FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Lung computed tomography image synthesis using generative adversarial networks

José Miguel Ferreira Mendes

Mestrado Integrado em Engenharia Informática e Computação

Supervisor: Hélder Filipe Pinto de Oliveira

Co-Supervisor: Tânia Maria Pereira Lopes

Co-Supervisor: António Manuel Trigueiros da Silva Cunha

July 31, 2020

Lung computed tomography image synthesis using generative adversarial networks

José Miguel Ferreira Mendes

Mestrado Integrado em Engenharia Informática e Computação

July 31, 2020

Resumo

A quantidade de dados desempenha um papel crítico no sucesso ou fracasso de aplicações de aprendizagem de máquina. Este problema é especialmente prevalente em domínios onde a aquisição de dados relevantes, de alta qualidade e heterogéneos é complicada por factores externos.

Recentemente, a engenharia biomédica tem sido apontada como um potencial candidato a investigação em aplicações de aprendizagem de máquina, com o objetivo de detectar ou diagnosticar diferentes doenças. No entanto, taxas, questões de privacidade e a grande quantidade de tempo e esforço necessários para enviar um protocolo aos comitês éticos para obter aprovação tornam os dados clínicos extremamente difíceis de serem obtidos.

Como tentativa de resolver este problema, os modelos generativos ganharam recentemente um maior interesse na comunidade de visão computacional, devido à sua capacidade de aumentar a quantidade de dados, gerando novas amostras de alta qualidade a partir do conjunto de dados inicial. Três tipos de modelos generativos dominaram recentemente a geração de imagens: modelos *auto-regressive*, *variational autoencoders* e *generative adversarial networks*. A maioria das aplicações de imagem médica utilizam *generative adversarial networks* devido à sua capacidade de gerar imagens de alta qualidade com base em mapas de anotações semânticas.

Tendo isto em consideração, o objetivo deste trabalho é sintetizar imagens pulmonares artificiais a partir de anotações posicionais e semânticas, usando bancos de dados de exames reais e técnicas do estado da arte. Para isso, primeiro exploramos três possíveis algoritmos de segmentação baseados em *threshold* para extrair os pulmões dos exames de tomografia computadorizada, de forma a criar imagens semelhantes às que seriam usadas noutros modelos de *deep learning*. Posteriormente, exploramos a *framework pix2pix* para gerar imagens de pulmões a partir de mapas de segmentação e implementamos uma versão alternativa que permite o uso adicional de anotações semânticas, que são especialmente comuns no domínio médico.

Adicionalmente, contribuímos para a investigação da avaliação de modelos generativos, facilitando a interpretação da *Fréchet Inception Distance* através de uma representação visual das distribuições reais e sintéticas reduzidas espacialmente e do cálculo de *domain-specific Fréchet Inception Distance*. Os nossos resultados demonstram que a *Fréchet Inception Distance* pode ter resultados inconsistentes quando usada no domínio médico, onde o domínio de imagens são muito diferentes das encontrados na base de dados do ImageNet.

Abstract

Data sample size plays a critical role in the success or failure of machine learning applications. This problem is especially prevalent in domains where acquiring relevant, high-quality, heterogeneous data is complicated by external factors.

In recent years, biomedical engineering has been targeted as a potential research candidate for machine learning applications, with the purpose of detecting or diagnosing various diseases. However, fees, privacy issues and the large amount of time and effort to submit a protocol to ethical committees to get approval, make clinical data extremely difficult to obtain.

As an attempt to solve this issue, generative models, have recently gained growing interest in the computer vision community, due to their ability to increase dataset size by generating new high-quality samples from the initial dataset. Three types of generative models have recently dominated image generation: auto-regressive models, variational autoencoders and generative adversarial networks. The majority of medical imaging applications make use of generative adversarial networks due to their ability to generate images of high-quality based on semantic label maps.

With this in mind, the goal of this work is to synthesize artificial lung images from corresponding positional and semantic annotations using databases of real exams and state of the art generative modeling techniques. To achieve, we first explore three possible threshold-based segmentation algorithms to extract the lungs from the full Computed Tomography exams, in order to create images of the same modality that would be used in other deep learning models. Subsequently, we explore the *pix2pix* framework to generate lung images from the segmentation maps, and implement a modified version that enables the additional use of semantic labels, which are especially common in the medical domain.

Additionally, we contribute to the ongoing research of generative model evaluation by facilitating the interpretation of the Fréchet Inception Distance by means of a visual representation of spatially reduced real and generated distributions and the computation of domain-specific Fréchet Inception Distance. Our results show that the original Fréchet Inception Distance may show inconsistent results when used in the medical domain, where the domain of images are much different from the ones found in the ImageNet dataset.

Keywords: Deep Learning, Generative Adversarial Networks, Medical Imaging

Acknowledgements

I would like to thank my supervisors, Professors Hélder Oliveira, Tânia Pereira and António Cunha for the guidance and knowledge shared during this work.

My family, for the unconditional love and support.

My friends, for all the fun times, even when being together was not possible.

My girlfriend, for the patience and for encouraging me to be and do better.

José Mendes

This work is financed by the ERDF – European Regional Development Fund through the Operational Programme for Competitiveness and Internationalisation - COMPETE 2020 Programme and by National Funds through the Portuguese funding agency, FCT - Fundação para a Ciência e a Tecnologia within project POCI-01-0145-FEDER-030263.

Contents

List of Figures

List of Tables

Abbreviations

ACGAN	Auxiliary Classifier Generative Adversatial Network
AE	Autoencoder
CCGAN	Conditional Generative Adversarial Network with Classifier
CGAN	Conditional Generative Adversarial Networks
CNN	Convolutional Neural Network
CT	Computed Tomography
D	Discriminator
FID	Fréchet Inception Distance
G	Generator
GAN	Generative Adversatial Network
LIDC-IDRI	Lung Image Database Consortium image collection
MSE	Mean Square Error
NSCLC	Non Small Cell Lung Cancer
PPV	Positive Predictive Value
ReLU	Rectified Linear Unit
SCLC	Small Cell Lung Cancer
SSIM	Structural Similarity Index
VAE	Variational Autoencoder
VTT	Visual Turing Test

Chapter 1

Introduction

With approximately 9.6 million deaths per year, cancer remains one of the leading causes of death in the world. Lung cancer remains the most common cause of cancer death, accounting for 18.4% of annual cancer deaths [1].

Low dose computed tomography (CT) scans are widely used by radiologists for lung cancer screening and early diagnosis, however, radiologists face many challenges in analyzing lung nodules that often lead to misdiagnosis. One of those challenges is the variability in the appearance of lung nodules: shape, size, texture and other characteristics vary regardless of being benign or malignant nodules.

As such, there is significant effort in pushing Computer-Aided Diagnosis (CAD) to reliably identify and characterize lung nodules, in hope to improve lung cancer diagnosis. Recent advances by Hussein et al. [2] have shown promising results in this regard by using supervised learning to classify various characteristics of nodules (calcification, lobulation, sphericity, speculation, margin, and texture). However, with only 635 benign and 509 malignant nodules, it's noted in that study and is the main topic of [3] that high-quality medical data in large numbers is extremely difficult to acquire. Moreover, annotated medical data requires contributions by medical experts, which is expensive and time consuming and therefore, many available datasets lack annotations that would potentially improve the quality of machine learning applications. Ultimately, fees, privacy issues, and the large amount of time and effort to submit a protocol to ethical committees to get approval, make clinical data extremely difficult to obtain.

Recently, generative models, have gained popularity as a potential data augmentation technique to allow for additional synthetic data to be sampled and used to augment the real training data [4]. Three types of generative models have dominated image generation in the last few years: auto-regressive models, variational autoencoders (VAE) and generative adversarial networks (GAN). Regressive models are mostly used to complete partially obfuscated images and are, currently, very inefficient to train and to generate images. VAEs are faster to train but produce images with the lowest quality of the three types. Finally, GANs take a game theory approach to

synthesize images, and currently produce state of the art images in terms of quality, depite being very unstable to train. Proposed by Goodfellow et. al in 2014 [5], GANs consist of two networks, pitched against each other in an adversarial process where one generates new, fake data (Generator Network, **G**) and the other characterizes images as real or fake (Discriminator Network, **D**). This process creates a competition where the two models are led to improve each other: **G** will, over time, generate images of increased realism, while **D** will improve its capability of distinguishing real and fake data. Many variations of the original GAN have since been proposed [6, 7, 8, 9, 10] with interesting applications in generating images of one style from another style (image-to-image translation) [11] or image inpainting [12].

In medical imaging, the majority of studies employ image-to-image translation to generate labels into segmentations, segmentations to images or medical cross modality generation [13, 14, 15, 16]. For example, Salman Uh Dar et al. [16], trained a conditional GAN to generate the missing contrast in a Multi-Contrast brain MRI. This type of generation makes use of pairs of images to train a GAN to generate the "missing pair" in a new, not seen before, image. Fewer cases involve the generation of images with the starting point of a random **z** noise vector, the original architecture of GAN. OMaayan Frid-Adar et al. [17], used a Deep Convolutional GAN (DCGAN) to generate liver lesions of three labeled types (cysts, metastases, hemangiomas) in order to augment the initial dataset size of a liver lesion classifier. Using a DCGAN to augment data resulted in an improvement in the accuracy of the classifier, proposing that augmentation through generative models is a viable approach.

1.1 Motivation

Dealing with small, unlabeled data has been a recurring issue for machine learning applications in the medical domain. As a result, many models fall short of the expected results since, typically, learning algorithms require large amounts of annotated data.

Large quantities of medical data are often barred behind large fees, privacy issues or slow bureaucratic processes and annotated data requires the availability of experienced radiologists which is both expensive and time consuming.

Due to the high incidence of lung related pathologies, there has been a great push for the use of deep learning techniques to automatically detect and diagnose these pathologies, which are highly dependant on the amount of available data. Traditionally, deep learning models employ data-augmentation techniques such as rotation, translation and cropping to produce additional training samples, however, these samples are highly correlated with the existing samples, since they are directly derived from them. Generative models have proven useful to increase data by generating, new, realistic data that is indistinguishable from real data.

1.2 Goals and Contributions

The purpose of this work is to research and implement a reliable method to synthesize lung images from both object position annotations and semantic characteristics, in order to help reduce the impact of data scarcity in the clinical use of lung cancer classification models.

The contributions of this work are the following:

- Implement a generative model capable of generating lung images from lung position annotations;
- Implement a generative model capable of generating lung images from lung position annotations and semantic annotations;
- Provide a deeper exploration of the Fréchet Inception Distance metric by using domain-specific encoders;

1.3 Structure

The remainder of this report is organized as follows: chapter 2 presents an overview of the anatomy of the lung, followed by information regarding computed tomography and its uses and a brief overview of lung cancer and pulmonary nodules. Chapter 3 explores a literature review of the most commonly used generative model architectures, evaluation metrics, current lines of research and uses in medical imaging. Chapter 4 details the datasets, segmentation methods, generative models and evaluation metrics used in this work, as well of the results of the generated samples. Chapter 5 concludes the dissertation with an overview of the accomplished goals and contributions and suggestions of possible future work.

Chapter 2

Medical Background

2.1 Lung Anatomy

The lungs are a paired, cone-shaped organ lying in the thoracic cavity, connected to the trachea by the left and right bronchi[18]. The diaphragm, a flat, dome-shaped muscle is located at the base of the lungs and thoracic cavity. The lungs are enclosed by the pleurae, which are attached to the mediastinum. The left lung, occupies a smaller volume than the right, which is shorter and wider. The indentation on the surface of the left lung, called cardiac notch, allows space for the heart. The costal surface of the lung borders the ribs and the mediastinal surfaces faces the midline [19] (Figure 2.1).

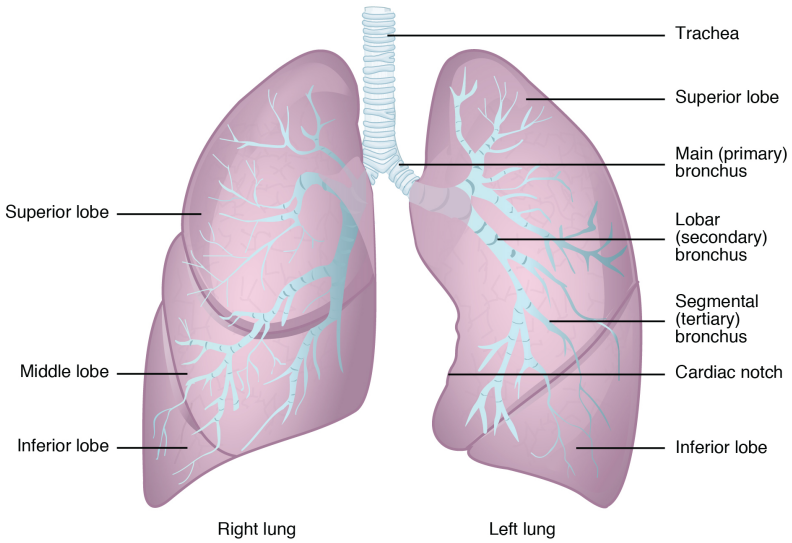

Figure 2.1: Lung anatomy [18].

The lungs are composed of a set of lobes, separated by fissures. The right lung contains three lobes: superior, middle and inferior lobe. While the left lung consists of only two: superior and inferior lobe. Each lobe can then be divided into several bronchopulmonary segments. Ten for the right lung: apical, anterior and medial, located in the right upper lobe. Medial and lateral in the

middle lobe. Superior, medial, anterior, lateral, and posterior in the lower lobe. Eight for the left lung: apicoposterior, anterior, superior lingula, and inferior lingula in the left upper lobe, superior, anteromedial, lateral, and posterior in left lower lobe (Figure 2.2).

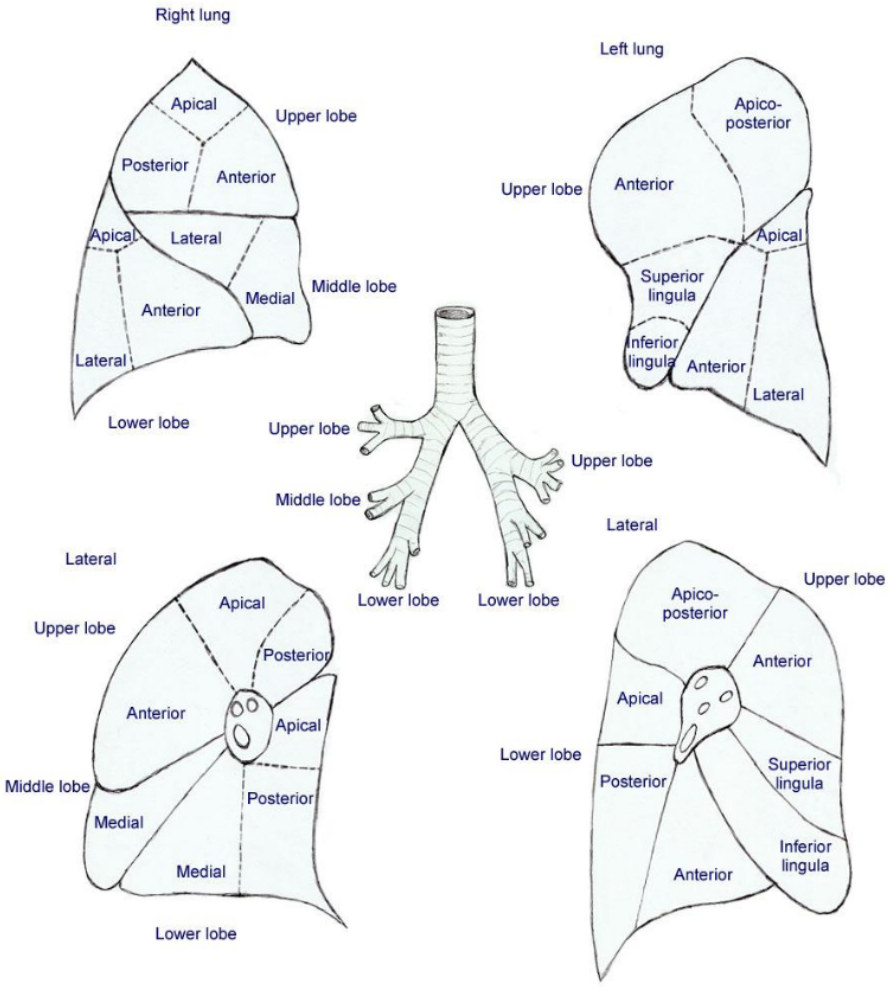

Figure 2.2: Bronchopulmonary segments [19].

2.2 Computed Tomography

Developed in 1972 by Godfrey Hounsfield and Allan McLeod Cormack, the computed tomography (CT) is a non-invasive imaging technique that combines multiple X-ray images, based on the principle that different tissues reflect and absorb X-rays at different levels. To capture theses images, a patient lies in a motorized, ring-shaped platform while a computerized axial tomography (CAT) scanner rotates 360 degrees, taking X-ray images. These images are then combined into a two-dimensional view of the scanned area named "slice", which in turn are used to reconstruct a 3D volume CT image. Figure 2.3 shows an example of one of the mentioned slices, on a thoracic CT scan.

Figure 2.3: An example thoracic CT image [20].

CT scans can be used for a variety of diagnostic procedures such as identification of tumors, cysts or infections and have the advantage of being painless and low-risk since the amount of radiation a patient is subjected during the CT scan is minimal. The most common CT scan related problem is an adverse allergic reaction when, occasionally, contrast materials are administered intravenously or through other routes, in order to improve image quality related with structural relationships of the spine, the spinal cord, and its nerves, which can result in allergic reactions.

2.3 Lung Cancer

According to the National Cancer Institute[1], cancer is a term for diseases in which malignant cells grow and divide without control. Moreover, the cells can spread into different regions of the body through blood and lymph systems, invading other tissues.

Lung cancer can be classified in two different categories: Small Cell Lung Cancer (SCLC) and Non Small Cell Lung Cancer (NSCLC), with the latter being the most common, accounting for around 84% of all cases, according to the American Cancer Society [21]. Different sub-types of cancer fall into the NSCLC category, each starting in a different type of lung cell, such as:

- **Adenocarcinoma**: most common sub-type of cancer, starts in mucus cells;

[1] https://www.cancer.gov/

- **Squamous cell carcinoma**: cells that line airways of lungs;

- **Large cell (undifferentiated) carcinoma**: can appear in any part of the lung.

After its initial appearance, cancer can spread to other parts of the body, making treatment much more difficult. The SEER[2] database [22], tracks 5-year survival rates in the United States for both NSCLC and SCLC, based on how far the cancer has spread. **Localized** indicates no sign of cancer outside its initial tissue, **regional** implies cancer has spread to nearby structures and **distant** means that the cancer has spread to distant parts of the body. For lung cancer the 5-year survival rates are 57.4%, 30.8% and 5.2% for **localized**, **regional** and **distant** respectively, indicating that early diagnosis can lead to a greater chance of surviving.

2.4 Pulmonary Nodules

A pulmonary nodule is a small, round or oval-shaped growth in the lung that is smaller than 3 cm in diameter and can be either benign or malignant (Figure 2.4). Often named a "spot on the lung" or a "coin lesion", pulmonary nodules vary in shape and size, with larger ones having higher likelihood of being malignant. Structures larger than 3cm in diameter are named pulmonary masses due to their much higher likelihood of being malignant.

A nodule can have a number of different origins such as ongoing infections, lesions from past infections or from benign/malignant tumors. Management of a detected pulmonary nodule should aim at identifying malignancy as fast as possible, since early identification of malignant lung nodules is an important factor in increasing survival chance, leading to a potential 60-80% 5-year survival rate, in stage I NSCLC [23].

