Modeling anomaly detection for imbalanced datasets with deep generative models

Santos Buitrago, N.R.

Award date:
2018

Technische Universiteit
Eindhoven
University of Technology

Department of Mathematics and Computer Science
Data Mining Research Group

Modeling Anomaly Detection for Imbalanced Datasets with Deep Generative Models

Master Thesis

Nazly Rocio Santos Buitrago

Supervisors:
Vlado Menkovski
Dimitrios Mavroeidis

A thesis presented for the degree of Master in Data Science and Engineering

Eindhoven, August 2018

Modeling Anomaly Detection for Imbalanced Datasets with Deep Generative Models

Nazly Rocio Santos Buitrago
Department of Mathematics and Computer Science
Eindhoven University of Technology
Eindhoven, The Netherlands
Email: n.r.santos.buitrago@student.tue.nl

***Abstract*—Imbalanced datasets, when positive samples are scarce, present a challenge for traditional binary classifiers. We address this problem with an Anomaly Detection framework. We define an anomaly as a datapoint that is *unlikely* generated by the density distribution function of a learned model trained only with normal samples. We use state-of-the-art deep generative models (GAN and VAE) for the estimation of a likelihood lower bound from which we obtain the distribution of the data. We define an evaluation protocol using a logistic regression classifier for mapping the log likelihood space to the probability space. The method is applied to MNIST dataset and to the use case of lung cancer detection with nodule extractions from NLST 3D dataset. The results show GANs and VAEs have some robustness for modeling a density function in the MNIST case using the normal datapoints and ability to separate abnormal samples. For the NLST case, neither GANs nor VAEs were able to capture the complexity of the data and discriminate anomalies at the level that this task requires.**

I. Introduction

One of the greatest challenges for machine learning is to deal with small datasets and insufficient amount of labelled data[1]. This is particularly true when there is a level of imbalance in the classes to predict. A classic classification task, in the Computer Vision domain, would require annotated images at a semantic level, and preferably, a similar amount of samples of each class[2]. In a binary classification problem, with negative and positive classes, having a skewed setup implies adjusting the chosen methods and the metrics of evaluation. Our research focuses on the problem of imbalanced datasets using an Anomaly Detector Framework. The purpose is to train a model with only one class, from which we have more samples, and treat the other class as an anomaly: a sample with low probability of being produced by such a learned model.

Anomaly detection has been an important research topic for several years and applied statistical methods have tried to address it in a conceptual way[3][4] . However, the interpretation of an *anomaly* in a machine learning task could fall in ambiguity, being context dependent or changing according to the type of data[5]. Furthermore, there are not clear evaluation standards and some methods with flaws in their protocol could be found[6]. There are different approaches for the design of an anomaly detector[7][8][9]. In particular, we consider it as a Probability Density Estimation process in which the goal is to discover the probability distribution of the *normal* data p_{data}, by finding the optimal parameters that define it. Computing these optimal parameters θ, means getting the values that maximize the likelihood of the observed data.

In the presence of relatively small number of samples, a Generative Model is best suited to learn the distribution of the features from each class[10]. Given a group of training datapoints $x_1, x_2, ..., x_n$, with labels y_i, a generative model would learn the join probability distribution $p(x, y)$[11].An explicit generative model estimates p_{data} by giving a log likelihood function, that can be seen as the density function of the data. Traditionally in machine learning, the model creation is based on Maximun Likelihood Estimation (MLE) for the parameters θ. Having the likelihood function $p_{model}(X|\theta)$, the MLE is defined by:

$$\theta_{MLE} = \underset{\theta}{\operatorname{argmax}}\, p_{model}(X|\theta) = \underset{\theta}{\operatorname{argmax}} \prod_i p_{model}(x_i|\theta) \tag{1}$$

When computing the MLE, we find the parameters that maximize the likelihood that our model p_{model} produced the data that was observed. The assumption is that the samples are independent and identically distributed, furthermore, computing MLE is equivalent to obtaining the log likelihood (since logarithm is a strictly increasing function, the logarithm of a function f achieves its maximum value at the same points as the function itself). This notion is particularly important to be able to train algorithms and define optimization methods. Equation 1 can be seen as a parametric function $f(x; w)$, in which x are the inputs and w the weights (parameters). When the weights w change, the function f responds differently for a given input. The quantification of how well the model performs w.r.t the input is given by the loss function $E(w)$, we, therefore, want to minimize the loss of the model $\operatorname{argmin}_w E(w)$. This optimization process could be seen as the computation of the negative log likelihood given by $E(w) = -\sum_n \log p[x_n|f(x_n; w)]$[12]. By having this form, the task becomes an optimization process that can be solved using Stochastic Gradient Descent (SGD).

In our anomaly detection framework the observed samples are the *normal*, whereas an abnormal sample is based on how *unlikely* it is to be generated by the model. Usually, the boundary to distinguish anomalies is given by the definition of a threshold ϵ, with respect to the learned distribution. It is

also the case that this cut-off is not obvious to identify and relies entirely on experts' opinions[4].

Deep Generative Models are the current unsupervised methods with strong capacity for feature representation, data generation and learning of the data distribution. Their structure, using neural networks, allows them to construct powerful functions from the training and generate new *alike* samples, particularly for high dimensional data, for which density estimation is a long standing problem.

This research was done at the Data Science department of Philips Research in Eindhoven, The Netherlands. The department has worked in the scope of cancer detection and screening analysis. Lung cancer alone was responsible for 1.69 million deaths in 2015[1]. Early cancer detection and diagnosis of abnormal anatomies, by means of Computer Tomography (CT), has been a recurrent research topic specially in the Computer Vision domain[13][14]. Screening a patient using CT techniques is a common procedure for examination of possible tumors (nodules) and abnormal tissue. Although there have been efforts to automatize the interpretation of the resultant images using CADe (computer-aided detection)[15], this task is still highly dependant on the opinion of experts and the use of historical data of the individual. The manual assessment of the data brings constraints in terms of time and quality, increasing the risk of error and delays in the clinical pipeline. As a resource, they provided a nodule detector[16] that could be potentially used as a base for posterior techniques aiming the identification of malign samples. Due to the relevance of the task, we would like to find a proper method and metric for the evaluation that can deal with the complexity of this particular data and that is able to find a differentiation between positive samples (cancer) and negative (non-cancer). Given the imbalanced nature of the dataset (25% - 75%), this scenario is a relevant application case for our Anomaly Detector Framework.

The contributions of this work are the following:

- Design of a comparative anomaly detection framework using the state-of-the-art deep generative models: GAN and VAE.
- Definition of an evaluation metric for abnormal samples relying on the likelihood estimation.
- Application of such a framework for the MNIST 2D dataset and the nodule extractor over NLST lung cancer 3D dataset.

