



Thoracic MRI emulation through texture synthesis and ground truth manipulation

Mariana Ribeiro Dias

Mestrado Integrado em Bioengenharia

Supervisor: Prof. Helder Filipe Pinto de Oliveira

Co-supervisor: Prof. João Pedro Fonseca Teixeira

July 18, 2019

Thoracic MRI emulation through texture synthesis and ground truth manipulation

Mariana Ribeiro Dias

Mestrado Integrado em Bioengenharia

July 18, 2019

Resumo

O cancro da mama é uma patologia vastamente difundida na população feminina. Exames imagiológicos desempenham um papel fundamental no seu diagnóstico, o que motivou o desenvolvimento de métodos automáticos para detetar lesões na mama, nomeadamente em Ressonância Magnética e Mamografia. Contudo, estes métodos ainda não são amplamente usados na prática clínica devido à falta de imagens em quantidade suficiente para os validar, dado que há uma verdadeira dificuldade na aquisição de *datasets* médicos abrangentes. Este problema também afeta outras tarefas de deteção em imagem médica, tendo motivado publicações recentes relativas à síntese de imagens médicas artificiais através de modelos geradores.

Tendo isto em consideração, o nosso objetivo é sintetizar imagens de Ressonância Magnética torácica artificiais a partir de anotações médicas manuais usando, para tal, exames reais e técnicas do estado da arte. Isto permitir-nos-á equipar investigadores da área da imagiologia médica com as imagens que têm carecido para validar os seus algoritmos. Pelo que sabemos, não existem quaisquer publicações anteriores relativas a este tema.

Nesta dissertação, investigamos o uso de três diferentes *frameworks* – a *pix2pix*, a *pix2pixHD* e a CycleGAN – todas elas baseadas em redes geradoras adversárias, para atingir o nosso objetivo. Para verificar se alguma das *frameworks* pode ser melhorada, propomos várias variantes para cada uma, nomeadamente quanto à arquitetura do gerador, função de custo e quantidade de canais de *input* – anotações categóricas, com um canal, versus anotações multicanal, nas quais cada canal corresponde a uma imagem binária referente a cada classe presente na anotação categórica. Os resultados obtidos indicam que arquiteturas de gerador que usam blocos de *residual learning* são eficientes para a síntese de imagem e que anotações categóricas, com um canal, codificam informação de forma mais eficaz do que anotações multicanal. Adicionalmente, foi provada a eficácia da arquitetura de gerador *coarse-to-fine* proposta na *pix2pixHD*, bem como a importância dos componentes de *feature matching* e *perceptual loss* da função de custo proposta no mesmo modelo. A *pix2pixHD* e a CycleGAN foram as *frameworks* mais eficazes para a síntese.

Adicionalmente, contribuímos para o debate acerca da avaliação das redes geradoras adversárias ao propor o uso da Informação Mútua Normalizada para avaliar a semelhança entre imagens, e ao usar *Global Convolutional Network* para segmentar e comparar imagens verdadeiras e falsas. Também fornecemos uma análise detalhada da métrica *Fréchet Inception Distance*, recentemente proposta para avaliar a performance de modelos generativos mas, até à data, escassamente estudada.

Abstract

Breast cancer is a widespread pathology amongst the female population. Imageology exams play a key role in its diagnosis, which motivated the development of automated methods to perform that detection, mainly in thoracic Magnetic Resonance Imaging and Mammography exams. Nevertheless, these methods are not yet broadly used in a clinical context due to the lack of sufficient images to validate them, since comprehensive medical image datasets are hard to collect. This problem also affects other medical detection tasks and motivated recent works concerning the synthesis of artificial medical images using generative models.

Bearing this in mind, our goal is to synthesize artificial thoracic Magnetic Resonance Imaging exams from corresponding annotations using real exams and state of the art techniques, equipping medical-imaging researchers with the data they are lacking to validate their algorithms. To the best of our knowledge, no previous works concerning this particular topic exist.

In this dissertation, we investigate the use of three different frameworks - the *pix2pix*, the *pix2pix_{HD}* and the CycleGAN - all based in generative adversarial networks, to achieve our goal. To assess if any of the frameworks can be improved, we propose several variants of each, namely concerning the generator network's architecture, loss function and the number of input channels - single-channel, categorical annotations versus one-hot encoded annotations, in which each channel corresponds to a binary image referring to each label of the categorical annotation. Our results indicate that generator architectures with residual learning blocks are efficient for the image synthesis and that single-channel annotations convey information with more efficacy than the one-hot encoded ones. Furthermore, we proved the greater efficacy of the coarse-to-fine generator architecture proposed in the *pix2pix_{HD}*, as well as the importance of the feature matching and perceptual loss components proposed in the same framework. The CycleGAN and the original *pix2pix_{HD}* frameworks were the most successful for the task at hands.

Additionally, we contribute to the debate concerning the evaluation of generative models by proposing the use of the normalized mutual information to assess image similarity and the use of the Global Convolutional Network for the segmentation and comparison of real and fake images. We also provide a deeper analysis into the Fréchet Inception Distance metric, a recently proposed but under-studied measure to evaluate generative models.

Acknowledgments

Mais do que um trabalho, esta dissertação representa o culminar de um percurso de cinco anos na Faculdade de Engenharia, que foi marcado por várias pessoas às quais gostaria de agradecer.

Primeiramente, os meus pais e a minha irmã, pelo seu apoio incondicional, conselhos e companhia nos bons e maus momentos.

Aos meus amigos, pelas mil e umas aventuras vividas e por terem estado sempre presentes para ouvirem os altos e baixos da execução desta dissertação.

Aos meus orientadores, pelas oportunidades que me proporcionaram ao longo dos últimos dois anos, pelos ensinamentos, e por terem estado sempre disponíveis a ouvir e debater as minhas ideias, tanto as que tinham sentido como as que não.

A todos, um gigante obrigada.

Mariana Dias

Contents

1	Introduction	1
1.1	Motivations	1
1.2	Goals and Contributions	2
1.3	Document structure	3
2	Medical background	5
2.1	Anatomy and physiology of the chest wall	5
2.1.1	Skeletal composition	6
2.1.2	Musculature	6
2.1.3	Vasculature	8
2.1.4	Surface anatomy: skin and breasts	8
2.2	Breast imageology	9
2.2.1	Magnetic Resonance Imaging	9
2.2.2	Thoracic Magnetic Resonance Imaging	9
2.3	Summary	10
3	Texture analysis and synthesis	11
3.1	Pixel-based synthesis	12
3.2	Patch-based synthesis	13
3.3	Surface texture synthesis	13
3.4	Solid texture synthesis	14
3.5	Deep-Learning methods	14
3.6	Summary	17
4	Generative Modeling	19
4.1	Generative Adversarial Networks	19
4.1.1	Principles	19
4.1.2	Conditional Generative Adversarial Networks	23
4.1.3	Image-to-image translation applications	24
4.2	Variational Autoencoders	31
4.2.1	Conditional Variational Autoencoders	33
4.2.2	Adversarial Autoencoders	33
4.3	Applications for medical data	34
4.4	Evaluation of generative models	35
4.4.1	Perceptual studies	35
4.4.2	Inception Score	35
4.4.3	Fréchet Inception Distance	36
4.4.4	Peak Signal-To-Noise Ratio	37

4.4.5	Structural Similarity Index	37
4.4.6	Semantic Segmentation Metrics	38
4.5	Summary	38
5	Thoracic MRI synthesis	41
5.1	Methods	41
5.1.1	Synthesis	41
5.1.2	Evaluation	44
5.2	Dataset	48
5.3	Implementation details	49
5.4	Results and discussion	49
5.4.1	Fréchet Inception Distance study	49
5.4.2	Thoracic MRI segmentation	52
5.4.3	Thoracic MRI synthesis	54
5.5	Summary	63
6	Conclusion	65
6.1	Overview and contributions	65
6.2	Future work	66
A	Tables	67
B	Visual results	71
	References	75

List of Figures

2.1	Anatomical directional references.	5
2.2	The thoracic skeleton	6
2.3	Thoracic muscles	7
2.4	Breast anatomy.	8
3.1	Orientation field creation, for surface texture synthesis	13
3.2	Architecture proposed by Leon Gatys	15
3.3	Results of the synthesis using the method proposed by Leon Gatys	16
4.1	GANs framework.	20
4.2	SGANs architecture.	22
4.3	CGANs framework	24
4.4	Exemplifying result for the application of the <i>pix2pix</i> for the generation of images from the Inria Aerial Image dataset.	26
4.5	Coarse-to-fine generator of the <i>pix2pix_{HD}</i> framework.	27
4.6	Comparative results for the application of the <i>pix2pix</i> , <i>pix2pix_{HD}</i> and CRN frameworks for the generation of images from the Cityscapes dataset.	28
4.7	Cycle GAN framework.	29
4.8	Exemplifying result for the application of the CycleGAN for photo to painting conversion, for different painting styles.	30
4.9	Comparative result for the application of the CycleGAN and <i>pix2pix</i> frameworks for the synthesis of images of the Inria Aerial Image dataset.	30
4.10	Autoencoder architecture	31
4.11	Comparison of the mapping from the input to the latent space on standard Autoencoders and VAEs	31
4.12	VAE framework.	33
5.1	U-Net based generator architecture.	43
5.2	Residual learning block from the ResNet.	43
5.3	ResNet based generator architecture.	44
5.4	Coarse-to-fine generator architecture.	44
5.5	Architecture of the 30×30 PatchGAN.	45
5.6	Comparison of the general architectures used for classification and segmentation with the architecture proposed by the GCN.	46
5.7	FCN architecture variants	47
5.8	GCN architecture.	48
5.9	Illustration of the automatic label generation for the thoracic cavity.	48
5.10	Reference and tampered <i>Lena</i> images.	50
5.11	Reference and tampered MRI images.	51

5.12	Examples of less accurate breast segmentations in the validation set.	53
5.13	Examples of less accurate thoracic cavity segmentations in the validation set. . .	53
5.14	Examples of clavicle segmentation in the validation and test sets.	54
5.15	Synthetic images produced by <i>pix2pixV1</i> , <i>pix2pixV1_{OHE}</i> , <i>pix2pixV2</i> , <i>pix2pixV2_{OHE}</i> and <i>pix2pixV3</i> for the same input annotation. On the bottom right, the correspond- ing real MRI image.	56
5.16	Synthetic images produced by <i>pix2pix_{HD}V1</i> and <i>pix2pix_{HD}V2</i> for two different input annotations.	57
5.17	Synthetic images produced by the CycleGAN.	58
5.18	Synthetic images produced by the top 3 best performing models according to the FID: the <i>pix2pixV3</i> , the <i>pix2pix_{HD}V1</i> and the Cycle GAN.	59
5.19	Graphics indicating the loss progression for the generator and discriminator for the <i>pix2pixV1</i> model with the hyper-parameter sets HV_1 , HV_2 and HV_3	61
5.20	Graphics indicating the loss progression for the generator and discriminator for the <i>pix2pixV2</i> model with the hyper-parameter sets HV_1 , HV_2 and HV_3	62
5.21	Graphics indicating the loss progression for the generator and discriminator for the <i>pix2pixV3</i> model with the hyper-parameter sets HV_1 , HV_2 and HV_3	62
5.22	Manipulated annotations and corresponding synthetic MRI images produced by the CycleGAN.	63
B.1	Less successful cases in the synthesis of the clavicle.	71
B.2	Less successful cases in the synthesis of the nipple.	72
B.3	Less successful cases in the synthesis of the sternum.	73

List of Tables

4.1	Generative modeling applications for medical imaging	34
5.1	Summary of all models used for the synthesis task	45
5.2	Reference FID values	51
5.3	GCN validation and test Dice scores	52
5.4	Global image evaluation scores	54
5.5	Normalized Mutual Information scores of each labeled structure	55
5.6	Dice scores of each labeled structure	55
A.1	Pixel-based texture synthesis methods	67
A.2	Patch-based texture synthesis methods	67
A.3	Surface texture synthesis methods	68
A.4	Solid texture synthesis methods	69
A.5	Deep-Learning methods for texture synthesis	70

Abbreviations

BC	Breast Cancer
MRI	Magnetic Resonance Imaging
2D	Two dimensions
3D	Three dimensions
CNN	Convolutional Neural Network
GAN	Generative Adversarial Networks
CGAN	Conditional Generative Adversarial Networks
VAE	Variational Autoencoder
CVAE	Conditional Variational Autoencoder
AAE	Adversarial Autoencoder
KL	Kullback-Leibler
MI	Mutual Information
NMI	Normalized Mutual Information
SSIM	Structural Similarity Index
FID	Fréchet Inception Distance
IS	Inception Score
PSNR	Peak Signal-To-Noise Ratio
VTT	Visual Turing Test
GCN	Global Convolutional Network
ReLU	Rectified Linear Unit

Chapter 1

Introduction

The word Cancer is used to describe an unusual cell modification and its uncontrolled proliferation, which frequently generates cell masses formally known as tumors. Breast Cancer (BC) can either originate in the breast tissue or in the ducts that connect the breasts' glands to the nipples [Ame17]. Its most common symptoms are palpable and painless lumps. In its early stages, the BC tumors are small, easily treatable, but asymptomatic, making them usually go undetected until the disease progresses further (BC cells tend to spread to the underarm lymph nodes). As so, periodic screening is crucial for timely detection and successful treatment of BC.

Imageology exams play a central role in BC screening and diagnosis. The American Cancer Society's recommendations for the early detection of BC [SBB⁺07] differ according to the woman's age and family history - for women with a high risk of developing BC and/or with dense breast tissue, periodic Mammography or thoracic Magnetic Resonance Imaging (MRI) exams are recommended.

According to the World Health Organization, BC is the most frequent cancer among women, affecting 2.1 million women each year, which makes it the deadliest cancer within the female population [Wor19]. In 2018, it is estimated that BC caused the death of approximately 627,000 women, which corresponds to 15% of all cancer deaths amongst women. Despite BC incidence being higher in developed countries, incidence rates are rising globally. This has served as motivation for the development of automatic methods for BC detection, as well as for the evaluation of benign nodules, both in Mammography and MRI, such as [ZYM18]. Nevertheless, these methods lack satisfactory accuracy for clinical use, which reveals the necessity of further strategies to improve them.

1.1 Motivations

Coping with small, unbalanced and poorly annotated datasets has been a recurring problem for medical image analysis researchers. Consequently, this has limited their success in validating supervised learning algorithms for real-life application, which requires extensive amounts of labeled data.

The paucity of comprehensive and annotated medical data is due to several factors - on one hand, acquiring medical images often requires expensive and invasive procedures; on the other hand, annotating medical images is a time-consuming task that requires the labor of experienced radiologists, specialized in a particular image type. Medical image datasets are usually very unbalanced since abnormal images are captured less frequently than normal ones.

Traditional data-augmentation techniques (e.g. rotation, translation, crop) produce training samples highly correlated with already existing ones which, by itself, is insufficient to counteract the consequences of data scarcity. This, together with the data needs that accompanied the expansion of Deep Learning, served as motivations for the development of techniques to generate meaningful synthetic data and fast-forwarded research on this topic in recent years.

Due to the high incidence of breast-related pathologies, a great amount of progress has been made in the automatic detection of breast lesions in thoracic MRI. The development and validation of those methods, as well as others, depends immensely on the availability of data which, to this date, keeps being scarce. This comes to show that the artificial generation of data is a long coming potential solution in the medical image analysis field and is a key step in the advancement of engineering and medical practices.

1.2 Goals and Contributions

The purpose of this research is to implement an efficient method to synthesize artificial thoracic MRI exams from manually drawn annotations, helping to overcome the data scarcity that has been delaying the clinical use of automated algorithms to detect breast lesions. We use generative models to synthesize the MRI textures, and generate exams from new annotations, created by manipulating existing ones. Specifically, our contributions to the scientific community are the following:

- Research and application of Deep Learning methods, namely generative models, for the synthesis of thoracic MRI exams;
- Research and application of Deep Learning methods for the segmentation of the key structures existing in thoracic MRI exams, namely state of the art semantic segmentation networks;
- Provide a deeper analysis into the Fréchet Inception Distance metric for the evaluation of generative models;
- Research and development of novel metrics to evaluate the performance of generative models;
- Article submission for the 20th International Conference on Image Analysis and Processing.

1.3 Document structure

The remaining of this dissertation is organized as follows:

- Chapter 2 overviews the anatomy of the thorax, focusing on the chest wall and breasts, finishing with a brief analysis of the principles behind MRI and its use for breast imaging;
- Chapter 3 reviews general concepts concerning texture analysis and synthesis;
- Chapter 4 provides a detailed literature review concerning generative models, with focus on Generative Adversarial Networks and Variational Autoencoders, and their respective subclasses, addressing the already existing applications of generative models for the synthesis of medical data. It also overviews the most frequent metrics used to evaluate the performance of generative models, as well as novel strategies to do so;
- Chapter 5 details all models and metrics used for the practical work executed, as well as the dataset and concrete implementation details. This is followed by a presentation of all results obtained and posterior analysis;
- Chapter 6 concludes this dissertation with an overview of our accomplishments and contributions, and with suggestions for future work;
- Appendix A contains several tables, each one summarizing the main works of each class of texture synthesis methods approached in Chapter 3;
- Appendix B contains additional synthetic images for further analysis by the reader.

Chapter 2

Medical background

This chapter reviews the fundamentals of the chest wall and breasts anatomy, from a macroscopic standpoint. We address separately the skeletal, muscular and vascular components of the region. This anatomy analysis is based on the descriptions presented in [GL16], [PG16], [RG18], [CEMA11], and [SS17]. Additionally, we overview the principles behind MRI, as well as the motivations and protocols to perform thoracic MRI for the evaluation of breast pathologies. Figure 2.1 illustrates some of the most common terms used in anatomical descriptions such as ours.

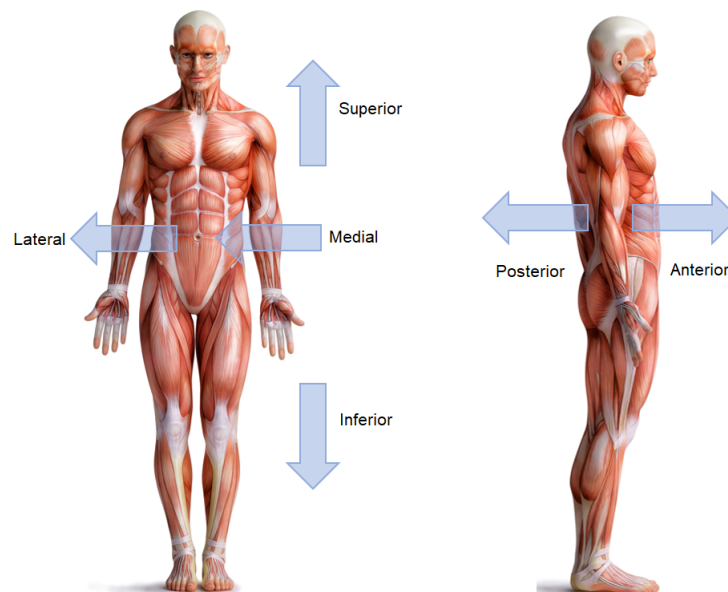


Figure 2.1: Anatomical directional references. Adapted from: [Sci]

2.1 Anatomy and physiology of the chest wall

The chest wall is a complex system that encases the chest organs (e.g. heart, lungs, and liver), stabilizes arm and shoulder motion, and supports respiratory movements. Below, we examine the chest wall's skeletal, muscular, vascular and surface anatomy, including the breasts' anatomy.