[2]Surveillance, Epidemiology, and End Results

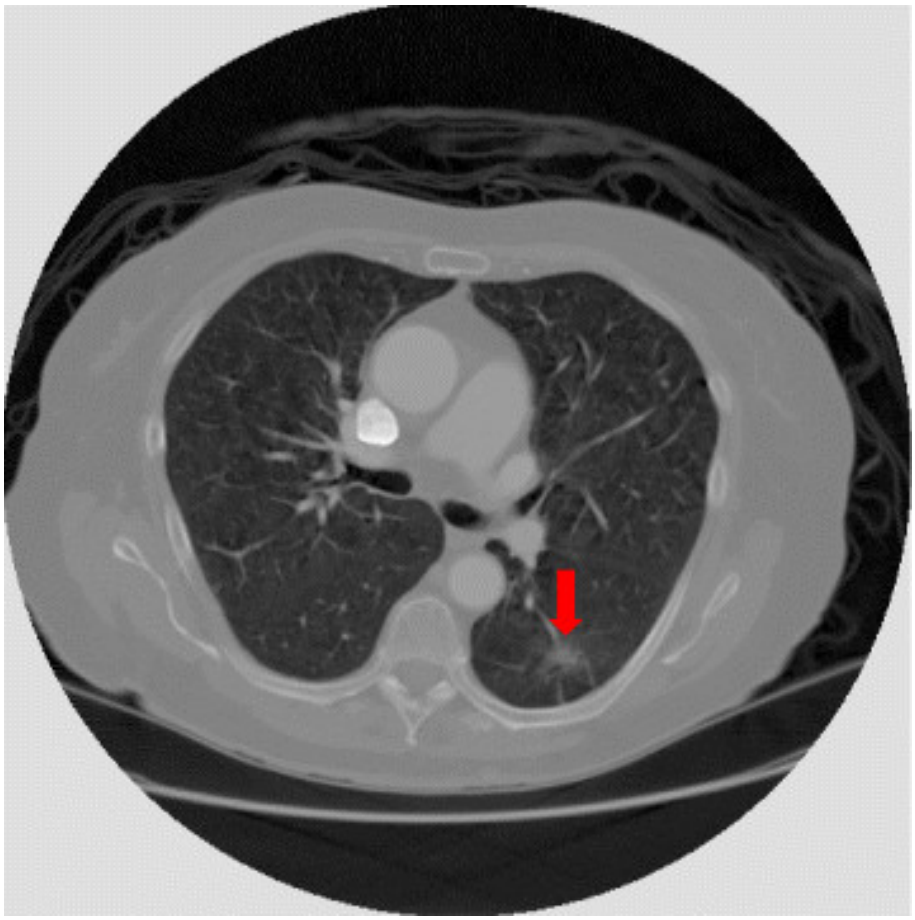

Figure 2.4: CT image presenting a pulmonary nodule (red arrow). [24]

According to the results of the National Lung Screening Trial, a United States based lung cancer screening trial [25], the diameter of the nodule played a significant part in the malignancy results of screened patients, as shown in table 2.1.

Diameter	PPV
4-6mm	0.5
7-10mm	1.7
11-20mm	11.9
21-30mm	29.7
>30	41.3

Table 2.1: The positive predictive value (PPV) is defined as the proportion of patients with confirmed lung cancer among those with a positive result on screening whose lung-cancer status was known [25].

Additionally, when assessing the likelihood of malignancy, other nodule characteristics are typically considered, when examining a CT exam, such as: calcificaton, spiculation and ragged margins, intranodular fat, among others [23].

2.5 Emphysema

Emphysema is a lung condition, part of a larger group of diseases known as chronic obstructive pulmonary disease (COPD), that occurs more commonly in smokers or people who regularly breathe in irritants, that causes shortness of breath. Emphysema causes the air sacs in the lungs to be damaged and, over time, the inner walls of the air sacs weaken and rupture, which reduces the surface area of the lungs and, in turn, the amount of oxygen that reaches the bloodstream[26]. Studies have shown that the presence of emphysema may increase the risk of lung cancer by 3.8-fold[27] and as such, the diagnosis of this disease could lead to early cancer diagnosis and reduce the risk of mortality.

Emphysema is a condition that over time becomes more visible in lung CT imaging. An example of a lung CT with emphysema can be seen in Figure 2.5

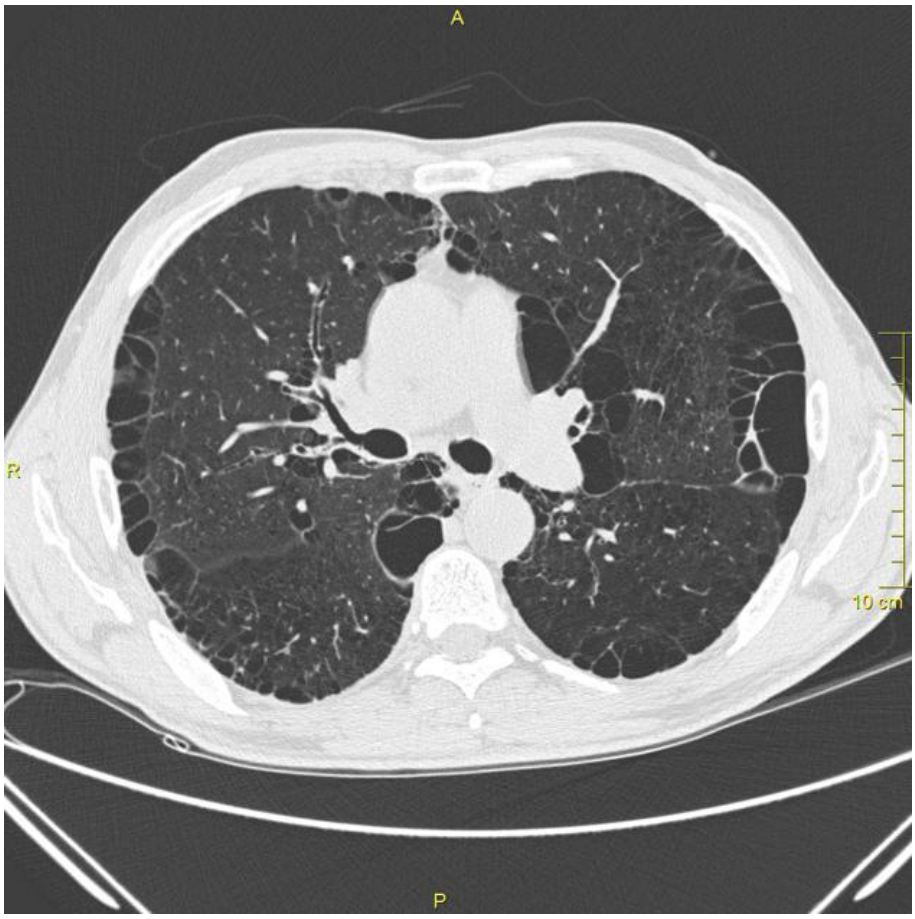

Figure 2.5: CT scan of a lung with emphysema. [28]

Chapter 3

Generative Models - Literature Review

Generative models are a subset of unsupervised learning where given some training data, they generate new samples from that same distribution. To achieve this, generative models perform density estimation, where from an unknown training data distribution p_{data}, they return an estimate of that distribution p_{model}. There are two ways to achieve this result: **explicit density estimation**, where generative models explicitly define and solve for p_{model}, or **implicit density estimation** where the model samples directly from p_{model} without explicitly defining it. Figure 3.1 shows the taxonomy of various generative models.

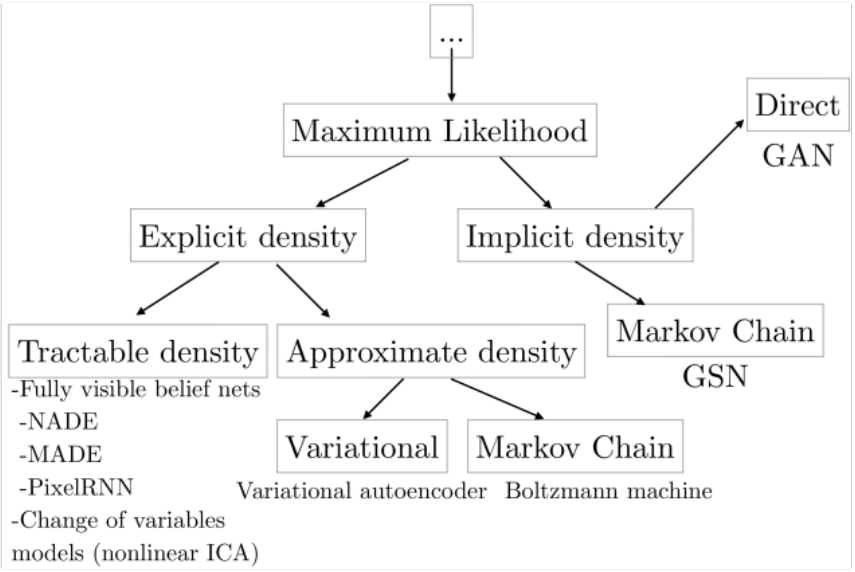

Figure 3.1: Taxonomy of Generative Models. [29]

3.1 Auto-Regressive Models

Auto-regressive generative models implicitly define a distribution over sequences using the Chain Rule for Conditional Probability, where in each step, the distribution of the next sequence element is predicted given the previous elements. PixelRNN and PixelCNN[30] are two examples of auto-regressive models where the first uses a recurrent neural network and the second a convolutional neural network, the two main architectures for auto-regressive generative models.

Pixel recurrent neural networks (PixelRNN), generate each pixel of an image in a sequence, where each one is dependant on all previously generated pixels. This technique employs a simple and stable training process to produce very sharp images, sacrificing efficiency during sampling.

Formally, the goal is to assign a probability $p(x)$ to each image x of size $N \times N$ pixels. This probability is the product of conditional probabilities for each pixel:

$$p(x) = \prod_{i=1}^{n^2} p(x_i | x_1, ..., x_{i-1}) \tag{3.1}$$

where the probability of pixel x_i, $p(x_i)$, is conditioned on the probability of previously generated pixels, for each color channel (red, green and blue).

Two PixelRNNs were designed, each composed of up to twelve, fast two-dimensional Long Short-Term Memory (LSTM) layers: *Row LSTM*, applies each convolution along each row, capturing a triangular area above the pixel, as shown in Figure 3.2.

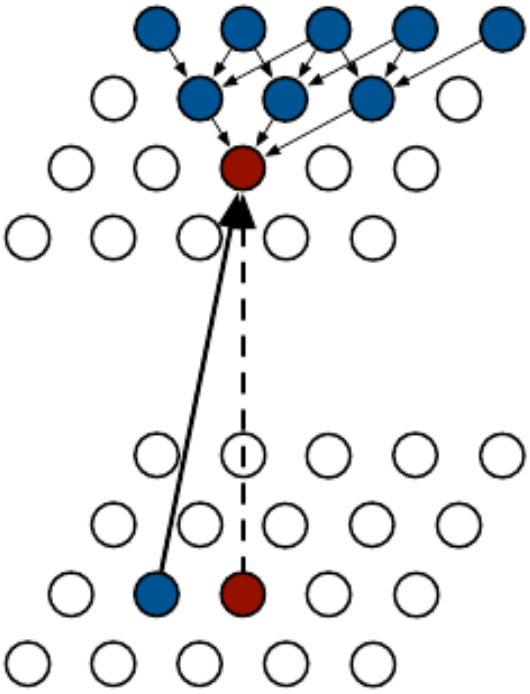

Figure 3.2: Row LSTM. [30]

Since this process does not consider all previously generated pixels, an additional architecture was proposed, *Diagonal BiLSTM*, where convolutions are applied along the diagonal of an image (Figure 3.3).

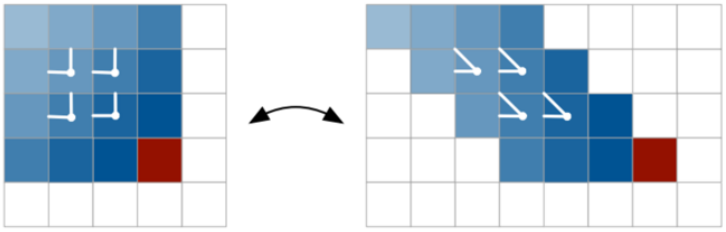

Figure 3.3: Diagonal BiLSTM. [30]

Both row LSTM and diagonal BiLSTM have an unbounded dependency range, meaning that, during training, each pixel could potentially be using information from every pixel before itself, which can be computationally inefficient. To help overcome this issue, the PixelCNN uses convolutional layers to capture a bounded receptive field and compute features for all pixels at once, which lowers training time considerably. However, image generation is still sequential in PixelCNN and since the receptive field will be smaller, the quality of the image may be affected.

Since auto-regressive models are based on context to generate pixels, they are typically used to complete partially occluded images, making them the model that achieves better results in this type of problem. Figure 3.4 showcases results of PixelRNN in this type of problem.

Figure 3.4: PixelRNN results when completing an image. [30]

3.2 Variational Autoencoders

An Autoencoder is an unsupervised learning technique that apply Neural Networks to the task of representation learning. They consist of a network that encodes/decodes training data to/from a bottleneck region. The general attributes in the bottleneck are referred to as latent attributes of the input data and compose a latent space. Figure 3.5 illustrates these concepts.

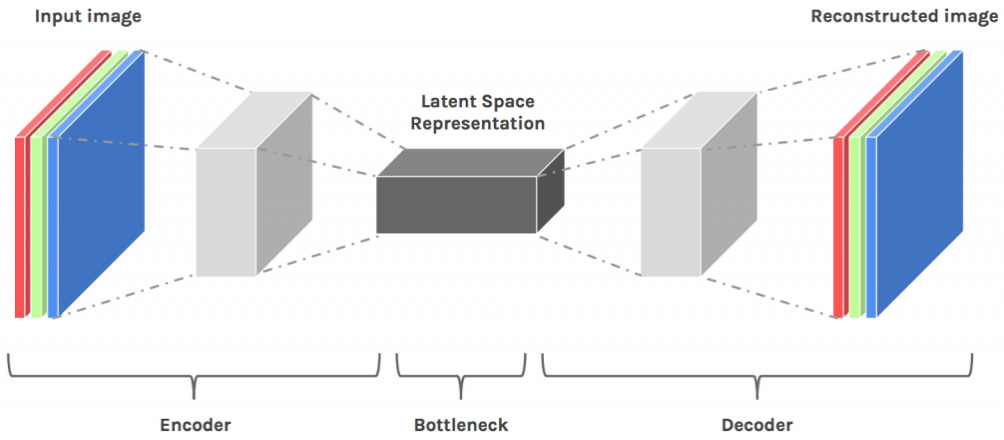

Figure 3.5: Autoencoder architecture. [31]

In terms of data generation, Autoencoder are limited since the encoder outputs a single value for each encoding dimension, producing a non-continuous latent space, which doesn't allow interpolation. To overcome this, Variational Autoencoders (VAE)[32], employ a probabilistic spin on Autoenconders and enable the model to generate new data: instead of encoding an input as a single point, we encode it as a distribution over the latent space. The model is then trained as follows:

- Encode input as distribution over the latent space;
- Point from the latent space is sampled from that distribution;
- Decode sample point and compute reconstruction error;
- Backpropagate error through the network;

Formally, we can define the "probabilistic decoder" as $p(x|z)$, that is, the distribution of the decoded variable given the encoded one, whereas the "probabilistic encoder" is defined by $p(z|x)$, that describes the distribution of the encoded variable given the decoded one and can be calculated by the Bayes' rule:

$$p(z|x) = \frac{p(x|z)p(z)}{p(x)} = \frac{p(x|z)p(z)}{\int p(x|z)p(z)dz} \tag{3.2}$$

The integral in the denominator implies evaluating all possible configurations of latent variables for each candidate z which is intractable. To overcome this issue, $p(z|x)$ is instead approximated through a distribution $q_\lambda(z|x)$, where λ refers to the parameters of the type of distribution, for example, $\lambda = (\mu, \sigma^2)$ in a Gaussian distribution. The Kullback-Leibler (KL) divergence can then be used to evaluate how well the chosen distribution approximates $p(z|x)$, and the goal is to obtain the λ of distribution q that minimize the KL divergence:

$$argmin_\lambda KL(q_\lambda(z|x)||p(z|x)) = argmin_\lambda \mathbb{E}_q[\log q_\lambda(z|x)] - \mathbb{E}_q[\log p(x,z)] + \log p(x) \tag{3.3}$$

This minimization is also intractable since it involves computing $p(x)$. Since the KL divergence is always equal or greater than zero and $\log p(x)$ does not depend of q, we can instead maximize the following expression, known as evidence lower bound (ELBO):

$$ELBO = \mathbb{E}_q[\log p(x,z)] - \mathbb{E}[\log q_\lambda(z|x)] \tag{3.4}$$

The network is then trained to minimize the loss function shown in equation 3.5, where $\mathscr{L}(x,x_1)$ indicates the previously mentioned reconstruction loss.

$$arg\min_{\lambda} \mathscr{L}(x,x_1) + ELBO \tag{3.5}$$

3.3 Generative Adversarial Networks

Generative Adversarial Networks (GANs), first proposed in [5], have since gained increased popularity as a data generation technique. Figure 3.6 shows the cumulative number of uniquely named GAN papers released monthly since they were first proposed in 2014, to September 2018.

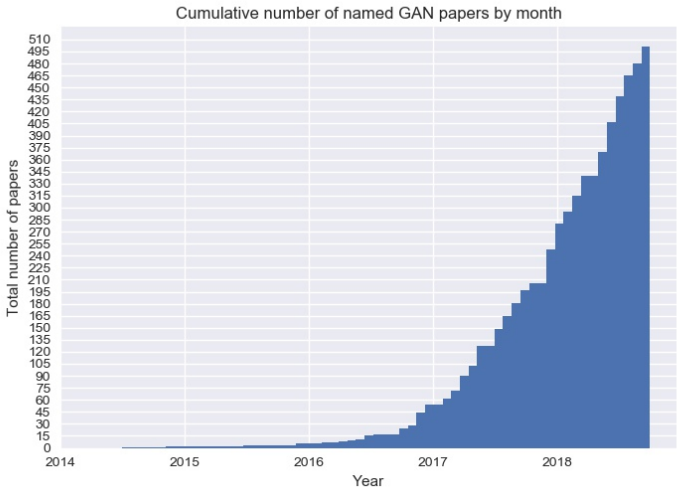

Figure 3.6: Cumulative number of named GAN papers by month. [33]

GANs pair two neural networks (tipically, convolutional neural networks) in an adversarial, two-player game where player one, the generator, has the goal of tricking the other player by generating realistic samples. Player two, the discriminator, has to guess if an image it receives as input is real or fake. In other words, the discriminator's goal is to correctly guess if the images originate from the original training dataset or if they were synthesized by the generator. This competition leads the two models to improve each other: the generator will, over time, sample images with increased realistic features, while the discriminator will improve its ability to distinguish real from fake images. The resulting system creates a scenario where if a discriminator learns the data features and is accurate at identifying real images and a generator is still able to trick it, then the

generated images can be considered of high-quality and could be part of the original dataset. This interaction is summarized in Figure 3.7.

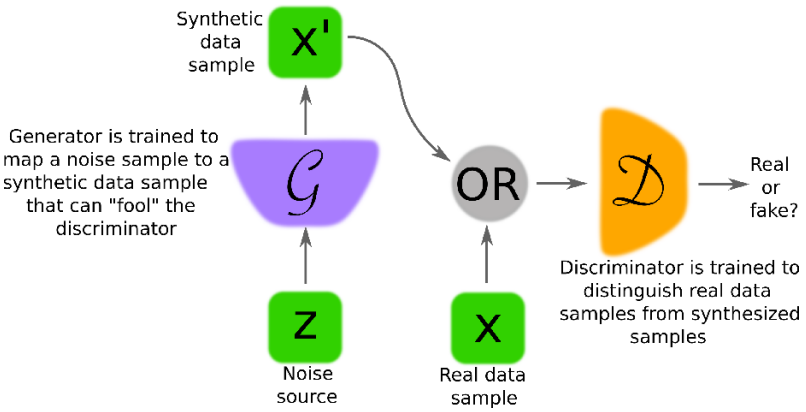

Figure 3.7: GAN architecture. [34]

Formally, the goal of the generator network (from now on referred to as **G**), is to learn the mapping of some representation space (latent space) $p_z(Z)$, where Z is a a sample from that latent space, to the space of the data $p_{data}(x)$, where x is a real sample and produce a synthetic data sample $G(z)$. The goal of the discriminator network (from now on referred to as **D**), is to learn the mapping of some data sample to the probability of it being real (closer to p_{data}, probability value closer to 1) or fake (closer to p_z, probability value closer to 0).

The process of training a GAN is often described as a zero-sum game (also known as min-max game), where the generator and the discriminator have differentiable functions. The cost of training is evaluated using a value function, V(G,D) that depends on both the generator and the discriminator:

$$\max_D \min_G V(G,D) \tag{3.6}$$

where,

$$V(G,D) = \mathbb{E}_{x\sim p_{data}(x)} \log D(x) + \mathbb{E}_{z\sim p_z(Z)} \log(1 - D(G(z))) \tag{3.7}$$

The cost used for G in Equation 3.7 is useful for theoretical analysis but does not usually perform well in practice. This is because when D successfully rejects G samples with very high confidence, which occurs frequently in the early stages of training when the fake samples are much different from the training data, G will likely go into the state of vanishing gradients which throttles or completely stops training. To solve this, the most common approach is to change G's cost function to:

$$\max_G [\mathbb{E}_{z\sim p_z(Z)} \log(D(G(z)))] \tag{3.8}$$

The difference is, in Equation 3.7, G minimizes the log-probability of D being correct. In Figure 3.8, the new optimized objective function, G maximizes the log-probability of D being mistaken. This approach is known to improve results and is considered the standard in practice.

The training process consists of simultaneous Stochastic Gradient Descent. Ideally, D is trained until optimization with respect to current G, then D is updated again. In practice, this is prone to over-fitting and so D and G are, tipically, alternately optimized: D optimized for k steps and G optimized 1 step. The number of k steps is a hyper-parameter. Although some authors achieved better results running more k steps on D, Goodfellow[29], is of the opinion that the protocol works best with the value of 1. That is, D and G are trained iteratively, with one step for each one.

