POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Executive Summary of the Thesis

Generative Adversarial Networks And Real-Time Anomaly Detection In Texture Images

Laurea Magistrale in Computer Science and Engineering - Ingegneria Informatica

Author: Stefano Pecchia

Advisor: Prof. Giacomo Boracchi

Co-advisor: Dr. Diego Carrera

Academic year: 2020-2021

1. Introduction

Anomaly detection (AD) in images can be defined as the task of identifying anomalous regions inside the images themselves. This is a critical task which is very difficult to tackle using machine learning models: anomalies can appear in huge variety of different forms, and collecting a representative dataset for all the anomalies is an expensive and cumbersome task that is almost impossible to perform in real-life scenarios.

A promising approach to perform AD in images is to use Generative Adversarial Networks (GANs) [2] exploiting their ability in capturing and modeling the distribution of natural images. Indeed, GANs have been able to outperform many other models in the complex task of synthetic images generation, super-resolution and domain translation. More recently, GAN-based methods have been also used to tackle the problem of AD in images. However, most of the existing solutions have focused on the over-simplistic one-class anomaly detection in images, whose goal is labeling the whole image as anomalous or normal, rather than finding subtle anomalous regions within the image. While still being important, this problem is simpler to solve, since different images have global features that can be opportunely used by the anomaly detector.

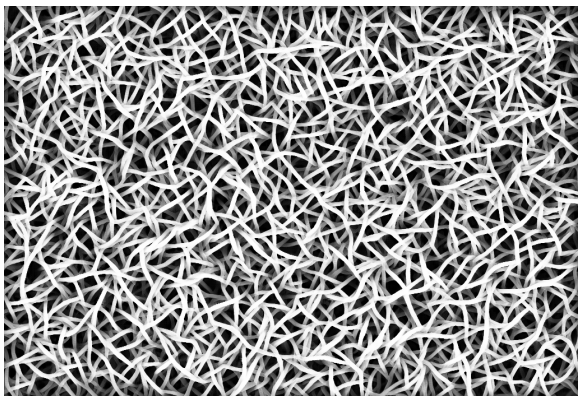


Figure 1: This Nanofiber Image is not real: it is generated by our model.

Hence, we design a new model, the Fully Convolutional Bidirectional Generative Adversarial Network (FCBiGAN), which is capable of providing an end-to-end solution for detecting anomalies in high resolution texture images. We define different anomaly scores, that can be assigned to each pixel in an image, in order to locate the anomalous regions within it. Our model, while still retaining its generative capabilities, achieves impressive results in terms of

computational efficiency for the AD task and is capable of matching the performances of a state-of-the-art GAN-based AD method, the fast-AnoGAN [5].

2. Problem Formulation

Anomaly detection in images can be defined as follows: given an image $\boldsymbol{x}_{test}$, defined over the pixel domain $\mathbb{Z}_p \in \mathbb{Z}^2$, locate any anomalous region in $\boldsymbol{x}_{test}$, estimating an unknown anomaly mask M, defined as

$$M(c) = \begin{cases} 0, & \text{if } c \text{ is normal} \\ 1, & \text{if } c \text{ is anomalous} \end{cases} \tag{1}$$

where $c \in \mathbb{Z}_p$ is the pixel inside the image that must be labeled as normal or anomalous. The estimated anomaly mask $\hat{M}$ needs to cover the anomalous regions in $\boldsymbol{x}_{test}$, in order to properly segment them.
In our case, the anomaly mask is produced by a machine learning model $\mathcal{M}$ which is trained with a semi-supervised training algorithm. In semi-supervised learning, the training dataset TR only contains normal images that are used by $\mathcal{M}$ to estimate the distribution of normal images, p_{data}. At test time, $\mathcal{M}$ can be used to assign an anomaly score to each pixel, to measure its degree of anomaly. The anomaly mask can be obtained by setting a decision rule to the pixelwise anomaly score, that is usually based on a threshold.