II. Related Work

Two main frameworks gained popularity and acceptance in the deep learning community: Generative Adversarial Networks[17] (GAN) and Variational AutoEnconders[18] (VAE). Since their appearance in 2013-2014, strong research moved into their interpretation, application and development. Currently there are more 200 variations of GANs in terms of training, architecture, loss function, objective and applications[2]. Some of the impressive applications involve super resolution image conversion [19], image inpainting[20], cross-domain transfer (CycleGAN)[21], pose guided person generation[22] among several others. GANs are known for being unstable to train, with several hyperparameters to tune. However, the results are sharp, and could fool the human eye when producing new image samples. The community has worked in different tips and tricks to make it work[23] and there are several baselines for configurations according to specific applications. This continuous research has made them the state-of-the-art approach in unsupervised learning. VAEs are known for producing blurry results in the new samples. However, their training setup is well defined and has an explicit generation of the learned probability distribution. The applications with VAEs involve more particular domains, including different types of data, sequential, discrete or continuous, like creation of music sequences[24]. Due to their proved performance when dealing with high dimensional datasets in an unsupervised setup, deep generative models are suitable for design of the anomaly detection.

Other density estimation methods which rely in autoregression models include; the Neural Autoregressive Distribution Estimator (NADE)[25], the real-valued neural autoregressive density-estimator (RNADE)[26], the real-valued non-volume preserving method[27] or the Masked Autoregressive Flow estimation[10]. An improved VAE approach with inverse autoregressive flows[28] has also demonstrated strong capacity for density estimation. Since these models focus on a specific setup for training, they are out of scope of this project, but could be considered as future work.

Figure 1: MNIST dataset class separation for Anomaly Detection

III. Applications

A. MNIST dataset, 2D benchmark setup

For making the results comparable, also to be able to have a perceptual and qualitative evaluation of the images, initially we trained and formulated the anomaly detection framework over the computer vision benchmark dataset MNIST[3]. For our experiments we splitted the dataset in a binary classification problem, having an imbalanced setup. We trained only using the Negative Samples (a subset of 9216 images containing equal samples of 0-8). Then we tested the approach using some Positive samples (images of 9). Figure 1 shows the distribution of the classes.

[1] http://www.who.int/news-room/fact-sheets/detail/cancer

[2] https://github.com/hindupuravinash/the-gan-zoo

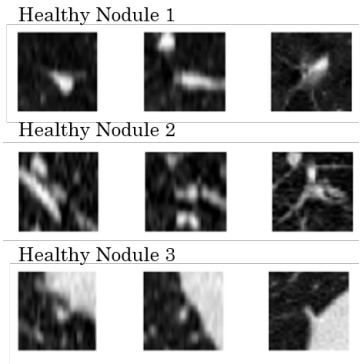
[3] http://yann.lecun.com/exdb/mnist/

B. Lung cancer detection, nodules from NLST 3D dataset

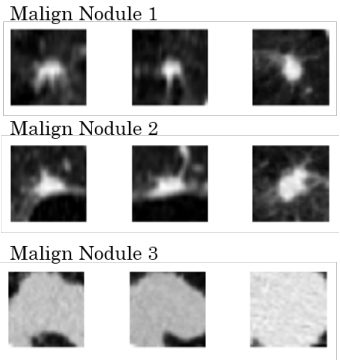
The medical imaging domain has specific challenges when using machine learning techniques. This is more evident for the segmentation of organs from 3D volumes into 2D slices. Lung cancer detection usually requires annotated images (cancer, non-cancer) at a nodule (tumor) level, with its additional information such as malignancy, diameter, spiculation or lobulation, and a preferably amount of samples of each class. Recent efforts[4] leveraged from the use of the publicly available datasets with considerable nodule annotations, achieving good performance. However, this supervised approach does not seem to be easily scalable due to the lack of new, equally rich data. In this particular application, the benign nodules of the lung do not share specific characteristics. They are diverse in size, texture, shape, and location, as a consequence, the differentiation between malign nodules is not evident for human perception. Due to the high complexity of the data, we are not sure how far the abnormal samples are from the normal samples. We would like to test our anomaly detection framework over this scope and evaluate whether the generative models are able to understand class related particularities such as shapes, edges or spacial position, plus additional hidden features. The raw dataset is provided by the NLST (National Lung Screening Trial), consisting of high resolution chest tomographies. The input for our models is the result of a nodule detector. We are dealing with 3D cubes of 32x32x32 mm^3 with a voxel size of $1mm^3$. Although many computer vision problems in the medical domain lie in a 3D image scope, most of the research found[29][30][31][32][33], deals with them in a 2D domain due to the high structure complexity of the data, and the computational limitation. For our problem, treating the nodules in a 2D scheme could imply missing information with respect to the space in the lung and connection with additional tissue. Also, healthy nodules have many variations between them, not visible when the nodule is sliced in 2D sections. In our research we designed and implemented 3D models for handling the data, and investigated whether this approach helped for the robust estimation of the probability estimation. Figure 2 shows some nodules example, the variation between the data and the difficulty for humans to discriminate healthy from abnormal samples.

For our experiments the input of the models is a 3D cube of 28x28x28 pixels, result of a data augmentation process that produces sub patches from the original shape of 32x32x32. This is an important step in the training, since it enriches the features to be seen by the model, and reduce probability of overfitting[32][34][35]. For convenience in display, figure 3 shows the 3D image as a set of 25 slices of 28x28 pixels. This convention will be used in the remaining of this document when referring to a 3D image from this dataset. The data augmentation method produces an automatic split of the dataset intro training, validation and test. Table I shows the distribution.

[4]https://github.com/dhammack/DSB2017/blob/master/dsb_2017_daniel_hammack.pdf



(a) Healthy Samples



(b) Abnormal Samples

Figure 2: Examples of samples in the dataset with their axial, coronal and sagital perspective. Figure 2a shows 3 different healthy nodules. Figure 2b shows 3 different nodules identified as abnormal (positive for cancer).

	Label 0 Healthy	Label 1 Cancer	Total
Training	1722	460	2182
Validation	431	115	546
Testing	539	143	682

Table I: Lung nodule dataset after data augmentation

IV. Deep Generative Models Background

A. Lower Bound of Likelihood Estimation

Having a likelihood function $p(x|\theta)$ of a complex high dimensional data, could yield in an intractable function. That is, the computational complexity for solving the operation is exponential and it is not possible to get the solution. In the Variational Bayes approach, we introduce latent (hidden) variables[5] $z \in Z$, in which $Z \rightarrow X$, that help to make an approximation of the likelihood. By Bayes' rule, the posterior distribution of the hidden variables could be defined as:

$$p(z|x) = \frac{p(x|z)p(z)}{p(x)} = \frac{p(x|z)p(z)}{\int_z p(x,z)} \tag{2}$$

For computational convenience, and since the logarithm is a monotonically increasing function, maximizing a function is

[5]https://xyang35.github.io/2017/04/14/variational-lower-bound/

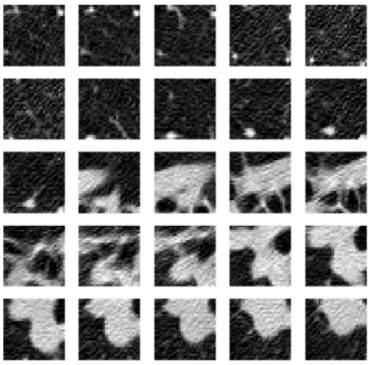


Figure 3: Displaying 25 slices of 28x28 pixels, as a representation of the cube of 28x28x28 pixels used for training the models.

equal to maximize the log of it, then we can express $p(x)$ (the distribution of our observed data) as $\log p(x)$. By variational inference[36] we can have:

$$\begin{aligned}\log p(x) &= \log \int_z p(x,z)\, dz \\ &= \log \int_z p(x,z)\frac{q(z)}{q(z)}\, dz \\ &= \log\left(\mathbb{E}_q\left[\frac{p(x,z)}{q(z)}\right]\right) \\ &\geq \mathbb{E}_q[\log p(x,z)] - \mathbb{E}_q[\log q(z)]\end{aligned} \tag{3}$$

In which $\mathbb{E}_q[\log p(x,z)] - \mathbb{E}_q[\log q(z)]$ is the Evidence Lower Bound (ELBO). A function representing the marginal likelihood of observations.