3.3.1 Deep Convolutional GAN

GANs are often affected by a number of different issues such as: mode collapse (G maps different Z values to the same output, resulting in similar synthesized images), diminished gradients (D becomes rapidly successful, resulting in gradient vanishing in G), non-convergence (G and D fail to stabilize and converge) and high sensitivity to hyper-parameters. In order to solve these issues, several architecture variants have been proposed. Most notably, the Deep Convolutional GAN[10] made great progress in improving the quality of generated images by replacing maxpoolings in the original architecture with strided convolutions in D and fractionally-strided convolutions in G (figure 3.8), allowing the networks to learn their own spatial downsamplings. Despite this progress, the mentioned issues still remain an open problem.

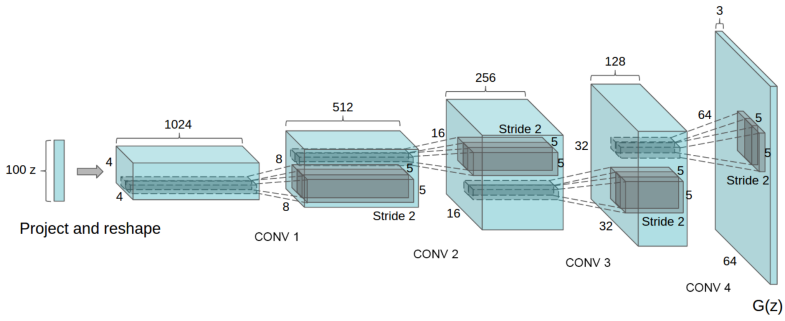

Figure 3.8: DCGAN Generator architecture. [10]

Other key changes in the DCGAN architecture were:

- Using batch normalization[35] in both G and D to help gradient flow in deeper models and prevent mode collapse.
- Removal of fully connected hidden layers for deeper models.
- Using ReLU activation in G for all layer except output, which uses Tanh.
- Using LeakyReLU in all layers of D

- Adam optimizer[36] instead of Stochastic Gradient Descent with momentum

As an extension to this architecture, Wu et al.[37], presented GANs that were able to synthesize 3D images of cars, tables and chairs using volumetric convolutions.

3.3.1.1 Loss Functions

Most GAN architectures are now based on the DCGAN architecture with hundreds of new named GANs being developed since its inception. The majority of new GANs focus on researching improvements of the original loss function by adding new penalties or by creating new ways to compute costs. Popular variations are the Wasserstein GAN (wGAN)[6] and the Least Squares GAN (LSGAN)[38].

wGAN consists on a modification to the original loss function in which D does not classify instances. Instead, for each instance it outputs a number corresponding to how real an image is and so the training goal is to have a higher value for real images than to fake images. This way, D is no longer simply "discriminating" between real or fake images but critiquing the "realness" of an image. This means that the inputs to the loss functions don't have to be probabilities. The loss functions for D and G are then respectively:

$$\max_D D(x) - D(G(z)) \tag{3.9}$$

$$\max_G D(G(z)) \tag{3.10}$$

The benefit of Wasserstein loss is that it provides useful gradient almost everywhere, allowing for the continued training of the models. It also means that a lower Wasserstein loss correlates with better fake image quality, meaning that we are explicitly seeking a minimization of generator loss.

Also motivated by the issue of vanishing gradients caused by the use of binary cross entropy, it was also proposed penalizing G when the generated images are very different from the real images, reducing the chances of the gradients to vanish. D is then modified to minimize the sum squared difference between predicted and expected values for real and fake images:

$$\min_D (D(x) - 1)^2 + (D(G(z)))^2 \tag{3.11}$$

The generator seeks to minimize the sum squared difference between predicted and expected values as though the generated images were real:

$$\min_G (D(G(z)) - 1)^2 \tag{3.12}$$

In practice, this involves maintaining the class labels of 0 and 1 for fake and real images respectively, minimizing the least squares, also called mean squared error or L2 loss:

$$l2loss = \sum (y_{predicted} - y_{true})^2 \tag{3.13}$$

The benefit of the least squares loss is that it gives more penalty to larger errors, in turn resulting in a large correction rather than a vanishing gradient and no model update. Many other GAN loss functions have been proposed such as the DRAGAN[39] or the BEGAN[40], however, it should be noted that there is a growing discussion[41] on whether modifications to the loss functions effectively improve GAN results.

3.3.2 Stacked GANs

Stacked GANs (SGAN)[42] are another variant of the original GAN architecture that instead of using a Generator and Discriminator, uses an Encoder, and a Decoder network. The decoder works as the generator in a GAN model and as the name "stacked" implies, the decoding and the encoding are done in a stack.

The stack of encoders is fed with image x, predicting label y at the end of the stack. Each encoder in the stack creates an intermediate prediction and feeds it as conditional input to the corresponding generator along with noise. The output of the generator is fed as input to the same encoder creating a new prediction. Each level is trained individually, and then joint training is performed as schematised in Figure 3.9.

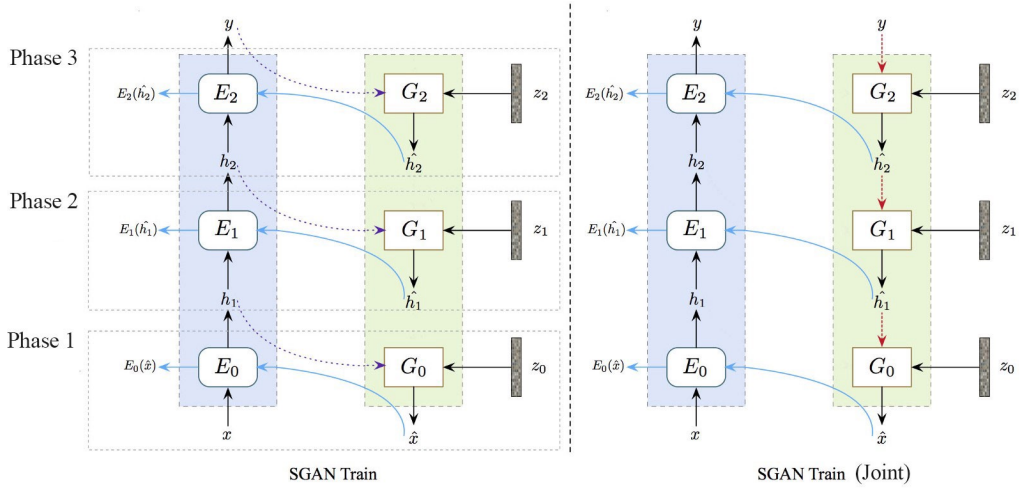

Figure 3.9: SGAN architecture. [42]

3.3.3 Progressive Growing of GANs

Most GAN research has focused on low-resolution images. During the early stages of training, if the real images are of low-resolution and so have less detail, D will have more difficulty in

identifying real from fake images. In other words, low-resolution images make the early G's distribution closer to the real images distribution, which improves convergence.

Progressive growing of GANs[7] (PGGAN), were introduced by NVIDIA[1] as a way to tackle this limitation. The main idea of this architecture is to grow G and D progressively, starting from simpler, low-resolution images, and adding new layers that introduce higher-resolution details as training progresses. The goal is to initially discover large scale (low frequency) information and incrementally learn more fine scale (higher frequency) information. To achieve this, G and D are mirrors and are increased in synchrony, as shown in figure 3.10.

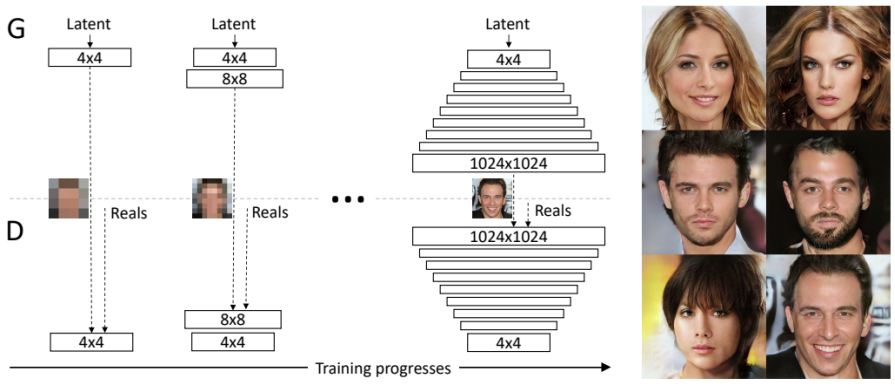

Figure 3.10: PGGAN architecture & results. [7]

Figure 3.11 shows how each layer is faded in to G and D. This process is used to avoid "shocks" to the model when a new layer is added with initialised parameters, which would destabilise the model. After a certain number of training iterations (800k images shown to D in the publication), the new layer is added to D and G and progressively more images are passed through this layer, until it is completely faded in (fade in takes another 800k images in the publication). In figure 3.11, the fade in is controlled by α which starts at 0. As it increases, more images are passed through the new 32×32 layer (b), until it reaches the value of 1 and all images start passing through it (c). This process repeats until all layers are faded in and the images are outputted in the desired resolution.

[1] https://www.nvidia.com/

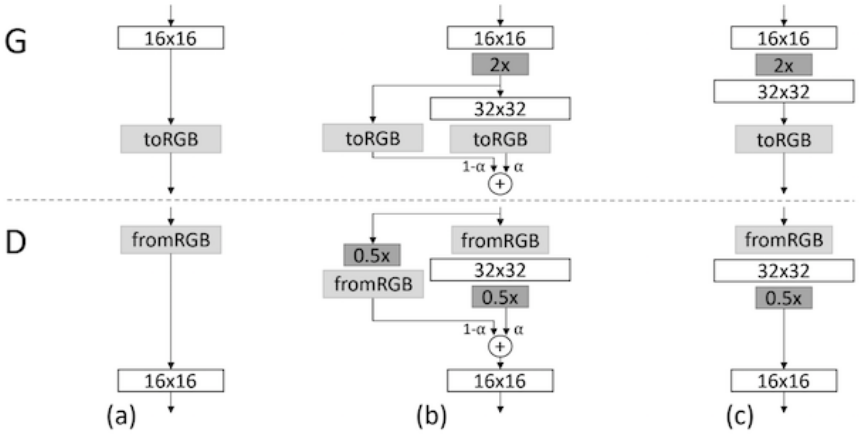

Figure 3.11: PGGAN layer fade-in. [7]

In terms of loss functions, a WGAN-GP and LSGAN were used, achieving similar results, and noting that the choice of loss function should have minimal impact on this training method.

3.3.4 Conditional GAN

First proposed in [8], conditional GANs (cGAN) extend the original architecture by adding a class conditional label to *G* and *D*, as shown in Figure 3.12. Conditional GANs provide better representations of multimodal data, for example, in the MNIST dataset[43], using a cGAN would allow the specification of which digits the generator should output. In the original GAN (or DCGAN), the generator randomly outputs digits and there is no control over which digits are generated.

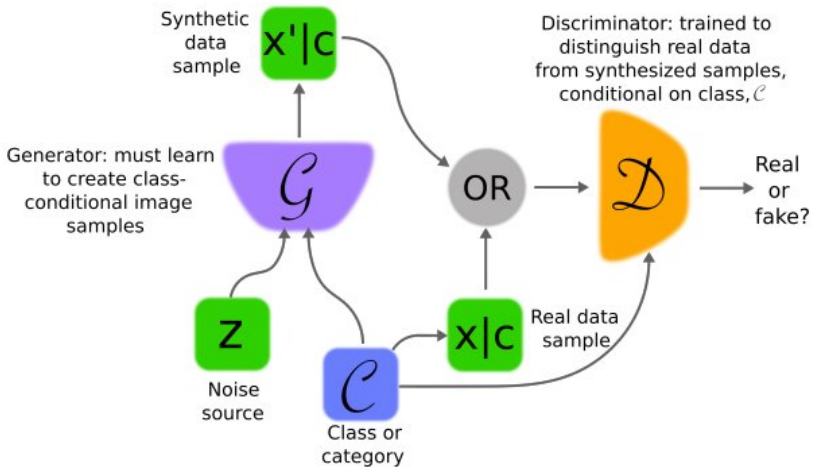

Figure 3.12: cGAN architecture. [34]

The loss function for the cGAN shown in Equation 3.14 is a variation of the original minmax equation for GANs shown in Equation 3.7.

$$\max_D \min_G V(G,D) = \mathbb{E}_{x\sim p_{data}(x)} \log D(x|y) + \mathbb{E}_{z\sim p_z(Z)} \log(1 - D(G(z|y))) \tag{3.14}$$

where y represents a class label or data from other modalities such as semantic label maps, as shown in section 3.3.5.

Another type of conditional GAN, the Auxiliary Classifier GAN (AC-GAN), first proposed in[9], showed that tasking D with reconstructing side information, instead of directly being fed side information, improved the overall quality of the generated images on the ImageNet dataset[44].

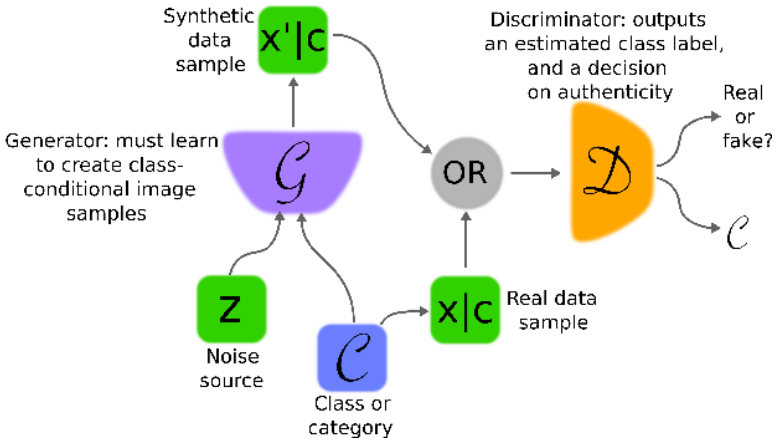

Figure 3.13: AC-GAN architecture. [34]

This architecture implies a modification to the original GAN loss function of Equation 3.7, in order to add the additional term L_c, related with D's probability of correctly classifying labels:

$$\max_D \min_G V(G,D) = \mathbb{E}_{x\sim p_{data}(x)} \log D(x) + \mathbb{E}_{z\sim p_z(Z)} \log(1 - D(G(z))) + L_c \tag{3.15}$$

where,

$$L_c = \mathbb{E}[\log P(C = c|X_{real})] + \mathbb{E}[\log P(C = c|X_{fake})] \tag{3.16}$$

3.3.5 Image-to-image Translation: pix2pix & pix2pixHD

First introduced in [45], image to image translation refers to the conversion of an image of a certain domain, into an image of a different domain. The idea is to learn the mapping between an input image and an output image using a training set of image pairs.

The first work that pioneered image-to-image using GANs was pix2pix [11], a cGAN based architecture that uses U-Net[46] for G and a Convolutional PatchGAN as D. Instead of classifying each image as real or fake, as in the cGAN architecture, the PatchGAN applies a classification strategy to $N \times N$ patches of the images it receives as input. The training process of this framework is similar to the cGAN architecture, with the addition of the L_1 distance as one of terms of the loss function of G to measure the similarity between corresponding real and synthetic images. The L_2

loss was also tested but produced blurrier results. The resulting loss function is the adversarial loss function presented in the previous chapters, with the added L_1 loss:

$$\arg\min_G \max_D \mathbb{E}_{x,y}[\log D(x,y)] + \mathbb{E}_{x,z}[\log(1 - D(x, G(x,z))) + \lambda \mathbb{E}_{x,y,z}[||y - G(x,z)||_1] \quad (3.17)$$

where λ is a hyper-parameter that controls the importance given to the L_1 distance term. In terms of the practical implementation, G was trained to maximize $log(D(x, G(x,z)))$, to prevent vanishing gradients.

This framework does not directly input noise to G. Instead, it is introduced through dropouts in the network, since initial experiments showed that G simply learned to ignore the noise when it was provided as a direct input tot he network. Additionally, D's objective function is divided by 2, to slow down it's learning rate.

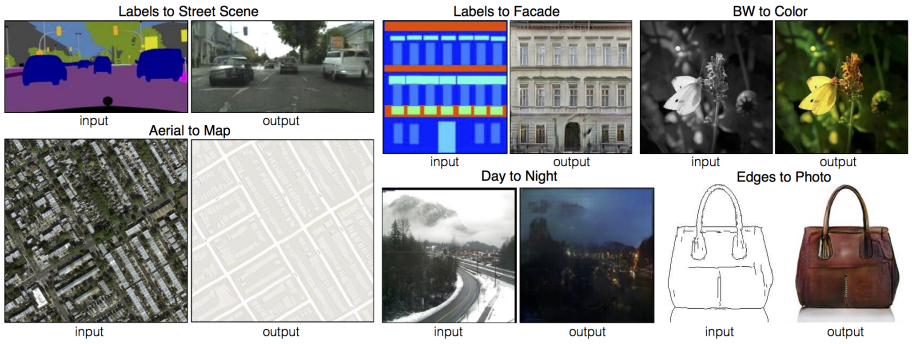

Figure 3.14: pix2pix results. [11]

A variant of this architecture known as $pix2pix_{HD}$ is proposed in [47] where the model is modified to use a coarse-to-fine G and multi-scale discriminators, achieving better results at producing images of higher resolution.

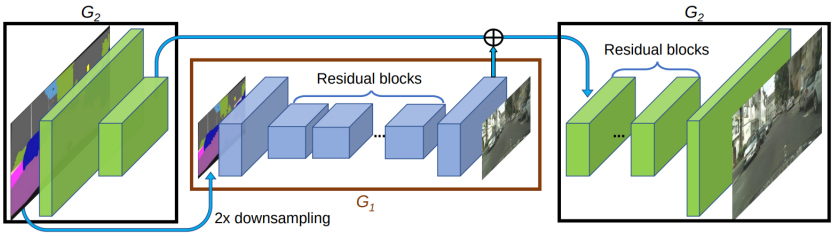

Figure 3.15: Coarse-to-fine generator of $pix2pix_{HD}$. [47]

The coarse-to-fine generator is composed of two sub-networks: global generator G_1 and local enhancer G_2 that interact in the following way: G_2 first receives a semantic label map and outputs a set of feature maps with half the size of the original input. The semantic label map is downsampled to half size, concatenated with the feature map and fed as input to G_1. The last set of feature map

produced by $G1$ is summed pixel-wise to the set of feature maps outputted by G_2, and the resulting set of feature maps is fed to G_2's residual blocks, finishing the forward pass through G2.

To discriminate real and fake images, a multi-scale discriminator architecture was proposed. Three identical discriminators are trained to detect real and fake images at different resolutions. Through experiments, it was noted that the multi-scale discriminators not only resulted in higher-quality images, but also reduced repeated patterns that would often appear in the generated images, making the images visually incoherent.

The objective function used to train the model combines a modified version of the classical GAN loss that incorporates multi-discriminator optimization with **feature matching** (Equation 3.18) and **perceptual loss** (Equation 3.19).

$$\mathscr{L}_{FM}(G,D_k) = \mathbb{E}_{(s,x)} \sum_{i=1}^{T} \frac{1}{N_i} [||D_k^{(i)}(s,x) - D_k^{(i)}(s,G(s))||_1] \tag{3.18}$$

where $D^k_{(i)}$ indicates the i^{th} layer feature maps of the discriminator D_k, T refers to the total number of layers and N_i refers to the number of elements of the i_{th} layer.

The **perceptual loss** component consists of computing the L_1 distance between the same intermediate feature maps of the VGG network[48] obtained using the real image and it's corresponding fake counterpart:

$$\mathscr{L}_{perceptual}(x,G) = \sum_{i=1}^{N} \frac{1}{M_i} [||F^{(i)}(x) - F^{(i)}(G(s))||_1] \tag{3.19}$$

where $F^{(i)}$ denotes the i_{th} layer with M_i elements of the VGG network.

The final objective function is then given by:

$$\min_G((\max_{D_1,D_2,D_3} \sum_{k=1,2,3} \mathscr{L}_{GAN}(G,D_k)) + \lambda \sum_{k=1,2,3} \mathscr{L}_{FM}(G,D_k) + \lambda \mathscr{L}_{perceptual}(x,G)) \tag{3.20}$$

Additionally, **binary boundary maps** were used to differentiate between multiple objects of the same class that were partially in front of each other. Figure 3.16 showcases the results of this implementation compared with the original *pix2pix*.