3. Related works

3.1. GANs Background

Our work focuses on designing a practical model for AD that is part of the GANs' family. Firstly introduced by Goodfellow et al. in 2014, a GAN is model composed of two neural networks, $\mathcal{G}$ and $\mathcal{D}$. The Generator $\mathcal{G} : \Omega_z \to \Omega_I$ learns the mapping from the latent space to the image space. The latent space is usually defined as a known prior distribution p_z, such as the Gaussian distribution, from which a latent code $\boldsymbol{z}$ is drawn and used to generate the fake image $\mathcal{G}(\boldsymbol{z})$. The Discriminator $\mathcal{D} : \Omega_I \to (0, 1) \subset \mathbb{R}$, learns the mapping from the image space to a probability. Given an input image $\boldsymbol{x}$, $\mathcal{D}$ learns to distinguish images coming from TR, also called real images, from the one generated by $\mathcal{G}$. The output of $\mathcal{D}$ corresponds to the probability that the given image comes from the real data.
$\mathcal{D}$ and the $\mathcal{G}$ parameters are optimized by letting the two neural networks play an adversarial zero-sum game, described by the following optimization problem:

$$\begin{aligned} \min_{\mathcal{G}} \max_{\mathcal{D}} V(\mathcal{D}, \mathcal{G}) = \mathbb{E}_{\boldsymbol{x} \sim p_{data}}[\log \mathcal{D}(\boldsymbol{x})] \\ + \mathbb{E}_{\boldsymbol{z} \sim p_z}[\log (1 - \mathcal{D}(\mathcal{G}(\boldsymbol{z})))] \end{aligned} \tag{2}$$

By optimizing Equation (2), $\mathcal{D}$ tries to find the best decision boundary that separates samples coming from the training set to the one produced by $\mathcal{G}$. On the other hand, $\mathcal{G}$ uses the $\mathcal{D}$'s output as a feedback to improve the quality of its generated images: by increasing the probability of being real assigned to the fake images, the distribution of fake images p_g distance to p_{data} is reduced. Goodfellow et al. proved that in the optimal point of convergence is when $p_g = p_{data}$. The ability of modeling the distribution of real data is what makes GAN interesting to use in the AD task. $\mathcal{D}$, which must know how to properly separate samples from the training set from every other kind of image that $\mathcal{G}$ produces, can be indeed used to score how distant a test image is from the real images in TR. $\mathcal{G}$ instead, can be potentially used to reconstruct images, by inverting its mapping: $\mathcal{G}^{-1} : \Omega_I \to \Omega_z$. Indeed, by finding the latent code associated to an image, the $\mathcal{G}$ can be used to reconstruct the image itself, similarly to what happens with autoencoders. Anomalous images should yield higher reconstruction errors, since they are not part of the manifold of images learnt by $\mathcal{G}$.

3.2. Bidirectional Generative Adversarial Networks

During the recent years many GAN based AD methods have been proposed, and most of them have been analyzed to build our solution: some of them replace the generator with an autoencoder, trading the GAN generative capabilities for the reconstruction ones. We are not interested in this kind of models, which in our opinion are closer to the autoencoder family than to the GAN one. Indeed, we refer to them as Autoencoders Trained with Adversarial Losses. We report here the model that we used as the main reference for the FCBiGAN.
In 2017 the Bidirectional Generative Adversarial Network (BiGAN) was first introduced by Don-

ahue et al.: this model extends the vanilla GAN architecture by adding an Encoder $\mathcal{E} : \Omega_I \to \Omega_z$, that learns to inver the mapping of $\mathcal{G}$. In the BiGAN architecture, $\mathcal{D}$ takes as input the pair image $\boldsymbol{x}$ and corresponding latent code $\boldsymbol{z}$. In case of real images, $\boldsymbol{z}$ is the one provided by $\mathcal{E}(\boldsymbol{x})$, while for fake images, $\boldsymbol{z}$ is the vector used as input of $\mathcal{G}$ that gives as output the fake image $\mathcal{G}(\boldsymbol{z})$. The BiGAN's adversarial game value function can be thought as an extension of the original GAN's one, and is defined as follows:

$$\begin{aligned}\min_{\mathcal{G},\mathcal{E}} \max_{\mathcal{D}} V(\mathcal{D},\mathcal{E},\mathcal{G}) = \\ \mathbb{E}_{\boldsymbol{x}\sim p_{data}}[\log \mathcal{D}(\boldsymbol{x},\mathcal{E}(\boldsymbol{x}))] \\ + \mathbb{E}_{\boldsymbol{z}\sim p_z}[\log(1-\mathcal{D}(\mathcal{G}(\boldsymbol{z}),\boldsymbol{z}))]\end{aligned} \tag{3}$$