Kullback-Leibler Divergence (KL) is used to measure how close the approximation q is to the posterior of the distribution:

$$KL[q(z)||p(z|x)] = \int_z q(z)\log\frac{q(z)}{p(z|x)}\, dz = \mathbb{E}_q\left[\log\frac{q(z)}{p(z|x)}\right] \tag{4}$$

By applying log properties we can express equation 4 as:

$$\begin{aligned}KL[q(z)||p(z|x)] &= \mathbb{E}_q\big[\log\frac{q(z)}{p(z|x)}\big] \\ &= \mathbb{E}_q[\log q(z)] - \mathbb{E}_q[\log p(z|x)] \\ &= \mathbb{E}_q[\log q(z)] - \mathbb{E}_q[\log p(z,x)] + \log p(x) \\ &= -(\mathbb{E}_q[\log p(z,x)] - \mathbb{E}_q[\log q(z)]) + \log p(x)\end{aligned} \tag{5}$$

In which the final expression is the previous defined ELBO with an additional constant that does not depend on q. Maximizing the ELBO is equivalent to minimize the KL divergence to the posterior. In practical purposes, there is not direct computation of the KL divergence, but instead a maximization of the ELBO over densities $q(z)$.

B. Generative Adversarial Networks

Since its first appearance in 2014, GANs[17] have demonstrated interesting results in unsupervised tasks over images[34], its adversarial training consists in a Generator G producing images and a Discriminator D validating them.

In a GAN we have an adversarial training between Generator G and Discriminator D. First, we have a probability distribution that we define, represented by p_z. A sample from this distribution is $z \backsim p_z$, normally noise. The Generator G is a function that takes the distribution and generates samples from it, $G(z)$. The goal is to define G in a way that, taking a vector from p_z, it will be able to return an image sample. The definition of G is done using deep neural networks. In the original paper this is a Multilayer Perceptron definition. However, the main concern is how to train this neural network and its latent variables.

That is the role of Discriminator D. Now, let us represent the unknown probability distribution of the original images used for training as p_{data}. $G(z)$ draws images or samples following another distribution, notated as the generative probability distribution p_g. In this training process we would like to end up having $p_{data} = p_g$.

The Discriminator $D(x)$ takes an image x as input and returns the probability that x is sampled from p_{data}. As a main concept for the implementation of the loss, when x comes from p_{data} the Discriminator returns a value closer to 1, and when x comes from p_g, a value closer to 0. This evaluation determines which image is real and which one is fake.

DCGAN training setup

The literature review showed that a DCGAN architecture[37], formed by 3D convolutional layers and 3D transposed convolutions, is able to learn from 3D images composed by voxels[38]. This is why, our first approach was to design and train a DCGAN with the standard loss function and some of the recent tips and tricks[6].

Equation 6 is used as a way to evaluate the training of DCGAN.

$$\begin{aligned}\min_G \max_D V(D,G) = \mathbb{E}_{x\backsim P_{data(x)}}\log(D(x)) \\ +\mathbb{E}_{z\backsim P_z(z)} log(1 - D(G(x)))\end{aligned} \tag{6}$$

In practical terms, during training, to apply Gradient Descent, we optimize:

$$\begin{aligned}\nabla_{\theta d}\frac{1}{m}\sum_{i=1}^{m}[\log D(x^{(i)}) \\ -\log(D(G(z^{(i)}))]\end{aligned} \tag{7}$$

The training setup is done as suggested by the original paper. It is expected that in the learning process, initially, the Discriminator D would reject most of the samples generated by the generator, at the same time $G(z)$ is going to produce mainly noisy images. At later steps, D would have problems distinguishing fake samples from real ones. As a consequence, a good performance of the GAN will show the value of the Discriminator loss close to 0.5.

[6]https://github.com/soumith/ganhacks

The experiments using this approach showed that making a GAN learn from this data is a fragile process; the loss function computed during training did not converge, and the proposed architecture of the referent paper did not produce results (non understandable outputs) for the NLST dataset.

Alternatively, looking into different improvements in the training of GANs, and motivated by previous results in a 2D setup, using similar data[39], our experiments were focused in designing and training a WGAN-GP architecture.

WGAN-GP training setup

The main concept behind the WGAN approach is to change the role of the Discriminator and use it as a *critic*. The loss function that we are trying to optimize is giving by the earth mover distance (or Wasserstein distance):

$$W(P_{data}, P_g) = \inf_{\gamma \in \Pi(P_{data}, P_g)} \mathbb{E}_{(x,y)\backsim\gamma} \|x - y\| \tag{8}$$

In the initial steps of training, we would expect the distance between real p_{data} and fake p_g images to be big. Then, at later steps, this distance is decreased, getting close to the total loss function

In practical terms[40], to apply Gradient Descent, we optimize:

$$\nabla_w \frac{1}{m} \sum_{i=1}^{m} [f(x^{(i)}) - f(G(z^{(i)})] \tag{9}$$

where f is a 1-Lipschitz function.

C. Variational Autoencoders

In computer vision, Pixel distance or reconstruction metric, is far from a perceptual distance, a sample needs to be remarkably close in pixel distance to a datapoint X before it can be considered evidence that X is likely under the model. Because of that, it is very difficult to measure likelihood of images under a model using only sampling. Then, we would like to use a Generative Model with stronger capacity and less parameters to configure. The training framework used in the VAE has a strong mathematical conceptualization. They are based on function approximators and can be trained using SGD. VAEs includes assumptions, as many generative models in machine learning, but the error is consider small given its capacity[41].. The main objective in the VAE training is to estimate:

$$P(X) = \int P(X|z;\theta) P(z) dz \tag{10}$$

in which there is a dependency between X and z. This is the framework for the *Maximum Likelihood*, the criteria that can be used to measure if a sample is likely to be from the distribution $P(X)$ or if is unlikely. In VAEs, we can assume that $P(X|z;\theta) = \mathcal{N}(X|f(z;\theta), \sigma^2 * I)$, this is, that the output is a Gaussian distribution with mean $f(z;\theta)$ and covariance I multiplied by a scalar σ. VAEs preserve the similarity metric, but focus in the sampling procedure. In order to sample values of z that are likely to have produced x, VAEs use the variational inference described in equation 5, allowing to compute $\mathbb{E}_{z(q)} P(x|z)$. In the same way, we can express our model as:

$$\begin{aligned} KL[q(z)||p(z|x)] = \mathbb{E}_{z(q)}[\log q(z) - \log p(x|z) \\ - \log p(z)] + \log p(x). \end{aligned} \tag{11}$$