2.1.1 Skeletal composition

The thoracic skeleton is composed, posteriorly, by 12 thoracic vertebrae (T1 through T12) and, anteriorly, by the sternum and 12 pairs of ribs and costal notches. The ribs have an arched shape, wrapping around laterally and anteriorly, providing flexibility and allowing the absorption of kinetic energy in case of trauma. Figure 2.2 details the components of the thoracic skeleton.

The first 7 pairs of ribs (known as true ribs) are directly connected to the sternum through their own costal cartilages, while the 8th, 9th, and 10th pairs (known as false ribs) are not: their costal cartilages are merged, forming an indirect connection to the sternum.

The sternum is a long and flat bone that marks the medial anterior limit of the chest. It is composed of three differently sized sections - from top to bottom, the manubrium, the body, and the xiphoid process. It has a convex curvature and serves as a platform for the attachment of muscles of the thorax, neck, back, abdomen and upper limbs. The transition between the manubrium and the body, in continuity with the second rib, forms an angle known as sternal angle or angle of Louis. It is located at the same level of the tracheal bifurcation and near the upper border of the atria of the heart.

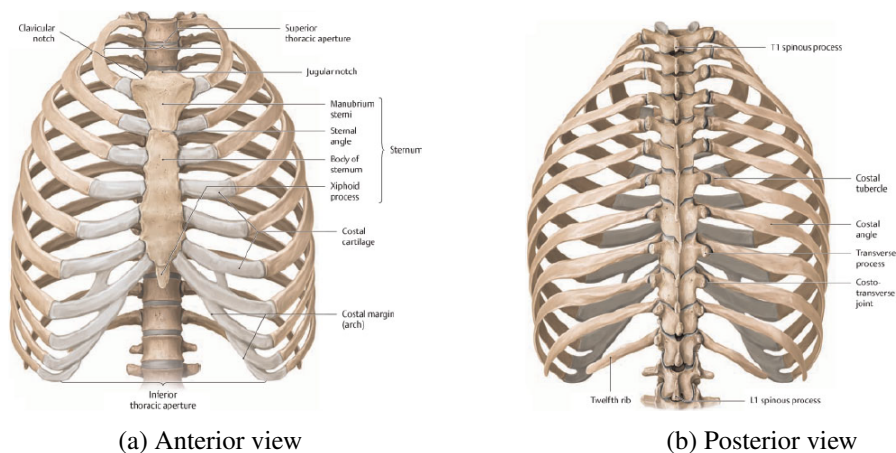


Figure 2.2: The thoracic skeleton. Source: [CEMA11].

2.1.2 Musculature

There are two sets of chest muscles: the inspiratory and the expiratory. The inspiratory set is responsible for elevating the rib cage and expanding the chest during inspiration, comprising the sternocleidomastoid, the scalenes, the external intercostal (for rib elevation), and the *pectoralis minor*. The expiratory set constricts the rib cage, deflating the lungs in a downward motion during expiration, and is composed by the *rectus abdominis*, the internal intercostals (for rib depression), and the internal and external obliques. The diaphragm belongs to both sets. Other relevant muscles are the *pectoralis major*, the *latissimus dorsi*, the *serratus anterior* and the trapezius. Below, we detail the location and insertions of the most relevant muscles of the thorax. Figure 2.3 shows the musculature of the thorax.

Pectoralis major and pectoralis minor: the *pectoralis major* covers the anterior superior portion of the chest, having proximal insertions at the medial clavicle, lateral sternum, and superior six costal cartilages, and a distal insertion at the intertubercular groove of the humerus. The *pectoralis minor* is inserted, proximally, in the 3rd to 5th ribs, near their costal cartilages and, distally, on the medial border and superior surface of the coracoid process of the scapula.

Intercostal muscles: from the exterior to the interior, each intercostal space is covered by the external intercostals, the internal intercostals and a thin layer of muscle with the same orientation as the previous layer. Neurovascular bundles separate the middle and innermost layers. Both the internal and external are obliquely oriented, but in different directions: respectively, anterior-inferiorly and posterior-inferiorly.

Latissimus dorsi: it is largest muscle of the body and has insertions on the lower six thoracic vertebrae, crest of the ileum, and intertubercular sulcus of the humerus.

Serratus anterior: it is composed of several muscle slips which are inserted on the upper borders of the top eight ribs and on the surface of the medial scapula, covering the anterolateral region of the chest wall. It contributes for the abduction and flexion of the shoulder, besides holding the scapula against the rib cage.

Rectus abdominis: it is an extensive, vertical muscle of the abdominal wall, and has insertions on pubic symphysis, xiphoid process and 5th to 7th costal cartilages.

Oblique muscles: the external oblique muscle is a broad muscle that is spreads along the anterolateral abdomen and chest wall. It has insertions on the 6th through 12th, along the anterior half of the iliac crest, on the linea alba, and on the pubic and inguinal ligaments. The internal oblique has attachments along the iliac crest, on the lateral half of the inguinal ligament, and on the inferior borders of the 10th to 12th ribs. Its fibers run in an inferior lateral direction, oppositely to the external oblique. It is placed in the middle of the 3 abdominal muscles (external obliques, internal obliques, and *rectus abdominis*).

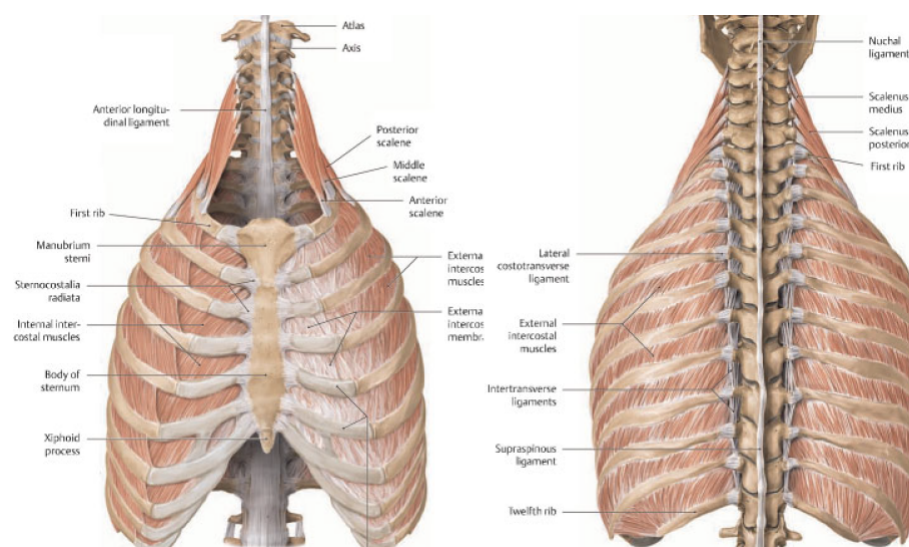


Figure 2.3: Thoracic muscles

2.1.3 Vasculature

The blood supply to the sternum is provided by the internal mammary arteries (from above), and by the acromiothoracic and transverse cervical arteries (laterally). The internal mammary artery emerges from a branch of the subclavian artery and spreads behind the costal cartilages alongside the sternum. The ventral skin and muscles are also irrigated by branches of the subclavian vessels, and also by deep epigastric arteries. Cutaneous perforators that originate from the above-mentioned vessels irrigate the *pectoralis major* muscle, the costal margin, and the region of intersection of the *serratus anterior* muscle with the midaxillary line.

2.1.4 Surface anatomy: skin and breasts

The **chest skin** comprises a moderate dermal layer, irrigated by arterial perforators. In younger women, the nipples lie above the inframammary creases, are aligned with the fourth intercostal space, and are placed laterally to the midclavicular line. The axillary tail of the breast prolongs obliquely upwards until the medial wall of the axilla.

Concerning the **breasts**, their base lies over the *pectoralis major* muscle, someplace between the 2nd and 6th ribs, as described in [PG16]. The base is attached to the *pectoralis major* fascia through the Cooper ligaments, which are directly anchored to the dermis. The breasts are also bordered medially by the sternum, superiorly by the clavicles, laterally by the *latissimus dorsi* and inferiorly by the *rectus abdominis* muscle. Figure 2.4 summarizes the breasts' anatomy.

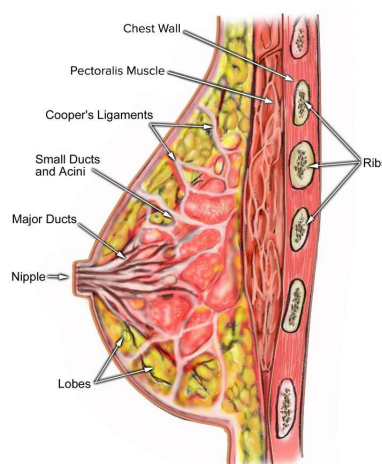


Figure 2.4: Breast anatomy. Source: [GL16].

The **breasts are composed** of fatty and glandular tissues intercalated with fibrous ligaments, which allows their suspension. These ligaments relax with weight and age, which results in the ptosis of the breasts. The ratio between the fatty and glandular tissues depends mostly on the women's age and hormonal exposure (the ratio reduces in menopause since the estrogen production decreases).

The **breasts' blood supply** is ensured by the internal mammary perforators (which derive from the internal mammary arteries and provide 60% of the breasts' irrigation), the lateral thoracic artery, the thoracoacromial artery, the vessels involved in the irrigation of the *serratus anterior* muscle, and by terminal branches of some of the intercostal perforators.

2.2 Breast imageology

Breast-related pathologies can be evaluated through several imageology techniques, including Mammography, MRI and Ultrasound. As stated in Chapter 1, our interest is in MRI, so we will perform a brief overview of the principles behind this technique as well as the guidelines for its use for breast imaging.

2.2.1 Magnetic Resonance Imaging

MRI is an imageology technique that exploits the existence of induced nuclear magnetism in the human body [CS18]. It is based on the fact that atoms that possess an odd number of protons have a magnetic moment. These nuclear moments are, normally, randomly oriented. However, when placed under a constant external magnetic field, such as the one generated by an MRI machine magnet, they become aligned with the field, producing a measurable magnetic moment.

The interaction between the magnetic moment of the nucleus and the external field causes each spinning nucleus to change its rotation orientation - this phenomenon is known as nuclear precession. Each nucleus precesses at a characteristic (resonant) frequency, proportional to the strength of the external magnetic field, that can be calculated using the Larmor equation [CMP06]. The density of protons and the relaxation times (the rate at which the magnetization returns to its equilibrium) are used to acquire the images.

Due to the high abundance of hydrogen nuclei in the tissues of the human body compared with other atomic nuclei, the proton signal from the hydrogen atom is the main source of signal used to acquire the images. As so, structures with higher hydrogen concentration, such as tissues rich in lipid content (e.g. the breasts) will be displayed brighter than the ones with a lower concentration (e.g. bones).

2.2.2 Thoracic Magnetic Resonance Imaging

Thoracic MRI is a non-invasive imaging technique used to captures high-quality images of the breasts. It is a complimentary exam to traditional breast imaging techniques (Mammography and Ultrasound) [GL15].

Initially, it was performed without injecting any contrast agent, therefore T1 and T2 weighted protocols were necessary to obtain relevant acquisitions. However, it quickly became clear that T2 relaxation rates of benign and malignant tissues overlap, so *in situ* cancers were poorly detected. Some of the developments that allowed to overcome this poor detection include dedicated breast coils, rapid 2D gradient-echo imaging, stronger magnets (>1T, enabling spectral fat-suppression),

k-space filling methods (which provided higher resolution and speed), and the injection of gadolinium dimeglumine as a contrast agent. Gadolinium-based contrast agents shorten T1 and increase tissue relaxation rates, leading to a higher signal intensity.

Contrast-enhanced MRI is significantly more sensitive to breast malignancies than conventional breast imaging techniques. Non-contrast MRI is solely recommended to assess the integrity of breast implants.

According to guidelines provided by the American Cancer Society, yearly thoracic MRI should only be performed in women with high-risk factors for breast cancer [Ame17]. It is also recommended for the assessment of breast cancer progression, to characterize lesions which diagnosis was uncertain after other exams, to monitor tumor recession after chemotherapy, and for post-operative evaluation of patients with positive margins.

The main advantages of performing MRI in detriment of the remaining exams are the use of non-ionizing radiation, its high spatial resolution, the possibility of multiplanar acquisition, the capture of images of the entire breast volume and chest wall, better soft tissue contrast, and the detection of occult, multifocal and residual malignancies. However, it is a costly technique that requires injection of a contrast agent, doesn't have a standard protocol, enhances *in situ* carcinomas non uniformly, generates more false positives and generates a large number of images, which interpretation requires a more specialized radiologist than mammography and US. It is also less widely available than Mammography and Ultrasound.

2.3 Summary

This chapter presented a brief overview of the anatomy of the chest wall and breasts, which are the regions captured in the dataset used for our work. Some of the key aspects reviewed are the relative position of the structures that compose the thorax, as well as how their tissue composition affects their representation in MRI, which are features we'll have to make sure that are maintained when synthesizing artificial thoracic MRI exams.

Lastly, we reviewed the fundamental principles behind MRI and the guidelines for its application for breast imaging, showing the relevance of this imageology technique to evaluate breast pathologies.

Chapter 3

Texture analysis and synthesis

There is no formal consensual definition of texture. In general terms, texture refers to a complex visual pattern composed of primitives - local entities with specific characteristics of color, slope, luminosity, size, among others [TJ98]. The repetition of primitives often incorporates some randomness. Textures may be characterized by a wide range of features, namely lightness, regularity, roughness, frequency, phase, direction, randomness, coarseness, smoothness, granulation, amid others.

Textures characterize areas, so they must be analyzed over a spatial neighborhoods. The size of the neighborhood must be in conformity with the size of the primitives. Different primitives in different texture patterns, or even in the same pattern, can have different sizes, so a multi-scale analysis is often necessary.

Texture analysis techniques include structural, statistical, model-based and transform-based methods. **Structural methods** describe textures as hierarchies of primitives (or micro-textures), which implies the prior definition of the primitives and their placement rules. Placing each primitive may be conditioned by location or the neighbor primitives of that location. Despite providing a structured symbolical description of the images, structural methods fail in reflecting the variability of both macro and micro-textures, making the inaccurate for the analysis of natural textures.

Statistical methods describe textures by non-deterministic properties that express the relation between pixel intensities, especially second order statistics. They are inspired in research works concerning texture discrimination by the Human Visual System, such as [JGSF73], which shows that regions of an image with different second-order statistics are instantly distinguished, while regions with equal second order moments but different third order ones require a superior cognitive effort to be differentiated. It has been shown that statistical methods that rely on second order statistics provide better discrimination results than both structural and transform-based methods. The most popular second order statistical features are energy, entropy, contrast, homogeneity, and correlation, and can all be extracted through co-occurrence matrices.

Model based methods create image models that can be used not merely to describe textures but also to synthesize them. The parameters of the models must capture the perceived traits of the texture. The two principal classes of models are Random Field Models and Fractals. Fractals refer

to patterns composed of repeated occurrences of a primitive or motif at different scales, which occurs frequently in Nature. Markov Random Fields (MRF) are widely used to model images, as they are able to capture local contextual information in an image. These models encompass two properties: locality (the intensity of each pixel depends only on the intensities of the neighboring pixels) and stationary (the locality property is true for every pixel) [TJ98].

Concerning **Transform methods**, research in the Psycho-physical field has proven that the brain’s visual cortex performs a frequency analysis when inspecting images and that, within that analysis, different cell groups respond to different frequencies and orientations [CR68]. It has been observed that those responses are similar to Gabor functions, which is a strong motivation to perform frequency analysis in the computer vision field. The most common methods for frequency analysis are the Fourier, Gabor, and Wavelet transforms.

Most classic texture synthesis methods are by-example methods in which, given an input texture sample, a new texture image of arbitrary size must be synthesized in such a way that the input texture and the artificially generated one look different from one another, while appearing to be generated by the same stochastic process. Most texture synthesis algorithms assume that the synthesis process complies with the locality and stationary properties described above. Hence, texture synthesis methods must generate new pixels in a way that allows each new output pixel to have a spatial neighborhood similar to at least one neighborhood of the input sample, ensuring the perceptual resemblance between the input and output textures.

Throughout this chapter, we will build upon the previous analysis and briefly explain the main classes of classical by-example texture synthesis methods, detailing with greater detail how Deep Learning approaches handle this task. Our goal is to assess if texture synthesis methods can be useful to synthesize thoracic MRI images. Only static texture synthesis methods will be analyzed since dynamic texture synthesis (time-varying textures) doesn’t fall into the scope of our goals.

3.1 Pixel-based synthesis

Pixel-based synthesis consists of selecting a seed region from an input texture sample and using it as a starting point to grow the rest of the texture. New layers of pixels are grown starting at the seed and progressing outwards. The synthesis of each output pixel usually relies on a neighborhood search. Table A.1 compiles the most important works found in the literature regarding this class of methods. The work presented in [EL99] pioneered this class of methods and proposes an inside-out synthesis starting with a small, randomly selected, texture patch. The texture is grown one pixel at the time and it requires a time-consuming and exhaustive neighborhood search, which may result in be non-uniform textures or simply copies of the starting patch. Subsequent works improve upon these limitations. For instance, [WL00] synthesizes new textures starting from a noise image, with the desired size. Both the texture sample and the noise image are decomposed through multilevel image pyramids, and each noise level is modified to match the features of the corresponding texture level, also using a time-consuming neighborhood search - however, it produces textures with less artifacts.

3.2 Patch-based synthesis

Patch synthesis emerged to improve the quality and speed limitations of pixel-based methods. This class of methods focuses on synthesizing patches of pixels rather than individual pixels, improving local coherence. Patch-based synthesis can be interpreted as an extension of pixel-based synthesis since the construction of new patches also relies on a neighborhood search, which means that the associated computational burden still exists, but at a lower scale. However, the patch copying process is not as simple as the pixel copy since there is overlap between new patches and previously synthesized ones. Algorithms differ in the way they handle this overlap, and two main situations can happen: either new patches overwrite older ones, as in [PFH00] (patches are stitched together using the minimum cost path of the overlap region), or the overlapped regions are blended together, as in [EF01] and [KSE03] (uses graph-cuts). While the first strategy may produce visible seams, the second may cause blurriness. Table A.2 summarizes the patch-based synthesis methods discovered in the literature review and their principal aspects.