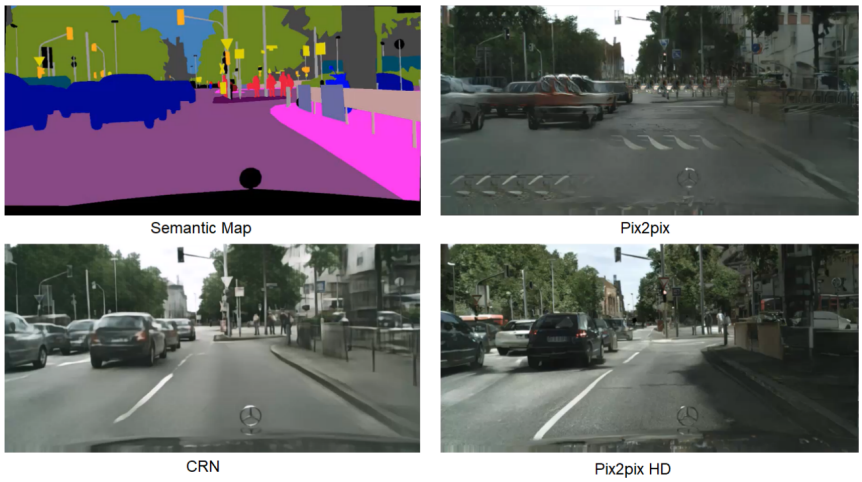

Figure 3.16: Comparative results of $pix2pix$, $pix2pix_{HD}$ and CRN. [47]

3.3.6 Image-to-image Translation: CycleGAN

The two mentioned frameworks of the previous section, require paired data which can be difficult and expensive to prepare. For this reason, the CycleGAN[49] was proposed as an unsupervised model, using a collection of images that do not need to be related in any way. This simple, yet powerful technique resulted in image translation of high quality on a wide range of domains, Figure 3.17 shows some of the published results.

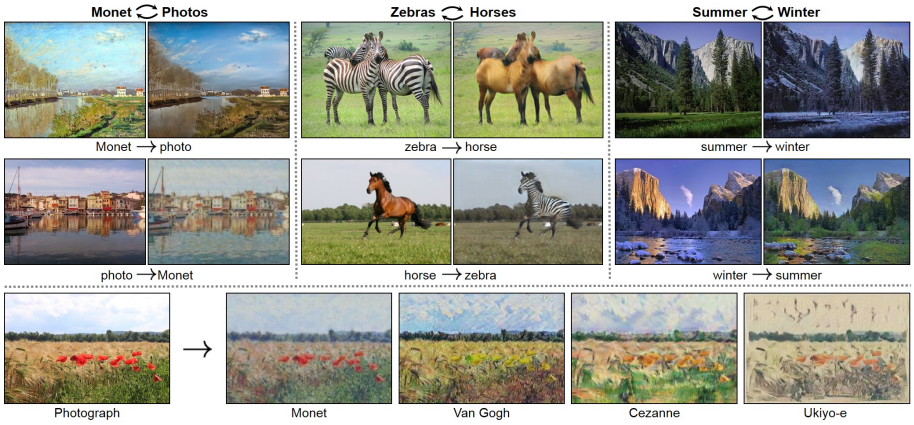

Figure 3.17: CycleGAN results. [49]

The framework is composed of four networks: two generators G and F, and two discriminators D_x and D_y. These networks work in pairs where D_y evaluates images translated from domain X to Y by G, and D_x evaluates the inverse transformation (Y to X) performed by F. The goal is to learn the appropriate G and F mappings according to the training data comprised in each domain. This interaction is shown in Figure 3.18.

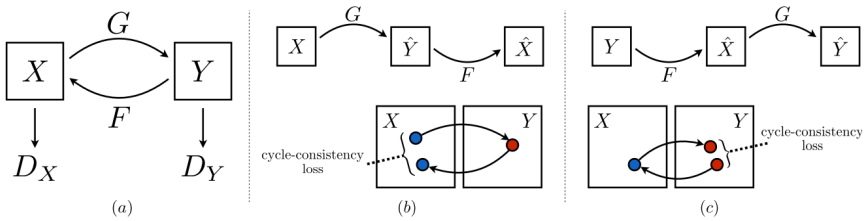

Figure 3.18: CycleGAN framework. [49]

The loss function used in this framework combines the classic GAN loss with an added **cycle consistency** term. The goal of this term is to evaluate whether an image that is translated from domain X to Y to X, remains identical to the original image. **Cycle consistency** is computed through the L_1 distance between each image and its reconstruction for each domain:

$$\mathscr{L}_{cc}(G,F) = \mathbb{E}_{x\sim p_{data}(x)}[||F(G(x)) - x||_1] + \mathbb{E}_{y\sim p_{data}(y)}[||G(F(y)) - y||_1] \tag{3.21}$$

The final loss function is then given by:

$$\mathscr{L}(G,F,D_x,D_y) = \mathscr{L}_{GAN}(G,D_y,X,Y) + \mathscr{L}_{GAN}(F,D_x,Y,X) + \lambda\mathscr{L}_{cc}(G,F) \tag{3.22}$$

where λ controls the importance of the cycle consistency term.

3.3.7 Optimization Techniques

Training a GAN is not a simple task and finding ways to improve and facilitate this process is an ongoing research topic. Non-convergence, mode collapse and diminished gradients are some of the main issues that affect GAN training and several publications have highlighted possible techniques that reduce the impact of these issues:

- **Experience replay**[50]: D updated using the newest generated samples as well as with previous ones, preventing it from over-fitting to a certain time instance of the generator.
- **Historical averaging**: track previous model parameters and penalize changes that are too different from the average changes, which improves convergence.[51]
- **Feature matching**: Train G to produce fake data that matches the statistics of real data.[51]
- **One-sided label smoothing**: Penalize D by replacing 0 and 1 targets with smoothed values, such as 0.2 and 0.9 respectively. Some authors advise not smoothing fake labels.[51]
- **Mini-batch discrimination**: Instead of training D with one image at a time, train with one mini-batch of fake images and one mini-batch of real images. This not only improves convergence but can also be used to detect mode collapse by analyzing the similarity of the images in the mini-batch of fake images.[51]

- **Virtual batch normalization**: Normalize each input sample in relation to the statistics of a fixed, reference batch, defined beforehand.[51]

3.3.8 Evaluation of generative models

Standardising GAN evaluation is currently an open problem. Although several methods and metrics have been proposed, researchers are yet to agree on which better capture the advantages and shortcomings of GAN models. Current methods can be divided into two groups: **perceptual studies**, which relate with using human observers analyze and compare real and fake images, and **objective metrics**, which contain both traditional similarity scores and tasks such as classification, detection or feature extraction and comparison.

3.3.8.1 Perceptual Studies

Perceptual studies consist on asking human annotators to attempt to distinguish between generated data and real data and evaluate those results to extract a metric that quantifies the quality of the model. Naturally, since these evaluations are based on subjective evaluation, the results can vary depending on the motivations of the subject, the setup of the task or by the use of hand-picked samples.

The Visual Turing Test (VTT)[52], proposes a standard to perform this evaluation. It consists of posing binary questions to assess a system's ability to recognise objects and identify attributes and relationships in images. This method is commonly used in medical generation, for example in [53], since objective methods are typically not available in this domain.

3.3.8.2 Inception Score

The Inception Score (IS)[51], was proposed as a way to overcome the downsides of perceptual studies while still maintaining correlation with human evaluation. The IS measures two things simultaneously: whether the images have variety and if each image contain meaningful objects.

Calculating the IS involves applying one of the most widely used classification networks, the Inception network[54], pre-trained on the ImageNet dataset, to obtain the conditional label distribution $p(y|x)$. If the generated images are varied, the marginal $\int p(y|x = G(z))dz$ will have high entropy and if an image contains meaningful objects, its corresponding $p(y|x)$ will have low entropy. The combination of these two values results in the Inception Score:

$$\text{IS} = e^{\mathbb{E}_x KL(p(y|x)||p(y))} \tag{3.23}$$

where $p(y)$ corresponds to the marginal class distribution.

Despite being a novel method of GAN evaluation, the IS score still has several drawbacks[55, 56]:

- IS is limited by what the Inception network can classify, which depends on the data used to train it, which makes it impractical in domains where classifying images is difficult.

- The Inception network is sensitive to small variations on the pre-trained weights, resulting in different scores for the same set of test images. The process of training a network for classification has inherent randomness, which causes different training procedures to produce different weights.
- The IS has no component for evaluating intra-class variability, so high scores will be produced even if the generator only synthesizes one type of image per class.
- The IS is unable to detect if the generator learns to replicate the training images instead of generating different ones, producing high values.
- Due to its conception, the IS may favor models that generate good objects rather than realistic images.
- The IS is an asymmetric measure and is affected by image resolution.

3.3.8.3 Fréchet Inception Distance

The Fréchet inception distance (FID)[57] measures the generator's performance by calculating the Fréchet distance between two multivariate Gaussians, created by the 2048-dimensional features of the pool3 layer of the Inception-v3 model:

$$\text{FID} = ||\mu_r - \mu_g||^2 + \text{Tr}(\Sigma_r + \Sigma_g - 2(\Sigma_r\Sigma_g)^{1/2}) \tag{3.24}$$

where the μ_r and μ_g refer to the feature-wise mean of the real and generated images and Σ_r, Σ_g are the covariance matrix for the real and generated feature vectors.

FID has 0 as its lower bound but has no upper bound. A lower FID value indicates higher similarity between a real image and its corresponding synthetic counterpart and, therefore, a high quality synthetic image. Similarly to IS, this metric is also dependant on what the Inception network can classify.

3.3.8.4 Structural Similarity Index

First used in[58] as an image evaluation technique, the Structural Similarity Index (SSIM) uses three image characteristics to compare two images:

- **Luminance and contrast distortion:** Image distortitions are less visible in bright or textured regions.
- **Loss of structural correlation:** Spatially close pixels are considered to have strong interdependence.

The SSIM is calculated through the product shown in equation 3.25 where l, c and s represent luminance, contrast and structural correlation, respectively.

$$\text{SSIM(x,y)} = l(x,y) \cdot c(x,y) \cdot s(x,y) \tag{3.25}$$

where,

$$l(x,y) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1} \tag{3.26}$$

$$c(x,y) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2} \tag{3.27}$$

$$s(x,y) = \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3} \tag{3.28}$$

where μ_x, μ_y, σ_x, σ_y, and σ_{xy} are the local means, standard deviations, and cross-covariance for images x, y. C_1, C_2 and C_3 are constants.

3.3.9 GANs in medical imaging

Although the use of GANs in the medical domain is extremely recent, the popularity of GANs and the quality of the results shown in benchmark datasets, have encouraged medical imaging researchers to experiment with this technology over the last years. Figure 3.19 shows the distribution of released GAN papers in medical imaging in terms of (a) canonical tasks, (b) image modality and (c) number of papers published from 2016 to 2018.

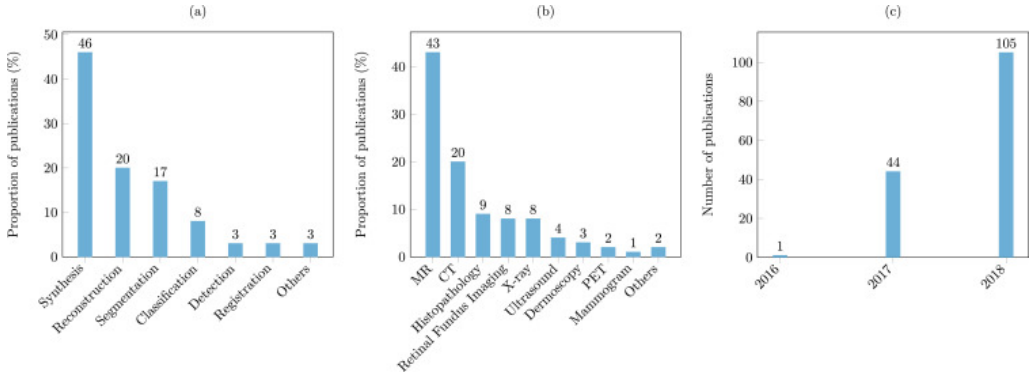

Figure 3.19: (a) Categorization of GAN related papers according to canonical tasks. (b) Categorization of GAN related papers according to imaging modality. (c) Number of GAN related papers published from 2016. Note that some works performed various tasks and conducted evaluation on datasets with different modalities. We counted these works multiple times in plotting these graphs. Works related to cross domain image transfer were counted based on the source domain. The statistics presented in figure (a) and (b) are based on papers published on or before January 1st, 2019. [59]

3.3.9.1 Noise-to-image applications

Noise-to-image refers to GANs that generate images that are not conditioned on a semantic label map. From graph (c) of figure 3.19, only 21 publications fall into this category with the large majority of publications employing image-to-image synthesis.

On the use of GANs as a potential augmentation technique, Maayan Frid-Adar et al.[17], used three DCGANs (section 3.3.1), to generate three types of liver lesions (cysts, metastases, and hemangiomas). The generated samples where used to train a liver lesion classifier with additional synthetic data, improving the accuracy of the classifier when compared with training with traditional augmentation techniques (rotation, cropping, etc.) by 7.1%. This work is fundamental in further supporting the hypothesis that GANs can effectively be used to lessen the impact of data scarcity in the medical domain, and increase the accuracy of classifiers. Furthermore, the same experiment was performed with an ACGAN (section 3.3.4), but the generated images did not improve classification. This result was unexpected since typically, other experiments performed by the computer vision community, suggested that using labels to train GANs resulted in higher quality images[9][51].

Christopher Bowles et al.[60], also used GANs to increase data size to improve the performance of a CNN. In this publication, a PGGAN (section 3.3.3) was used due to its ability to synthesize images of larger resolutions and the additional synthetic data was used to test whether a segmentation network could be improved. Several experiments were performed with varying percentages of available real data. The results showed that in all cases the segmentation network benefited from the increased synthetic data, with more significant results with lower percentages of available real data.

In another interesting experiment, Sarfaraz Hussein et. al[53], used a DCGAN to generate lung nodules and used a Visual Turing Test (section 3.3.8.1) to evaluate whether two radiologists could identify which nodules were real or not. Figure 3.21 shows the ratio of nodules correctly identified by the radiologists and Figure 3.20 showcases the generated nodules. In this publication, it's noted that the radiologists would occasionally notice unnatural characteristics in the lung nodules, such as nodules having both malignant and benign characteristics that would stand out and tip them into recognizing them as fake.

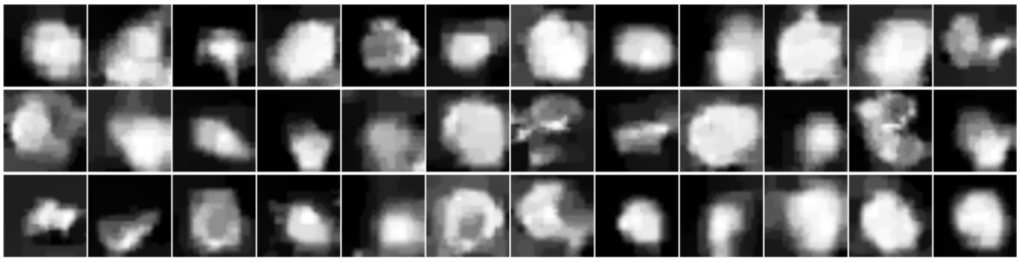

Figure 3.20: Example of synthetic lung nodules from DCGAN. [53]

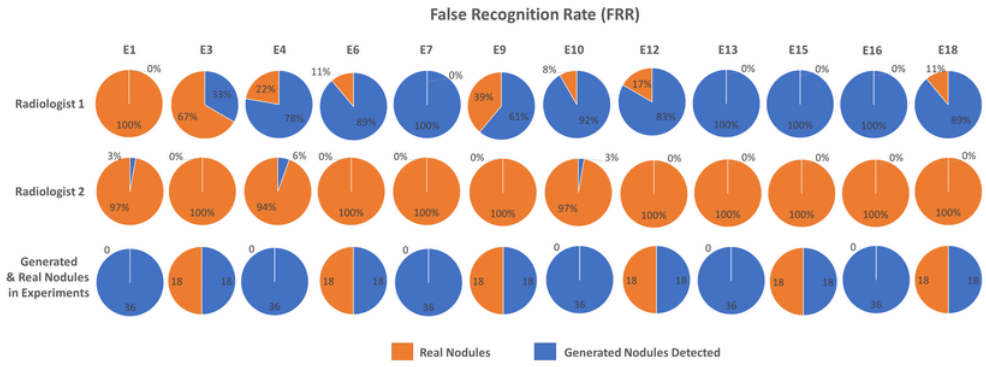

Figure 3.21: FRR for the two radiologists and contain either all generated nodules or a mixture of real and generated nodules. The FRR shown in the pie charts indicate the percentage of nodules which radiologists recognized as generated (fake). [53]

This experiment showed that a DCGAN architecture can be used to generate samples that pass the VTT test. Nevertheless, it was noted that some samples contained characteristics of both malignant and benign nodules, which would stand out to the radiologists and lead them to correctly identify the synthetic image as fake.

3.3.9.2 Image-to-image applications

The majority of image-to-image applications employ the pix2pix and cycleGAN frameworks for cross modality synthesis. This type of synthesis relates to generating an image of a certain domain, from an image of a different domain for example, T1 weighted MRI to T2 weighted MRI or MRI to CT.

In the medical imaging context, annotating medical data requires highly specialized professionals. This means that paired images are often not available, which encourages many researchers to employ the CycleGAN framework. Jelmer M. Wolterink et al.[14], used this architecture to translate brain MR images into brain CT images, experimenting with paired and unpaired data and achieving better results with the latter (figure 3.22).

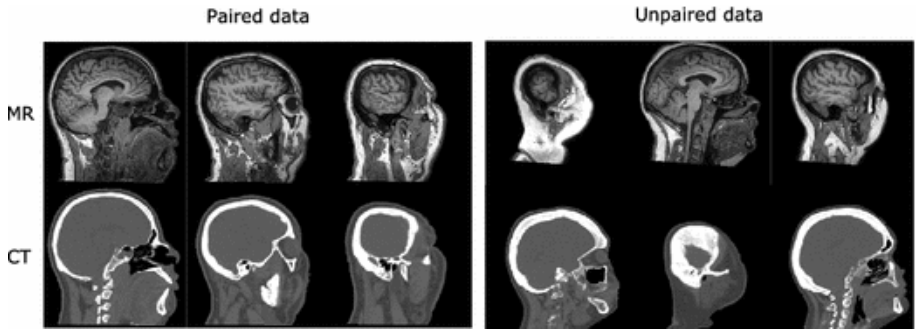

Figure 3.22: Paired data results vs unpaired data results. [14]

In another interesting application, Zizhao Zhang et al.[61], found that improved results could be achieved when adding a **shape consistency loss** obtain from two segmentation networks to

the cycleGAN loss function. This added term provided an extra level of regularization on the generators, preserving pixel-wise semantic label ownership and guarantee the anatomical structure invariance in medical volumes.

3.4 Summary

In this chapter, we explored various architectures of the three main types of generative models, their strengths and shortcomings. Auto-regressive models are a relatively simple model to train, produce images of high-quality but are very inefficient and, since they are based on generating pixels through context, are mostly used for completing partially occluded images, which is not within the scope of this project. Variational autoencoders are efficient both during training and sampling but produce blurry images due to the reconstruction error of the decoder. Improving image quality in VAEs is currently an open problem with VAE-GAN hybrids being proposed recently, such as the adversarial autoencoder[62] that produce improved results. However, these results are still not on par with the quality of images generated by GANs and are also yet to be tested in real-world scenarios. GANs currently offer state of the art image quality and their popularity as sprouted a number of different architectures aimed at improving both efficiency (DCGAN) and image resolution (PGGAN). The biggest disadvantage of GANs when compared to the other two methods is the lack of viable evaluation metrics, which often leads to researchers using subjective visual evaluation.

Chapter 4

Lung Image Synthesis

4.1 Datasets

Choosing viable datasets is a crucial task in any deep learning application, especially when dealing with unstable models such as GANs and in domains where high-quality data is difficultly to acquire. The following sections will describe the two used datasets, one public (LIDC) and one private (NLST), their details and train/test distributions.

4.1.1 Lung Image Database Consortium

The Lung Image Database Consortium image collection (LIDC-IDRI) [20] is a public dataset consisting of 1018 diagnostic and lung cancer screening thoracic computed tomography (CT) scans with an associated XML file containing the results of an annotation process performed by up to four radiologists related with the position and characteristics of lung nodules.

For the final dataset, we decided to use both examples with and without nodules, and, in order to balance the dataset, for each image containing an annotated nodule, a random slice guaranteed to not contain a nodule was added to the final dataset.

Of the total 1018 exams, 713 (70%) were used in the training dataset while the remaining 305 (30%) were used for the test set, resulting in 20553 images for training and 7885 for testing. All images remained with the original size of 512×512.

4.1.2 National Lung Screening Trial

The NLST Dataset [63] is a large scale dataset that collected data between August 2002 and April 2004. It includes patient clinical data, characteristics, screening exam results from approximately 54,000 participants. The CT dataset contains 75,000 screening exams complete with various annotations such as presence of abnormalities (for example, *emphysema* and *fibrosis*).

Using the entire database is unfeasible due to its size, so the NLST Query tool was used to select 400 total CTs where 200 contained emphysema and 200 did not contain emphysema. Since there were no annotations for the lung nodules, the masks from this dataset only contain the

position of the lungs and the entire CT was used for the final dataset. In total, 35829 images were used for the training set and 16383 for the test set.