Rearranging and expressing into a KL-divergence terms, equation 12 is the core of the training in the VAE. The left side is the term we want to maximize plus an error. The left side is trained using SGD.

$$\begin{gathered} \underbrace{\log p(x)}_{\substack{\text{Distribution} \\ \text{of the Model}}} - \underbrace{KL[q(z|x)||p(z|x)]}_{\text{error}} = \\ \underbrace{\underbrace{\mathbb{E}_{z(q)}[\log p(x|z)]}_{\text{Reconstruction Loss}} - \underbrace{KL[q(z|x)||p(z)]}_{\text{Latent loss}}}_{\text{Evidence Lower BOund (ELBO)}} \end{gathered} \tag{12}$$

V. Anomaly Detection Framework

A. Anomaly Detection with GANs

The reference paper for Anomaly Detection[42], based on work from[20], proposed a framework composed by three steps: 1. learn a manifold $\mathcal{X}$ of a corpus of *normal* images, 2. Map images back to the latent space, and, 3. Detect abnormal samples using a visual and perceptual component.

The visual component is the residual loss, which compares similarity of images at pixel level through the generator G. The residual loss is defined by:

$$\mathcal{L}_R(z_\gamma) = \sum |x - G(z_\gamma)| \tag{13}$$

Where x is the query image and $G(z_\gamma)$ is optimally, the closest generated image after γ steps in the latent space. If the generator is able to generate a perfect looking image with respect to the query, the residual loss is $\mathcal{L}_R(z_\gamma) = 0$. The perceptual component is defined as a discriminator loss, based on the discriminator D.

$$\mathcal{L}_D(z_\gamma) = \sum |f(x) - f(G(z_\gamma))| \tag{14}$$

where f is a layer from the discriminator. The features learned from the query image $f(x)$ are compared to the ones of the closest image in the latent space $f(G(z_\gamma))$.

The method for detecting an abnormal sample consists in using the overall loss composed by the sum of the residual and the discriminator loss. A λ parameter is set to give more weight to each loss and calibrate it according to the training:

$$\mathcal{L}(z_\gamma) = (1 - \lambda)\mathcal{L}_R(z_\gamma) + \lambda \mathcal{L}_D(z_\gamma) \tag{15}$$

z_γ is a value obtained after starting at some random point in the latent space z_1, which generates an image $G(z_1)$, then, *walking* towards the direction of the query image x. After $z_1, z_2, ..., z_\gamma$ steps, if the query image x belongs to the learned distribution of the model, we would expect $G(z_\gamma) \approx x$.

It has been proven that it is possible to walk in the latent space in GANs using techniques such as interpolation[20], to obtain intermediate points and check the respective sample generation.

Equation 15 is used to update and optimize z_γ through stochastic gradient descend (SGD) with momentum,

$$\Delta z_i = -\alpha \frac{\partial \mathcal{L}(z_i)}{\partial z_i} - \beta \Delta z_{i-1}$$
$$z_{i+1} = z_i + \Delta z_i$$

After the training, we obtain the closest image to the query x, generated by $G(z_\gamma)$ and the loss value $\mathcal{L}(z_\gamma)$. As suggested in the paper, we use equation 15 to set a threshold ϵ for Anomaly Detection. The reasoning is that if the query image x is close to the learned representation, it is consider *normal* and will have a low score. If x is *abnormal*, it will have a higher score, above the defined threshold. For our approach We considered the loss value as a likelihood lower bound.

B. Anomaly Detection with VAEs

After the training of a VAE, we would have the learned distributions for computing the integral in equation 10. Similar approaches[43] use the reconstruction error for anomaly detection. However, just by using that part of the loss given by the VAE, we are not really computing $p(x)$ as explained in section IV-A. Our goal in Anomaly Detection is to estimate how likely is an image query to belong to the learned distribution.

In our approach, we considered that our latent space z follows a multivariate normal distribution. We can assume then, that

$$p(\boldsymbol{x}|\boldsymbol{z}) = \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}, \sigma.I) = \prod_{j=1}^{p} \mathcal{N}(x_j|\mu_j, \sigma)$$

Where p corresponds to each pixel in the dimension of the image. By moving into log space, we can then transform $p(x|z)$ in terms of the PDF of a Gaussian with mean μ_j and σ_j as standard deviation:

$$\begin{aligned}
\log p(x|z) &= \sum_{j=1}^{p} \log \mathcal{N}(x_j|\mu_j, \sigma) \\
&= \sum_{j=1}^{p} \log \frac{1}{\sigma\sqrt{2\pi}} e^{-(x_j-\mu_j)^2/2\sigma^2} \\
&= \sum_{j=1}^{p} -\frac{1}{2}\log 2\pi\sigma^2 - \frac{(x_j-\mu_j)^2}{2\sigma^2} \\
&= C - \frac{1}{2\sigma^2}\sum_{j=1}^{p}(x_j-\mu_j)^2
\end{aligned} \tag{16}$$

The previous equation (eq. 16), is the reconstruction loss used in the training. For computation purposes, we assumed the expressions that do not depend on p to be constant, and $\frac{1}{2\sigma^2} = 1$. Notice that this reconstruction could be seen as the commonly used Summed Squared Error loss. Implementation wise, we could compute it as a Mean Squared error, multiplied by the number of pixels.

Based on equation 5 of the Background section IV-A for the definition of likelihood lower bound, we will estimate the likelihood by sampling from z, L times for each datapoint i.e. each image. Thus, for a lower bound estimation at a datapoint level, we have:

$$\mathbb{E}_{q(z)}[\log p(x|z)] \approx \frac{1}{L}\sum_{l=1}^{L} \log p(x|z^{(l)}), z^{(l)} \backsim q(z)$$

The result value for each data point is considered the likelihood estimation.

C. Evaluation

The anomaly detection framework is based on the learning capacity of the chosen Deep Generative Models. After training, we assume the model learned specific features from the data and the resultant loss values can be seen as a likelihood value for each data point measuring how likely is for that sample of p_{model} to belong to the distribution p_{data}. We can also infer that the likelihood values for abnormal samples are far (greater) than the observed data. To evaluate this assumption, we took equal number of *normal* and *abnormal* samples and computed the likelihood value for all the datapoints. That is, we passed n_normal images through the trained GAN and VAE, and then we repeated the process with $n\_abnormal$ images.