3.3 Surface texture synthesis

Many texture synthesis applications require the placement of texture over curved surfaces. Table A.3 lists the methods of this type found in the literature and summarizes their hallmarks. Early approaches to this problem created flat textures and attempted to wrap them onto the surfaces. However, these texture mapping strategies originate issues such as distortion and noticeable seams. These limitations gave rise to procedures that synthesize textures tailored to particular surfaces, which is the current standard approach to texture synthesis over surfaces. All procedures of this type normally involve two steps:

- Creation of an orientation field over the surface (or another equivalent representation), specifying the direction of the texture as it runs across the surface. A vector field associates to each point on the surface, a vector that is tangent to the surface. Only a few vectors must be defined by the user, as these will guide the creation of the remainder of the field, as Figure 3.1 illustrates.

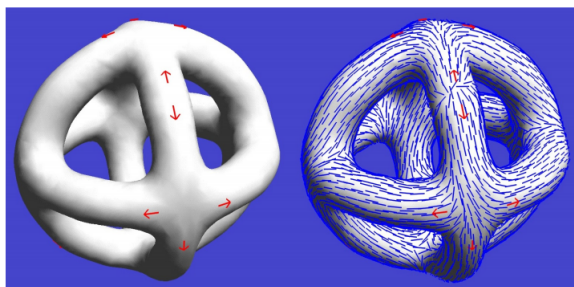


Figure 3.1: Orientation field creation, for surface texture synthesis. On the left, the user defined vectors; on the right, the final vector field. Source:[WLKT09].

The user-defined constraints can be converted into a complete orientation field through different tactics, such as parallel transport (moving each vector along a path that allows the vector to always maintain the same angle with the path), used in [PFH00], or by treating the user defined vectors as boundary conditions for vector-valued diffusion, as performed in [Tur01].

- Texture synthesis according to the orientation field. This comprises three classes of methods:
 - **Classical pixel-by-pixel synthesis**, which relies on surfaces with a high point density. These points can either be placed randomly, such as in [Tur91], or through an hierarchical order. Some of the methods that follow this strategy are [WL01] and [Tur01];
 - **Mapping between regions of the plane and the surface**, which allows synthesis to be carried out in the plane. [YHBZ01] and [LH06] both use this strategy, creating a texture map;
 - **Treating triangles of a mesh as texture patches**, as in [SCA02].

3.4 Solid texture synthesis

Solid texture synthesis aims to generate a textured volume from a 2D texture sample. Each slice of the volume must be visually similar to the 2D sample. As so, algorithms try to generate volumes in which each 2D slice looks like the 2D image sample while also displaying continuity between slices, which is considerably more challenging than simply generating a new 2D texture. Some methods, such as [DLTD08], take as input several 2D images, representative of different planes of the volume, to achieve more realistic results. Table A.4 summarizes the most important solid texture synthesis works found in the literature.

3.5 Deep-Learning methods

With the advent of Deep-Learning, new strategies for texture synthesis and artificial data generation arose. Table A.5 summarizes the most relevant Deep-Learning based methods found in the literature, focusing on CNN based approaches, and excluding generative modeling methods, since these will be explored in Chapter 4.

One of the first Deep-Learning approaches to texture synthesis was proposed in [GEB15]. This work introduces a parametric texture model that results from the combination of a CNN and spatial statistics. The architecture outline is presented in Figure 3.2.

Concretely, the framework presented comprises two sections: one for texture analysis and another for synthesis. Both use a VGG-19 CNN, pre-trained for an object recognition task. Only 16 convolutional layers and 5 pooling layers from the network were used and all fully connected layers were discarded. The authors modified two aspects of the conventional VGG-19 architecture:

- Max poolings were changed to average poolings since average poolings improved gradient flow and produced cleaner results;
- The weights of the network were re-scaled so that the mean activation of each filter was one.

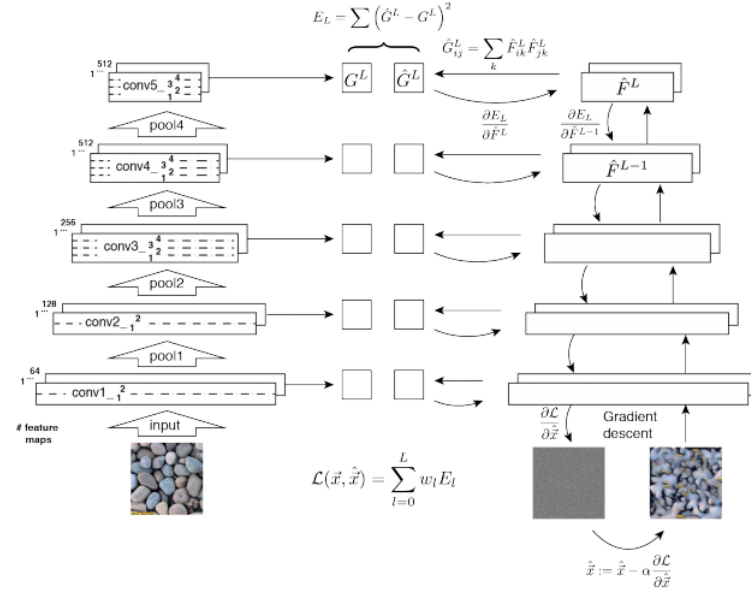


Figure 3.2: Architecture proposed by Leon Gatys. Source: [GEB15].

The making of a new texture involves two stages:

Stage 1: Texture Analysis

Firstly, the authors extract differently sized features from an input texture image. To do so, the input texture is passed through the CNN, and the filter maps that result from the filter responses are gathered. The amount of feature maps generated by a layer is equal to the number of filters that compose that layer. The feature maps of layer l are stored in F^l , in which F^l_{jk} is the activation of the j^{th} filter at position k in layer l . A stationary texture descriptor - the Gram matrices - is computed using the previously acquired maps. The Gram matrices express the correlation between the responses of different features. They are computed through the formula presented in Equation 3.1.

$$G^l_{i,j} = \sum_k F^l_{ik} F^l_{jk} \quad (3.1)$$

The authors of this work believe that the set of all Gram matrices acquired in response to the input texture produce a stationary description of the texture, and fully describe it. The left side of Figure 3.2 refers to the above described process.

Stage 2: Texture Synthesis

To generate a new texture, this method performs gradient descent from a white noise image, producing a new image that matches the Gram-matrix representation of the input texture. The optimization strategy used consists of minimizing the mean squared distance between the entries of the Gram-matrix of the input texture and the entries of the Gram-matrix of the noise image.

Equation 3.2 expresses the contribution of the Gram matrices of each layer to the loss, and the loss is computed as stated in Equation 3.3.

$$E_l = \frac{1}{4N_l^2 M_l^2} \sum_{i,j} (G_{ik}^l - \hat{G}_{ij}^l)^2 \quad (3.2)$$

$$\mathcal{L} = \sum_{l=0}^L \omega_l E_l \quad (3.3)$$

The gradient $\frac{d\mathcal{L}}{dx}$ is used as input for an optimization strategy, namely the Fortran subroutines for large scale bound constrained optimization, described in [ZBLN97], which is adequate for high-dimensional optimization problems. Sets of forward and backward passes are performed in order to optimize the noise image, as one would do to optimize a network. The right side of Figure 3.2 outlines the above described synthesis process.

This architecture provides substantial improvements when compared to other recent parametric texture synthesis methods. It produces natural textures of similar quality to non-parametric texture synthesis methods but at a higher computational cost. Figure 3.3 shows some results obtained with this method. In [Sne17], an improvement upon the previous method is proposed. This new method also matches the corresponding feature maps of an input texture and a noise image; however, the matching is performed at several levels of a Gaussian pyramid.

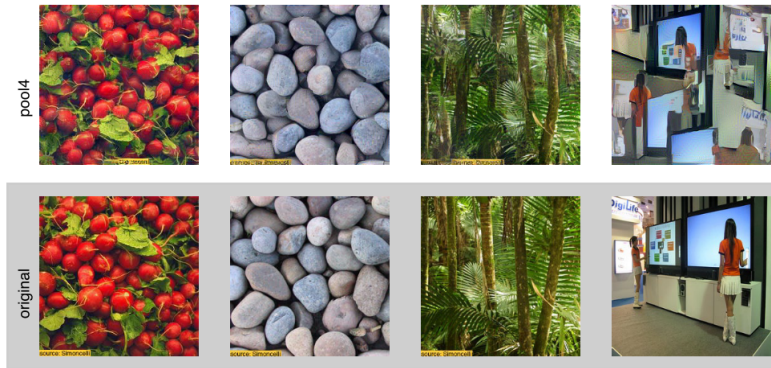


Figure 3.3: Results of the synthesis using the method proposed in [GEB15]. On the top bottom row, four sample textures; on the top row, the corresponding synthesized textures. Source: [GEB15].

Other relevant texture synthesis works include:

- [LGX16], which uses CNNs to synthesize texture, applying constraints to condition both the Fourier spectrum and the statistical features learnt by CNNs;
- [ULVL16], which proposes an alternative approach to relief the computational burden imposed by the strategy proposed in [GEB15]. It introduces a compact feed-forward CNN which takes as input a single example texture and generates multiple versions of the same texture with arbitrary size using them, posteriorly, for style transfer. This approach is further improved by a posterior work, by the same authors, proposed in [UVL17];

- [UBGB16], which introduces a new parametric texture model based on a single-layer CNN with random filters, showing that neither the hierarchical texture representation nor the trained filters are crucial for high-quality texture synthesis.

3.6 Summary

This chapter described the key concepts behind texture analysis, as well as the state of the art on static, by-example texture synthesis. We presented a brief analysis of classic texture synthesis algorithms, approaching pixel-based and patch-based methods, as well as surface and solid texture synthesis. There are few applications of these traditional methods to medical imaging datasets; in fact, the applications found only address Mammography [BAE99] and [Cas09]) and Ultrasound images ([MRB⁺17]). In the context of MRI, applications for detection and segmentation of lesions and structures, such as [CGB06], [GMKM⁺15] and [Gm13] are significantly more common.

In light of the new advancements in the Deep-Learning field, we addressed novel methods for texture synthesis. Most of the Deep-Learning approaches for the generation of artificial MRI exams use Generative Adversarial Networks, which are explored further in Chapter 4. The Deep-Learning approaches explored in this chapter were neither conceived or applied yet for MRI tissue synthesis. Once again, Deep-Learning applications for detection and segmentation using Mammography images (such as [ZYM18]) are more abundant.

Most importantly, from the literature review conducted throughout this chapter, we concluded that none of the texture synthesis methods found is sufficient to synthesize a coherent and complete MRI image from a corresponding annotation with the quality we intend.

Chapter 4

Generative Modeling

According to [Goo16], a generative model is any model that is capable of taking a training set, consisting of samples drawn from a prior distribution, and learning to represent an estimate of that distribution, outputting either a probability distribution or samples from it. Generative models are useful for a great variety of tasks:

- They can be incorporated into model-based reinforcement learning methods [FGL16];
- They can be trained with missing data and produce predictions on incomplete input data, as in [SPS⁺17];
- Some generative models, such as Generative Adversarial Networks (GANs), enable machine learning to work with multi-modal outputs, which is useful for tasks such as predicting the next frame of a video, in which a single input may produce several acceptable outputs [LKC15];
- They are useful for tasks that require a realistic generation of samples from some distribution.

There are many classes of generative models, however, we'll only review the ones we believe to be the most useful to our purposes, which are GANs and Variational Autoencoders. We'll give more emphasis to GANs since their efficiency for a large variety of tasks, ranging from the synthesis of natural images [RMC15] to super-resolution methods [LTH⁺17] or style-transfer applications [ZPIE17], has been extensively proven. We'll also overview the most popular strategies for the evaluation of generative models

4.1 Generative Adversarial Networks

4.1.1 Principles

Generative Adversarial Networks were first proposed in [GPAM⁺14] and gained popularity for data generation in recent years. GANs pitch two different models, a generative one and a dis-

criminating one, against each other. While the generative model creates new, fake data, the discriminating examines the new data and tries to understand if it is real or false. The competition established between the two models leads them to improve each other: the generative model learns how to generate increasingly realistic data, while the discriminating model improves its capability of distinguishing real from fake data. Figure 4.1 summarizes this interaction.

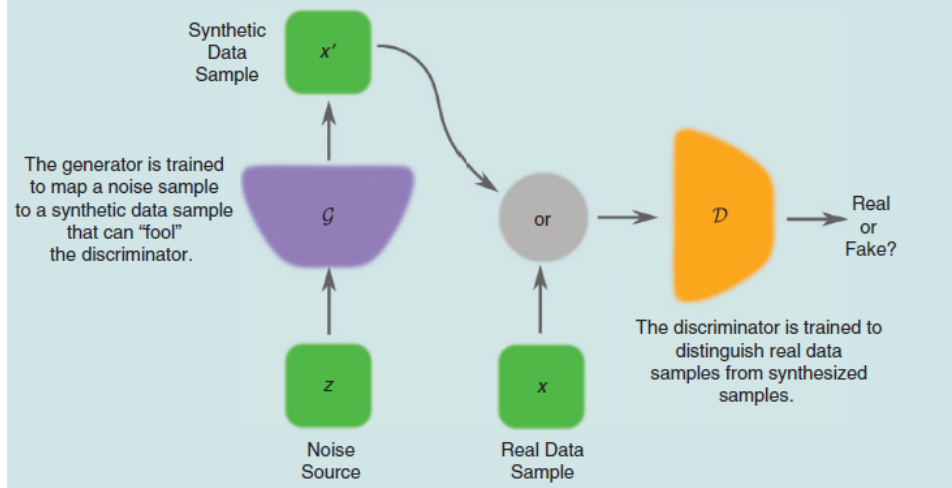


Figure 4.1: GANs framework. Source: [CWD⁺18].

The generator and the discriminator must be differentiable functions, and are both commonly represented by deep neural networks. The goal of the generator network G is to learn the distribution p_g over the data x . To do so, it maps an input noise sample z , taken from the prior $p_z(z)$, to the data space, producing a synthetic data sample x' - formally, we are referring to the mapping $G(z; \theta_g)$, where θ_g represents the network's parameters. The discriminator network D maps either real (x) or fake (x') samples into probability values indicating whether each sample is closer to being from the data distribution p_{data} (probability closer to 1) or from p_g (probability closer to 0). Formally, this mapping is $D(x; \theta_d)$, where θ_d represents the parameters of the network and x the sample.

D is trained to maximize the probability of x coming from the data, while G is trained to minimize $\log(1 - D(G(z)))$, which depends on the parameters of the discriminator. The previous optimization can be accomplished by applying a zero-sum game (also known as minmax game), to the value function $V(G, D)$, as detailed in Equation 4.1. As Equation 4.1 shows, G has no direct access to the real data, learning solely through its interaction with D , while D has access to both synthetic and real samples. Other optimization strategies could also be used, as presented in [Goo16]. The minimax game has a global optimum for $p_g = p_{data}$, which is demonstrated in [GPAM⁺14].

$$\min_G \max_D V(D, G) = \min_G \max_D \mathbb{E}_{x \sim p_{data}(x)} [\log D(x)] + \mathbb{E}_{z \sim p_z(z)} [\log(1 - D(G(z)))] \quad (4.1)$$

Training a GAN is not trivial. D cannot be optimized to completion before the optimization of G , since this is computationally not feasible and prone to over-fitting. As so, D and G must

be alternately optimized: D is optimized k steps and then G is optimized 1 step. This way, D is maintained near its optimal solution, as long as G changes slowly enough. The number of steps, k , is a hyper-parameter. This is a mini-batch stochastic gradient descent training process.

In the early learning stages of training, when G is poor, D can reject samples confidently since they are noticeably different from the training data. This means that $\log(1 - D(G(z)))$ saturates easily, producing low gradients. To obtain stronger gradients in the context of early learning, G can be optimized by maximizing $\log(D(G(z)))$.

The main disadvantages of using GANs are the absence of an explicit representation of $p_g(x)$, and the difficult training process, due to the need of synchronizing the training of D and G . Still, adversarial models have a statistical advantage since the generator is not updated directly with data examples, but instead through the gradients flowing through the discriminator, making it, in this sense, less prone to over-fitting.

Some of the most common issues in GAN training are mode collapse (the generator produces a limited variety of samples), diminished gradients (if the discriminator gets too successful, the gradients vanish and the generator does not learn), absence of convergence (despite oscillating, the network's parameters fail in converging) and high sensitivity to the hyper-parameters. All these issues motivated researchers to improve the training process, through varying the networks' architecture, the cost functions applied and even the optimization process.

Concerning the **network design** for image applications, the Deep Convolutional GAN (DC-GAN) [RMC15] made great progress in improving the quality of synthetic images by replacing maxpoolings by strided convolutions (in the discriminator) or fractional-strided convolutions (in the generator), allowing the networks to learn their own spatial downsamplings. Other improvements to both the generator and discriminator include:

- Abolishment of fully connected layers in deep architectures;
- Use of batch normalization to stabilize training, help gradient flow in deeper models, and prevent mode collapse;
- The choice of adequate activation functions. In the generator, the Rectified Linear Unit (ReLU) is applied in all layers except the last, in which the hyperbolic tangent function is used - contrarily to the ReLU, this is bounded and, therefore, allows the network to learn more quickly to saturate and cover the color space of the training distribution. In the discriminator, the Leaky ReLU is used in all layers, since it was found to work especially well for high-resolution images.

Stacked GANs (SGANs) [HLP⁺17] and Progressive GANs (PGANs) [KALL17] also propose important architectural improvements. SGANs propose a stack of encoders, a stack of generators and a stack of discriminators, all with the same amount of networks. The stack of encoders is fed with image x , predicting the label y at the end of the stack. Intermediate predictions are produced after each encoder. At each level, the output of the encoder is fed as conditional input to the corresponding generator, together with noise. The output of the generator is fed as input to the

same encoder producing yet another prediction. The discriminator is fed with the encoder's input and the generator's output. Each level is trained individually, and then joint training is performed. Figure 4.2 schematizes the SGANs architecture. PGANs also propose a staged training strategy: in the beginning, both G and D are networks with low spatial resolution, and as the training progresses new layers are added to both networks, increasing the spatial resolution.

