}{n_d} \cdot \|f(x) - f(G(E(x))\|^2$$

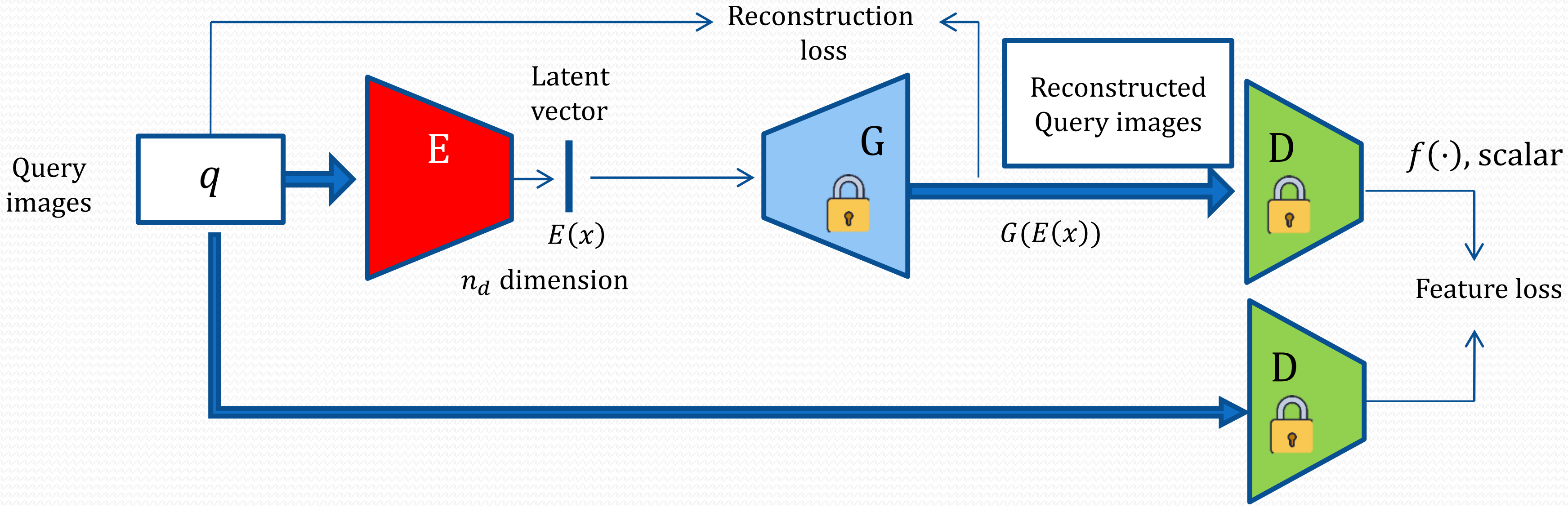


Detect Anomaly

Step 3

- Anomaly score (AS) for image-level detection is the sum of reconstruction loss and feature loss of query image.
- If the query image is a normal image, the AS shall be as smaller as possible.
- The AS of anomalous query image shall be significantly different from the AS of normal images.
- For the pixel-level anomaly localization, we use the absolute value of pixel-wise residual, $\hat{A}_R = |x - G(E(x))|$

$$\underbrace{AS(q)}_{\text{Anomaly Score}} = \underbrace{\frac{1}{n} \cdot \|x - G(E(q))\|^2}_{\text{Reconstruction loss}} + \underbrace{\frac{\kappa}{n_d} \cdot \|f(q) - f(G(E(q))\|^2}_{\text{Feature loss}}$$



MVTec Dataset

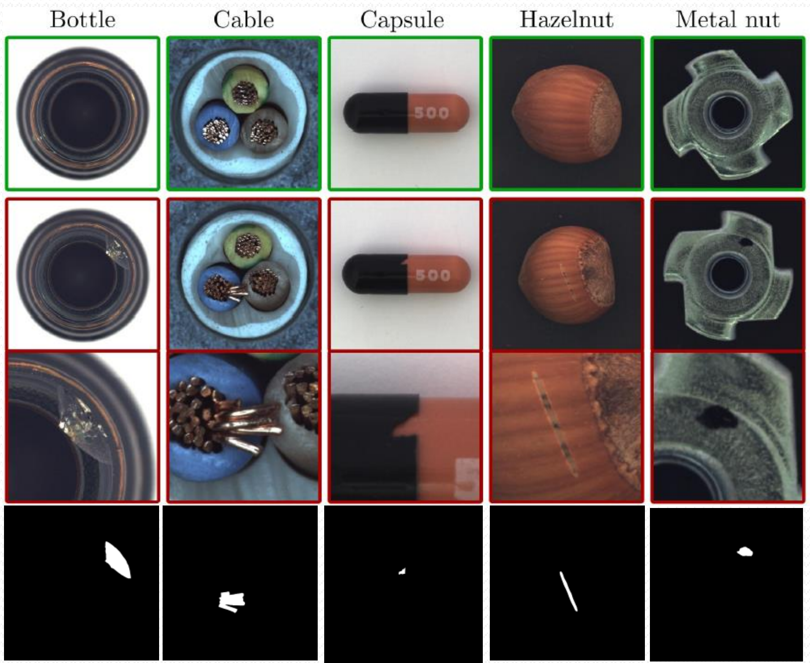
- A real-world data for unsupervised anomaly detection
- The dataset is released by MVTec Software GmbH (Munich, Germany)
- It contains 15 categories with 3,629 images for training and validation and 1,725 images for testing.
- It has normal (detect-free) images for training, and normal & anomalous images for testing
- There are 73 types of defects such as scratch, dent, distort, bent or contamination, not shown in training images.
- Pixel-level ground truth labels for defect region are provided in test images.
- The dataset is for image-level anomaly detection and pixel-level defect localization.