For the resultant values, we analyzed their density distribution, and compared visually the curve for normal samples against the abnormal. The densities are not linearly separable since the values are not expressing a probability by default. Our method for giving a probabilistic meaning to our likelihood scores is to interpret the results as a single feature (or representation) of the data. We have then, a one-dimensional feature composed by the likelihood values (lower bound) of both normal and abnormal samples, with the same number for each class. The evaluation method is a classifier which is going to learn a separation of the data. For our approach we tried tried different classifiers and chose Logistic Regression to establish the capacity of the detector. We noticed that this simple classifier was able to discriminate the input. The AUROC of the classifier was used as evaluation metric for all the experiments.

VI. Results

The performed experiments could be divided in two frameworks: Anomaly Detection framework with GAN architectures (GAN-AD) and Anomaly Detection framework with VAE (VAE-AD). For both phases the tests were performed for the application cases described in section III, over 2D MNIST and 3D NLST dataset. As a general structure, we present:

- High level defined architecture for the GAN-AD and VAN-AD over NLST dataset,
- Qualitative performance of the models in terms of the generation/reconstruction of samples,

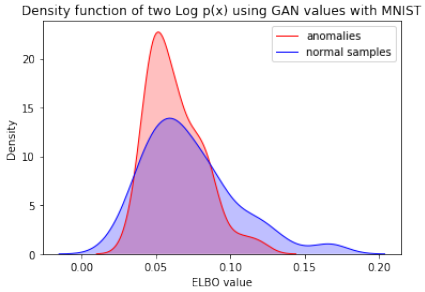


Figure 4: Density distribution for the scores (taken as an ELBO) obtained with GAN-AD with MNIST dataset. We can perceive that there is not clear separation or threshold between normal and abnormal samples

- Visual evaluation of the Anomaly Detector output using plot of density distributions,
- The AUROC score obtained after passing the Anomaly Detector output as input of a Logistic Regression classifier.

A. *MNIST 2D setup*

For both models we trained with a subset of 9216 images containing normal samples, (images of numbers 0-8). Then we evaluated the approach using equal number of samples from both classes: 450 Positive samples (images of number 9) and 450 additional normal samples.

1) GAN-AD: There have been several architectures for GAN implementations[44][40]. For a dataset like MNIST with 784 dimensions, we used the proposed DCGAN[37], with similar configuration of the convolutional layers, and the same recommendations for training. These type of implementations are widely explored by the community, so there was not need to tune the hyper-parameters with rigor. We trained for 50 epochs and we computed the anomaly scores using 100 backpropagation steps for finding the optimal z mapping back to latent space. We chose $\lambda = 0.5$ in equation 15, after empirical experimentation.

After the training, we performed the Anomaly Detection framework, obtaining the loss scores. We treated them as a likelihood value. Figure 4 shows the distribution of the values, corresponding to each class. We found out high fragility in the results when we: increased the trained epochs, increased the number of backpropagation steps for finding the optimal z in latent space, and, changed the number of samples used both for training and for evaluation.

For quantitative evaluation, we took the values as input for the Logistic Regression classifier, figure 5 plots the resultant AUROC. The value of 0.66 shows that the classifier is still able to separate normal from abnormal samples, although not in a powerful way. As explained before, this value was fragile for number of samples used in the evaluation and the previous training. We plotted the result obtained when using the same configuration than VAE-AD, as a fair comparison.

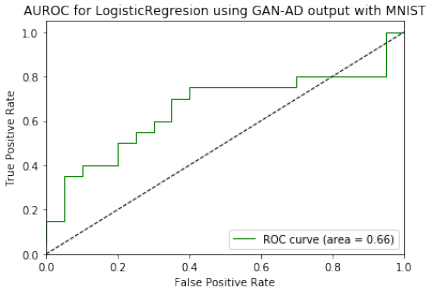
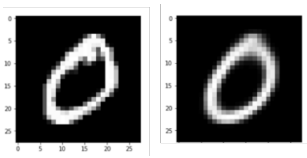


Figure 5: AUROC score for the Logistic Regression classifier of GAN-AD with MNIST

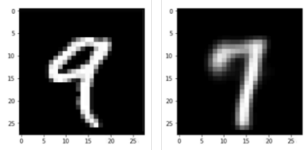
2) VAE-AD: For the 2D context of MNIST, we use a simple VAE architecture with 2D convolutions and 2D upsamplings. For this dataset there was no need of a deep level of convolutions or number of units. We trained the VAE for 30 epochs, using the same 9216 images labeled as *normal* (numbers from [0-8]). The training last aprox. 1 min. The model is able to visually reconstruct with high quality a normal sample, and tries to approximate an abnormal sample with the information it got during training. See Fig.6.

The metrics of our model were used for the computation of the likelihood lower bound. For the VAE-AD, we took the trained VAE and passed new samples through it. For this step we used 450 normal samples and 450 abnormal samples. The result is a density graph composed by the ELBO values (eq. **??**). Figure 7 shows the distribution of the results for both type of samples.

Moving from log space into a probability space, we passed the obtained values to the logistic regression algorithm. Figure 8 shows the AUROC graph and ROC score results. The score of 0.84 shows a high potential for the features as differentiation between sample types.



(a) Reconstruction of *negative* samples (seen during training)



(b) Reconstruction of *positive* samples (anomalies)

Figure 6: Reconstruction of images using the trained VAE. The first column is the query image given to the VAE. The second column is the reconstructed image. We can notice that for 6b, the model tries to reconstruct a number 9 using a similar image seen during training (most likely a number 7).

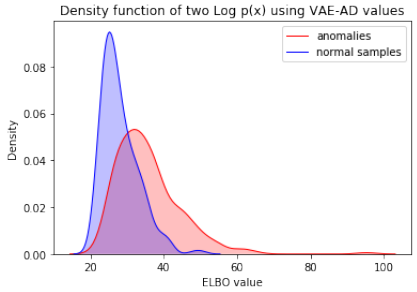


Figure 7: Density distribution of VAE-AD with MNIST dataset. We can see how the values create a differentiation in the densities. Values greater than 50 are highly probably to be anomalies.

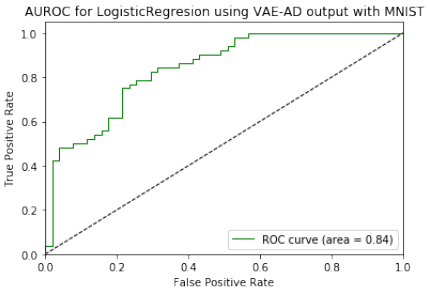


Figure 8: AUROC score for VAE-AD for MNIST. The result implies the classifier has potential to discriminate normal from abnormal samples, using the output of the model

.