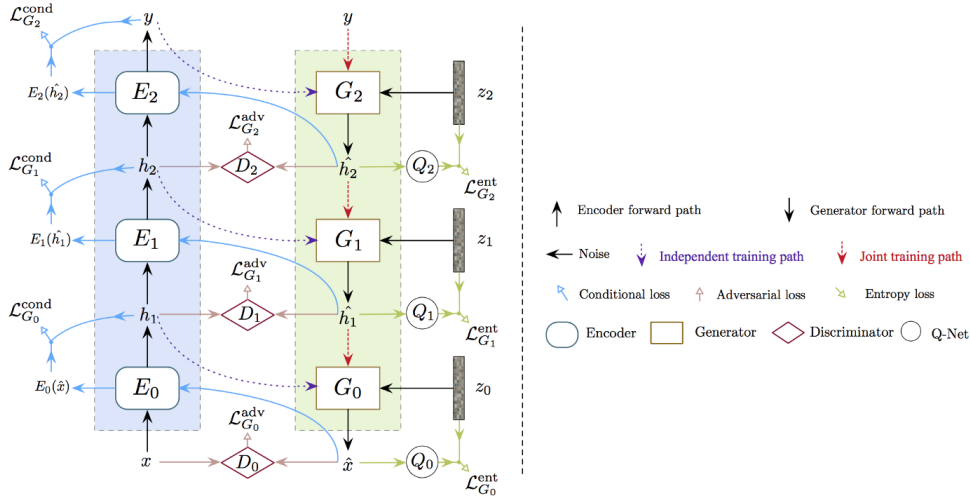


Figure 4.2: SGANs architecture. Adapted from: [HLP⁺17].

Concerning the improvement of the loss function, plenty of research has already been done, from which emerged variations of existing loss functions (e.g. addition of new penalties), as well as entirely new ways to compute costs. One of the most popular variations is the Least Squares GAN (LSGAN) [MLX⁺17]. Its development was motivated by vanishing gradient issues associated with the use of the sigmoid cross entropy loss function, commonly used in practical implementations of GANs. The LSGAN's authors show that penalizing samples that lie far away from the decision boundary generates more gradients when updating the generator, reducing the chances of the gradient to vanish, and propose objective functions for the generator and discriminator based on the least mean square - those functions are given in Equations 4.2 and 4.3 respectively for the generator and the discriminator. In Equations 4.2 and 4.3, a and b are, respectively, the labels for the fake data and real data, and c is the value that G wants D to predict for the false data.

$$\min_G V_{LSGAN}(G) = \frac{1}{2} \mathbb{E}_{z \sim p_z(z)} [(D(G(z)) - c)^2] \quad (4.2)$$

$$\min_D V_{LSGAN}(D) = \frac{1}{2} \mathbb{E}_{x \sim p_{data}(x)} [(D(x) - b)^2] + \frac{1}{2} \mathbb{E}_{z \sim p_z(z)} [(D(G(z)) - a)^2] \quad (4.3)$$

There are many other loss function modifications besides the LSGAN, such as the Wasserstein GAN [ACB17], the Energy-based GAN [ZML16], and the Boundary Equilibrium GAN [BSM17].

Regarding the **optimization process**, several improvements have been proposed, of which we highlight the ones presented bellow, based on the analysis in [SGZ⁺16]:

- **Experience replay** [PV16] - the discriminator is updated using the current generated sample as well as with previous ones, preventing it from over-fitting for a certain time instance of the generator;
- **Historical averaging** - tracking of previous model parameters and penalizing changes that are different from the average changes, which improves convergence;
- **Feature matching** - to prevent over-training the generator, which can happen when the discriminator's output is directly optimized, a new objective function was developed, consisting of training the generator to produce fake data that matches the statistics of real data. To this end, the generator is trained to match the expected value of the features on an intermediate layer of the discriminator, since the discriminator provides the features most worthy of being matched (the most discriminating features between real and fake data).
- **One-sided label smoothing** - as the training progresses, if the discriminator learns to depend solely on a small subset of features to detect real images, the generator may learn to only synthesize those features. To avoid this, the discriminator can be penalized by replacing 0 and 1 targets with smoothed values, such as 0.1 and 0.9 respectively.
- **Mini-batch discrimination** - one factor that highly contributes to mode collapse is the discriminator processing each image independently. Consequently, the similarity between the generator's outputs for different input samples is not assessed, and the discriminator has no way of detecting if the generator is only producing a small variety of outputs. To avoid this, the discriminator can be fed with a batch of several fake images instead of individual images. Moreover, one can compute the similarity between an image and the remaining ones from the same batch, and append the similarity in one of the dense layers of the discriminator to aid in the classification of the image as real or false. If the mode starts to collapse, the similarity between the fake images will increase.
- **Virtual batch normalization** - despite improving greatly the optimization of CNNs, batch normalization makes the output of a CNN for an input sample to strongly depend on the batch's remaining samples. Virtual batch normalization avoids this issue by normalizing each input sample in relation to the statistics of a fixed, reference batch, defined beforehand and normalized on its own statistics, as well as in relation to the input sample itself. As this is computationally expensive, it is usually only applied for the generator.

4.1.2 Conditional Generative Adversarial Networks

Conditional Generative Adversarial Networks (CGANs), proposed in [MO14], are an extension of the GANs algorithm which allows the conditioning of the data to be generated by feeding both the generator and discriminator networks with extra information y , which can be any kind of auxiliary information, such as class labels or data from other modalities. We can perform the conditioning by feeding y into both the discriminator and generator as additional input layers. The general

architecture of CGANs is shown in Figure 4.3. In CGANs, the minmax game equation, given in Equation 4.4, is a variant of minmax game equation for GANs, given in Equation 4.1.

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim p_{data}(x)} [\log D(x|y)] + \mathbb{E}_{z \sim p_z(z)} [\log(1 - D(G(z|y)))] \quad (4.4)$$

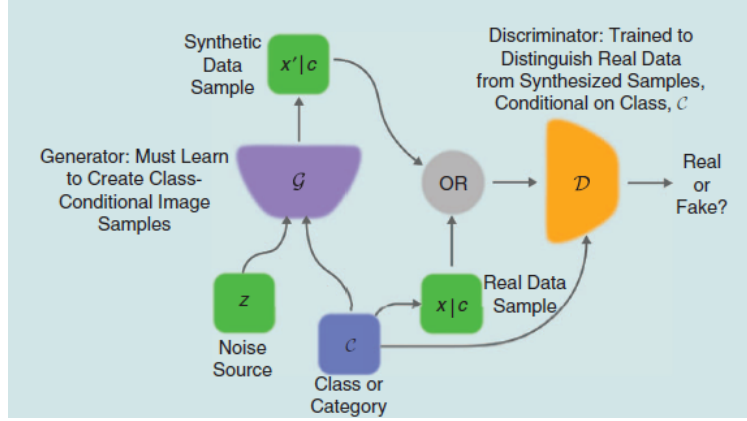


Figure 4.3: CGANs framework. Source: [CWD⁺18].

4.1.3 Image-to-image translation applications

Image to image translation, a concept firstly introduced in [HJO⁺01], refers to the conversion of a scene from a form of representation into another - in other words, a pixel-to-pixel conversion of an image from one domain to another. Converting MRI annotations into MRI exams, as well as the opposite operation, are both goals that can be accomplished through this concept. There are several works that apply the GANs framework to this purpose, namely [IZZE17] and [WLZ⁺18]; however, not all applications re-purpose the GANs baseline, namely [ZPIE17], which proposes simply a CNN.

4.1.3.1 Baseline

The work that pioneered the image-to-image translation topic was [IZZE17], which introduces a framework commonly known as *pix2pix*. It consists of a CGAN that uses a U-Net based generator [RFB15] and a Convolutional PatchGAN, similar to the one proposed in [LW16], as the discriminator. Instead of classifying each synthetic image as real or fake, the PatchGAN applies the same classification strategy to $N \times N$ patches of the synthetic image. The authors of this work defend that this discriminator architecture promotes a more realistic synthesis of high-frequency details.

To train this framework, the traditional loss function for the CGAN is applied, with the following two improvements:

- The discriminator has access to the input images, besides the synthesized ones;

- Addition of the L_1 distance as one of terms of the loss function to measure the similarity between corresponding real and synthetic images. This has been proven to aid generators to produce more realistic images. With this extra term, the discriminator's training will remain unchanged, while the generator will be trained to fool the discriminator and to be near the output in a L_1 sense. The L_1 loss captures low frequency information and promotes less blurring than the L_2 loss. The discriminator is left to focus on high frequency content, which is coherent with the choice of the PatchGAN for its architecture.

The resulting objective function is indicated in Equation 4.5, where λ controls the importance given to the L1 distance term.

$$\arg \min_G \max_D \mathbb{E}_{x,y} [\log D(x,y)] + \mathbb{E}_{x,z} [\log(1 - D(x, G(x,z)))] + \lambda \mathbb{E}_{x,y,z} [\|y - G(x,z)\|_1] \quad (4.5)$$

Concerning the practical implementation of this framework, no noise is provided as a direct input to the generator - instead, it is introduced through dropouts in the network, both in the training and test stages. Initial experiments showed that the generator simply learned to ignore the noise when it was provided as a direct input to the network. Regarding the optimization process, D and G are updated with gradient descent alternately, one step each; as suggested in [GPAM⁺14], G is trained to maximize $\log(D(x, G(x,z)))$; finally, the objective function for D is divided by 2, to slow down the rate at which D learns with respect to G . For the inference, besides the maintenance of the dropout, batch normalization is applied using the statistics of the test batches instead of aggregate statistics of the training batches - these two aspects differ from usual inference strategies. When the batch normalization strategy described is applied with unitary batch size, it is known as instance normalization, and it was already shown to be useful for image generation in [UVL16].

The *pix2pix* framework was applied to a great variety of tasks (e.g. conversion of semantic label maps to corresponding image, inpainting) and datasets, and an exemplifying result, obtained with the Inria Aerial Image dataset [MTCA17], is shown on Figure 4.4. The main shortcomings detected in the fake images were artifacts in regions where the input image was sparse and when the inputs were unusual. Its performance was also evaluated for several variations of the loss function, generator and discriminator architectures, and the main conclusions gathered were the following:

- **Loss:** by itself, the L_1 loss produces blurry results and grayish colors, especially when the generator is unsure of the presence or absence of an edge or of the region's color, respectively; using the CGAN loss without the L_1 term generates much better results, but with artifacts, which are corrected when the L_1 term is added;
- **Generator:** both the U-Net and a general encoder-decoder, without skip connections, were tested, and the U-Net always had a superior performance, independently of the loss function used;

- **Discriminator:** several variation of the patch size ($N \times N$) were tested, ranging from pixel-sized patches to image-sized patches. Pixel-sized patches do not promote sharpness, but promote colorfulness; 70×70 patch size allowed satisfactory sharpness and less artifacts than inferior patch sizes, while image-sized patches didn't improve either qualitative or quantitative results, most likely by corresponding to a network with much more parameters and, consequently, harder to train.



Figure 4.4: Exemplifying result for the application of the *pix2pix* for the generation of images from the Inria Aerial Image dataset. From left to right, the input image, the output image and the ground truth. Source: [IZZ17]

4.1.3.2 Variations upon the baseline

To this date, several variations upon the *pix2pix* baseline have been proposed. For instance, in [WLZ⁺18], the *pix2pix* is modified by using a coarse-to-fine generator and multi-scale discriminators, with the goal of producing images with higher resolution. This framework is known as *pix2pix_{HD}*.

The coarse-to-fine generator comprises two sub-networks: the global generator network (G_1) and the local enhancer network (G_2). Both consist of a convolutional front-end (G_1^F and G_2^F), followed by a sequence of residual blocks (G_1^R and G_2^R), finishing in a transposed convolutional back-end (G_1^B and G_2^B) - however, G_2 operates at a greater resolution than G_1 . The two networks are linked together, and they interact as follows: a semantic label map is given as input to G_2^F , which outputs a set of feature maps with half the size of the input, which is concatenated with a $2\times$ downsampled version of the semantic label map and fed as input to G_1 . The last set of feature map produced by G_1^B is summed pixel-wise to the set of feature maps outputted by G_2^F , and the resulting set of feature maps is fed to G_2^R , finishing the forward pass through G_2 . The interaction between the global generator and local enhancer is detailed in Figure 4.5, where the previous terminology is used.

Concerning the discrimination stage, a set of three generators is used, all with similar architectures but operating at different resolutions. Each discriminator is trained with images at different scales, obtained by downsampling twice the real and fake images, forming a 3 level image pyramid. This strategy avoids the use of a single discriminator network with a high receptive field, which would be needed due to the high resolution of the images and that would require either

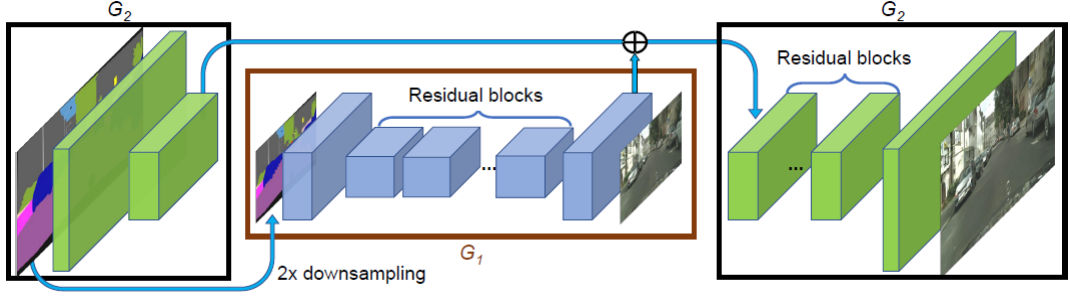


Figure 4.5: Coarse-to-fine generator of the *pix2pix_{HD}* framework. Source: [WLZ⁺18].

a deep architecture or large convolutional kernels, both of which are prone to over-fitting. This multi-scale discrimination strategy avoids pattern repetition.

The objective function used to train the networks combines the classical GAN loss, modified to contemplate the optimization of the three discriminators, and two extra terms, one which evaluates feature matching and a perceptual loss term. The **feature matching loss** component consists of computing the L_1 distance between the same intermediate feature maps of the discriminator obtained using the real image and the corresponding fake one. The calculation of this loss component for discriminator D_k is given in Equation 4.6, 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.

$$\mathcal{L}_{FM}(G, D_k) = \mathbb{E}_{(s,x)} \sum_{i=1}^T \frac{1}{N_i} [\|D_k^{(i)}(s, x) - D_k^{(i)}(s, G(s))\|_1] \quad (4.6)$$

The **perceptual loss** component consists of computing the L_1 distance between the same intermediate feature maps of the VGG network [SZ14] obtained using the real image and the corresponding fake one. Its calculation is described in Equation 4.7, where $F^{(i)}$ denotes the i_{th} layer with M_i elements of the VGG network.

$$\mathcal{L}_{perceptual}(x, G) = \sum_{i=1}^N \frac{1}{M_i} [\|F^{(i)}(x) - F^{(i)}(G(s))\|_1] \quad (4.7)$$

The final objective function is expressed in Equation 4.8.

$$\min_G \left(\max_{D_1, D_2, D_3} \sum_{k=1,2,3} \mathcal{L}_{GAN}(G, D_k) \right) + \lambda \sum_{k=1,2,3} \mathcal{L}_{FM}(G, D_k) + \lambda \mathcal{L}_{perceptual}(x, G) \quad (4.8)$$

The *pix2pix_{HD}* framework proposed another two improvements:

- Semantic label maps do not differentiate objects from the same class, which may cause different objects of the same class to be merged in the map. This issue is addressed by the computation of a **binary boundary map** in which the pixels belonging to the edges of all object instances of all classes in the semantic label map are 1. The instance boundary map

is concatenated with a one-hot vector representing the label map, and fed as input to the generator;

- An ideal synthesis algorithm should be capable of performing a one-to-many mapping, producing various images from the same semantic label map. To this end, an encoder-decoder network is trained to embed visual features for each object instance in the ground truth image. New images are synthesized by concatenating the output of the encoder-decoder with the semantic label map and provide this as input to the generator. Varied results can be produced by manipulating the visual features before the concatenation with the label map. This process is called **instance-level feature embedding**.

The training procedures were similar to those used for the *pix2pix* and Figure 4.6 shows an exemplifying result, obtained with the Cityscapes dataset [COR⁺16], where it is possible to visually see the superior performance of the *pix2pix_{HD}* compared with the *pix2pix*.

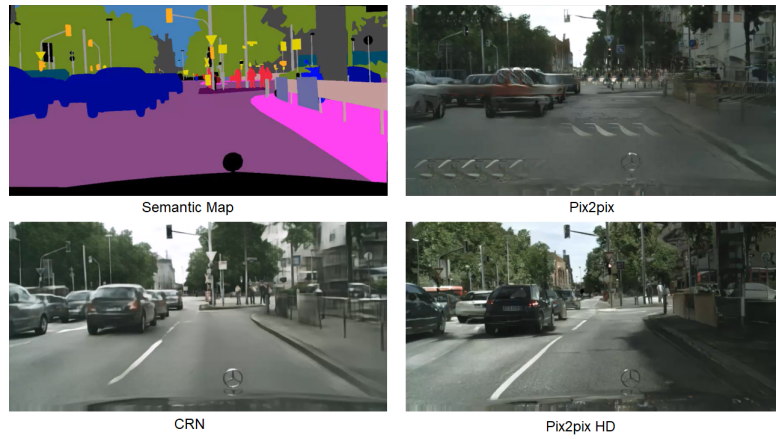


Figure 4.6: Comparative results for the application of the *pix2pix*, *pix2pix_{HD}* and CRN frameworks for the generation of images from the Cityscapes dataset. Source: [ZPIE17].

The authors conducted studies to analyze the loss function, the impact of the use of instance maps and the architectures of the generator and discriminator, and their main findings were the following:

- **Loss:** combining the GAN loss with the feature matching loss produces considerably better results than using the only the GAN loss. Adding the perceptual loss further improves the results, despite this component not being essential to produce images with satisfactory realism;
- **Instance maps:** the use of instance maps increases the realism of the synthetic images and refines the objects' boundaries;
- **Generator:** the coarse-to-fine generator produces better results than the CRN [ZPIE17] and the U-Net applied in this same framework;

- **Discriminator:** the multi-scale generators allow the synthesis of images with greater quality and also make the training more stable.

Both *pix2pix* and *pix2pix_{HD}* require paired training data, which is often hard to obtain, especially in the medical imaging context, since annotating medical data requires highly specialized professionals. This motivated the development of unsupervised GAN based architectures for image-to-image translation that don't require paired data, namely the CycleGAN [ZPIE17].

The CycleGAN framework is composed of four networks: two generators, G and F , and two discriminators, D_X and D_Y . G translates each image of the domain X to domain Y and D_Y assesses if the translation result resembles the remaining images that belong to the Y domain; F performs the opposite transformation and is judged by D_X . The overall goal of the CycleGAN is to learn appropriate G and F mappings according to the training data comprised in each domain. The interaction between the four networks is detailed in Figure 4.7 a).

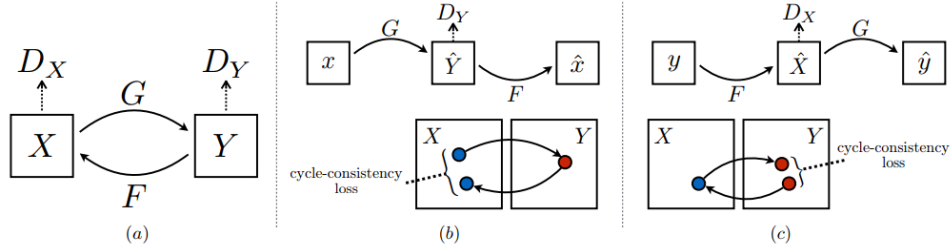


Figure 4.7: Cycle GAN framework. X and Y are two different image domains while G and F are two distinct generator networks mapping, respectively, X to Y and Y to X . D_X and D_Y are two distinct discriminator networks which evaluate, respectively, synthetic images produced by F and G . Source: [ZPIE17].