4.2 Pre-processing

Considering that one of the main goals of this project is to explore a potential solution to data scarcity issues when building machine learning models in the medical domain, it is logical to fit our generated images to the kind of data that these models would use as input. Therefore, both datasets were first passed through a segmentation algorithm that extracted the lung from the CT images. This process also allows the creation of a mask of the position of the lungs that will later be used in our image-to-image GANs.

Three different threshold-based segmentation algorithms were tested: a mixture of pa mixture of implementations from various Kaggle contributors to the Data Science Bowl 2017 compiled by H.Chen[64], a lung segmentation algorithm developed to increase the likelihood of inclusion of juxta-pleural lung nodules developed by Moreira Aresta[65] and an algorithm developed during this dissertation.

Starting with H.Chen's algorithm[64], the CT is first normalized and then a threshold value is found by using *kmeans* clustering to separate soft tissue/bone from lung/air. Then, after two morphological operations to erode and dilate the mask, the lungs candidates are chosen by analyzing their position relatively to the center of the image. Figure 4.1 details the steps taken to segment a slice using H.Chen's[64] method.

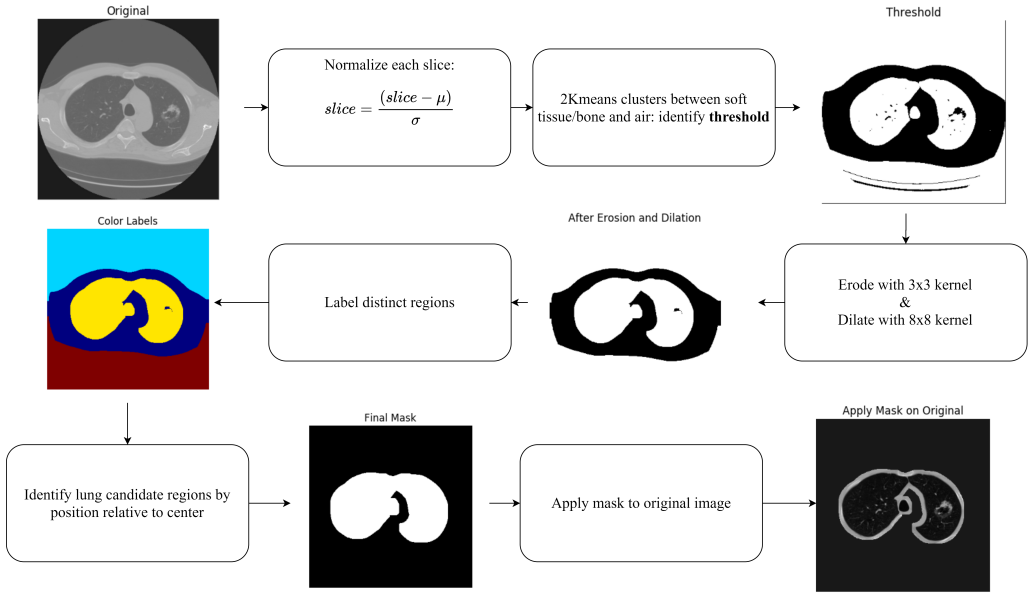

Figure 4.1: Flowchart of H.Chen[64] algorithm.

Although this algorithm presented a good starting point, it failed to segment slices that the other two algorithms could segment without any additional changes, likely due to the kmeans

method failing to find the correct threshold value. Figure 4.2 shows an example where the algorithm successfully segmented a slice.

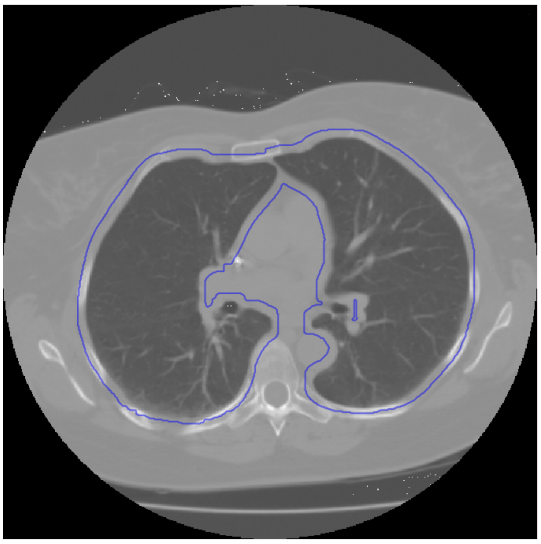

Figure 4.2: Segmented slice using H.Chen[64] algorithm on the LIDC dataset.

Moreira Aresta's[65] algorithm, takes a region growing-based approach to segment the entire lung volume at once. Initially, a voxel near the fat/muscle area of the volume is selected and, in an iterative process, each neighboring voxel is included in the final mask if its intensity is no less than 35% of the intensity of the initial voxel. Additionally, in order to increase the probability of the inclusion of juxta-pleural nodules in the final mask, a morphological operation is used to expand the borders of the resulting mask. Figure 4.3 and 4.4 show the algorithm flowchart and a successful result of this technique, respectively.

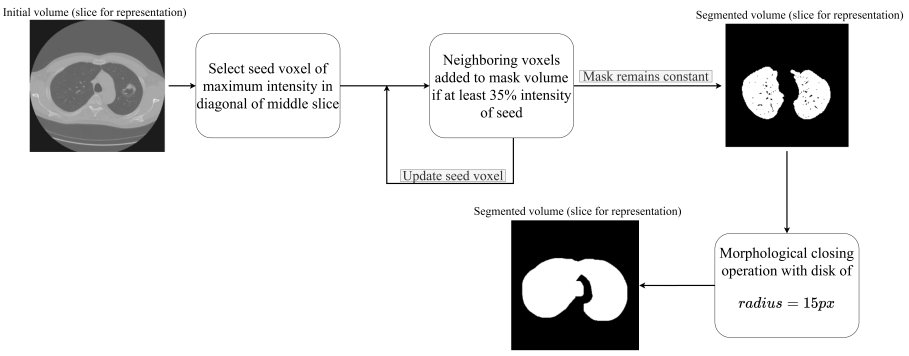

Figure 4.3: Flowchart of Moreira Aresta's[65] algorithm.

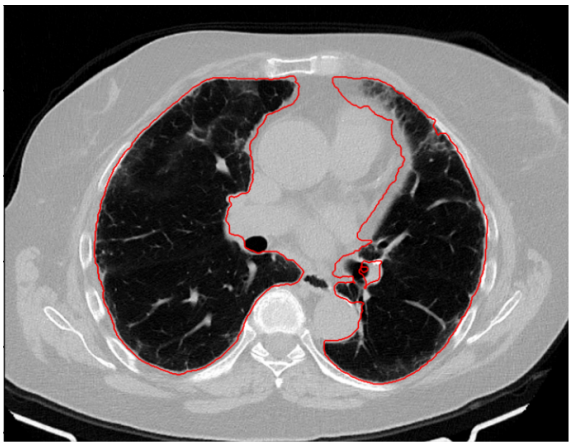

Figure 4.4: Segmented slice using Moreira Aresta's[65] on the LIDC dataset.

Lastly, the developed algorithm uses the dynamic threshold technique shown in figure 4.5 to calculate the optimal threshold for each slice. The threshold is calculated by finding the lowest point between between the two maximums which are typically lung and fat/muscle pixels. Then, possible lung masks are ordered by area size, and the largest one is picked for the final mask. Finally, each successive area is then evaluated on whether it has at least 20% of the previous accepted area, until either a certain minimum area threshold is reached or there are no more areas. After the areas are aggregated in a final mask, a morphological disk dilation is applied to fill out holes. This greedy approach is effective when the lungs occupy a large portion of the image, and when the lung is split in several smaller "blobs". Additionally, the minimum area can be seen has a hyper-parameter that can be easily fine-tuned if vanishing gradients occur. Figure 4.6 shows a flowchart that illustrates this method and Figure 4.7 shows different results of this algorithm.

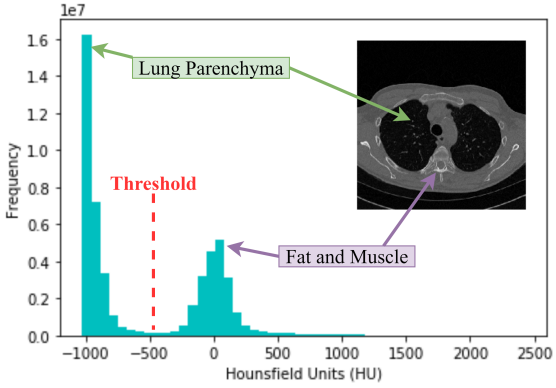

Figure 4.5: Dynamic threshold - image adapted from[66].

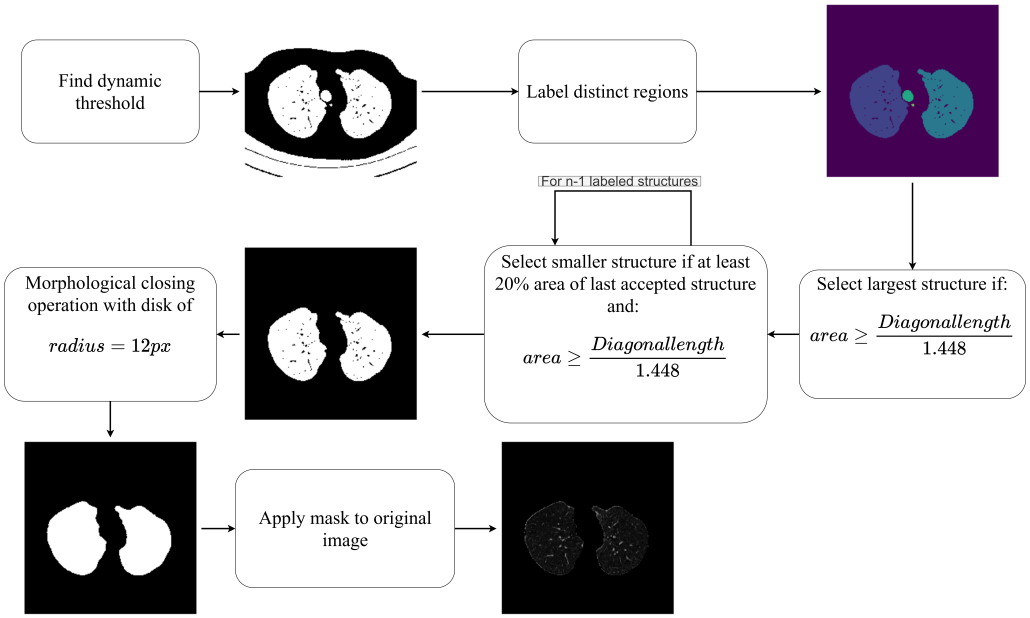

Figure 4.6: Flowchart of the developed algorithm.

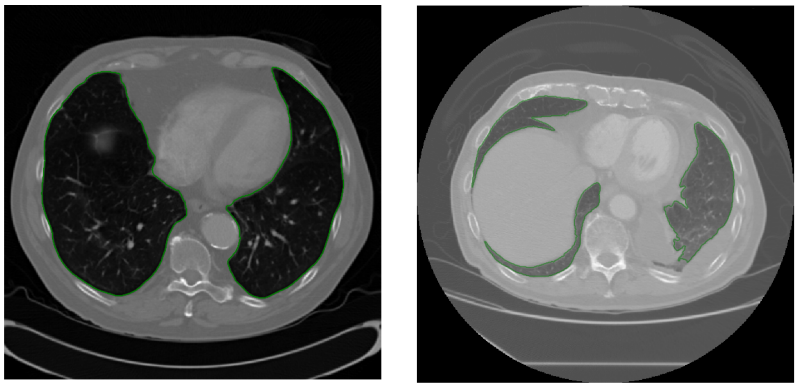

Figure 4.7: Segmented slices with developed algorithm on the LIDC dataset.

After several experiments with the three algorithms, the last two seemed to perform similarly well for the goals of this project. Although more robust, the main focus of the second algorithm was to guarantee the inclusion of juxta-pleural nodules, which is not within the scope of this project. After initial experiments with image-to-image GANs, it was a common occurrence that when there was a low variance between images, the model would hit vanishing gradients which caused the Generator to collapse and fail to train. This means that there is a disadvantage to training image-to-image GANs with sections of lungs that are small, relative to the rest of the image. Because the developed algorithm makes controlling the minimum size of the lungs more practical, it was chosen for this project. Figures 4.8 and 4.9 showcase an example of segmentation by all three algorithms and fully segmented lungs using the third algorithm, respectively.

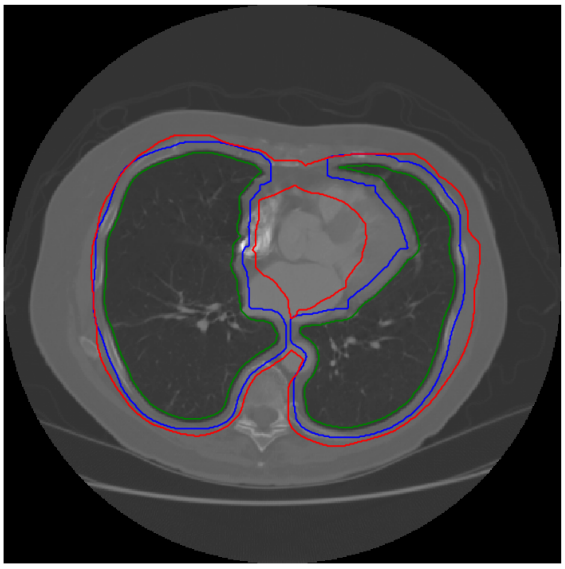

Figure 4.8: A slice from the LIDC dataset segmented by all three algorithms. Blue - algorithm 1, red - algorithm 2, green - algorithm 3

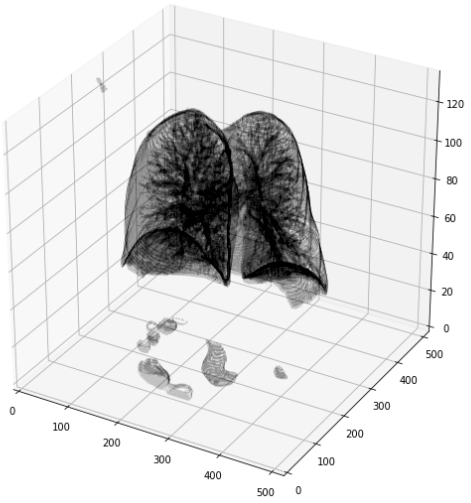

Figure 4.9: Fully segmented lungs using the developed algorithm.

For the LIDC dataset presented in section 4.1.1, an additional pre-processing step was added to include the annotated positions of the nodules. Each lung nodule mask was added to the corresponding lung mask generated by the segmentation algorithm and, since each nodule has up to four different positional masks pertaining to the annotation of each radiologist, an average mask was computed by adding each pixel to the final mask if it was contained in at least 50% of all masks, in other words, the final mask is the result of a 50% pixel-wise consensus from all radiologists. This process was done by the Python package *pylidc*, created by Hancock et al.[67] and

an example of a final result can be seen in Figure 4.10 as well as a final lung/nodule mask and corresponding lung segmentation in Figure 4.11.

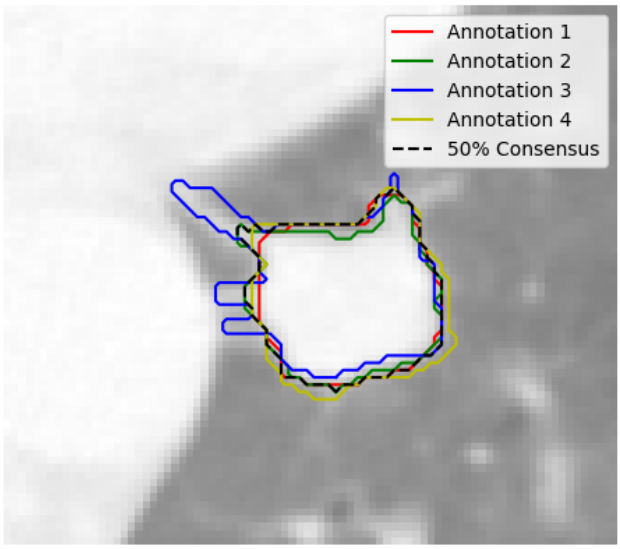

Figure 4.10: Nodule position annotation of four radiologists and the resulting 50% consensus mask.[67]

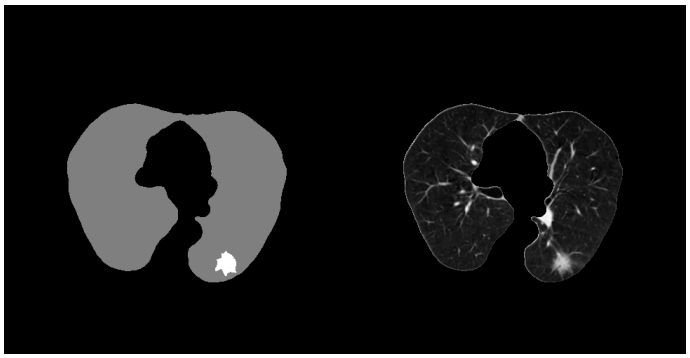

Figure 4.11: Mask of lung with nodule and corresponding segmented lung from the LIDC dataset.

The NLST dataset used only the lung position masks and so no additional pre-processing was needed. The emphysema presence/absence was later used internally when loading data into the models. Figure 4.12 shows an example of a mask and resulting segmented lung from the NLST dataset.

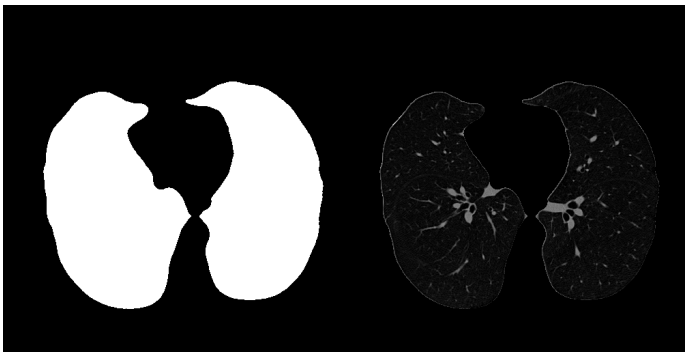

Figure 4.12: Mask of lung and corresponding segmented lung from the NLST dataset.

4.3 Model architectures

This chapter will describe the models used for synthesizing lung images. Starting with the Pix2Pix model that was used as baseline, followed by a modified Pix2Pix version that allows conditioning the synthesized image on a semantic label.

4.3.1 Pix2pix

The baseline Pix2Pix architecture was implemented as recommended in the original publication[11]. The generator utilizes an architecture similar to U-Net, without max-pooling layers, where an encoder comprised of convolutional layers increases the depth of an image by computing feature maps and reducing the dimensions of the image until a bottleneck is reached. The decoder then has the opposite task of reducing the depth of the image, while increasing the dimension of the image. Additionally, skip-connections, connect each encoder layer with its mirrored decoder layer, which improves gradient flow, stabilizes training and improves results. Considering that the notation C_k and TC_k denote a convolution and transposed convolution with k filters respectively, the specific encoder and decoder architectures are the following:

- encoder: Ci_{64}-C_{128}-C_{256}-C_{512}-C_{512}-C_{512}-C_{512}-C_{512}
- decoder: TC_{512}-TC_{512}-TC_{512}-TC_{512}-TC_{256}-TC_{128}-TC_{64}-TCf_1

All C_k layers apply batch normalization followed by a *LeakyReLU* layer with slope of 0.2. The first layer of the encoder, Ci_{64}, does not use batch normalization. The decoders TC_k layers use *ReLU* and the final layer, TCf_1, maps the image to the number of output channels (in this case 1 since the images are grayscale) and applies a *Tanh* function. All convolutions use a kernel size of 4×4 and stride of 2, with the exception of the first encoder layer which uses a stride of 1. Figure 4.13 shows a scheme of this architecture.