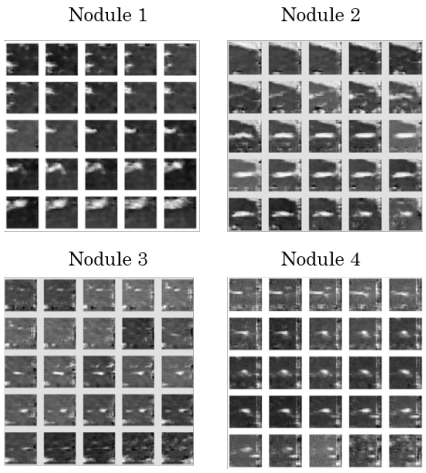


Figure 9: Four nodules generated by the 3DGAN

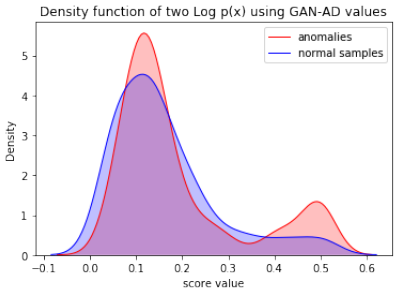


Figure 10: Density distribution of values using GAN-AD with NLST

B. NLST 3D nodules dataset

1) GAN-AD: After exhaustive parameter tuning and attempts in training, figure 11 shows the 3D WGAN-GP[45] architecture that was able to learn from the nodule data and produced some visually understandable results.

Training was configured with a seed z of size 100 following a uniform distribution. We trained for 100 epochs, since the loss function for the critic showed optimization around epoch 50 and stops learning from epoch 60. Figure 9 shows examples of new data produced by the GAN. The samples, however keep improving visually until 100 epochs.

While comparing different generated images from the variations in training of WGAN-GP, we notice a partial mode collapse[23] in the samples. We can see that the images look similar and they are not able to create complex shapes as seen in the training data. This was the case for less sharp images generated from simpler architectures or with change on the random seed z. Even when the generator is able to construct simple shapes, they are very similar between them.

With the final trained architecture, fig.11, we compute the metrics proposed in our methodology.

We used the testing split; 120 normal samples and 120 abnormal samples for calculation of the loss score (taken as a likelihood value). We ran 100 backpropagation steps for mapping images into the latent space, and we chose $\lambda = 0.5$ in equation 15, after empirical experimentation. The experiment setup showed that backpropgating in the latent space was resource consuming, taking almost 1 minute per image for 200 steps. Also, giving more weight λ to one loss did not improve the resultant optimization. Figure 10 shows the distribution of the results. Visually, it is clear that the model is not able to differentiate the distribution of normal samples from abnormal, as they overlap.

As evaluation step, and for moving into a probability space as suggested in the methodology, we passed the obtained score to the simple logistic regression algorithm. Figure 12 shows the AUROC graph and ROC score results. A value of 0.58 means that the input features were not relevant enough, and the classifier was able to perform just better than random chance.

2) VAE-AD: Based on the performance of the 3D WGAN-GP architecture, we trained a 3DVAE using a similar setup of 3D convolutional layers and Upsampling3D. Figure 15 shows the architecture used for the encoder and the generation of the latent variables z.

Using the same 1722 normal nodules as for GANs, we trained the model for 100 epochs. The model has strong capacity of reconstruction of normal samples. Figure 16 shows some of the reconstructions, for both benign and malign nodules.

As for the VAE-AD, we used the resultant metrics for com-

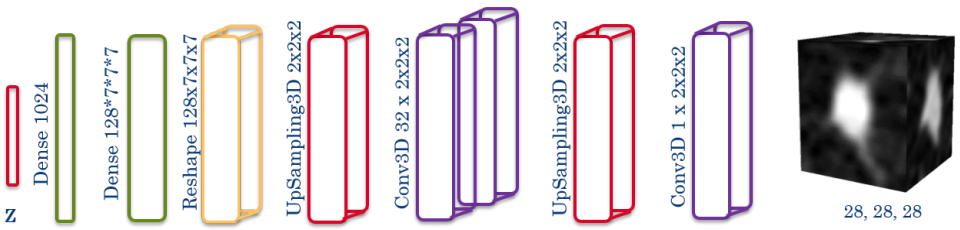


Figure 11: Trained 3D WGAN-GP architecture for Generator

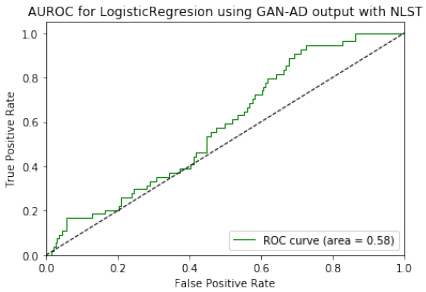


Figure 12: AUROC score for GAN-AD. The result implies the classifier was not able to discriminate any feature from normal to abnormal samples.

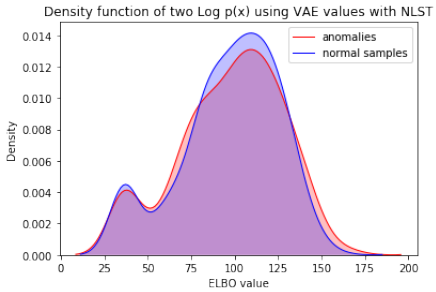


Figure 13: Density distributions of VAE-AD outputs for NLST dataset

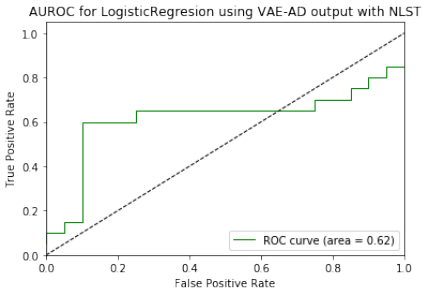


Figure 14: AUROC score for VAE-AD for NLST. The result implies the classifier performs better than random choice, but still, it does not have the capacity for discriminating normal from abnormal samples.

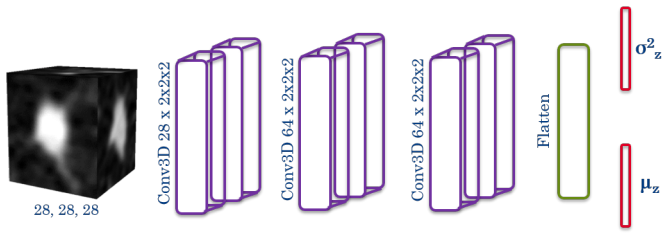


Figure 15: Trained 3D VAE architecture for Encoder

putation of the likelihood lower bound. We used the trained 3DVAE and passed new samples through it. We used 115 normal samples and 115 anomaly samples. The distribution of the values is shown in Fig. 13. Visually it is clear that the distributions overlap, not making a clear separation between the samples, but with little differences in the density in the middle.

For comparing the results, and for evaluation, we gave the scores as a single feature to the logistic regression classifier. Figure 14 shows the resulting AUROC. We can see that even if the classifier performs better than random, the given representation was not enough for make a clear distinction between normal and abnormal samples. Empirically, we noticed that increasing the number of samples could improve this score. We used samples from the validation split to perform more experiments, but the AUROC was not greater than 0.62. In presence of additional data, more experimentation could give better performance.

VII. Conclusion

This work defined a comparative Anomaly Detection framework for two state-of-the-art deep generative models. We used a metric based on likelihood estimation, and created an evaluation protocol for identification of anomalies. The concept of likelihood estimation is closer related to the VAE implementation. GAN computes a loss score that was taken as a likelihood value to be used in our logistic regression classifier, used as evaluation criteria.