The quality of the translations performed by G and F are evaluated by a loss function that combines the classic GAN loss with a **cycle consistency** term. Ideally, if an image is translated from domain X to domain Y , and back again to domain X , the reconstructed image obtained should be identical to the image fed at the beginning (as shown in Figure 4.7 b), and the same should also happen for the complementary operation ($Y \rightarrow X \rightarrow Y$, shown in Figure 4.7 c). This goal is called cycle consistency and it prevents contradictions between the mappings learned by G and F . Cycle consistency is achieved by the computation of the L_1 distance between each image and its reconstruction for each domain, as shown in Equation 4.9.

$$\mathcal{L}_{cyc}(G, F) = \mathbb{E}_{x \sim p_{data}(x)} [\|F(G(x)) - x\|_1] + \mathbb{E}_{y \sim p_{data}(y)} [\|G(F(y)) - y\|_1] \quad (4.9)$$

Bearing the two loss components in mind, optimization goal is given by Equation 4.10, where λ controls the weight given to the cycle consistency loss term.

$$G^*, F^* = \arg \min_{G, F} \max_{D_X, D_Y} \mathcal{L}_{GAN}(G, D_Y, X, Y) + \mathcal{L}_{GAN}(F, D_X, X, Y) + \lambda \mathcal{L}_{cyc}(G, F) \quad (4.10)$$

For applications such as painting-photo conversion, an extra loss term - the **identity loss** - is recommended to improve color preservation. This is achieved by computing the L_1 distance between an image of the target domain and the output produced when that image is given as input to the generator that translates to that domain (e.g. L_1 distance between an image y , from domain Y , and the output of $G(y)$), as Equation 4.11 shows.

$$\mathcal{L}_{identity}(G, F) = \mathbb{E}_{y \sim p_{data}(y)} [\|G(y) - y\|_1] + \mathbb{E}_{x \sim p_{data}(x)} [\|F(x) - x\|_1] \quad (4.11)$$

The training of this framework uses experience replay - the discriminator is updated using the 50 most recently generated images. Upon the analysis of the results and comparison with baseline approaches, the main findings were that the CycleGAN outperforms quantitatively the *pix2pix*, provides good reconstructions, and that the conjugation of the GAN loss and the cycle consistency loss is essential for satisfactory results - simply removing the cycle consistency loss in one of the directions increases instability during training and leads to mode collapse. Figure 4.8 shows the results obtained using the CycleGAN for style transfer (image to painting style), while Figure 4.9 shows results obtained for the Inrial Aerial Image dataset, comparing with *pix2pix*.

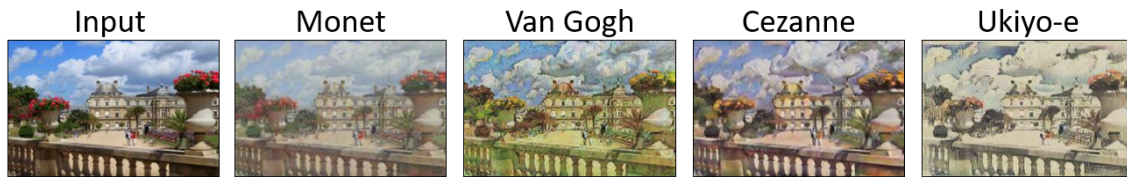


Figure 4.8: Exemplifying result for the application of the CycleGAN for photo to painting conversion, for different painting styles. Source: [ZPIE17]

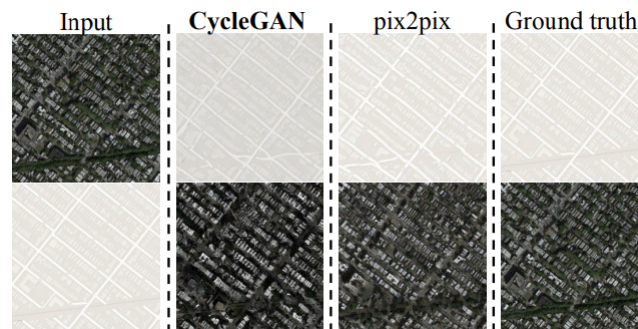


Figure 4.9: Comparative result for the application of the CycleGAN and *pix2pix* frameworks for the synthesis of images of the Inrial Aerial Image dataset. Source: [ZPIE17]

Many more frames for image-to-image translation exist, such as [CCK⁺18], [AATVG18], [ARS⁺18] and [ZZP⁺17], most of them exploring multi-modal frameworks.

4.2 Variational Autoencoders

Yet another recent and popular approach to data generation is Variational Autoencoders, first proposed in [KW13]. An Autoencoder is an unsupervised learning technique that applies Neural Networks to the task of representation learning. Concretely, it consists of a network which encodes the training data into a compressed representation, in a bottleneck region, and also interprets and decodes that compressed representation, as shown in Figure 4.10. We refer to the general attributes encoded in the bottleneck as latent attributes of the input data, and they compose a latent space.

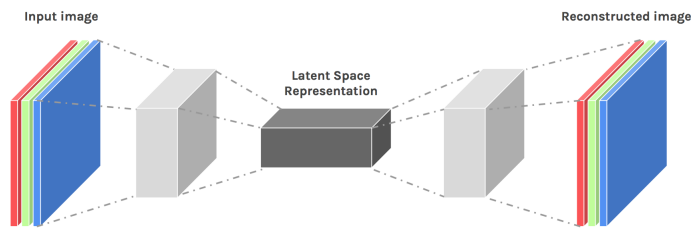


Figure 4.10: Autoencoder architecture. Source: [Sha].

The main limitation of autoencoders when applied to data generation is that the encoder outputs a single value for each encoding dimension, producing a non-continuous latent space, which doesn't allow interpolation. To overcome this issue, Variational Autoencoders (VAEs) were created. VAEs describe the observations in the latent space in a probabilistic manner. As so, the encoder outputs a probability distribution (usually Gaussian shaped) for each attribute of the latent space instead of a single scalar value, as shown in Figure 4.11.

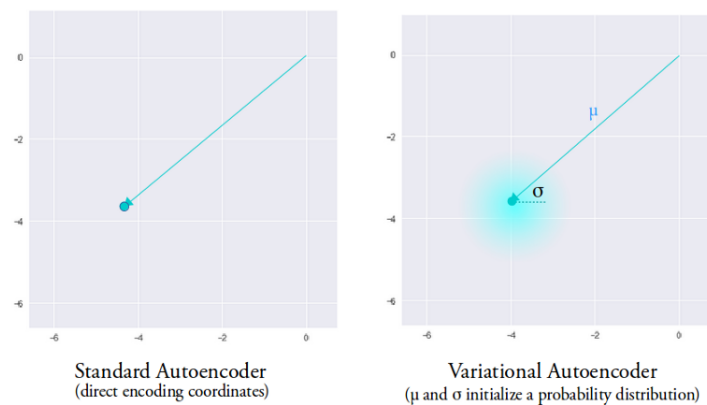


Figure 4.11: Comparison of the mapping from the input to the latent space on standard Autoencoders and VAEs. Source: [Mis].

The mathematical foundations of VAEs are very different from the ones of classical Autoencoders (such as Sparse or Denoising Autoencoders) since they are statistically motivated. Besides being useful for image generation tasks, VAEs also enable the exploration of specific variations of the existing data through the latent space.

In VAEs, the data x and latent variables z are modeled by the prior probability distributions $p(x)$ and $p(z)$, respectively. The goal is to infer values for the latent variables that represent well the observed data, which translates into calculating $p(z|x)$. This can be achieved by applying Bayes' rule, as stated in Equation 4.12.

$$p(z|x) = \frac{p(x|z)p(z)}{p(x)} \quad (4.12)$$

The denominator of Equation 4.12 can be calculated using Equation 4.13. Computing the integral given in Equation 4.13 implies evaluating all possible configurations of latent variables - for each candidate z , we have to assess if it can generate x - which isn't feasible.

$$p(x) = \int p(x|z)p(z)dz \quad (4.13)$$

To overcome this issue, $p(z|x)$ must be approximated through a distribution $q_\lambda(z|x)$, which will be used as a proxy of $p(z|x)$ to form predictions about future data. λ refers to the parameters of the type of distribution - for instance, for a Gaussian distribution, $\lambda = (\mu, \sigma^2)$. The Kullback-Leibler (KL) divergence can be used to evaluate how well the distribution chosen approximates $p(z|x)$, and our goal is to obtain the parameters λ of the distribution q that minimize this divergence, as stated in Equation 4.14.

$$\begin{aligned} \operatorname{argmin}_\lambda KL(q_\lambda(z|x)||p(z|x)) &= \operatorname{argmin}_\lambda \mathbb{E}_q[\log q_\lambda(z|x)] - \mathbb{E}_q[\log p(x,z)] + \log p(x) \\ &= \operatorname{argmin}_\lambda (\mathbb{E}_q[\log p(x,z)] - \mathbb{E}_q[\log q_\lambda(z|x)]) + \log p(x) \end{aligned} \quad (4.14)$$

The minimization proposed in Equation 4.14 is intractable since it implies, once again, computing $p(x)$. Since the KL divergence is always equal or greater than zero and $\log p(x)$ does not depend of q , minimizing the KL divergence is equivalent to maximizing the expression given in Equation 4.15, known as evidence lower bound (ELBO).

$$ELBO = \mathbb{E}_q[\log p(x,z)] - \mathbb{E}_q[\log q_\lambda(z|x)] \quad (4.15)$$

Summing up, the encoder will take x as input and learn the optimal distribution $q_\lambda(z|x)$, outputting the parameters that describe the distributions for all attributes of the latent space, while the decoder will learn the distribution $p(x|z)$ and output a reconstructed image, $\hat{x}$ from sampling the distributions of each attribute of the latent space. This entire process is summarized in Figure 4.12.

The full network can be trained by minimizing the loss function shown in Equation 4.16, where $\mathcal{L}(x, \hat{x})$ indicates the reconstruction loss - a measure of resemblance between x and $\hat{x}$.

$$\operatorname{argmin}_\lambda \mathcal{L}(x, \hat{x}) + ELBO \quad (4.16)$$

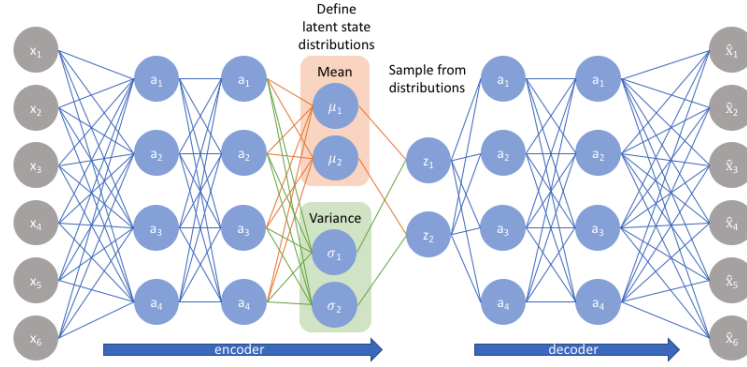


Figure 4.12: VAE framework. Source: [Jor18].

4.2.1 Conditional Variational Autoencoders

Similarly to the described in Chapter 4.1.2, the Conditional Variational Autoencoder (CVAE), addressed in [KRMW14, SLY15, YYS15], is an extension of the VAE that allows the conditioning of the region of the latent space which will be sampled, by adding an extra input to the encoder and to the decoder. This extra input can be either the class label (provided, for instance, by a one-hot vector) or data from other modalities. Concerning the optimization of a CVAE, Equation 4.17 shows the ELBO for the conditional context.

$$ELBO_{CVAE} = \mathbb{E}_q[\log p(x|y, z)] - \mathbb{E}_q[\log q_\lambda(z|x, y)] \quad (4.17)$$

4.2.2 Adversarial Autoencoders

Training VAEs comes with inescapable issues. Firstly, the integral of the KL divergence only has an analytic solution for some classes of distributions. Additionally, representing the latent space with a discrete variable doesn't allow back-propagation, hampering the training process and forcing reparametrization tricks. Adversarial Autoencoders (AAEs) [MSJ⁺15] avoid all these issues with by establishing an adversarial framework. In AAEs, besides having the autoencoder as the generative network, a new network is added to perform the task of the discriminator, similarly to what was described within the GANs context. The discriminator network will try to predict if a sample of the latent space was originated by the encoder portion of the network, or by an user-defined prior distribution $p(z)$.

Accordingly, the training of AAEs is divided into two phases. The first is the reconstruction phase, which consists of training the autoencoder to minimize the reconstruction loss, disregarding the discriminator. This is followed by the regularization phase, which consists of training the encoder and the discriminator to correctly classify the encoder's output and $p(z)$.

4.3 Applications for medical data

The application of generative models to the synthesis of medical images is extremely recent, however, several relevant studies have already emerged, and the most relevant are summarized in Table 4.1. This table also lists the metrics used for the evaluation of each work which, as can be observed, are varied. The evaluation of generative models will be further discussed in Chapter 4.4.

Table 4.1: Generative modeling applications for medical imaging

Paper	Year	Task	Model	Evaluation
[WDS ⁺ 17]	2017	2D brain MRI to CT conversion	Cycle GAN	MAE & PSNR
[OSDU18]	2018	MRI(T1/T2w) to MRA conversion	SteerableGAN	PSNR & Dice
[NTL ⁺ 18]	2018	Brain MRI (T1/T2w, T1c & Flair) synthesis from annotations	<i>pix2pix</i>	Dice score
[CJGT18]	2018	Brain MRI (T1/T2w, T1c & Flair) modality conversion	Multi-input, multi-output CNN	PSNR, SSIM & Dice score
[FADK ⁺ 18], [FAKA ⁺ 18]	2018	Liver CT synthesis from annotation & lesion classification	DCGAN	Accuracy, Specificity & Sensitivity
[SV18]	2018	Brain MRI (T1w) super-resolution	3D adaptation of SRGAN	PSNR & SSIM
[CGM ⁺ 18]	2018	Retina fundus image synthesis from vessel tree annotations	<i>pix2pix</i> & Adversarial Autoencoder	ISC
[NTL ⁺ 17]	2018	Brain and pelvis MRI to CT conversion	GAN with 3D FCN generator	MAE & PSNR
[HHR ⁺ 18]	2018	Brain MRI (T1/T2w, T1c & Flair) synthesis	DCGAN & WGAN	VTT & Accuracy
[DYK ⁺ 19]	2018	Brain MRI (T1/T2w, T1c & Flair) synthesis	PGAN & Cycle GAN	SSIM & PSNR
[AJF ⁺ 18]	2018	PET-CT conversion, MRI motion correction & PET denoising	MedGAN	SSIM, PSNR, MSE, VIF, UQI & LPIPS
[KA19]	2019	MRI to CT conversion	U-Net & <i>pix2pix</i>	MAE, MSE & PSNR

MAE - Mean Absolute Error; MSE - Mean Square Error; PSNR - Peak Signal-To-Noise Ratio; MRA - Magnetic Resonance Angiography; SSIM - Structural Similarity Index; ISC - Image Structure Clustering; VTT - Visual Turing Test; VIF - Visual Information Fidelity; UQI - Universal Quality Index; LPIPS - Learned Perceptual Image Patch Similarity; T1w/T2w - T1/T2 weighted; CNN - Convolutional Neural Network; FCN - Fully Convolutional Neural Network.

The majority of the works published use brain MRI images, either for synthesis [NTL⁺18, DYK⁺19, HHR⁺18], super-resolution [SV18] or modality conversion (e.g. T1 weighted MRI to T2 weighted, MRI to CT, among others) [WDS⁺17, OSDU18, CJGT18, NTL⁺17]. Most works rely on one or several of the models reviewed throughout this chapter - for instance, the *pix2pix* is

applied in [NTL⁺18, CGM⁺18, KA19] and the Cycle GAN is used in [WDS⁺17, DYK⁺19]. No work concerning thoracic MRI synthesis was found in the literature.

4.4 Evaluation of generative models

Standardizing the evaluation of generative models is critical and has been a topic of great debate in recent years. There are two main trends for evaluating generative models: **objective metrics**, using explicit numerical criteria, and **perceptual studies**, which rely only on human judgment. Both will be discussed below.

Objective metrics encompass both traditional similarity scores and tasks such as detection, classification or feature extraction and comparison. Within this class, we highlight the Peak Signal-To-Noise Ratio, Structural Similarity Index, Inception Score, the Fréchet Inception Distance, and metrics based on Semantic Segmentation. As varied as these methods may be, none of them evaluate the human understanding of synthetic data. **Perceptual studies** fill that gap by having human observers analyze and compare real and fake images. These studies are driven by the human ability to provide much richer interpretations of fake data than any algorithm. Nevertheless, they are prone to inconsistencies since they rely on subjective analysis.

4.4.1 Perceptual studies

Perceptual studies consist of having human annotators classify a set of real and fake images and using those classifications to extract a metric quantifying how accurately the images were classified [SGZ⁺16]. The main limitations of this strategy are that the results produced often vary with the motivation of the subjects and the setup of the task. It produces considerably worse results when the subjects are given feedback on their performances, as they learn to detect the flaws of the synthetic images.

The Visual Turing Test, (VTT), introduced in [GGHY15], proposed a standardized way of performing this evaluation. The VTT poses binary questions to assess a system's ability to recognize objects and identify attributes and relationships in images. Several medical image generation works, such as [CHBB18], have presented the VTT to radiologists to judge the quality of their images and models.

4.4.2 Inception Score

The Inception Score (IS), introduced in [SGZ⁺16], was proposed as an alternative to perceptual studies for the evaluation of GANs, overcoming their downsides but still presenting high correlation with human evaluation. To judge the quality of a GAN, the IS evaluates if the generator is able to synthesize a great variety of images and if the images contain meaningful objects.