$\mathcal{E}$ learns to invert $\mathcal{G}$ by sharing a common objective in the adversarial game against $\mathcal{D}$, as proved by Donahue et al.. Even if it wasn't originally designed for AD purposes, the BiGAN can also be used for AD by defining the following anomaly score:

$$A(\boldsymbol{x}_{test}) = (1-\alpha)\cdot L_R(\boldsymbol{x}_{test}) + \alpha \cdot L_D(\boldsymbol{x}_{test}) \tag{4}$$

where $L_R(\boldsymbol{x}_{test}) = \|\boldsymbol{x}_{test} - \mathcal{G}(\mathcal{E}(\boldsymbol{x}_{test}))\|_1$, which is the distance between the image $\boldsymbol{x}_{test}$ and its reconstruction $\mathcal{G}(\mathcal{E}(\boldsymbol{x}_{test}))$ and L_D is the Discriminator-based loss, that can be either defined as the distance of the features obtained by an intermediate $\mathcal{D}$'s layer $\boldsymbol{f}$ respectively from an image and its reconstruction $L_D(\boldsymbol{x}_{test}) = \|\boldsymbol{f}(\boldsymbol{x}_{test},\mathcal{E}(\boldsymbol{x}_{test})) - \boldsymbol{f}(\mathcal{G}(\mathcal{E}(\boldsymbol{x}_{test})),\mathcal{E}(\boldsymbol{x}_{test}))\|_1$, or as the crossentropy loss for the real class $L_D(\boldsymbol{x}_{test}) = \log(1-\mathcal{D}(\boldsymbol{x}_{test},\mathcal{E}(\boldsymbol{x}_{test})))$.
Despite its well founded theoretical background, the BiGAN architecture often fails in reconstructing the input, since $\mathcal{E}$ and $\mathcal{G}$ have no explicit constraints that the networks can use to improve the reliability of the reconstructed images $\mathcal{G}(\mathcal{E}(\boldsymbol{x}))$.

4. Our Contribution

The goal of our thesis is to perform efficient AD on full scale images, by using a GAN-based method. Unfortunately, to the best of our knowledge, this problem has never been a priority for GAN-based approach, thus we focused on improving an existing GAN-based method architecture. Indeed, our main objective is performing AD directly on full scale images without the inefficient step of subdividing the test image in patches, which is commonly applied by the existing GAN-based approaches. Due to its strong theoretical foundations and its closeness to the original GAN model, we decided to improve the BiGAN. We totally revamped the original architecture implementation, and we built a new model that we call the Fully Convolutional BiGAN (FCBiGAN).

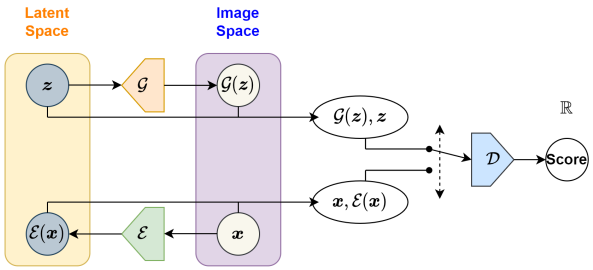


Figure 2: FCBiGAN architecture overview.

4.1. Improving The BiGAN

We modified different aspects of the original BiGAN:

- We improved the quality of generated samples by training our model with the Least Squares loss function,
- We added a constraint to $\mathcal{E}$'s loss, the Reconstruction Penalty, to improve the quality of the reconstructed samples,
- We built a new fully convolutional architecture for all the sub-networks in the model.

Least Square Loss The first issue that we dealt with is the instability of the training procedure of the BiGAN model. During the recent years, several new GAN's loss functions have been proposed in order to improve the training stability and the quality of synthetic images, avoiding problems such as the mode collapse or the vanishing gradient. Our first contribution is adapting, one of these new functions, the Least Squares (LS) adversarial loss [4], for the BiGAN architecture:

$$\begin{aligned}\min_{\mathcal{D}} L_{\mathcal{D}} = \frac{1}{2}\mathbb{E}_{\boldsymbol{x}\sim p_{data}}[(\mathcal{D}(\boldsymbol{x},\mathcal{E}(\boldsymbol{x}))-1)^2] \\ + \frac{1}{2}\mathbb{E}_{\boldsymbol{z}\sim p_z}[(\mathcal{D}(\mathcal{G}(\boldsymbol{z}),\boldsymbol{z}))^2] \\ \min_{\mathcal{G}} L_{\mathcal{G}} = \frac{1}{2}\mathbb{E}_{\boldsymbol{z}\sim p_z}[(\mathcal{D}(\mathcal{G}(\boldsymbol{z}),\boldsymbol{z})-1)^2] \\ \min_{\mathcal{E}} L_{\mathcal{E}} = \frac{1}{2}\mathbb{E}_{\boldsymbol{x}\sim p_{data}}[(\mathcal{D}(\boldsymbol{x},\mathcal{E}(\boldsymbol{x})))^2]\end{aligned} \tag{5}$$