For the first use case with MNIST, GAN approach is fragile and it is dependant on different hyper parameter tuning and number of samples to train. When evaluated in a probability space, using the logistic regression classifier, it did not show the expected performance as for the reference paper[42]. Results showed an AUROC of 0.66 when training with less than 10.000 samples of the normal class. Since our scope was imbalance and scarcity of samples, this was a realistic

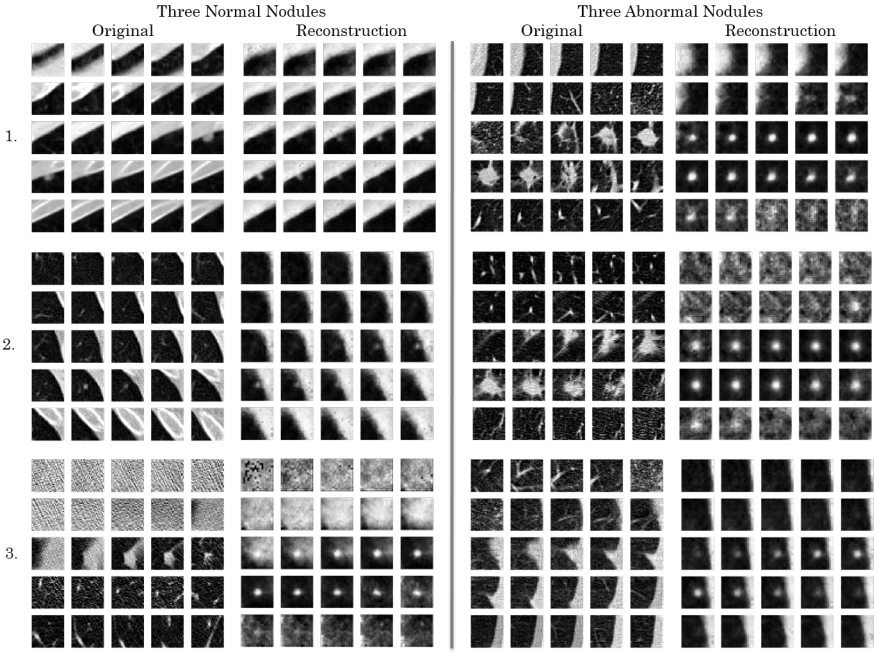


Figure 16: Reconstructed samples of VAE, using NLST nodules from both normal and abnormal type.

scenario for evaluating the model. For VAE approach, it is easy to train, not time consuming and the scores are obtained in a straight forward manner. The resultant AUROC shows potential in separation of abnormal samples with a value of 0.84.

The use case of lung cancer detection at a 3D image nodule level showed that neither of the generative models are not able to capture the feature complexity of the data. GAN approach evaluation showed a performance just better than random with a 0.58 AUROC. VAE presented a 0.62, which we consider not significantly relevant due to the importance of the abnormal samples.
Previous work[39] showed that GAN-AD did not perform well in a NLST 2D setup. We performed experiments over 3D architectures, expecting a richer model. However, we saw that deep generative models are still not robust enough in cases such as CT data of lung cancer at a nodule level.

VIII. Discussion

The results showed the deep generative models are not strong enough to capture high complexity of data and produce a probability density estimation that can differentiate normal from abnormal samples. For cancer detection at a nodule level we would like to find models with much capability of true positive detection. We consider, for the use case with NLST dataset, the samples do not content enough feature information for such a model. Looking at VAE's nodule reconstruction (Fig. 16), it seems as if the encoder vanishes certain characteristics seen in both normal and abnormal samples, and the decoder is not able to reconstruct them, which could explain why it is not able to distinguish the anomalies as expected.

During additional experiments for evaluation of the GAN's performance, the Inception Score[46][47] showed an important potential to give extra capacity to the GAN. This inception network requires a pre training using richer data. We encourage to perform experiments in this matter by using other datasets, and creating a basic inception network for evaluation. Additionally, the nodule detector[16], also capture information at a feature level that could leverage the generative model performance. However,for future work, due to the literature review, we would suggest to explore the Autoregressive Models for density estimation presented in the background section (See IV).

Acknowledgment

The author would like to thank the supervisors Dr. Vlado Menkovski and Dr. Dimitrios Mavroeidis for their advice, collaboration and patience. Philips Research Eindhoven for providing the infrastructure to perform the experiments. The colleagues, classmates and friends for the intense discussions, corrections and, especially, the immense support. And the family, for never stopping being proud, even without understanding what this was about and why it was worth so many sleepless nights.

References

[1] Jun Shi, Shichong Zhou, Xiao Liu, Qi Zhang, Minhua Lu, and Tianfu Wang. Stacked deep polynomial network based representation learning for tumor classification with small ultrasound image dataset. *Neurocomputing*, 194:87 – 94, 2016.

[2] Sotiris B Kotsiantis, I Zaharakis, and P Pintelas. Supervised machine learning: A review of classification

techniques. *Emerging artificial intelligence applications in computer engineering*, 160:3–24, 2007.

[3] Markus M. Breunig, Hans-Peter Kriegel, Raymond T. Ng, and Jörg Sander. Lof: Identifying density-based local outliers. *SIGMOD Rec.*, 29(2):93–104, May 2000.

[4] S. Papadimitriou, H. Kitagawa, P. B. Gibbons, and C. Faloutsos. Loci: fast outlier detection using the local correlation integral. In *Proceedings 19th International Conference on Data Engineering (Cat. No.03CH37405)*, pages 315–326, March 2003.

[5] Parikshit Gopalan, Vatsal Sharan, and Udi Wieder. Faster anomaly detection via matrix sketching. *CoRR*, abs/1804.03065, 2018.

[6] Yongqin Xian, Christoph H. Lampert, Bernt Schiele, and Zeynep Akata. Zero-shot learning - A comprehensive evaluation of the good, the bad and the ugly. *CoRR*, abs/1707.00600, 2017.

[7] Mohammad Sabokrou, Mohsen Fayyaz, Mahmood Fathy, Zahra Moayed, and Reinhard Klette. Deep-anomaly: Fully convolutional neural network for fast anomaly detection in crowded scenes. 02 2018.

[8] Longin Jan Latecki, Aleksandar Lazarevic, and Dragoljub Pokrajac. Outlier detection with kernel density functions. In Petra Perner, editor, *Machine Learning and Data Mining in Pattern Recognition*, pages 61–75, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.

[9] Wei Wang, Baoju Zhang, Dan Wang, Yu Jiang, Shan Qin, and Lei Xue. Anomaly detection based on probability density function with kullbackleibler divergence. *Signal Processing*, 126:12 – 17, 2016. Signal Processing for Heterogeneous Sensor Networks.

[10] G. Papamakarios, T. Pavlakou, and I. Murray. Masked Autoregressive Flow for Density Estimation. *ArXiv e-prints*, May 2017.

[11] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg, 2006.