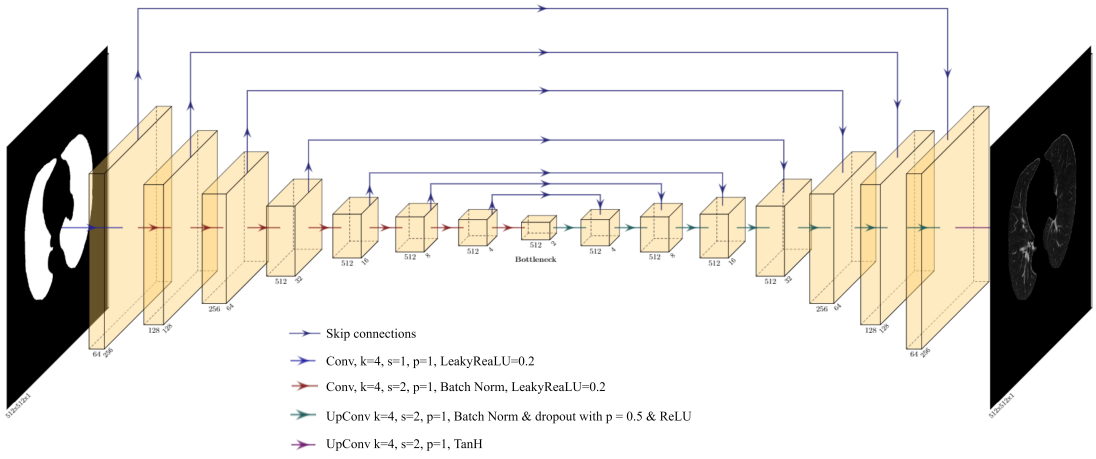

Figure 4.13: Pix2Pix Generator architecture

The architecture for the discriminator will follow the recommended in the publication for a 30×30 PatchGAN:

$$C_{64} - C_{128} - C_{256} - C_{512} - C_1$$

All convolutions use LeakyRelu with 0.2 slope, kernel size of 4×4 and padding of 1. The first three convolutions use a stride of 2, while the remaining two use a stride of 1. The first four layers use LeakyReLU as the activation function with 0.2 slope, while the last layer uses a Sigmoid function to determine whether an image is real of fake.

4.3.2 Semantic label conditioned Pix2Pix

The second implemented model was a variant of the Pix2Pix model that enables the use of an additional semantic label. The goal of this implementation is two-fold: control the type of image generated by the model and enable the possibility of utilizing semantic annotations that are very common in medical datasets, which could in turn produce better results.

In order to achieve this, the architecture will be modified to give the generator an additional class label and the discriminator the additional task of classifying images according to said label. This addition to the architecture is identical to the ACGAN described in section 3.3.4. The idea is to combine the contextual position of objects given by the image-to-image architectures and enable the generation of images based on additional semantic information. This implementation results in a modification to the loss function to include D's loss when guessing the class of an image, given by Equation 4.1.

$$\arg\min_G \max_D \mathbb{E}_{x,y}[\log D(x,y)] + \mathbb{E}_{x,z}[\log(1 - D(x, G(x,z))) + \lambda_1 \mathbb{E}_{x,y,z}[||y - G(x,z)||_1] + \lambda_2 L_c \quad (4.1)$$

where λ_2 controls the importance given to,

$$L_c = \mathbb{E}[\log P(C = c|X_{real})] + \mathbb{E}[\log P(C = c|X_{fake})] \tag{4.2}$$

Equation 4.2 indicates the loss for classifying a certain label c for real and fake images.

The architectures for D and G will remain the same, with the exception of an addition of an extra ouput layer with a Sigmoid function to D, in order to calculate the probability of a binary class, and the concatenation of the label to the input segmentation maps. Figure 4.14 shows an overview of the interaction of the two networks, with the addition of the class label.

This architecture is identical to the conditional generative adversarial network with classifier, ccGAN[68], where the same idea was implemented but different models were used for the generator and discriminator in a segmentation task.

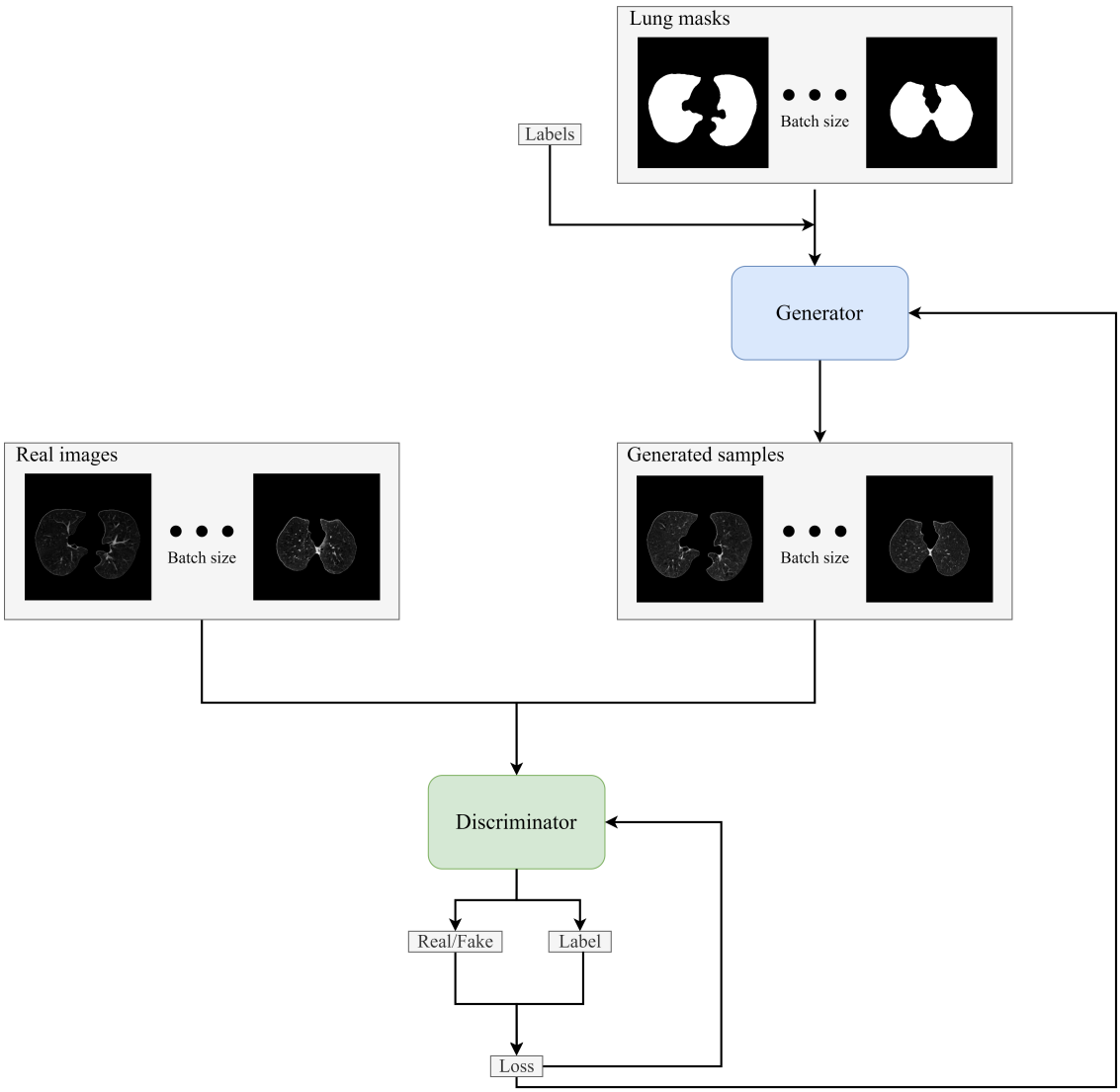

Figure 4.14: ccGAN architecture - the discriminator classifies images as real or fake and classifies the label. The combined loss is propagated to both the Discriminator and Generator.

4.4 Metrics

In order to evaluate the quality of the generated images, it is important to choose a set of metrics that cover the disadvantages of each other, since GAN evaluation is still an open problem. Considering that the Fréchet Inception Distance is currently widely used, it is logical to integrate it in this project. However, the FID is calculated by first computing the 2048-feature vector resulting from the pool3 layer of the Inception V3 network trained on the ImageNet dataset[44], which is composed of labeled RGB photographic images. This means that it is possible that the feature vector calculated for the lung images are not meaningful, since this domain images are not represented in the ImageNet dataset.

Liu et al.[69] suggested that FIDs-InceptionV3 extracted from images that are not represented in the ImageNet dataset, don't correlate with visual inspection, which was considered one of the strongest arguments for the use of FID. Instead, Liu et al. suggest the use of a domain-specific encoder to extract feature vectors for FID calculation.

This chapter will first explain a small test on the ccGAN architecture using the MNIST dataset, followed by an explanation and reasoning behind the choice of the evaluation metrics.

4.4.1 MNIST Test

Several studies have shown that evaluating GANs is not a simple task[56][41]. This problem is especially prevalent when the domain deals with images that are not easily evaluated through visual turing tests, such as images from the medical domain which require expertise to be visually evaluated. As such, it was important to guarantee that the implementation of the ccGAN was working as intended since it would be difficult to confirm that the medical annotations are being learned. In sum, by training the model in a domain where the generated images can be visually evaluated, we can rule out the possibility of an incorrect implementation. However, it is important to note that this test does not guarantee that the model will be able to learn medical labels, since the difficulty in training is vastly superior.

In order test if the ccGAN implementation was working as intended, a small experiment with the MNIST dataset[43] was conducted. The goal of the experiment was to test whether it was possible to train the ccGAN to generate the image of the desired digits, conditioned on the class label while the mask remained the same for all numbers. To achieve this, the training data is composed of masks of size $256 \times 256 \times 1$, with a central area representing the general position of the digit, concatenated with the respective digit label, resulting in a mask of size $256 \times 256 \times 2$. In order to fit the model to a multi-label classification task, the activation function on the label output layer was changed to a SoftMax function.

The model was trained for 200 epochs, using stochastic gradient descent with the ADAM optimizer[36] with momentum parameters $\beta_1 = 0.5$ and $\beta_2 = 0.999$ and learning rate of 0.0002. The learning rate was kept constant for 100 epochs and linearly decayed until 0 for the remaining epochs, as per the strategy introduced in the Pix2Pix publication[11]. The LSGAN goal was used to improve training stabilization, a batch size of 64 and an alternating gradient step strategy was

used by computing one gradient descent step in the discriminator followed by one step in the generator.

After training, the generator was able to produce realistic digit results conditioned on a class label. An example of the used masks and generated digits can be seen in Figure 4.15

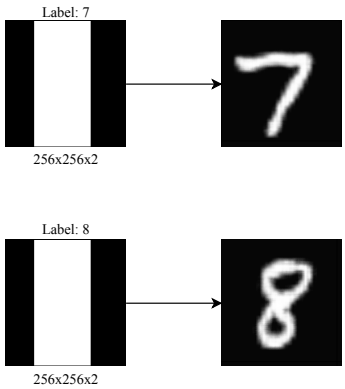

Figure 4.15: Example of two generated digits. Since the masks remain the same, it proves that the model can correctly be conditioned on class labels

4.4.2 Metrics

Along with training the GANs, each training set was also used to train two autoencoders. Then, the generated and real images from the test set are passed through the encoder where the distribution of fake and real images is calculated and used to compute the FID distance. Figure 4.16 summarizes this process. Additionally, for the ccGAN, a class-aware FID is also calculated for each label where the FID is calculated for each generated class.

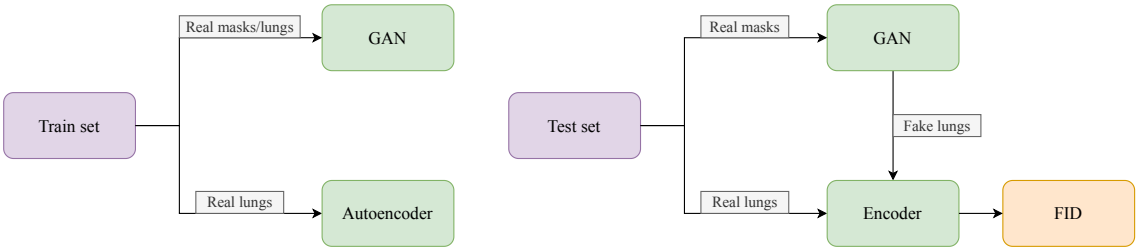

Figure 4.16: Train/test domain-specific FID calculation methodology.

Domain-specific FID is a relatively novel technique and at the moment there there are no publications that suggest the proper architecture for the encoder in benchmark datasets and no publications of this technique in the medical domain. Considering that the type of images in the medical domain are vastly different from other "real-world" image datasets, it would be a valuable contribution to create an encoder that extracts features from these datasets and use it for FID calculation, since it would create a viable way to compare GAN results in the medical domain.

For this purpose, two autoencoder architectures were trained. The first is a simple, fully-connected autoencoder that reduces images to 3 features and attempts to reconstruct the image.

This has the advantage of having the ability to create a visualization of real/fake distributions, since each image is reduced to a point in 3-dimensional space and we are able to visualize the distributions of the real and fake images. The encoder and decoder are composed of four fully-connected hidden linear layers each with ReLU as activation function, except for the final layers of the encoder and decoder which use Tanh. The FID values calculated from this autoencoder will be referred to as FID_{fc}.

The previous autoencoder serves as a way to visualize where the real/fake distributions lie, however, state of the art autoencoders typically make use of convolutions to map each image to thousands of feature maps, which usually results in a better reconstruction. The second autoencoder is the inverse of DCGAN[10] with a bottleneck of 1024 feature maps. Considering the notation used in the previous chapter, the specific encoder and decoder architectures are the following:

- encoder: Ci_{64}-C_{128}-C_{256}-C_{512}-Cf_{1024}
- decoder: TC_{512}-TC_{256}-TC_{128}-TC_{64}-TC_1

All C_k layers apply batch normalization followed by a LeakyReLU with slope of 0.2 layer. The first layer of the encoder, Ci_{64}, does not use batch normalization. The decoders TC_k layers use ReLU and the final layer, TCf_1, maps the image to the number of output channels (in this case 1 since the images are grayscale) and applies a Tanh function. All convolutions use a kernel size of 4×4 and stride of 2, except for the last encoder and the first decoder layer which use stride of 1. Figure 4.17 shows a scheme of this architecture. The FID values calculated from this autoencoder will be referred to as FID_{conv}.

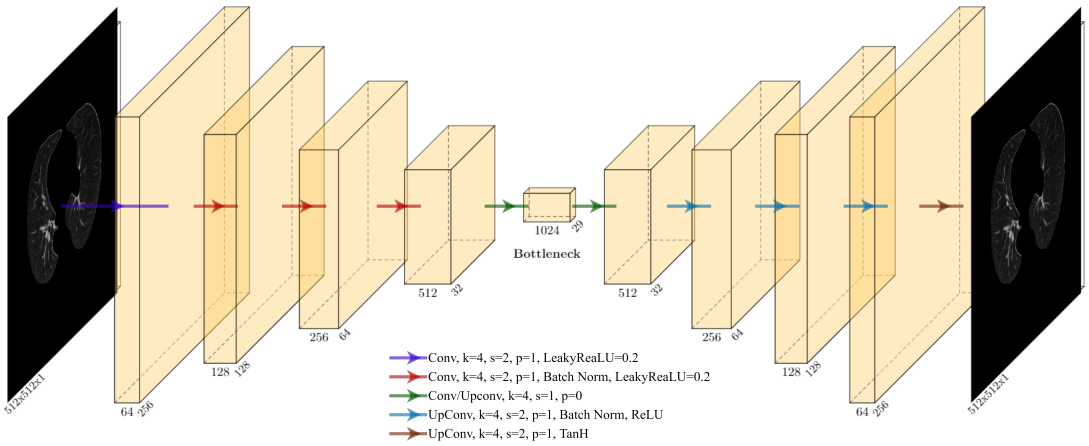

Figure 4.17: Convolutional autoencoder architecture

In sum, each GAN model is evaluated and compared using $FID_{inceptionV3}$, domain-specific FID_{fc}, domain-specific FID_{conv} and structural similarity index (SSIM).

4.5 Training

For the synthesis task, 3 GAN models were trained: Pix2Pix on the LIDC dataset ($P2P_{lidc}$), Pix2Pix and ccGAN on the NLST dataset ($P2P_{nlst}$ and $ccGAN_{nlst}$, respectively). The ccGAN used the presence/absence of *emphysema* as its classification label. The ccGAN was not trained on the LIDC dataset since the available labels refer to the lung nodules, which constitute very small parts of the image (between 3mm and 30mm), making it unlikely that the model could learn these labels. Additionally, two AEs (one fully-connected, one convolutional) were trained on each dataset for domain-specific FID computation. The same training datasets were used for GANs and AEs in order to not introduce bias when testing.

All GAN models were trained for 200 epochs, using stochastic gradient descent with the ADAM optimizer[36] with momentum parameters $\beta_1 = 0.5$ and $\beta_2 = 0.999$ and learning rate of 0.0002. The learning rate was kept constant for 100 epochs and linearly decayed until 0 for the remaining epochs. The LSGAN goal was used to improve training stabilization, a batch size of 64 and an alternating gradient step strategy was used by computing one gradient descent step in the discriminator followed by one step in the generator. The loss functions used Equation 3.17 with $\lambda = 10$, as per the strategy introduced in the Pix2Pix publication[11]. The ccGAN used $\lambda_2 = 10$ for the importance of the classification loss.

The AEs were trained for 200 epochs with the exception of the LIDC convolutional AE which was trained for 300 epochs to improve convergence, since the dataset was smaller. All AEs also used the ADAM optimizer with momentum parameters $\beta_1 = 0.5$ and $\beta_2 = 0.999$, learning rate of 0.00002 and used MSE loss.

4.6 Results and discussion

The intent behind FID is to create a reliable comparison between GAN results by method of feature extraction and comparison, which indicates that FIDs can only be compared when the model used for feature extraction is the same for all GANs. Table 4.1 shows the results of this project's $FID_{inceptionV3}$, compared with Pix2Pix $FIDinceptionV3$ results on benchmark datasets Facades, Maps, Edges2Shoes and Edges2Handbags from DeVries et al.[70] and with DCGAN $FIDinceptionV3$ results from MRI image synthesis from Haarburger et al.[71]. Additionally, for the LIDC dataset, all FID values are calculated for all images and separately, for only the subset of images that contain nodules and subset that do not contain nodules. Similarly, for the NLST dataset, all FID values are calculated for all images and separately, for only the subsets of images with and without emphysema. All $FIDinceptionV3$ from this project were estimated using the official implementation code provided by the authors of FID[57].

	$FID_{InceptionV3}$		
$P2P_{LIDC}$	All	With nodules	Without nodules
	12.82	16.84	13.43
$P2P_{NLST}$	All	Emphysema	No Emphysema
	11.56	15.21	13.75
$ccGAN_{NLST}$	All	Emphysema	No Emphysema
	10.82	12.35	12.62
$P2P_{Facades}$[70]	104		
$P2P_{Maps}$[70]	106.8		
$P2P_{Edges2Shoes}$[70]	47.3		
$P2P_{Edges2Handbags}$[70]	76.0		
$DCGAN_{MRI}$[71]	20.23		

Table 4.1: FID results from features extracted from the InceptionV3 network.

The results extracted from this metric, would indicate that the ccGAN is the overall superior model, surpassing the other two developed models and the literature results. However, as suggested by Liu et al.[69], it is possible that since the InceptionV3 model was trained on images that are far different from the domain of this work (RGB photographic images), the extracted features are not relevant enough to give an accurate representation of the distribution of real and generated images and thus the FID would be inconsistent with visual inspection. It is possible that the low FID values achieved by the developed models are caused by the extracted features being mainly related with the general position of the lungs, without much relevance to the detail inside them. Considering that the models use an image-to-image technique, with the position of the lungs as starting point, it would explain why the FID values are still considerably lower than the $DCGAN_{MRI}$[71] results, a non-conditional model that generates images from a latent space.

As mentioned in section 4.4.2, the fully-connected AE allows us to create a visualization of the distribution of real and fake images, since each image is reduce to a point in 3D space, in addition to calculating an FID value. The official implementation of FID was still used, but was modified to receive the features extracted from the encoder as input.

Figures 4.18 4.19 and 4.20 show the real and fake distributions of all test set images, images with nodules and images without nodules respectively, for the Pix2Pix model trained on the LIDC dataset.

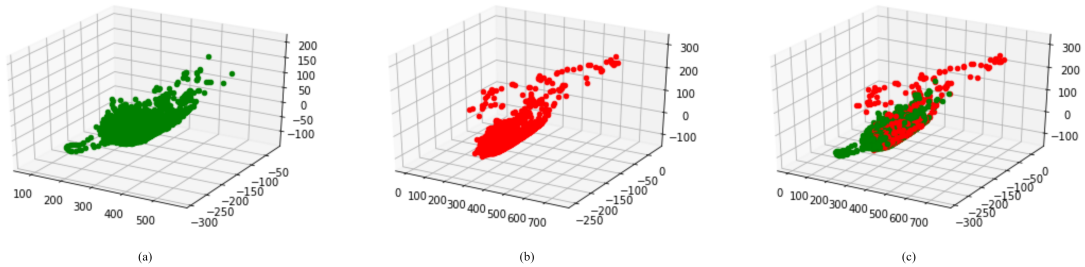

Figure 4.18: Distributions of all images for Pix2Pix on LIDC dataset. (a) real images distribution, (b) generated images distribution, (c) real and generated images.