The IS consists of applying the Inception network [SLJ⁺15], which is one of the most widely used classification networks, pre-trained on the ImageNet dataset, to obtain the conditional label distribution $p(y|x)$. On the one hand, if the model is able to generate a great variety of images,

the marginal $\int p(y|x = G(z)) dz$ will have high entropy. On the other hand, if an image contains meaningful objects, its corresponding $p(y|x)$ will have low entropy. Combining the previous two goals results in the IS, which is stated in Equation 4.18, where $p(y)$ indicates the marginal class distribution, which can be computed through the integral $\int_x p(y|x)p_g(x)$. The IS has a minimum value of zero, but has no upper bound. However, an upper bound can be defined by computing the IS over real images, as suggested in [Bor19]. Higher values of the IS are preferable since they indicate that the model is highly capable of generating both diverse and believable images.

$$IS = \exp(\mathbb{E}_x KL(p(y|x)||p(y))) \quad (4.18)$$

As innovative as it was, the IS has its own limitations, analyzed in [BS18]. The most impactful issues of this score are the following:

- 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 is limited by what the Inception network can detect, which depends on the network's architecture and on the training data used to train it. If the IS is used to evaluate synthetic images from a domain unrelated with the ones used to pre-train the network, low IS scores may be obtained even if the fake images have high quality. Low IS values will also be produced if the training and test labels are different;
- 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 [Bor19].
- The IS is an asymmetric measure and is affected by image resolution [Bor19];

4.4.3 Fréchet Inception Distance

The Fréchet inception distance (FID) [HRU⁺17] was developed to overcome the restraints of the IS. To evaluate the generator's performance, the FID measures the Fréchet distance between two multivariate Gaussians, modeled by the 2048-dimensional activations of the Inception-v3 pool3 layer for real and synthetic images. Equation 4.19 details the FID calculation, where $X_r \sim \mathcal{N}(\mu_r, \Sigma_r)$ and $X_g \sim \mathcal{N}(\mu_g, \Sigma_g)$ refer, respectively, to the activations of the pooling layer mentioned above for real and fake images.

$$FID = ||\mu_r - \mu_g||^2 + \text{Tr}(\Sigma_r + \Sigma_g - 2(\Sigma_r \Sigma_g)^{1/2}) \quad (4.19)$$

As in the IS, the FID has 0 as its lower bound but has no upper bound. A lower FID value will indicate higher similarity between the real image and the corresponding synthetic one and, therefore, a better synthesis quality. In [HRU⁺17] is shown that the FID is more robust to noise and blurring than the IS, besides also being more sensitive to mode collapse and capable of evaluating intra-class variability. Nevertheless, this metric depends of what the Inception network can detect, as explained for the IS. Additionally, the FID assumes that Gaussian distributions correctly model the feature maps extracted and only takes into consideration the first two order moments of the distributions, which may not be suitable approximations [Bor19]. Besides the FID, other improvements to the IS have been proposed, such as the Modified Inception Score [GSB17].

4.4.4 Peak Signal-To-Noise Ratio

Peak signal-to-noise ratio (PSNR) was first proposed for image quality assessment in [WBSS04]. According to a posterior analysis on the topic, presented in [HZ10], the PSNR between two images, f and g , of size $M \times N$, can be computed through the expression given in Equation 4.20, where MSE refers to the mean squared error, stated in Equation 4.21.

$$PSNR(f, g) = 10 \log_{10} \left(\frac{255^2}{MSE(f, g)} \right) \quad (4.20)$$

$$MSE(f, g) = \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N (f_{i,j} - g_{i,j})^2 \quad (4.21)$$

As the MSE tends to zero, the PSNR will approach infinity, indicating high image similarity. Oppositely, as the MSE increases, the PSNR will decrease, indicating that there are great differences between the images. Despite being simple to compute and interpret, according to [WBSS04], PSNR results do not always correlate well with perceived visual quality, which was expected since this metric has no way of assessing structural similarity.

4.4.5 Structural Similarity Index

The application of the structural similarity index (SSIM) to the image context was also first proposed in [WBSS04]. The SSIM was design to improve upon the PSNR and MSE (metrics based on error summation) and measures image dissimilarity through three criteria: luminance distortion, contrast distortion and loss of structural correlation. Image distortions are usually less visible in bright or textured regions, which motivated the use of the luminance and contrasts as criteria, respectively. Furthermore, spatially close pixels are considered to have strong interdependence, carrying important information concerning object structure, which justifies the use of a term to evaluate structural correlation.

Equation 4.22 states the SSIM calculation between to images f and g , where l , c and s refer, respectively, to the luminance, contrast and structure comparison functions, all of which are discriminated in Equation 4.23. In 4.23, μ_f and μ_g are average intensity values, σ_f and σ_g are standard deviation values, σ_{fg} is the co-variance matrix, and C_1 , C_2 and C_3 are constants. The

local statistics μ_f , μ_g , σ_f , σ_g and σ_{fg} are usually computed in neighborhoods using sliding windows, typically 8x8 square window or 11 x 11 circular symmetric Gaussian window. The SSIM values are bounded between -1 and 1, 0 meaning no correlation between the images, 1 meaning the images are equal and -1 indicating that local image structure is inverted [BXP08].

$$SSIM(f, g) = l(f, g)c(f, g)s(f, g) \quad (4.22)$$

$$\begin{cases} l(f, g) = \frac{2\mu_f\mu_g + C_1}{\mu_f^2 + \mu_g^2 + C_1} \\ c(f, g) = \frac{2\sigma_f\sigma_g + C_2}{\sigma_f^2 + \sigma_g^2 + C_2} \\ s(f, g) = \frac{\sigma_{fg} + C_3}{\sigma_f\sigma_g + C_3} \end{cases} \quad (4.23)$$

4.4.6 Semantic Segmentation Metrics

Several works defend that image realism can be quantified by segmenting the generated images with a semantic segmentation network, pre-trained with real images, and extracting metrics (e.g. Dice score) to evaluate the accuracy of the segmentation. If the synthetic images are realistic, the network will be able to segment them correctly. This strategy is especially used to evaluate image-to-image translation applications such as [IZZE17], [ZPIE17], and [WLZ⁺18], the first two applying the FCN-8s [SLD17] architecture for the segmentation network and the last applying the PSPNet [ZSQ⁺17].

4.5 Summary

Throughout this chapter, we addressed some of the motivations behind generative modeling, as well as the two most relevant classes of models for our purposes, which are CGANs and CVAEs. GANs and VAEs have different approaches to data generation: while GANs pitch two models against each other, a generator and a discriminator, VAEs generate data by encoding an input image into a latent space, sampling that same space and reconstructing the image using those samples. Both models are easily adapted to conditional variants. Nevertheless, GANs are harder to train than VAEs, but produce higher quality results, since VAEs can generate blurry results due to the reconstruction error.

Recently, GANs were successfully applied for image-to-image translation applications, which show great potential for the synthesis of medical images from input annotations. Despite GANs and VAEs being novel approaches for data generation, important works applying them for the synthesis of medical images have already emerged. However, most of them are dedicated to brain MRI synthesis or modality conversion, which shows the importance of the work we executed in filling this gap.

We also reviewed some of the most popular strategies for the evaluation of generative models. Despite perceptual studies being considered a gold standard for the assessment of graphical

realism, researchers strive to find a reliable quantitative measures that does not require human intervention. From those efforts resulted novel metrics such as the IS, which served as a baseline for further developments, such as the FID, which we find of great interest.

Many other evaluation metrics, which are not contemplated in this chapter, are reviewed in [Bor19]. Nonetheless, the scientific community has not reached a consensus of which ones capture better the strengths and shortcomings of generative models.

Chapter 5

Thoracic MRI synthesis

Throughout this chapter, we describe all models and metrics used for the practical work executed, as well as the specific implementation details and dataset. This is followed by the presentation and analysis of all results collected.

5.1 Methods

5.1.1 Synthesis

We adopted all three methods explored in Chapter 4.1.3 - *pix2pix*, *pix2pix_{HD}*, and CycleGAN - as baselines for the synthesis of 2D thoracic MRI images, and performed several variations to assess if they could be improved. For all methods, the loss functions used were the ones described in Chapter 4.1.3, except when indicated otherwise.

Concerning the *pix2pix*, the versions implemented differed in the generator network's architecture. The versions are the following:

- *pix2pixV1* - original generator architecture, based on the U-Net, as proposed in [IZZE17];
- *pix2pixV2* - generator architecture proposed for the CycleGAN in [ZPIE17];
- *pix2pixV3* - generator architecture proposed for the *pix2pix HD* in [WLZ⁺18].

These variations intend to assess if the basic *pix2pix* framework can be improved by changing the generator's architecture. We oppose an architecture with skip connections (*pix2pixV1*) with two architectures with residual learning blocks (*pix2pixV2* and *pix2pixV3*) to evaluate which of the strategies is more effective in perpetuating context information throughout the network. Opposing *pix2pixV2* to *pix2pixV3* allows the study of the efficacy of the coarse-to-fine architecture of *pix2pixV3* in relation to a similar but simpler architecture.

For *pix2pixV1* and *pix2pixV2*, two versions of the generator were conceived: one for a single channel input and another for a 7-channel input - respectively *pix2pixV1_{OHE}* and *pix2pixV2_{OHE}*. While the single-channel input is a simple categorical annotation, the one-hot encoded input is a multi-channel annotation in which each channel corresponds to a binary image referring to each

label of the categorical annotation. This will allow us to verify which of the input codifications is more efficient for the translation task.

Regarding the pix2pix_{HD} , the versions implemented differed in the loss function. The versions are the following:

- $\text{pix2pix}_{HD}\text{V1}$ - equal to the original architecture, proposed in [WLZ⁺18], but without the use of instance maps and instance-level feature embedding;
- $\text{pix2pix}_{HD}\text{V2}$ - equal to $\text{pix2pix}_{HD}\text{V1}$ but applying the loss function proposed for the $\text{pix2pix}\text{V1}$.

The variation $\text{pix2pix}_{HD}\text{V2}$ intends to evaluate if the feature matching and perceptual losses, proposed in [WLZ⁺18], indeed improve the learning process in relation to the traditional image-to-image translation loss function. None of the variants implemented use instance maps because, in the dataset used, different object of the same label never contact with each other. Instance-level feature embedding was also discarded for both versions because instance level manipulation is out of the scope of our goals for this work.

The **CycleGAN** was implemented exactly as it was proposed in [ZPIE17], but without batch discrimination.

The notation that will be used to detail the composition of all networks is the following:

- $\mathbf{C}_k$: layer with Convolution - Batch/ Instance Normalization - ReLU with k filters;
- $\mathbf{TC}_k$: layer with Transposed Convolution - Batch/ Instance Normalization - ReLU, with k filters;
- $\mathbf{TCD}_k$: layer with Transposed Convolution - Batch/ Instance Normalization - Drop Out with probability of 0.5 - ReLU, with k filters;
- $\mathbf{R}_k$: residual block consisting in Convolution - Batch/ Instance Normalization - ReLU - Convolution, with k filters for the input and output of all convolutions.

For $\text{pix2pix}\text{V1}$, as suggested in suggested in [IZZE17], we implemented a U-Net based architecture - an encoder-decoder with skip-connections - but without max-poolings. While the encoder is a compressing path that computes features, decreasing the size and increasing the amount of the feature maps, the decoder is an expanding path that locates the patterns intended, reducing the number but increasing the size of the feature maps. Skip-connections (concatenation of feature maps) are established between each layer i of the encoder and each layer $n-i$ of the decoder. They provide better gradient flow and transfer context information from the encoder to the decoder at different scales, helping to locate the patterns. The architectures of the encoder and decoder implemented are similar to the ones described in [IZZE17], and can be represented as follows:

Encoder: $\mathbf{C}_{64}-\mathbf{C}_{128}-\mathbf{C}_{256}-\mathbf{C}_{512}-\mathbf{C}_{512}-\mathbf{C}_{512}-\mathbf{C}_{512}$

Decoder: $\mathbf{TCD}_{512}-\mathbf{TCD}_{512}-\mathbf{TCD}_{512}-\mathbf{TCD}_{512}-\mathbf{TCD}_{256}-\mathbf{TCD}_{128}-\mathbf{TCD}_{64}-\mathbf{TCD}_{64}-\mathbf{C}_1$

All convolutions have a kernel size of 4×4 , padding of 1 and stride of 2, except in the first layer of the encoder ($\mathbf{C}_{64}$) which uses a stride of 1. $\mathbf{C}_{64}$ also does not apply batch normalization.

The activations of the encoder are Leaky ReLUs with slope 0.2, while the ones of the decoder are regular ReLUs. In the last layer of the decoder, C_1 , batch normalization is not applied and the activation function used is the hyperbolic tangent. This architecture is outlined in Figure 5.1.

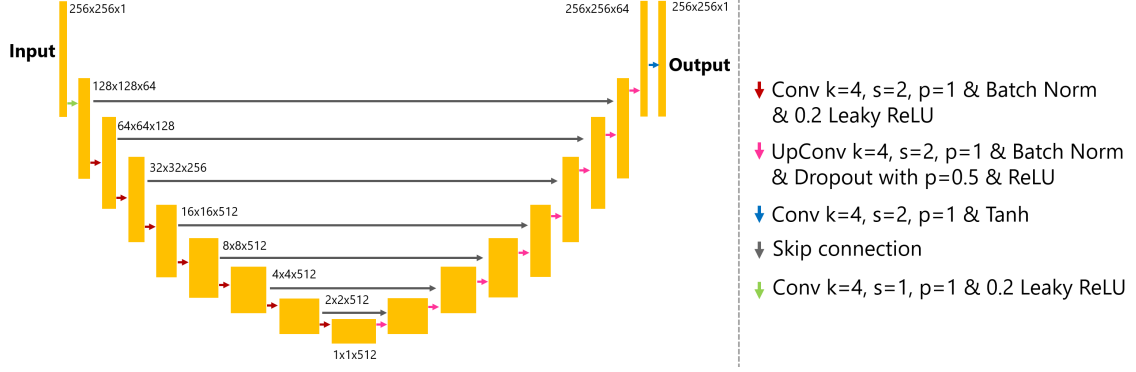


Figure 5.1: U-Net based generator architecture.

For *pix2pixV2* and the CycleGAN, we implemented the generator architecture suggested for the CycleGAN in [ZPIE17], which uses residual learning blocks [HZRS16]. These blocks perform identity shortcut connections and are illustrated in Figure 5.2. Using the notation presented above, the architecture is the following:

$$C_{64}-C_{128}-C_{256}-R_{256}-R_{256}-R_{256}-R_{256}-R_{256}-R_{256}-R_{256}-R_{256}-R_{256}-TC_{128}-TC_{64}-C_1$$

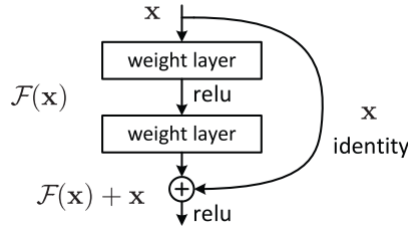


Figure 5.2: Residual learning block from the ResNet. Source: [HZRS16].

For the C_{64} and C_1 modules, the convolutions have a kernel size of 7×7 , a stride of 1 and padding of 3, while all other convolutions have a kernel size of 3×3 , a stride of 2 and a padding of 1. For C_1 , no batch normalization is used and the activation function applied is the hyperbolic tangent. This architecture is outlined in Figure 5.3.

For *pix2pixV3*, *pix2pix_{HD}V1* and *pix2pix_{HD}V2*, we implemented the generator architecture proposed in [WLZ⁺18] and explained in Chapter 4.1.3.2. The compositions of the global generator and local enhancer networks are the following:

Global generator: $C_{64}-C_{128}-C_{256}-C_{512}-R_{512}-R_{512}-R_{512}-R_{512}-R_{512}-R_{512}-R_{512}-R_{512}-TC_{256}-TC_{128}-TC_{64}-C_1$

Local enhancer: $C_{32}-C_{64}-R_{64}-R_{64}-TC_{32}-C_1$

For the Global Generator and the Local Enhancer, the first and last convolutions have a kernel size of 7×7 , a stride of 1 and a padding of 3, while all other convolutional layers have a kernel

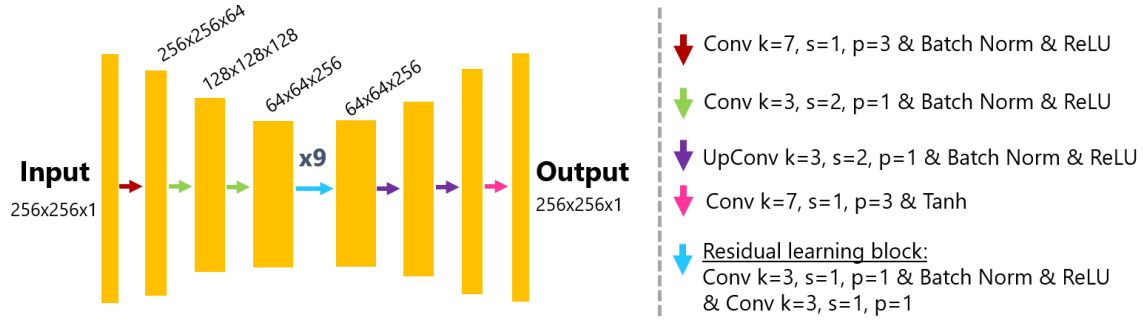


Figure 5.3: ResNet based generator architecture.

size of 3×3 , padding of 1 and stride of 2. This architecture is outlined in Figure 5.4. Both the CycleGAN and *pix2pix_{HD}* generators use reflection padding to reduce boundary artifacts.

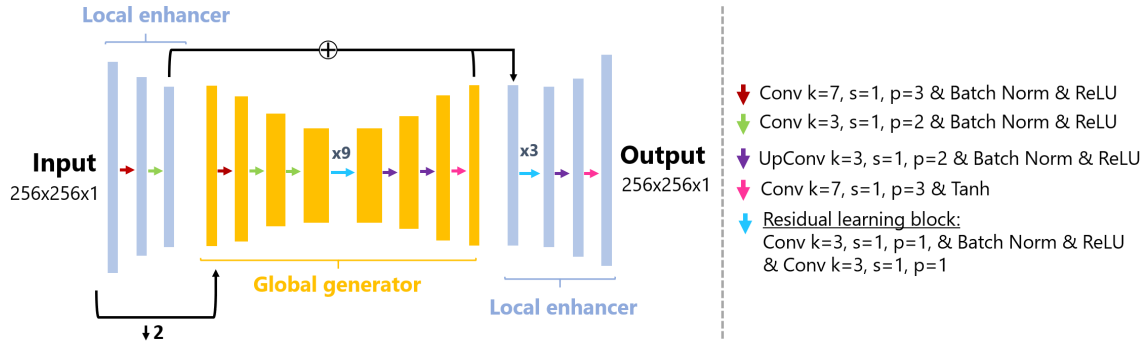


Figure 5.4: Coarse-to-fine generator architecture.

The discriminator network architecture used was, for all cases, a 30×30 PatchGAN. Its composition is the following:

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

All convolutions use a kernel size of 4×4 and padding of 1, however, C_{64} , C_{128} and C_{256} have a stride of 2, while C_{512} and C_1 have a stride of 1. The first and last convolutions, C_{64} and C_1 , are not followed by any type of normalization. All activations are leaky ReLUs with a slope of 0.2, except for C_1 , which applies a Sigmoid function after the convolution. Figure 5.5 contains a scheme that summarizes this architecture. For *pix2pix* and *pix2pix_{HD}*, the discriminator's input is a concatenation of the real or fake mri and the corresponding annotation.

Table 5.1 summarizes all approaches described above.