The networks are less prone to the vanishing gradient issue, since their losses are based on the distance to a target value, rather than being based on the probability of being real.

Reconstruction Penalty Unfortunately, $\mathcal{E}$ and $\mathcal{G}$ often fail in reconstructing the input images. Thus, we further improved its loss function by adding an additional weighted term to $\mathcal{E}$ loss, which we call Reconstruction Penalty, which corresponds to the absolute error between the input and its reconstruction:

$$\min_{\mathcal{E}} L_{\mathcal{E}} = \frac{1}{2}\mathbb{E}_{\boldsymbol{x}\sim p_{data}}[(\mathcal{D}(\boldsymbol{x},\mathcal{E}(\boldsymbol{x})))^2 + \lambda\|\boldsymbol{x}-\mathcal{G}(\mathcal{E}(\boldsymbol{x}))\|1] \quad (6)$$

Fully Convolutional Architecture The other major change that we introduced to the original BiGAN model is implementing a fully convolutional architecture for the BiGAN. Fully connected layers in $\mathcal{E}$ and in $\mathcal{G}$ can be transformed into convolutional ones, just by using a filter which has the same spatial dimensions of the input feature map, and a depth that is equal to the corresponding dense layer. The original BiGAN's $\mathcal{D}$ has an architecture that is harder to adapt into a fully convolutional one: the original architecture processed $\boldsymbol{x}$ with a stack of convolutional layers, the final convolutional layer's feature map is then flattened and concatenated with the output of the fully connected layers that take care of $\boldsymbol{z}$, since the concatenate operation can work only with input that have the same spatial dimensions. Instead, in the FCBiGAN $\boldsymbol{z}$ is first upsampled to match $\boldsymbol{x}$'s spatial dimensions, using only convolutional layers: in this way we can avoid to transform $\boldsymbol{x}$ into a vector to match $\boldsymbol{z}$'s dimensions and we can obtain a pixelwise prediction. We achieved this by efficiently upsampling $\boldsymbol{z}$, using only convolutions. In particular we take advantage of a technique that is recently used for super-resolution models, the sub-pixel convolution: this technique efficiently rearranges the feature maps dimensions, to upscale the spatial dimensions by a factor of r.

4.2. Training and testing phase

The AD models are efficiently trained using patches extracted from the training images. Once the FCBiGAN's training phase has finished, $\mathcal{G}$ is able to generate high resolution images, since the convolution operation can work with inputs of any size. In the test phase, we can adapt the losses used in Equation (4), to our model in order to assign a pixelwise anomaly score to $\boldsymbol{x}_{test}$:

$$A(\boldsymbol{x}_{test}) = (1-\alpha)\cdot L_R(\boldsymbol{x}_{test}) + \alpha \cdot L_D(\boldsymbol{x}_{test}) \quad (7)$$