[12] Peter Sunehag, Jochen Trumpf, S. V. N. Vishwanathan, and Nicol N. Schraudolph. Variable Metric Stochastic Approximation Theory. In David van Dyk and Max Welling, editors, *Proc. 12^{th} Intl. Conf. Artificial Intelligence and Statistics (AIstats)*, volume 5 of *Workshop and Conference Proceedings*, pages 560–566, Clearwater Beach, Florida, 2009.

[13] Holger R. Roth, Le Lu, Jiamin Liu, Jianhua Yao, Ari Seff, Kevin M. Cherry, Lauren Kim, and Ronald M. Summers. Improving computer-aided detection using convolutional neural networks and random view aggregation. *CoRR*, abs/1505.03046, 2015.

[14] H. Greenspan, B. van Ginneken, and R. M. Summers. Guest editorial deep learning in medical imaging: Overview and future promise of an exciting new technique. *IEEE Transactions on Medical Imaging*, 35(5):1153–1159, May 2016.

[15] Matthias Hammon, Peter Dankerl, Alexey Tsymbal, Michael Wels, Michael Kelm, Matthias May, Michael Suehling, Michael Uder, and Alexander Cavallaro. Automatic detection of lytic and blastic thoracolumbar spine metastases on computed tomography. *US National Library of Medicine National Institutes of Health*, https://www.ncbi.nlm.nih.gov/pubmed/23397381, 2013.

[16] Stojan Trajanovski, Dimitrios Mavroeidis, Christine Leon Swisher, Binyam Gebrekidan Gebre, Bas Veeling, Rafael Wiemker, Tobias Klinder, Amir Tahmasebi, Shawn M. Regis, Christoph Wald, Brady J. McKee, Heber MacMahon, and Homer Pien. Towards radiologist-level cancer risk assessment in CT lung screening using deep learning. *CoRR*, abs/1804.01901, 2018.

[17] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, David Warde-Farley Bing Xu, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Nets. 2014.

[18] D. P Kingma and M. Welling. Auto-Encoding Variational Bayes. *ArXiv e-prints*, December 2013.

[19] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew P. Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, and Wenzhe Shi. Photo-realistic single image super-resolution using a generative adversarial network. *CoRR*, abs/1609.04802, 2016.

[20] Raymond Yeh, Chen Chen, Teck Yian Lim, Mark Hasegawa-Johnson, and Minh N. Do. Semantic Image Inpainting with Deep Generative Models. *arXiv:1607.07539*, 2016.

[21] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *CoRR*, abs/1703.10593, 2017.

[22] Liqian Ma, Xu Jia, Qianru Sun, Bernt Schiele, Tinne Tuytelaars, and Luc Van Gool. Pose guided person image generation. *CoRR*, abs/1705.09368, 2017.

[23] Ian J. Goodfellow. NIPS 2016 tutorial: Generative adversarial networks. *CoRR*, 2017.

[24] Adam Roberts, Jesse Engel, and Douglas Eck, editors. *Hierarchical Variational Autoencoders for Music*, 2017.

[25] Hugo Larochelle and Iain Murray. The neural autoregressive distribution estimator. In Geoffrey Gordon, David Dunson, and Miroslav Dudk, editors, *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pages 29–37, Fort Lauderdale, FL, USA, 11–13 Apr 2011. PMLR.

[26] B. Uria, I. Murray, and H. Larochelle. RNADE: The real-valued neural autoregressive density-estimator. *ArXiv e-prints*, June 2013.

[27] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real NVP. *CoRR*, abs/1605.08803, 2016.

[28] Diederik P Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling. Improved variational inference with inverse autoregressive flow. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon,

and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 4743–4751. Curran Associates, Inc., 2016.

[29] Simon Kohl, David Bonekamp, Heinz-Peter Schlemmer, Kaneschka Yaqubi, Markus Hohenfellner, Boris Hadaschik, Jan-Philipp Radtke, and Klaus Maier-Hein. Adversarial Networks for the Detection of Aggressive Prostate Cancer. *arXiv:1702.08014*, 2017.

[30] Xin Yi and Paul Babyn. Sharpness-aware Low dose CT denoising using conditional generative adversarial network. *arXiv:1708.06453*, 2017.

[31] Mina Rezaei, Konstantin Harmuth, Willi Gierke, Thomas Kellermeier, Martin Fischer, Haojin Yang, and Christoph Meinel. Conditional Adversarial Network for Semantic Segmentation of Brain Tumor. *arXiv:1708.05227*, 2017.

[32] John T. Guibas, Tejpal S. Virdi, and Peter S. Li. Synthetic Medical Images from Dual Generative Adversarial Networks. *arXiv:1709.01872*, 2017.

[33] Dong Nie, Roger Trullo, Caroline Petitjean, Su Ruan, and Dinggang Shen. Medical Image Synthesis with Context-Aware Generative Adversarial Networks. *arXiv:1612.05362*, 2016.

[34] Yuusuke Kataoka, Takashi Matsubara, and Kuniaki Uehara. Image generation using generative adversarial networks and attention mechanism. 2016.

[35] Maayan Frid-Adar, Idit Diamant, Eyal Klang, Michal Amitai, Jacob Goldberger, and Hayit Greenspan. Gan-based synthetic medical image augmentation for increased CNN performance in liver lesion classification. *CoRR*, abs/1803.01229, 2018.

[36] Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques - Adaptive Computation and Machine Learning*. The MIT Press, 2009.

[37] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *CoRR*, abs/1511.06434, 2015.

[38] Jiajun Wu, Chengkai Zhang, Tianfan Xue, William T. Freeman, and Joshua B. Tenenbaum. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. *CoRR*, abs/1610.07584, 2016.

[39] Rodrigo Mendoza. Anomaly Detection with generative models. Master's thesis, Eindhoven University of Technology, 2018.

[40] Jonathan Hui. GAN Series. https://medium.com/@jonathan_hui/gan-gan-series-2d279f906e7b. Accessed: [30/07/2018].

[41] C. Doersch. Tutorial on Variational Autoencoders. *ArXiv e-prints*, June 2016.

[42] Thomas Schlegl, Philipp Seeböck, Sebastian M. Waldstein, Ursula Schmidt-Erfurth, and Georg Langs. Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. *CoRR*, abs/1703.05921, 2017.

[43] Christoph Baur, Benedikt Wiestler, Shadi Albarqouni, and Nassir Navab. Deep autoencoding models for unsupervised anomaly segmentation in brain MR images. *CoRR*, abs/1804.04488, 2018.

[44] Emily L. Denton, Sam Gross, and Rob Fergus. Semi-supervised learning with context-conditional generative adversarial networks. *CoRR*, abs/1611.06430, 2016.

[45] Ishaan Gulrajani, Faruk Ahmed, Martín Arjovsky, Vincent Dumoulin, and Aaron C. Courville. Improved training of wasserstein gans. *CoRR*, abs/1704.00028, 2017.

[46] Tim Salimans, Ian J. Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *CoRR*, abs/1606.03498, 2016.

[47] S. Barratt and R. Sharma. A Note on the Inception Score. *ArXiv e-prints*, January 2018.