5.1.2 Evaluation

To evaluate the quality of the synthetic images, we propose using two novel metrics: a normalized variant of Mutual Information, and the Dice score extracted after performing semantic segmentation on the synthetic images using the Global Convolutional Network [PZY⁺17].

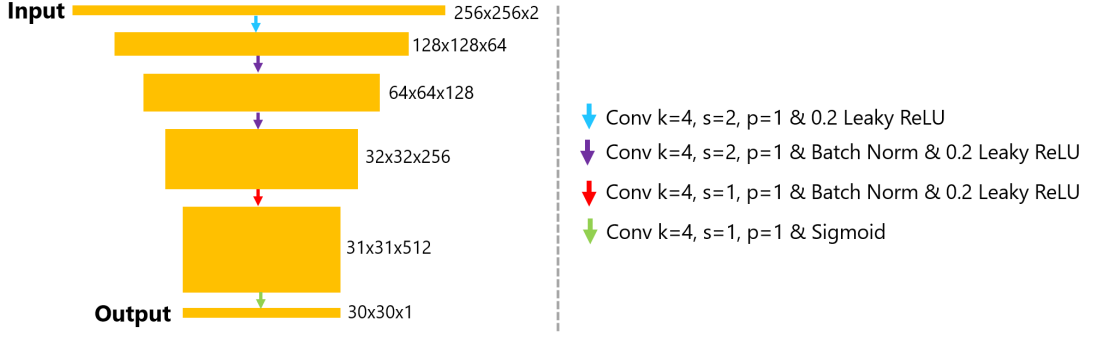
Figure 5.5: Architecture of the 30×30 PatchGAN.

Table 5.1: Summary of all models used for the synthesis task

Model	Baseline	Modification
<i>pix2pix</i> V1	<i>pix2pix</i>	None
<i>pix2pix</i> V1 _{OHE}	<i>pix2pix</i>	Multi-channel input
<i>pix2pix</i> V2	<i>pix2pix</i>	Generator architecture with residual learning blocks
<i>pix2pix</i> V2 _{OHE}	<i>pix2pix</i>	Generator architecture with residual learning blocks and multi-channel input
<i>pix2pix</i> V3	<i>pix2pix</i>	Generator architecture proposed in [WDS ⁺ 17]
<i>pix2pix</i> _{HD} V1	<i>pix2pix</i> _{HD}	No instance maps and instace-level feature embedding
<i>pix2pix</i> _{HD} V2	<i>pix2pix</i> _{HD}	No instance maps and instace-level feature embedding; uses the loss proposed for the <i>pix2pix</i>
CycleGAN	CycleGAN	No batch discrimination

5.1.2.1 Normalized Mutual Information

The maximization of mutual information (MI) is amply used for the registration of medical images [MVS03]. We use the normalized mutual information (NMI) to evaluate the similarity between real images and corresponding synthetic ones. For each real-fake image pair, the structures for the available labels will be aligned, as they should share the same anatomy, the same acquisition modality and the same annotation.

The MI between two images measures how much knowing one of them reduces uncertainty about the other. Conversely, as it is a pixel wise metric, it may dismiss valuable spatial information about the neighborhood of the pixel. Equation 5.1 details the calculation of the MI between two discrete random variables X and Y , where $p(x, y)$ refers to the joint probability mass function of $x \in X$ and $y \in Y$, and $p(x)$ and $p(y)$ correspond, respectively, to the marginal probability mass functions of x and y . A higher MI value will indicate a higher similarity between the real and corresponding synthetic image and, therefore, better synthesis quality.

$$I(X; Y) = \sum_{y \in Y} \sum_{x \in X} p(x, y) \log \left(\frac{p(x, y)}{p(x) p(y)} \right) \quad (5.1)$$

For the NMI, we normalize the MI by dividing it by the average of the entropy of X , $H(X)$, and the entropy of Y , $H(Y)$, as shown in Equation 5.2. This bounds the score to the $[0,1]$ range.

$$I(X;Y)_{norm} = \frac{I(X;Y)}{\mu(H(X), H(Y))} \quad (5.2)$$

5.1.2.2 Semantic Segmentation Metric

The use of segmentation related metrics to evaluate the quality of the synthesis of fake images was already introduced in Chapter 4.4.6. We propose training a semantic segmentation network, particularly the Global Convolutional Network (GCN) [PZY⁺17], for a multi-class semantic segmentation task using our dataset. We chose the GCN since it was developed as an improvement of the FCN, which was used to assess image quality in several image-to-image translation works, as stated in Chapter 4.4.6.

The major motivation supporting the GCN is the fact that semantic segmentation can be regarded as a per-pixel classification problem, which implies correctly classifying and locating each object of interest, which are tasks that require opposite models. On the one hand, for classification, the label predicted should always be the same independently of the object's position, so it requires transformation (e.g. rotation, translation) invariant models, usually with fully connected layers. On the other hand, localization seeks to precisely locate every pixel of each label, which requires transformation sensitive models, normally fully convolutional. This dichotomy is illustrated in Figures 5.6a and 5.6b.

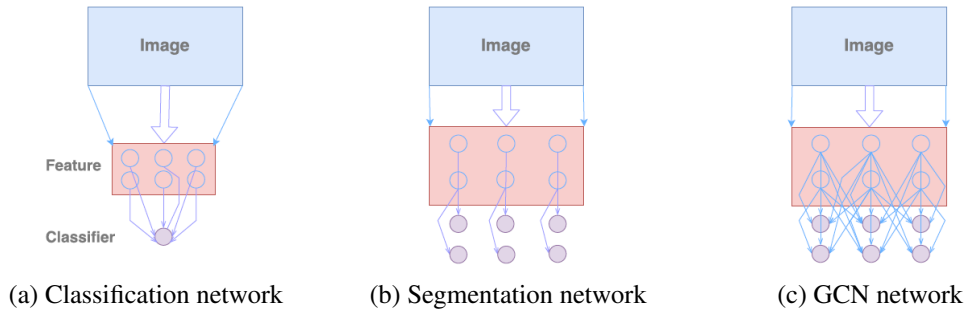


Figure 5.6: Comparison of the general architectures used for classification and segmentation with the architecture proposed by the GCN. Source: [PZY⁺17].

The GCN combines the two previous goals - for localization purposes, it is fully convolutional and has no fully connected or pooling layers that could discard location information; for classification purposes, it uses large kernels, which provides dense connections, making the model robust to transformations. This is illustrated in Figure 5.6c. To make the use of large kernels feasible, the GCN applies symmetric, separable filters, reducing the amount of parameters as well as the computational cost.

The GCN uses the FCN as a baseline and improves it using the above mentioned principles. As Figure 5.7 shows, the FCN applies a succession of convolutions and poolings, progressively

increasing the number of feature maps and reducing their dimensions until they reach $\frac{1}{32}$ of the original image's height and width. Predictions can be extracted by three different strategies:

- Directly upsampling the output of the last convolutional layer (conv7) by $32\times$, resulting in the original image dimensions, as shown in Figure 5.7. This generates the FCN-32s model;
- Summing the outputs of conv7, upsampled $2\times$, and the fourth pooling layer (pool4), as in Figure 5.7, and subsequent upsampling of $16\times$. This allows the prediction of finer details and is called the FCN-16s model;
- Summing the output of conv7, upsampled $4\times$, the output of pool4, upsampled $2\times$, and the output of the third pooling layer (pool3), followed by an $8\times$ upsampling. This is also shown in Figure 5.7 and further refines the prediction. This last version is named FCN-8s.

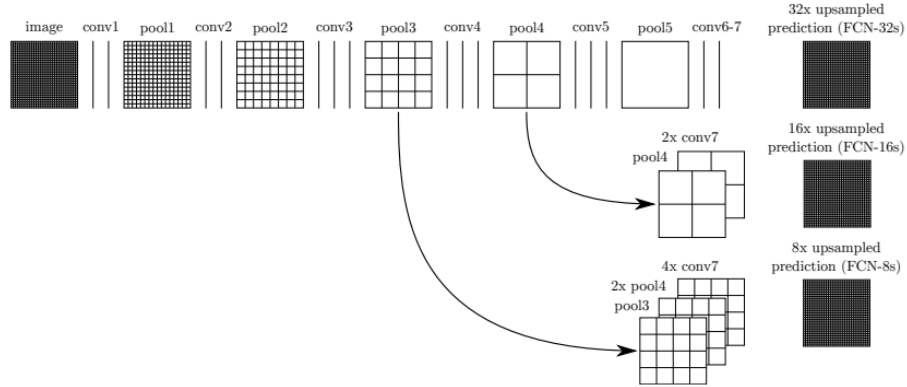


Figure 5.7: FCN architecture variants. Source: [SLD17].

Figure 5.8 illustrates the operations applied by the GCN. It first uses a pre-trained ResNet to extract features from the input images, resulting in feature maps of different scales. To each set of feature maps is applied the GCN module and a boundary refinement module, both illustrated in Figure 5.8. The GCN module applies a combination of $1 \times k + k \times 1$ and $k \times 1 + 1 \times k$ convolutions, which provides dense connections without the use global convolutions, resulting in a smaller amount of parameters. The boundary refinement module was designed to enhance the segmentation of the object's boundaries and consists of a parallel branch that applies a convolution, a ReLU and another convolution, adding this result to the module's input.

The GCN was implemented exactly as detailed in Figure 5.8 and optimized using the binary cross entropy loss, shown in Equation 5.3, where n refers to each class.

$$\mathcal{L}_n = -(y_n \log(p(y_n)) + (1 - y_n) \log(1 - p(y_n))) \quad (5.3)$$

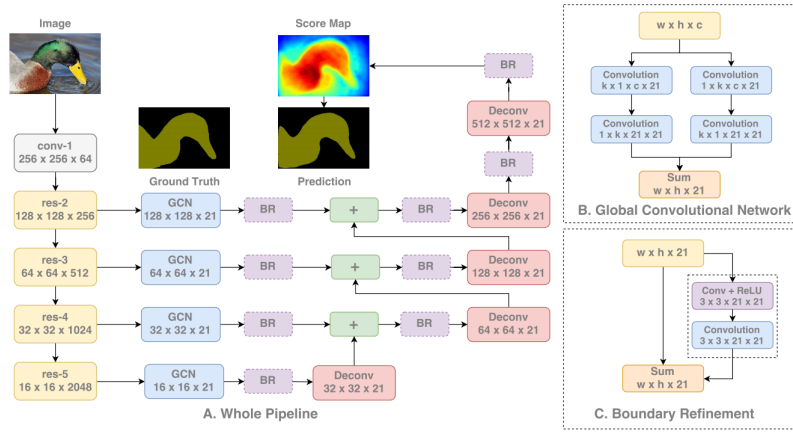


Figure 5.8: GCN architecture. Source: [PZY⁺17].

5.2 Dataset

The dataset used was provided by the BCCT.plan project¹ and consists of 25 T1-weighted thoracic MRI exams from breast cancer patients, obtained with a Philips Ingenia 3.0T MRI scanner. Each exam comprises 60 gray-scale axial images, with a 3mm thickness and a resolution of 720×720 pixels (0.3-0.5 mm/pixel).

All exams have corresponding annotations consisting of binary masks of the sternum, clavicle, nipple, pectoral muscle and breasts, all drawn manually by experts. Additionally, we automatically generated a label for the area occupied by the thoracic cavity between the rightmost and leftmost limits of the breasts label, as Figure 5.9 illustrates. Without this extra label, despite having very different intensity distributions, the entire thoracic cavity region and the background would have the same label, which would induce artifacts in the synthetic images.

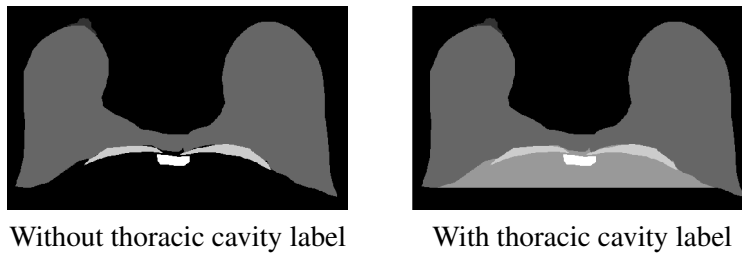


Figure 5.9: Illustration of the automatic label generation for the thoracic cavity.

Of the 25 exams available, 19 were assigned for the training set, 3 for the validation set and 3 for test set. Images with absent or incomplete annotations (e.g. 2D images with the sternum annotated but without the breast annotation) were excluded, resulting in 1152 images for training, 168 for validation and 170 for testing. All images were cropped into a square shape, excluding most of the background area, and resized to 256×256 pixels.

¹BCCT.plan - 3D tool for planning breast cancer conservative treatment - NORTE-01-0247-FEDER-017688

5.3 Implementation details

For the synthesis task, all models were trained during 200 epochs, using stochastic gradient descent with the *Adam* solver [KRMW14], with a learning rate of 0.0002 and momentum parameters $\beta_1 = 0.5$ and $\beta_2 = 0.999$. The learning rate was kept constant during the first 100 epochs, linearly decaying until 0 for the remaining epochs. This strategy was introduced for the training of the *pix2pix* framework in [IZZE17] and perpetuated in subsequent works. Additionally, we used the LSGAN goal instead of the classical GAN goal to stabilize the training, used a batch size of 1, and alternated between one gradient descent step in the discriminator and then one in the generator.

We used the loss function shown in Equation 4.5 for *pix2pixV1*, *pix2pixV1_{OHE}*, *pix2pixV2*, *pix2pixV2_{OHE}*, *pix2pixV3* and *pix2pix_{HD}V2* with $\lambda = 10$, as recommended in [IZZE17]. For *pix2pix_{HD}V1* we applied the loss function detailed in Equation 4.8 with $\lambda = 10$, as advised in [WLZ⁺18]. Lastly, for the Cycle GAN we combined the loss function from Equation 4.10, with $\lambda = 10$ (as suggested in [ZPIE17]), with the identity loss depicted in Equation 4.11, with a weight of 5.

Concerning the optimization of the GCN for the segmentation task, this network was trained during 100 epochs, also using *Adam* as the solver, having the same parameters except the learning rate, which was 0.0001. Each epoch was evaluated on a validation set by computing the Dice score between the predictions and the ground truths for the structures which we wanted to generate with the most quality (breasts, nipple, pectoral muscle, sternum and clavicle), and the weights of the best performing epoch were saved. The loss function used was the binary cross entropy, stated in Equation 5.3. We trained the network both with and without data augmentation, and the latter achieved greater Dice scores. The data augmentation consisted of randomly flipping the images horizontally, applying luminance and contrast changes, as well as salt and pepper noise.

All models were implemented in Python, using the Pytorch library.

5.4 Results and discussion

5.4.1 Fréchet Inception Distance study

Since the FID metric was only recently developed, it is neither widely used or studied. We find this metric of particular interest because it is based in feature extraction and comparison, which is a principle also applied in feature matching losses, which have been proven to greatly improve the performance of GANs. This, together with lack of reference values in the literature and the absence of an upper bound for this metric, motivated us to study it further. To do so, we computed the FID between a reference image and that same image tampered with specific artifacts. Concretely, we used as reference images the *Lena*, a benchmark for image processing algorithms, and a 2D thoracic MRI image from our dataset. The tamperings chosen reflect some of the issues that may be found in artificial images, which are the following:

- T_1 - horizontal flip;

- T_2 - 20 pixels horizontal translation;
- T_3 - 10° rotation;
- T_4 - 140 % zoom;
- T_5 - gamma correction with $\gamma = 2.5$, resulting in a image with lower intensity distribution;
- T_6 - gamma correction with $\gamma = 0.5$, resulting in a image with higher intensity distribution;
- T_7 - Gaussian blur with $\sigma = 2$;
- T_8 - median blur with size 15;
- T_9 - downsampling by $\frac{1}{4}$ of the images' resolution with aliasing, resulting in a blurry image;
- T_{10} - salt and pepper noise.

Figures 5.10 and 5.11 show, respectively, the reference and tampered *Lena* and MRI images. All FID values computed are presented in Table 5.2.

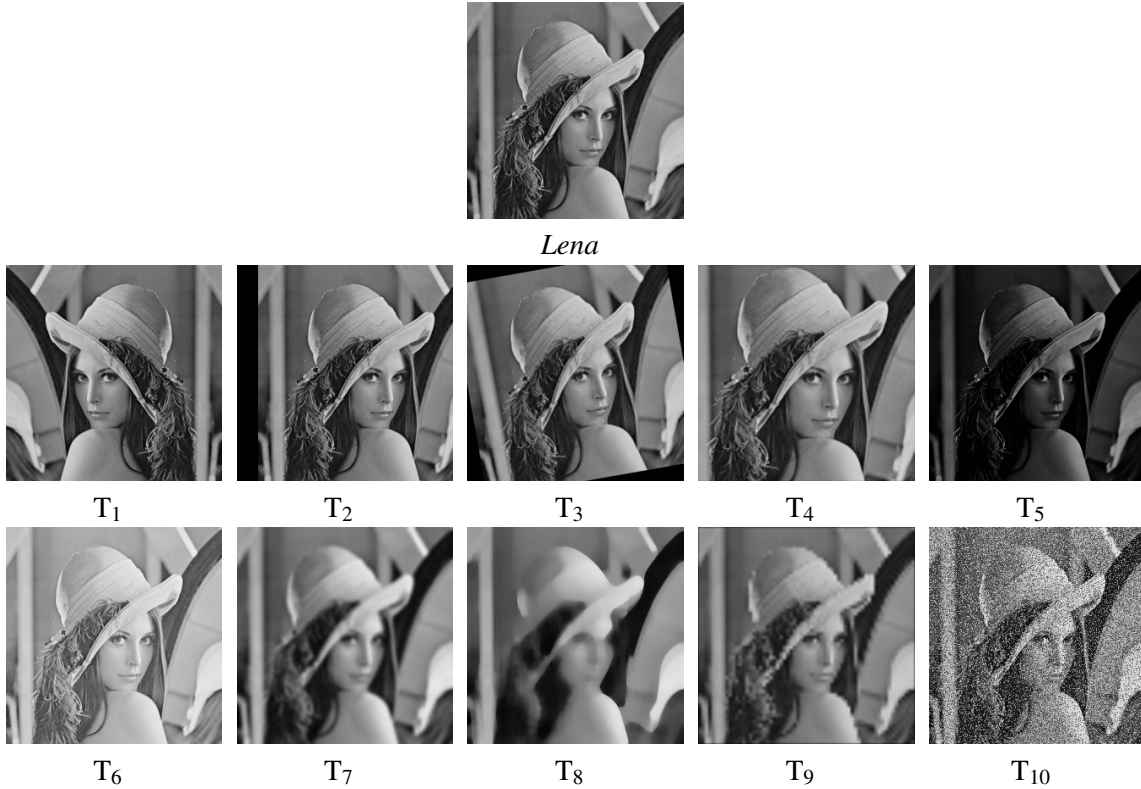


Figure 5.10: Reference and tampered *Lena* images, with the following transformations: T_1 - horizontal flip; T_2 - 20 pixels horizontal translation; T_3 - 10° rotation; T_4 - 140 % zoom; T_5 - gamma correction $\gamma = 2.5$; T_6 - gamma correction $\gamma = 0.5$; T_7 - Gaussian blur $\sigma = 2$; T_8 - median blur of size 15; T_9 - downsampling with aliasing; T_{10} - salt and pepper noise.