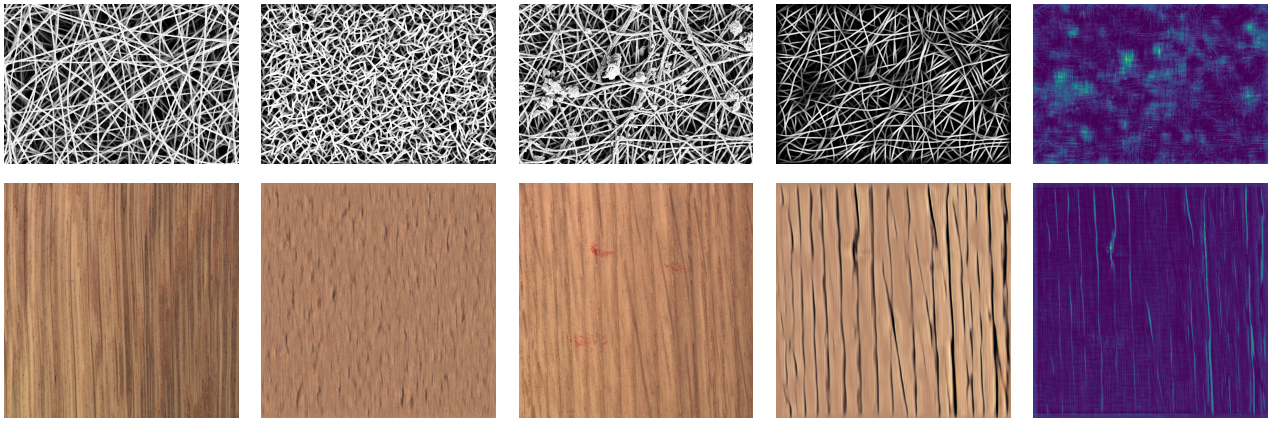
where the Reconstruction Error L_R measures the distance between $\boldsymbol{x}_{test}$ and $\mathcal{G}(\mathcal{E}(\boldsymbol{x}_{test}))$ and in our case is defined as $L_R = (\boldsymbol{x}_{test} - \mathcal{G}(\mathcal{E}(\boldsymbol{x}_{test})))^2$. The Discriminator Score L_D measures the distance that the pair $(\boldsymbol{x}_{test}, \mathcal{E}(\boldsymbol{x}_{test})))$ has from the target value of real images, $L_D = (\mathcal{D}(\boldsymbol{x}_{test}, \mathcal{E}(\boldsymbol{x}_{test})) - 1)^2$. In order to assign this loss for each input pixel in $\boldsymbol{x}_{test}$, we apply the shift-and-stitch trick, using dilated convolutions, as explained in [3].

These scores can be combined together using different aggregation strategies. In addition to the aggregation strategy already defined in Equation (7), the linear combination, we also define new aggregation strategies such as the Max operator, which takes the maximum among the aggregated losses or the Mahalanobis distance using the multivariate distribution estimated from the test losses on normal data.

5. Experimental Results

We evaluate our model with different types of experiments. The first task that the FCBiGAN must achieve, is to properly perform AD is generating high resolution images. We limit our experiments related to AD task only on texture images, since the FCBiGAN fails in generating and reconstructing images containing single objects. This limitation comes from the patchwise approach used in the training process of our model: in images containing objects the patches are semantically inconsistent between each other and the global properties of the image, such as the position of the eyes inside a face, are lost.

However for images containing a texture pattern, our model successfully learns to properly reconstruct normal data, while at the same time removing anomalies, as can be seen for the nanofibers' images in Figure 3. This is done without losing the generative capabilities of $\mathcal{G}$, since our model can still generate fake images.



(a) Real image (b) Fake image (c) Anomalous image (d) Reconstruction (e) Anomaly score map

Figure 3: Image examples from our benchmark dataset.

5.1. Datasets & Figures of Merit

We validate the FCBiGAN performance using two different datasets. The first dataset, the NanoTWICE dataset, contains 5 anomaly free grayscale images of nanofibers with a resolution of 1024×696 pixels, and 40 anomalous images, with the corresponding ground truth images, used to compute our evaluation metrics. This dataset is the reference dataset that we used to build the implementation of our model and that we used to explore the different anomaly scores. To evaluate our model's AD capabilities in addition to the NanoTWICE dataset, we use the MVTEC AD, which represents a more complex benchmark, since it contains different categories of texture materials, with images in the RGB space having a resolution that ranges from 800×800 to 1024×1024 pixels. Recently, this dataset is becoming a standard benchmark for AD methods. We use the MVTEC AD to measure the robustness of our architecture and to compare our model performance with the f-AnoGAN [5], which represents the current state-of-art for GAN-based AD.
We use as performance of metrics the reference threshold independent metrics previously used with the selected benchmark datasets. For the NanoTWICE dataset we compare the results using the Area Under the Receiving Operating Characteristic (AUROC) curve. The reference metric used with the MVTEC AD is instead the Area Under the Precision Recall (AUPR) curve. This metric is equivalent to the area under the curve that can be obtained by plotting the Precision and Recall at different thresholds. For both the AUROC and the AUPR, a perfect anomaly detector has an Area Under the Curve (AUC) of 1.0.