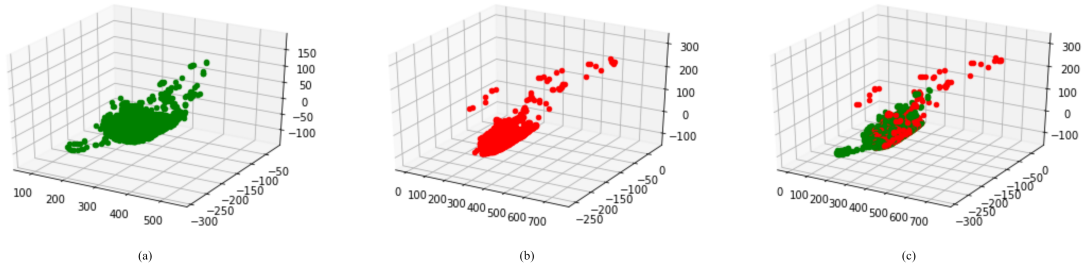

Figure 4.19: Distributions of images with nodules for Pix2Pix on LIDC dataset. (a) real images distribution, (b) generated images distribution, (c) real and generated images.

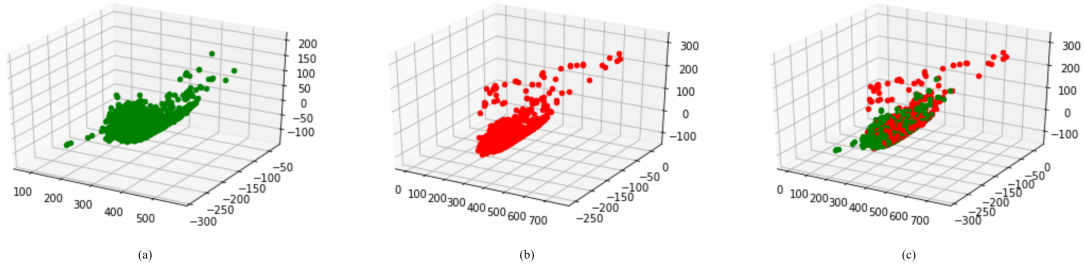

Figure 4.20: Distributions of images without nodules for Pix2Pix on LIDC dataset. (a) real images distribution, (b) generated images distribution, (c) real and generated images.

Analyzing the LIDC distributions, it appears that the differences between real images with and without nodules are very subtle, which is expected since the size of the nodules constitute an extremely small portion of the image and it is likely that this architecture is not deep enough to capture essential features to fully create differences in the representations. Additionally, it appears that most of the generated images are still encoded to the same space as the real images, with some outliers distancing themselves from the larger group of real images.

For the NLST dataset, the distributions and FID values were calculated for all images, images with emphysema, and images without emphysema. Figures 4.21, 4.22 and 4.23 show the distributions of all test images, images with emphysema and images without emphysema respectively, for the Pix2Pix model trained on the NLST dataset.

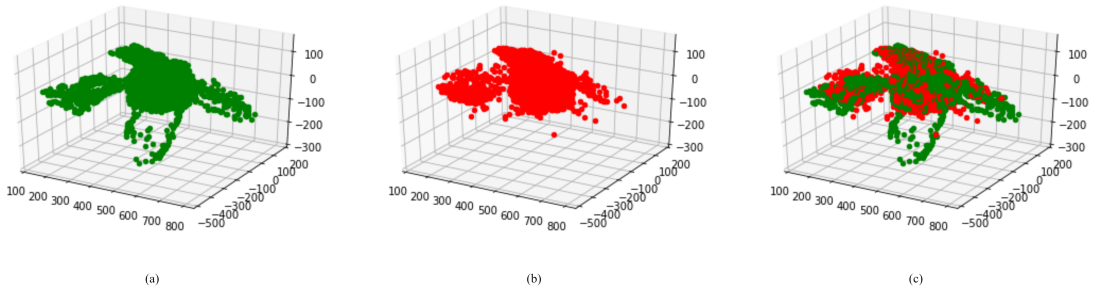

Figure 4.21: Distributions of all test set images for Pix2Pix on the NLST dataset. (a) real images distribution, (b) generated images distribution, (c) real and generated images.

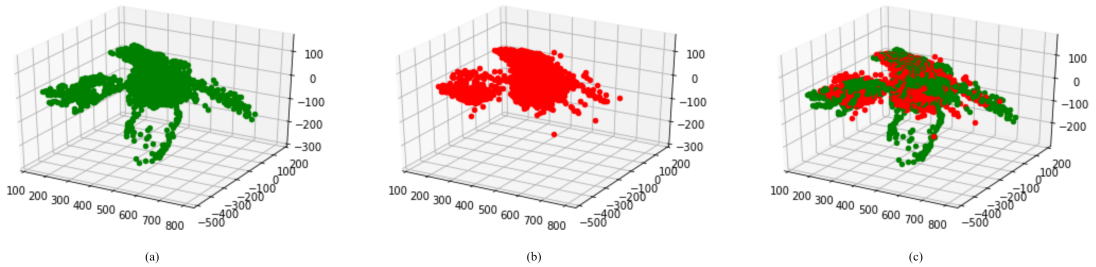

Figure 4.22: Distributions of test set images with emphysema label for Pix2Pix on the NLST dataset. (a) real images distribution, (b) generated images distribution, (c) real and generated images.

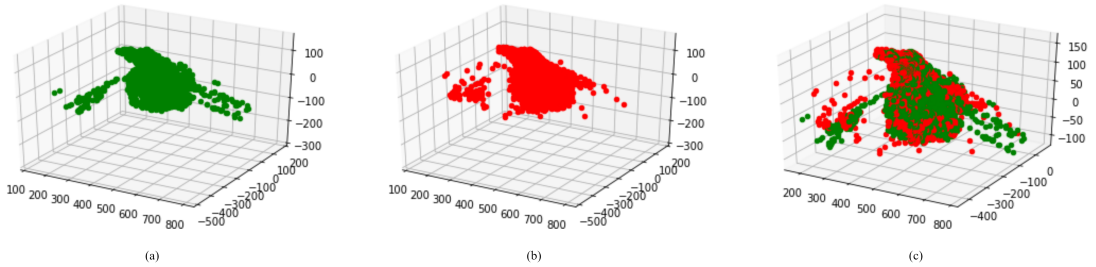

Figure 4.23: Distributions of test set images without emphysema label for Pix2Pix on the NLST dataset. (a) real images distribution, (b) generated images distribution, (c) real and generated images.

From these distributions, we can observe that although the real images with and without emphysema are slightly more visually different than the previous set of distributions, the majority of images are still encoded to the same general encoding space, likely due to the same reason. However, for the generated images, there seems to be more concentration of encodings around the real distribution, which should indicate better quality images and a lower FID.

Figures 4.24, 4.25 and 4.26 show the distributions of all test images, images with emphysema and images without emphysema respectively, for the ccGAN model trained on the NLST dataset.

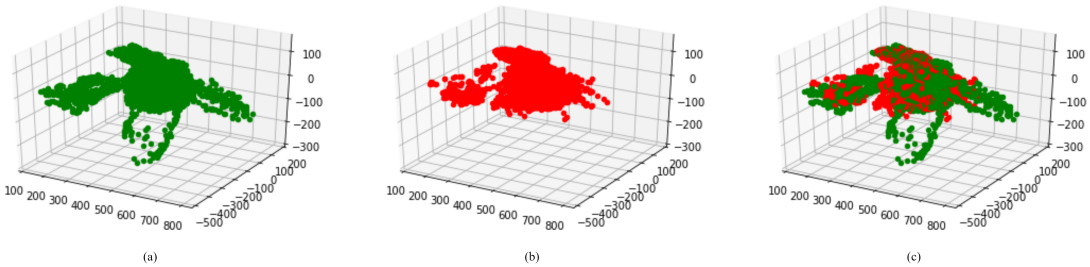

Figure 4.24: Distributions of all images for ccGAN on the NLST dataset. (a) real images distribution, (b) generated images distribution, (c) real and generated images.

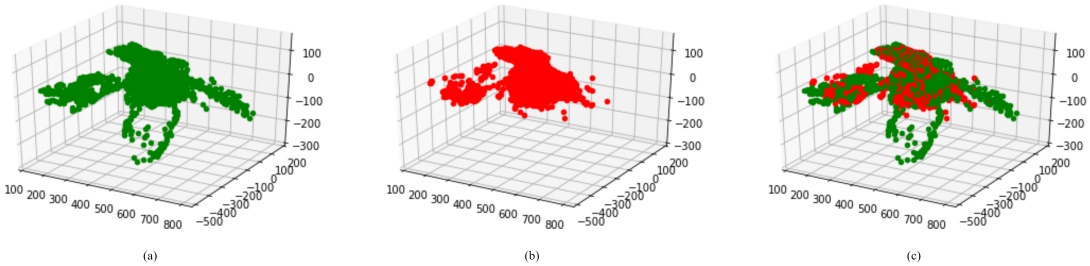

Figure 4.25: Distributions of images with emphysema label for ccGAN on the NLST dataset. (a) real images distribution, (b) generated images distribution, (c) real and generated images.

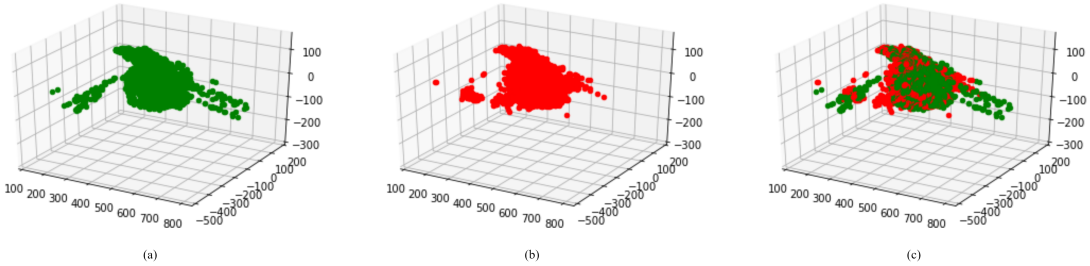

Figure 4.26: Distributions of images without emphysema label for ccGAN on the NLST dataset. (a) real images distribution, (b) generated images distribution, (c) real and generated images.

The distributions of the generated images appear similar to the previous, with the exception of the images without emphysema (Figure 4.26), where the fake and generated distributions appear to be closer. Additionally, it is worth noting that contrary to the LIDC dataset results, there are certain areas where the generated images failed to sample images to those areas, which could be an indicator of overfitting, since the model seems to not be able to generalize to the entire real distribution.

Table 4.2, shows the results of the FID for each of the previous distributions. Each value is the distance between the real and generated distributions.

	FID_{fc}		
$P2P_{LIDC}$	All	With nodules	Without nodules
	105.25	151.82	90.22
$P2P_{NLST}$	All	Emphysema	No Emphysema
	39.87	79.87	60.59
$ccGAN_{NLST}$	All	Emphysema	No Emphysema
	55.17	87.13	45.31

Table 4.2: FID values calculated from features extracted from the fully-connected AE

Being a distance measure, it is logical that the FID_{fc} values correlate more or less with visual inspection of the encoding distributions. From these metrics, the $P2P_{NLST}$ generates better general results than the rest, while the $ccGAN_{NLST}$ generates better images without emphysema. Moreover, it is worth noting that altough the LIDC dataset uses an additional mask label for the nodules, the FID values are much higher, likely due to differences in dataset size.

This experiment showed that although useful for a general view of the real/generated distributions, the simpler architecture might not be enough to capture the subtle differences between the labels of the two datasets. Hence, The encoder of the developed convolutional autoencoder outputs a feature map of size $1024 \times 29 \times 29$, for each image, creating a better likelihood of valuable encoded features. The values extracted from the convolutional autoencoder are displayed in Table 4.3. In order to reduce the spatial extent of the feature maps to 1×1, the features are first global average pooled to a vector before calculating the mean and covariance used in FID calculation.

	FID_{conv}		
$P2P_{LIDC}$	All	With nodules	Without nodules
	247.36	224.84	280.87
$P2P_{NLST}$	All	Emphysema	No Emphysema
	163.85	**162.12**	162.57
$ccGAN_{NLST}$	All	Emphysema	No Emphysema
	172.10	191.22	172.95

Table 4.3: FID values calculated from features extracted from the convolutional AE

The results for the entire datasets remain correlated with the previous encoder, where the $P2P_{NLST}$ generates the lower FID. However, we can see that for the labeled structures, the same architecture appears to be superior.

As a final metric, SSIM was used in order to compare each image with its groud-truth counterpart. The advantage of SSIM over other comparison metrics such as Mean Square Error and Peak Signal Noise Ratio, is that it compares the images in a structural approach rather than a pixel-wise approach.

Table 4.4 shows the SSIM results computed over the entire 512×512 images and on a crop of size 256×256 centered on the original images, in order to capture less background.

	$SSIM_{512}$		$SSIM_{256}$	
	μ	σ	μ	σ
$P2P_{LIDC}$	0.803	0.122	0.651	0.083
$P2P_{NLST}$	0.841	0.057	0.687	0.065
$ccGAN_{LIDC}$	**0.846**	0.057	0.696	0.064

Table 4.4: SSIM results for entire 512×512 image and with a central crop of 256×256

Additionally, Table 4.5 shows the SSIM results for a 128×128 and 64×64 centered window around pulmonary nodules generated on the LIDC dataset[1]. Since no other works were found in the literature concerning full lung synthesis, we are not able to compare these results with any reference.

	$SSIM_{128}$		$SSIM_{64}$	
	μ	σ	μ	σ
$P2P_{LIDC\_Nodules}$	**0.619**	0.050	0.601	0.064

Table 4.5: SSIM results for generated nodules in 128×128 and 64×64 window centered on the nodule.

As expected, the values for SSIM lower considerably when the full image is constricted to include less of the background and as shown by the previous results, the model has difficulty generating high-quality nodules, likely due to their smaller size. Figure 4.27 shows examples of generated nodules with a 128×128 window centered on the nodule.

[1] Padding equal to background value was added when necessary to fit the desired window sizes.

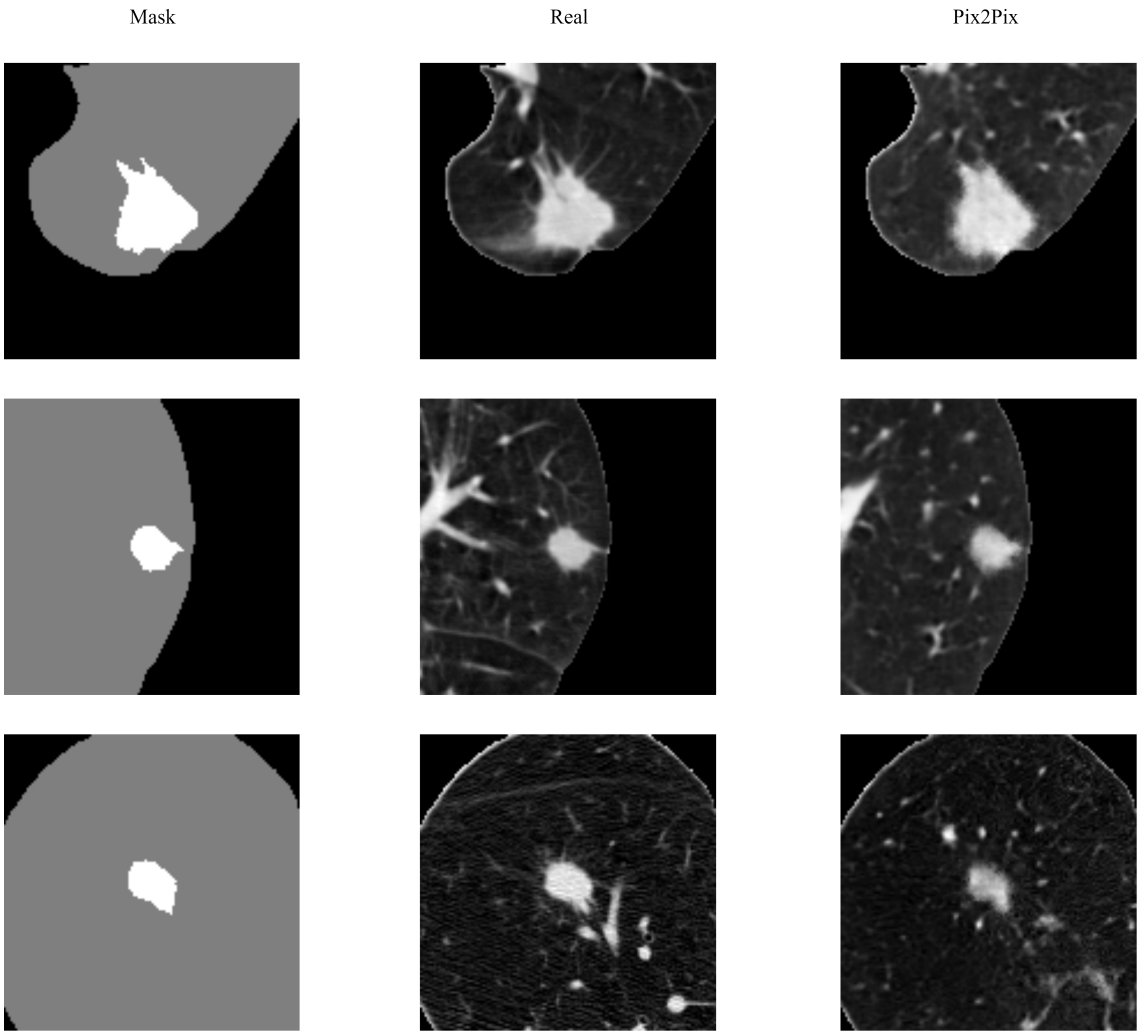

Figure 4.27: Generated images centered on a nodules with a 128×128 window.

Figures 4.28 shows examples of generated images of the Pix2Pix model on the LIDC dataset and Figure 4.29 shows examples of images generated using the Pix2Pix and ccGAN models on the NLST dataset.

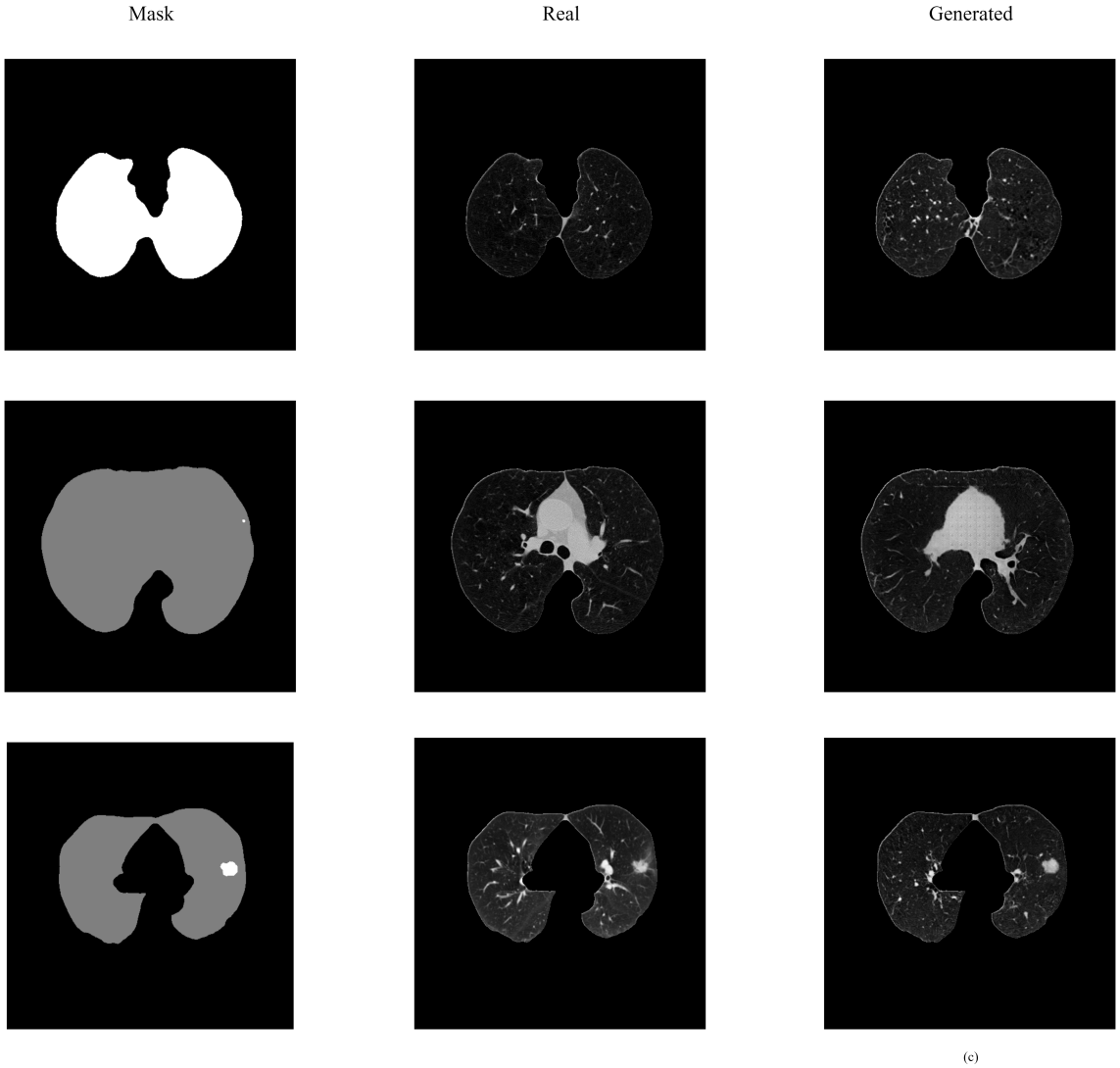

Figure 4.28: Generated images from the LIDC dataset usint the Pix2Pix model.