normal

anomalous

Close-up

ground truth labels



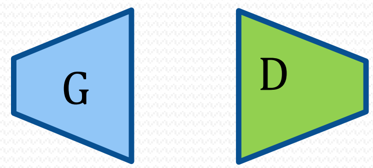
www.mvtec.com/company/research/datasets

Pytorch Code Explained

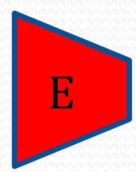
- We use https://github.com/A03ki/f-AnoGAN for the Exercise
- It is the implementation of f-AnoGAN in Pytorch

./mvtec_ad/model.py → Define Generator (G), Discriminator (D), Encoder(E)

./mvtec_ad/train_wgangp.py →



./mvtec_ad/train_encoder_izif.py →



./mvtec_ad/test_anomaly_detection.py
(Generate anomaly scores in ./result/score.csv) →

score.csv

label	img_distance	anomaly_score	z_distance
1	0.008094113	0.011250545	0.016097972
1	0.010277588	0.015948262	0.054447111
1	0.007859191	0.011685018	0.031970985
1	0.0131122	0.023521189	0.069101147
1	0.01315585	0.022864031	0.043429479
1	0.005859799	0.008997796	0.023142952
1	0.010663489	0.017291391	0.038698271
1	0.007316839	0.01005792	0.020571183
1	0.01030338	0.015590075	0.020919856
1	0.005117434	0.007185512	0.026372639
1	0.006431661	0.009215491	0.020542126

./mvtec_ad/save_compared_image.py

f-AnoGAN_MVTecAD.ipynb →

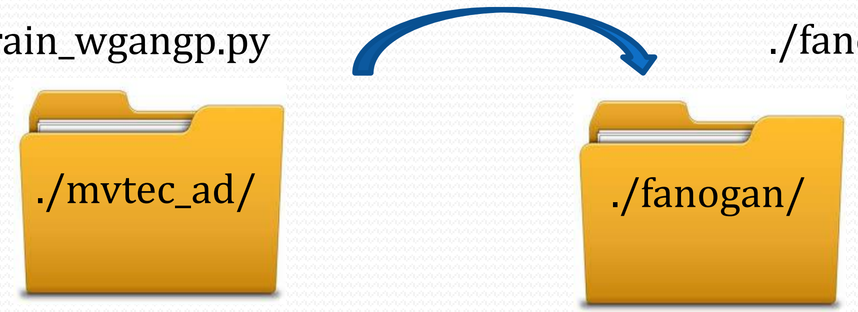
1. Training steps
2. Visualization: ROC-AUC and PR-AUC

Calling Procedure & Data Structure

- 4 driver files in ./mvtec_ad which calls the main function with the same file name in ./fanogan

./mvtec_ad/train_wgangp.py

./fanogan/train_wgangp.py



Training Procedure

1. python train_wgangp.py "bottle" --seed 1 --n_epochs 1500
2. python train_encoder_izif.py "bottle" --seed 1 --sample_interval 50 --n_epochs 200
3. python test_anomaly_detection.py "bottle"
4. python save_compared_images.py "bottle" --n_iters 0 --n_grid_lines 10

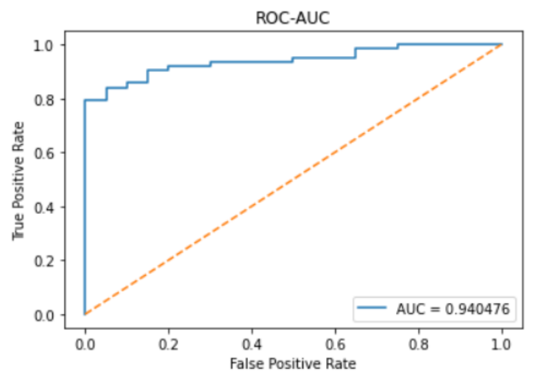
```
(base) kenny@zhoukaizhideMacBook-Air bottle % tree -dirsfirst
.
[        128]  ./train
[       6752]  ./train/good
[        224]  ./test
[        704]  ./test/good
[        736]  ./test/contamination
[        768]  ./test/broken_small
[        704]  ./test/broken_large
[        192]  ./ground_truth
[        736]  ./ground_truth/contamination
[        768]  ./ground_truth/broken_small
[        704]  ./ground_truth/broken_large

11 directories
```