5.2. Experimental Setting

We train our model by extracting 12000 patches of size 48×48 from the training images. The pixel intensity of the patches is first normalized into the range [-1, 1], and then we apply randomly horizontal and vertical flipping and counter-clockwise rotation of 90 degrees to the training images as data augmentation.
In the test phase we apply the same normalization on the pixel intensity, and after obtaining the anomaly score, a mean filter of size 48×48 is applied to smooth the prediction. We report here the results obtained by the anomaly score defined as $\text{Max}(\tilde{L}_D, \tilde{L}_R)$ on the z-score standardized $\tilde{L}_D = \text{z-score}(L_D)$ and $\tilde{L}_R = \text{z-score}(L_R)$, which is the anomaly score that achieved the most consistent results between the different categories of images.
Our implementation is built using Tensorflow 2.4 and Keras API, and is trained and evaluated using an NVIDIA RTX 2070 SUPER™.

5.3. Evaluation

As can be seen in Figure 3, in the case of the NanoTWICE dataset, our model not only is capable of reconstructing the original data, but it can also generate a fake image that has the same resolution of the normal one. As can be seen from the wood images, our implementation, that was primarily optimized for the NanoTWICE, can't properly reconstruct the original image.

Indeed, when training a GAN, the hyperparameters tuning is a critical aspect that must be considered. In particular, for the FCBiGAN, tuning the patch size used during the training process is important to capture the texture pattern inside the image, otherwise the model is not able to properly replicate the irregular or regular pattern composing the whole image. However, we decided to keep the same architecture implementation for all the different categories of images to have a fair comparison with the f-AnoGAN, that, as for our case, was not tuned for every single category of images.
The quality of the reconstructions is heavily connected with the anomaly detection results. Indeed for the NanoTWICE dataset the AUROC reached by our method reaches the same level of the best performing ones, despite not using pre-trained models and a heavily-tuned prediction pipeline, while at the same time improving by a large margin the computational efficiency.

	Sparse Dictionary	Feature Dictionary	**FCBiGAN**$(\text{Max}(\tilde{L}_D, \tilde{L}_R))$
AUROC	0.93	**0.97**	0.95
Time	50s	15s	**0.65s**

Table 1: AUROC and execution time for a single image, for the nanofibers in NanoTWICE.

Instead for the MVTEC AD, despite the poor quality of reconstructions, for certain categories of texture images our model outperforms the f-AnoGAN, as can be seen in Table 2.

Category	**f-AnoGAN**	**FCBiGAN**$(\text{Max}(\tilde{L}_D, \tilde{L}_R))$
Carpet	0.025	**0.074**
Grid	0.050	**0.098**
Leather	**0.156**	0.112
Tile	0.093	**0.138**
Wood	**0.159**	0.046

Table 2: AUPR for each texture image category in MVTEC AD.

6. Conclusions

GAN are among the most promising models in the context of AD in images. Our experimental results show that $\mathcal{G}$ can generate high resolution images for images containing texture patterns, and produce realistic images for certain categories of texture images. This property can be successfully used to locate anomalous regions in an image. Indeed, our model is able to compete with the best performing methods on the NanoTWICE dataset, while at the same time outperforming them by a large margin by in terms of execution time. The experiments on the MVTEC AD dataset, instead, show that our architecture is capable of competing also with the f-AnoGAN, which is considered as one of the best GAN-based AD methods. However, the results on the MVTEC AD dataset also show that our model architecture must be opportunely tuned for each dataset, since the performance do change depending on it.
Future work will eventually find a method to adapt the patchwise training also for images containing objects, helping this method in reaching an important milestone both for high resolution image generation and AD in images.

References

[1] J. Donahue, P. Krähenbühl, and T. Darrell. Adversarial feature learning, 2017.

[2] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial networks, 2014.

[3] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation, 2015.

[4] X. Mao, Q. Li, H. Xie, R. Y. K. Lau, Z. Wang, and S. P. Smolley. Least squares generative adversarial networks, 2017.

[5] T. Schlegl, P. Seeböck, S. M. Waldstein, G. Langs, and U. Schmidt-Erfurth. f-anogan: Fast unsupervised anomaly detection with generative adversarial networks. *Medical Image Analysis*, 54:30–44, 2019. ISSN 1361-8415. doi: https://doi.org/10.1016/j.media.2019.01.010. URL `https://www.sciencedirect.com/science/article/pii/S1361841518302640`.