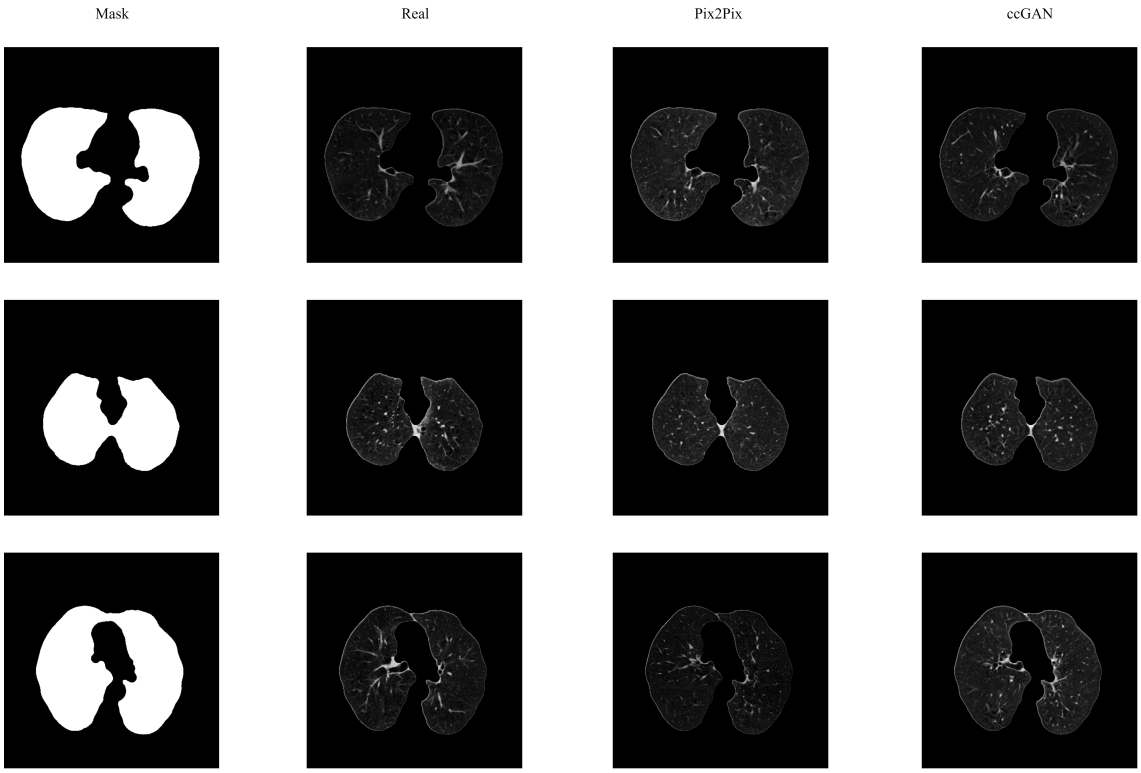

Figure 4.29: Generated images from the NLST dataset using the Pix2Pix and ccGAN models.

Visually, the models seem to generate images with similar quality, however, the results of the various metrics seem to indicate that the $Pix2Pix_{NLST}$ model generates images of higher quality, with only the $FID_{InceptionV3}$ clearly deviating from this conclusion.

4.7 Summary

At the beginning of this chapter, we gave an overview the chosen datasets for this work, namely, the National Lung Screening Trial (NLST) and Lung Image Database Consortium image collection (LIDC), their demographics and detailed information, followed by the explanation and analysis of the three different threshold-based segmentation algorithms explored to extract the lungs from thoracic CT scans, resulting in a modality of images closer to what is expected in lung-based deep learning models. Then, we reviewed the methods applied to synthesize artificial lung images namely, the Pix2Pix framework and a modified version that makes use of semantic labels and employs an auxiliary classifier to the Discriminator. Finally, we presented the chosen metrics to evaluate the quality of the generated samples, the widely used Fréchet Inception Distance (FID) that extracts and compares features resulting from the output of the pool3 layer of the InceptionV3 network, and derived domain-specific FID that uses AutoEncoders trained on the domain of the generated samples, in order to increase the relevance of the extracted and compared features. We presented the two different architectures used for this purpose, a fully-connected AE that reduces images to a three dimensional space, allowing the visualization of the distributions of the encoded

features, and a convolutional AE that maps each image to a $1024 \times 29 \times 29$ feature map, resulting in a more robust comparison.

Despite the advantages of the other two segmentation algorithms, the developed segmentation algorithm proved to be useful for the specific task of training GANs due to its ability to easily control the minimum accepted size of segmented objects and subsequently reducing the risk of inconsistent segmentation mappings.

Concerning the evaluation of the generative models, as expected, quantifying the realism and quality of the generated samples was a challenging task. The ambiguous nature of the FID metric prompted the use of an encoding visualization technique, that improved the overall understanding of the quality of the generated samples, and a domain-trained convolutional encoder for an increased trust in the result of this metric. The overall results seem to indicate that the original Pix2Pix architecture, trained on the NLST dataset, generated better samples, which is somewhat expected due to the larger dimensions of this dataset. However, it remains unclear whether these samples are of enough quality to be used as supplement to other deep learning models and, conversely, if the samples generated by the other models are not of enough quality for that sort of task. Additionally, it is possible that the inferior results of the ccGAN are a symptom of the chosen label since the emphysema annotation refers to the global volume, and the characteristics of this disease might not be present in all slices of a positive volume.

Chapter 5

Conclusion

This chapter summarizes the research performed throughout this dissertation, its main conclusions, contributions and lastly, a list of some possible research lines that could be explored to improve our work in the future.

5.1 Overview

The goal of this thesis, was to research and implemented an efficient method of synthesizing artificial lung images from annotation masks and semantic labels. The intent is to provide deep learning medical imaging researchers, with more comprehensive and balanced datasets, allowing for more robust methods of automatic detection or diagnosis of various lung diseases. Due to the domain of the used datasets, we started by reviewing the medical anatomy associated with the lungs and motivations behind the use of CT scans. We then reviewed the state of the art of image synthesis and evaluation, with special focus on generative adversarial networks and their applications in the medical imaging domain.

The main contributions of this work were the methods to synthesize and evaluate lung images. Concerning the generative models, three different approaches were explored: a pix2pix model on a smaller sized, but with two types of annotation mappings dataset (lung and nodule), a pix2pix model on a large dataset with only one type of annotation mapping (lung), and a ccGAN model on a large dataset with both the singular mapping and an additional semantic label (lung and emphysema label).

Regarding evaluation, the widely used FID metric was explored in a domain-specific environment, in order to improve the accuracy of results. To achieve this, two distinct autoencoders were trained to encode features to different spacial mappings, one allowing a 3D representation of real and fake distributions, and the other creating a larger spacial mapping, to closer resemble the encoding dimensions extracted from the InceptionV3, used in the original FID. Overall, the results computed from the used metrics indicate that the pix2pix trained on the NLST dataset, produces the higher quality samples.

5.2 Future work

This dissertation opens many different options regarding future research lines related with lung image synthesis. Below, we list a few possible improvements that we propose to implement in the future.

- Perform **Visual Turing Tests** with radiologists, which would give valuable insights regarding the realism of the generated samples.
- Explore the use of **semantic segmentation metrics**, which use pre-trained segmentation networks to evaluate the quality of the generated samples by computing the accuracy of the segmentation. If the samples are realistic, the network should be able to segment them correctly.
- Implement and evaluate lung generation using other GAN frameworks such as the **MedGAN**[72] or **SPADE**[73].
- Increase dataset size by using a larger subset of the NLST dataset.
- Create deeper autoencoders for better feature extraction used in the **domain-specific FID**.
- Use the features extracted from the used Autoencoders in domain-specific FID by using them in a separate task such as classification, which would guarantee that the compared features are important.
- Increase the amount of annotated structures such as surrounding muscles or generate the entire CT image

References

[1] The International Agency for Research on Cancer (IARC) report, "Latest global cancer data: Cancer burden rises to 18.1 million new cases and 9.6 million cancer deaths in 2018," *International Agency for Research on Cancer*, 2018.

[2] S. Hussein, K. Cao, Q. Song, and U. Bagci, "Risk stratification of lung nodules using 3D CNN-based multi-task learning," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2017.

[3] C. H. Lee and H. J. Yoon, "Medical big data: Promise and challenges," *Kidney Research and Clinical Practice*, 2017.

[4] K. Lata, M. Dave, and N. K.N., "Data Augmentation Using Generative Adversarial Network," *SSRN Electronic Journal*, 2019.

[5] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems*, 2014.

[6] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *34th International Conference on Machine Learning, ICML 2017*, 2017.

[7] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of GANs for improved quality, stability, and variation," in *6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings*, 2018.

[8] M. Mirza and S. Osindero, "CGAN," *CoRR*, 2014.

[9] A. Odena, C. Olah, and J. Shlens, "Conditional image synthesis with auxiliary classifier gans," in *34th International Conference on Machine Learning, ICML 2017*, 2017.

[10] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," in *4th International Conference on Learning Representations, ICLR 2016 - Conference Track Proceedings*, 2016.

[11] P. Isola, J. Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017.

[12] R. Yeh, C. Chen, T. Y. Lim, M. Hasegawa-Johnson, and M. N. Do, "Semantic Image Inpainting with Perceptual and Contextual Losses," *Arxiv*, 2016.

[13] B. Yu, L. Zhou, L. Wang, J. Fripp, and P. Bourgeat, "3D cGAN based cross-modality MR image synthesis for brain tumor segmentation," in *Proceedings - International Symposium on Biomedical Imaging*, 2018.

[14] J. M. Wolterink, A. M. Dinkla, M. H. Savenije, P. R. Seevinck, C. A. van den Berg, and I. Išgum, "Deep MR to CT synthesis using unpaired data," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2017.

[15] D. Nie, R. Trullo, J. Lian, C. Petitjean, S. Ruan, Q. Wang, and D. Shen, "Medical image synthesis with context-aware generative adversarial networks," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2017.

[16] S. U. Dar, M. Yurt, L. Karacan, A. Erdem, E. Erdem, and T. Cukur, "Image Synthesis in Multi-Contrast MRI With Conditional Generative Adversarial Networks," *IEEE transactions on medical imaging*, 2019.

[17] M. Frid-Adar, I. Diamant, E. Klang, M. Amitai, J. Goldberger, and H. Greenspan, "GAN-based synthetic medical image augmentation for increased CNN performance in liver lesion classification," *Neurocomputing*, 2018.

[18] OpenStax, "Anatomy & physiology," *OpenStax CNX. Feb 26 http://cnx.org/contents/14fb4ad7-39a1-4eee-ab6e-3ef2482e3e22@8.24*, 2016.

[19] E. A. Celis, "Lung anatomy," *Medscape*, 2017.

[20] S. G. Armato, G. McLennan, L. Bidaut, M. F. McNitt-Gray, C. R. Meyer, A. P. Reeves, B. Zhao, D. R. Aberle, C. I. Henschke, E. A. Hoffman, E. A. Kazerooni, H. MacMahon, E. J. Van Beek, D. Yankelevitz, A. M. Biancardi, P. H. Bland, M. S. Brown, R. M. Engelmann, G. E. Laderach, D. Max, R. C. Pais, D. P. Qing, R. Y. Roberts, A. R. Smith, A. Starkey, P. Batra, P. Caligiuri, A. Farooqi, G. W. Gladish, C. M. Jude, R. F. Munden, I. Petkovska, L. E. Quint, L. H. Schwartz, B. Sundaram, L. E. Dodd, C. Fenimore, D. Gur, N. Petrick, J. Freymann, J. Kirby, B. Hughes, A. Vande Casteele, S. Gupte, M. Sallam, M. D. Heath, M. H. Kuhn, E. Dharaiya, R. Burns, D. S. Fryd, M. Salganicoff, V. Anand, U. Shreter, S. Vastagh, B. Y. Croft, and L. P. Clarke, "The Lung Image Database Consortium (LIDC) and Image Database Resource Initiative (IDRI): A completed reference database of lung nodules on CT scans," *Medical Physics*, 2011.

[21] "Non small cell lung cancer.." https://www.cancer.org/cancer/lung-cancer/about/key-statistics.html.

[22] "Cancer stat facts: Lung and bronchus cancer." https://seer.cancer.gov/statfacts/html/lungb.html.

[23] D. Harzheim, R. Eberhardt, H. Hoffmann, and F. J. Herth, "The solitary pulmonary nodule," 2015.

[24] D. Lucena, J. Ferreira Junior, A. Machado, and M. Oliveira, "Automatic weighing attribute to retrieve similar lung cancer nodules," *BMC Medical Informatics and Decision Making*, vol. 16, pp. 135–149, 07 2016.

[25] "Results of initial low-dose computed tomographic screening for lung cancer," *New England Journal of Medicine*, 2013.

[26] "What is emphysema." webmd.com/lung/copd/what-is-emphysema.

[27] Y. Li, S. Swensen, L. Karabekmez, R. Marks, S. Stoddard, R. Jiang, J. Worra, F. Zhang, D. Midthun, M. Andrade, Y. Song, and P. Yang, "Effect of emphysema on lung cancer risk in smokers: A computed tomography-based assessment," *Cancer prevention research (Philadelphia, Pa.)*, vol. 4, pp. 43–50, 01 2011.

[28] "Paraseptal emphysema and subpleural bullae." https://radiopaedia.org/cases/paraseptal-emphysema-and-subpleural-bullae?lang=gb.

[29] I. Goodfellow, "NIPS 2016 Tutorial: Generative Adversarial Modeling," *arXiv*, 2017.

[30] A. van den Oord, N. Kalchbrenner, and K. Kavukcuoglu, "Pixel recurrent neural networks," *CoRR*, vol. abs/1601.06759, 2016.

[31] S. Deb, "How to perform data compression using autoencoders?,"

[32] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," 2013.

[33] A. Hindupur, "The gan zoo." https://github.com/hindupuravinash/the-gan-zoo.

[34] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath, "Generative adversarial networks: An overview," *IEEE Signal Processing Magazine*, vol. 35, pp. 53–65, Jan 2018.

[35] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *CoRR*, vol. abs/1502.03167, 2015.

[36] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 2015.

[37] J. Wu, C. Zhang, T. Xue, W. T. Freeman, and J. B. Tenenbaum, "Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling," *CoRR*, vol. abs/1610.07584, 2016.

[38] X. Mao, Q. Li, H. Xie, R. Y. K. Lau, Z. Wang, and S. P. Smolley, "Least squares generative adversarial networks," in *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2813–2821, Oct 2017.

[39] N. Kodali, J. D. Abernethy, J. Hays, and Z. Kira, "How to train your DRAGAN," *CoRR*, vol. abs/1705.07215, 2017.

[40] D. Berthelot, T. Schumm, and L. Metz, "BEGAN: boundary equilibrium generative adversarial networks," *CoRR*, vol. abs/1703.10717, 2017.

[41] M. Lucic, K. Kurach, M. Michalski, S. Gelly, and O. Bousquet, "Are gans created equal? a large-scale study," 2017.

[42] X. Huang, Y. Li, O. Poursaeed, J. E. Hopcroft, and S. J. Belongie, "Stacked generative adversarial networks," *CoRR*, vol. abs/1612.04357, 2016.

[43] Y. LeCun and C. Cortes, "MNIST handwritten digit database," 2010.

[44] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. S. Bernstein, A. C. Berg, and F. Li, "Imagenet large scale visual recognition challenge," *CoRR*, vol. abs/1409.0575, 2014.

[45] A. Hertzmann, C. E. Jacobs, N. Oliver, B. Curless, and D. H. Salesin, "Image analogies," in *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '01, (New York, NY, USA), p. 327–340, Association for Computing Machinery, 2001.

[46] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," *CoRR*, vol. abs/1505.04597, 2015.

[47] T. Wang, M. Liu, J. Zhu, A. Tao, J. Kautz, and B. Catanzaro, "High-resolution image synthesis and semantic manipulation with conditional gans," *CoRR*, vol. abs/1711.11585, 2017.

[48] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," vol. abs/1511.05644, 2014.

[49] J. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," *CoRR*, vol. abs/1703.10593, 2017.

[50] D. Pfau and O. Vinyals, "Connecting generative adversarial networks and actor-critic methods," *CoRR*, vol. abs/1610.01945, 2016.

[51] T. Salimans, I. J. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training gans," *CoRR*, vol. abs/1606.03498, 2016.

[52] D. Geman, S. Geman, N. Hallonquist, and L. Younes, "Visual turing test for computer vision systems," *Proceedings of the National Academy of Sciences*, vol. 112, no. 12, pp. 3618–3623, 2015.

[53] M. J. M. Chuquicusma, S. Hussein, J. Burt, and U. Bagci, "How to fool radiologists with generative adversarial networks? A visual turing test for lung cancer diagnosis," *CoRR*, vol. abs/1710.09762, 2017.

[54] C. Szegedy, Wei Liu, Yangqing Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–9, June 2015.

[55] S. Barratt and R. Sharma, "A note on the inception score," 2018.

[56] A. Borji, "Pros and cons of gan evaluation measures," *Computer Vision and Image Understanding*, vol. 179, pp. 41 – 65, 2019.

[57] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, G. Klambauer, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a nash equilibrium," *CoRR*, vol. abs/1706.08500, 2017.

[58] Zhou Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, pp. 600–612, April 2004.

[59] X. Yi, E. Walia, and P. Babyn, "Generative adversarial network in medical imaging: A review," *CoRR*, vol. abs/1809.07294, 2018.

[60] C. Bowles, L. Chen, R. Guerrero, P. Bentley, R. N. Gunn, A. Hammers, D. A. Dickie, M. del C. Valdés Hernández, J. M. Wardlaw, and D. Rueckert, "GAN augmentation: Augmenting training data using generative adversarial networks," *CoRR*, vol. abs/1810.10863, 2018.

[61] Z. Zhang, L. Yang, and Y. Zheng, "Translating and segmenting multimodal medical volumes with cycle- and shape-consistency generative adversarial network," *CoRR*, vol. abs/1802.09655, 2018.

[62] A. Makhzani, J. Shlens, N. Jaitly, and I. J. Goodfellow, "Adversarial autoencoders," *CoRR*, vol. abs/1511.05644, 2015.

[63] T. N. L. S. T. R. Team, "Reduced lung-cancer mortality with low-dose computed tomographic screening," 2011.

[64] "Dicom processing and segmentation in python." https://www.raddq.com/dicom-processing-segmentation-visualization-in-python/.

[65] G. Aresta, A. Cunha, and A. Campilho, "Detection of juxta-pleural lung nodules in computed tomography images," in *Medical Imaging 2017: Computer-Aided Diagnosis* (S. G. A. III and N. A. Petrick, eds.), vol. 10134, pp. 952 – 958, International Society for Optics and Photonics, SPIE, 2017.

[66] A. Farag, J. Graham, and A. Farag, "Robust segmentation of lung tissue in chest ct scanning," pp. 2249–2252, 09 2010.

[67] M. C. Hancock and J. F. Magnan, "Lung nodule malignancy classification using only radiologist-quantified image features as inputs to statistical learning algorithms: probing the Lung Image Database Consortium dataset with two statistical learning methods," *Journal of Medical Imaging*, vol. 3, no. 4, pp. 1 – 15, 2016.

[68] Y. Li and L. Shen, "cc-gan: A robust transfer-learning framework for hep-2 specimen image segmentation," *IEEE Access*, vol. 6, pp. 14048–14058, 2018.

[69] S. Liu, Y. Wei, J. Lu, and J. Zhou, "An improved evaluation framework for generative adversarial networks," 2018.

[70] T. DeVries, A. Romero, L. Pineda, G. W. Taylor, and M. Drozdzal, "On the evaluation of conditional gans," 2019.

[71] C. Haarburger, N. Horst, D. Truhn, M. Broeckmann, S. Schrading, C. Kuhl, and D. Merhof, "Multiparametric Magnetic Resonance Image Synthesis using Generative Adversarial Networks," in *Eurographics Workshop on Visual Computing for Biology and Medicine* (B. Kozlíková, L. Linsen, P.-P. Vázquez, K. Lawonn, and R. G. Raidou, eds.), The Eurographics Association, 2019.

[72] K. Armanious, C. Yang, M. Fischer, T. Küstner, K. Nikolaou, S. Gatidis, and B. Yang, "Medgan: Medical image translation using gans," *CoRR*, vol. abs/1806.06397, 2018.

[73] T. Park, M. Liu, T. Wang, and J. Zhu, "Semantic image synthesis with spatially-adaptive normalization," *CoRR*, vol. abs/1903.07291, 2019.