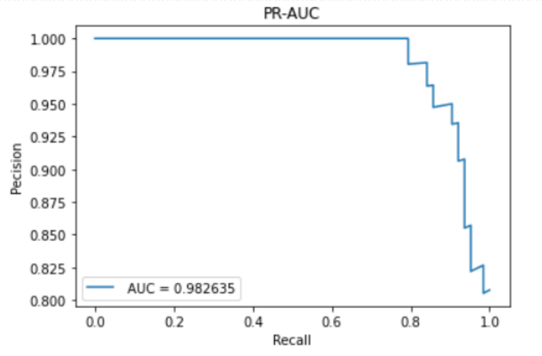
Visualization

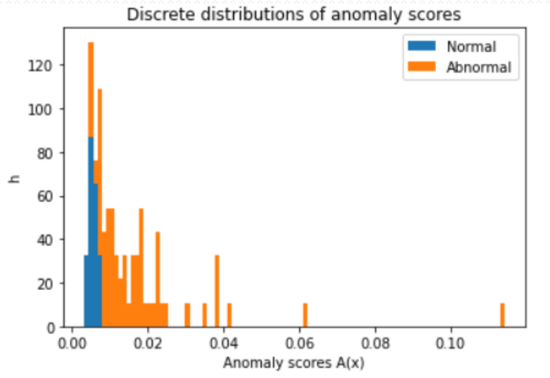
- Use *scikit-learn metrics* module to plot ROC-AUC and PR-AUC

```
from sklearn.metrics import roc_curve,  precision_recall_curve, auc
```

See the code in https://colab.research.google.com/drive/1_mIcP6k-706O6Bo70iIAoEIhYGYO4wAe?usp=sharing#scrollTo=fYgEsezABV0p







A(x) < 0.01 for normal data

Exercise

- In F-AnoGAN, the authors use MSE for the reconstruction loss.
- What if we use SSIM for the loss? Will it improve ROC-AUC or PR-ROC?
- Pick one of 15 categories in MVTec dataset for the exercise.

Hint: 1. Use pytorch-msssim   (pip install pytorch-msssim)

```
#  import SSIM
from pytorch_msssim import ssim, ms_ssim, SSIM, MS_SSIM
```

```
#  Add SSIM
    ssim_loss = SSIM(win_size=11, win_sigma=1.5, data_range=1, size_average=True, channel=3)
```

```
    # Scores for anomaly detection
    # img_distance = criterion(fake_img, real_img)

    # Replace MSE with SSIM
    img_distance = 1-ssim_loss(fake_img, real_img)
```

References

1. Paul Bergmann, Sindy Löwe, Michael Fauser, David Sattlegger, Carsten Steger. Improving Unsupervised Defect Segmentation by Applying Structural Similarity to Autoencoders In *arXiv:1807.02011v3*
2. Paul Bergmann, Michael Fauser, David Sattlegger, Carsten Stege. MVTec AD — A Comprehensive Real-World Dataset for Unsupervised Anomaly Detection *CVPR.2019.00982*
3. Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. *In European conference on computer vision, pages 694–711. Springer, 2016.*
4. Guansong Pang, Chunhua Shen, Longbing Cao, Anton van den Hengel. Deep Learning for Anomaly Detection- A Review. *arXiv:2007.02500*
5. Lukas Ruff, Jacob R. Kauffmann, Robert A. Vandermeulen, Grégoire Montavon, Wojciech Samek, Marius Kloft, Thomas G. Dietterich, Klaus-Robert Müller. A Unifying Review of Deep and Shallow Anomaly Detection In *arXiv:2009.11732*
6. Thomas Schlegl, Philipp Seeböck, Sebastian M. Waldstein, Georg Langs, Ursula Schmidt-Erfurth. f-AnoGAN: Fast unsupervised anomaly detection with generative adversarial networks In *Medical Image Analysis 54 (2019) 30–44*
7. Nina Tuluptceva, Bart Bakker, Irina Fedulova, Heinrich Schulz, and Dmitry V. Dylov. Anomaly Detection with Deep Perceptual Autoencoders In *arXiv:2006.13265v2*
8. Wang, L.; Zhang, D.; Guo, J.; Han, Y. Image Anomaly Detection Using Normal Data Only by Latent Space Resampling. *Appl. Sci.* **2020**, *10*, 8660. https://doi.org/10.3390/app10238660
9. Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing,* 13(4):600–612, 2004.
10. Weihao Xia, Yulun Zhang, Yujiu Yang, Jing-Hao Xue, Bolei Zhou, Ming-Hsuan Yang. GAN Inversion: A Survey In *arXiv:2101.05278v2*
11. f-ANoGAN Pytorch implementation https://github.com/A03ki/f-AnoGAN