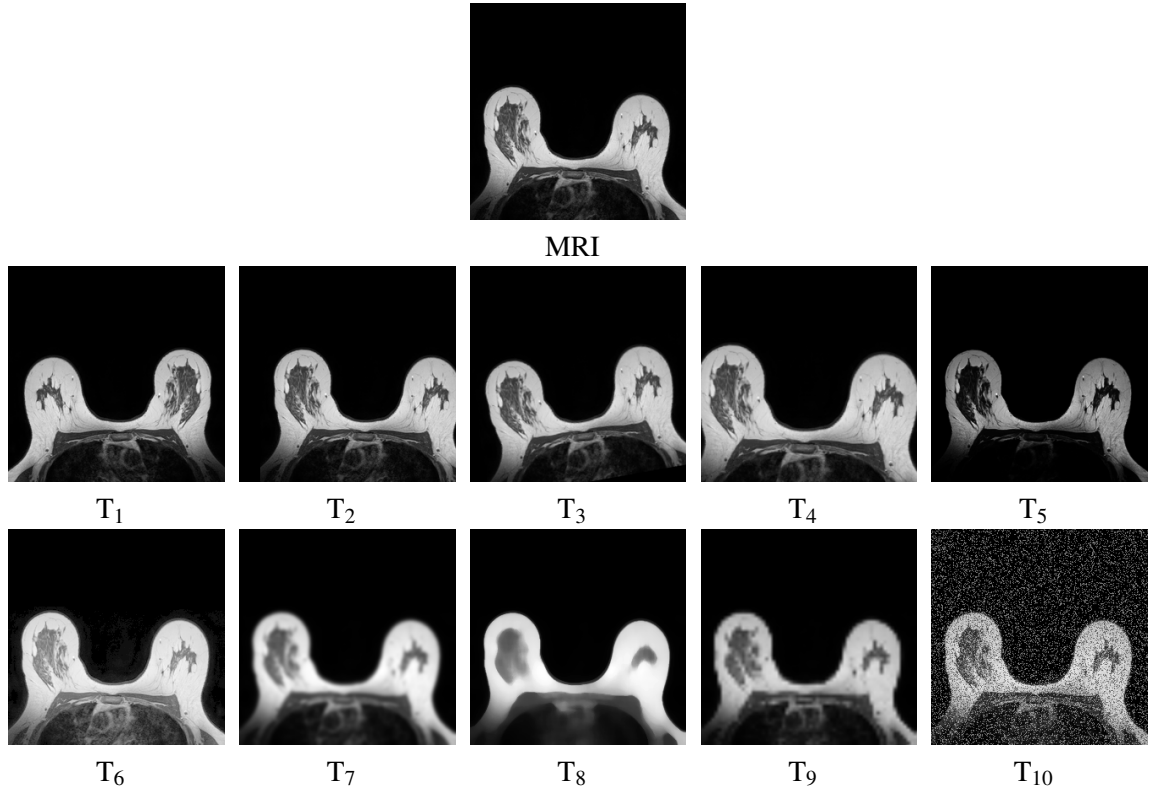


Figure 5.11: Reference and tampered MRI images, with the following transformations: T_1 - horizontal flip; T_2 - 20 pixels horizontal translation; T_3 - 10° rotation; T_4 - 140 % zoom; T_5 - gamma correction $\gamma = 2.5$; T_6 - gamma correction $\gamma = 0.5$; T_7 - Gaussian blur $\sigma = 2$; T_8 - median blur of size 15; T_9 - downsampling with aliasing ; T_{10} - salt and pepper noise.

Table 5.2: Reference FID values

	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9	T_{10}
<i>Lena</i>	30.376	50.713	269.132	52.324	36.064	16.906	173.874	264.458	287.841	431.582
MRI	49.840	69.891	129.482	92.639	107.296	133.346	293.852	229.086	273.175	367.131

Analyzing Table 5.2 shows that the FID scores obtained range from, approximately, 0 to 440. For both the *Lena* and the MRI references, the horizontal flip (T_1), horizontal translation (T_2) and zoom (T_4) transformations produce FID values under 100, which we consider to be low considering the range of FID values. This means that these transformations do not affect as much the features extracted by the Inception network as the remaining ones, which may occur due to the transformations affecting neither the pixel intensities or the image structure. Having small distances for these transformations also means that the FID is not very sensible to the detection of the miss-alignments produced by the transformations, which one may want to accurately detect in some applications.

For the *Lena* reference, both gamma correction transformations (T_5 with $\gamma = 2.5$ and T_6 with $\gamma = 0.5$) also produce FIDs under 100 (respectively 36.064 and 16.906), T_6 producing the lowest FID for this reference image. For the MRI reference, T_5 and T_6 produce FID scores of, respec-

tively, 107.296 and 133.346, which indicates that these transformations affect more the features extracted by the Inception network in our data than in the *Lena* reference image. Moreover, for the MRI reference, the lowest FID is achieved for the horizontal flip (T_1) transformation (49.840).

Transformations T_8 , T_9 and T_{10} (respectively median blur, downsampling with aliasing and salt and pepper noise) consistently produce high FID scores (over 200 for both reference images) considering the range of values obtained. We expected that all transformations that affect pixel intensities and overall image sharpness, namely blurs (T_7 , T_8 and T_9) and noises (T_{10}), to produce worse FID values, which is consistent with the results obtained. The salt and pepper noise (transformation T_{10}) produces the worst FID values for both reference images (431.582 for the *Lena* and 367.131 for the MRI) which leads us to conclude that this transformation deeply affects the features perceived by the Inception network.

Lastly, the rotation transformation (T_3) produced worse results than expected (FID of 269.132 for the *Lena* and of 129.482 for the MRI) considering that it does not affect the image's intensities and structure.

5.4.2 Thoracic MRI segmentation

To assess the success of the GCN network in segmenting our dataset's principal structures, we present in Table 5.3 the Dice similarity scores for the validation and test sets, using the weights of the epoch with the highest average validation Dice score, which was the 33rd epoch. To select the best training epoch, we computed the average Dice score of the structures we are most interested that our network segments efficiently, which are the nipple, breasts, pectoral muscle, sternum and clavicle. We applied the traditional Dice score formula, presented in Equation 5.4 and binarized the predictions using a threshold of 0.5. The threshold was chosen through experimentation.

$$DSC = \frac{2|X \cap Y|}{|X| + |Y|} \quad (5.4)$$

Table 5.3: GCN validation and test Dice scores

	Nipple		Breasts		Thoracic cavity		Pectoral		Sternum		Clavicle	
	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ
Validation set	0.157	0.241	0.873	0.157	0.627	0.259	0.661	0.184	0.601	0.319	0.298	0.231
Test set	0.073	0.188	0.955	0.041	0.745	0.226	0.695	0.194	0.689	0.320	0.602	0.289

The breasts are the structure with the highest Dice score for both the validation and the test sets which are, respectively, 0.873 and 0.955. This was expected since the breasts occupy large and visually well defined areas in the images, and are strongly represented throughout the training set. The inferior performance on the validation set may be due to, in some cases, the predictions including lateral skin sections, in continuity with the breasts, which are not included in the ground truth set. The validation set also includes some images in which the breasts' shape is less common and, therefore, harder to segment. Figure 5.12 illustrates both cases. Both situations occur less frequently in the test set.

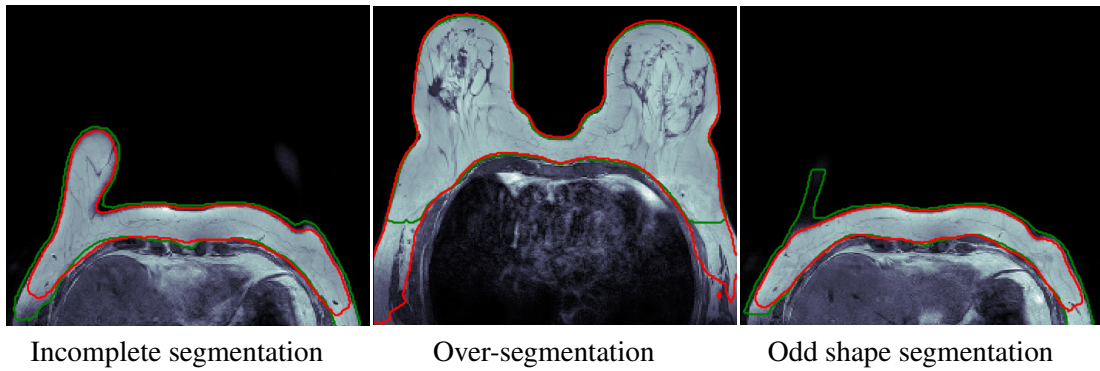


Figure 5.12: Examples of less accurate segmentations in the validation set. Each MRI image is overlaid with the real and predicted breasts edges in, respectively, green and red.

The sternum, pectoral muscle, clavicle and thoracic cavity also present higher average Dice scores for the test set, specially the last two, for which the difference in performance between the validation and test set are greater. Regarding the thoracic cavity, it is common for the prediction to be irregular in relation to the ground truth, which occurs more frequently in the validation set than in the test set. This case is illustrated in Figure 5.13.

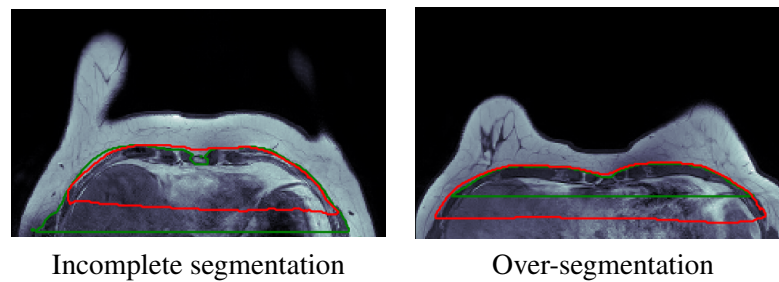


Figure 5.13: Examples of less accurate thoracic cavity segmentations in the validation set. Each MRI image is overlaid with the real and predicted thoracic cavity edges in, respectively, green and red.

The clavicle, sternum and pectoral muscle are not equally represented in the validation and test sets and occupy small and irregular areas within the images, which may explain the difference in performance. Concerning the clavicle, this structure has a more elongated shape in the validation set than in the training and test sets, in which the clavicles' shapes are more similar. This shape difference leads to a better clavicle segmentation in the test set than in the validation set, as can be observed in Figure 5.14.

Oppositely to the breasts, the nipple is the worst segmented structure in both sets due to its small representation in the training set, having Dice scores of 0.157 in the validation set and of 0.073 in the test set.

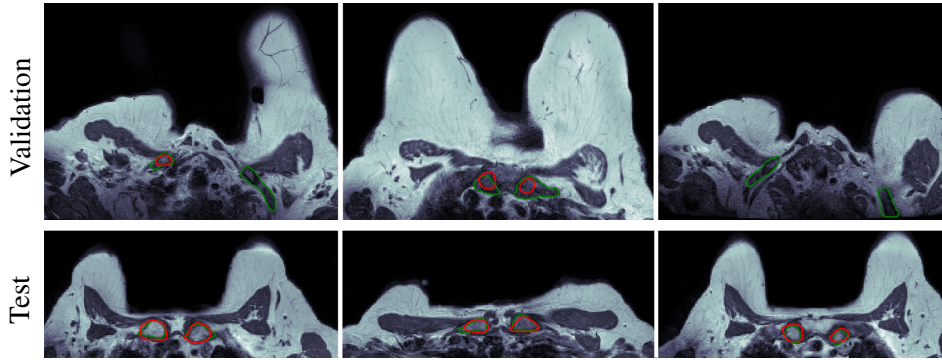


Figure 5.14: Examples of clavicle segmentation in the validation and test sets. Each MRI image is overlaid with the real and predicted clavicle edges in, respectively, green and red.

5.4.3 Thoracic MRI synthesis

To evaluate the quality of all models, we extracted quantitative measures from the synthetic images produced by each model. These measures were computed for the entire images and for each labeled structure separately. For the global image evaluation, we computed the SSIM, the NMI, and the FID. The FID was estimated using the official implementation code provided by the authors of [HRU⁺17], and the SSIM was computed using a 255×255 square window, since we wanted it to reflect the global aspect of the images. These results are presented in Table 5.4.

Table 5.4: Global image evaluation scores

	SSIM		NMI		FID
	μ	σ	μ	σ	
<i>pix2pix</i> V1	0.836	0.066	0.359	0.031	242.473
<i>pix2pix</i> V1 _{OHE}	0.753	0.109	0.314	0.060	261.040
<i>pix2pix</i> V2	0.785	0.081	0.314	0.037	218.662
<i>pix2pix</i> V2 _{OHE}	0.757	0.106	0.330	0.046	235.869
<i>pix2pix</i> V3	0.787	0.085	0.335	0.041	215.636
<i>pix2pix</i> _{HD} V1	0.848	0.065	0.376	0.033	210.650
<i>pix2pix</i> _{HD} V2	0.861	0.069	0.391	0.044	244.169
CycleGAN	0.813	0.085	0.334	0.029	186.320

SSIM - Structural Similarity Index; NMI - Normalized Mutual Information;

FID - Fréchet Inception Distance.

Concerning the individual structure evaluation, we computed the NMI and the Dice score for each labeled structure. The Dice score was extracted using the previously discussed GCN network. While the Dice score will evaluate how successfully the shape of each structure was generated, the NMI will evaluate if the content of each structure is coherent with the one from the corresponding real MRI image. The NMI and Dice values are shown, respectively, in Tables 5.5 and 5.6.

Table 5.5: Normalized Mutual Information scores of each labeled structure

	Nipple		Breasts		Thoracic cavity		Pectoral muscle		Sternum		Clavicle	
	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ
<i>pix2pixV1</i>	0.767	0.088	0.195	0.052	0.374	0.133	0.521	0.122	0.782	0.072	0.710	0.061
<i>pix2pixV1_{OHE}</i>	0.792	0.094	0.192	0.053	0.389	0.133	0.533	0.128	0.790	0.071	0.702	0.077
<i>pix2pixV2</i>	0.756	0.110	0.203	0.054	0.389	0.124	0.521	0.128	0.789	0.083	0.685	0.069
<i>pix2pixV2_{OHE}</i>	0.783	0.098	0.200	0.051	0.384	0.126	0.531	0.132	0.794	0.075	0.710	0.081
<i>pix2pixV3</i>	0.734	0.126	0.203	0.052	0.394	0.126	0.531	0.122	0.793	0.071	0.708	0.058
<i>pix2pix_{HD}V1</i>	0.783	0.089	0.206	0.054	0.385	0.129	0.515	0.132	0.784	0.075	0.705	0.065
<i>pix2pix_{HD}V2</i>	0.764	0.105	0.211	0.054	0.390	0.120	0.478	0.132	0.763	0.075	0.671	0.069
CycleGAN	0.787	0.083	0.203	0.053	0.382	0.127	0.510	0.117	0.787	0.084	0.692	0.079

Table 5.6: Dice scores of each labeled structure

	Nipple		Breasts		Thoracic cavity		Pectoral muscle		Sternum		Clavicle	
	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ
<i>pix2pixV1</i>	0.000	0.000	0.940	0.077	0.768	0.179	0.498	0.205	0.667	0.265	0.019	0.043
<i>pix2pixV1_{OHE}</i>	0.025	0.084	0.787	0.223	0.751	0.195	0.582	0.179	0.674	0.271	0.102	0.177
<i>pix2pixV2</i>	0.000	0.000	0.856	0.177	0.765	0.189	0.629	0.158	0.634	0.208	0.007	0.028
<i>pix2pixV2_{OHE}</i>	0.048	0.117	0.782	0.215	0.737	0.187	0.503	0.216	0.645	0.278	0.102	0.176
<i>pix2pixV3</i>	0.028	0.094	0.886	0.168	0.760	0.205	0.544	0.191	0.578	0.314	0.062	0.135
<i>pix2pix_{HD}V1</i>	0.104	0.188	0.955	0.059	0.791	0.177	0.657	0.187	0.732	0.284	0.168	0.215
<i>pix2pix_{HD}V2</i>	0.000	0.000	0.955	0.057	0.786	0.173	0.655	0.180	0.739	0.235	0.004	0.011
CycleGAN	0.080	0.153	0.921	0.125	0.776	0.190	0.673	0.154	0.685	0.291	0.037	0.128

Since no previous works concerning thoracic MRI synthesis were found in the literature, we will not be able to compare our results to any reference.

5.4.3.1 *Pix2pix* versions study

We will start by analyzing if any of the proposed versions of the *pix2pix* was able to outperform the baseline. The versions implemented varied in the generator network architecture and in the encoding of the annotation (single channel versus one channel per label). The main findings for both studies were the following:

- **Generator architecture:** according to the FID values presented in Table 5.4, the most successful architecture is *pix2pixV3* (FID=215.636), followed by *pix2pixV2* (FID=218.662) and finally *pix2pixV1* (FID=242.473), which is not in agreement with the SSIM and NMI results, which indicate that *pix2pixV1* is the best (SSIM=0.836, NMI=0.359), followed by *pix2pixV3* (SSIM=0.787, NMI=0.335) and *pix2pixV2* (SSIM=0.785, NMI=0.314). Considering all metrics equally reliable, our results are not enough to conclude whether the skip connections of the U-Net based generator used in *pix2pixV1* are more or less efficient for the synthesis than the identity connections of the residual learning blocks used in the generators of *pix2pixV2* and *pix2pixV3*. Nonetheless, *pix2pixV3* always outperforms *pix2pixV2*, proving that the coarse-to-fine generator architecture proposed in [WLZ⁺18] is more successful than a simpler version. The leftmost column of Figure 5.15 shows the results produced by

pix2pixV1 (on top), *pix2pixV2* (in the middle) and *pix2pixV3* (in the bottom), using the same input annotation. Visual quality seems similar between all three images.

- **Annotation encoding:** in Table 5.4 all metrics are higher for the single channel input versions, *pix2pixV1* and *pix2pixV2*, excepting the NMI which is superior for *pix2pixV2_{OHE}* than for *pix2pixV2* (0.330 versus 0.314). The top two rows of Figure 5.15 allow the visual comparison of the results produced by the single channel and multi-channel versions of each generator. For both generators, the results produced by the multi-channel architecture are visually more blurry, jittery and, globally, less realistic, than the ones produced by the single-channel architecture. Overall, single-channel inputs outperform multi-channel inputs and encode the information of the annotation more efficiently. As such, we did not implement multi-channel versions of *pix2pixV3*, *pix2pix_{HD}V1*, *pix2pix_{HD}V2* and CycleGAN.

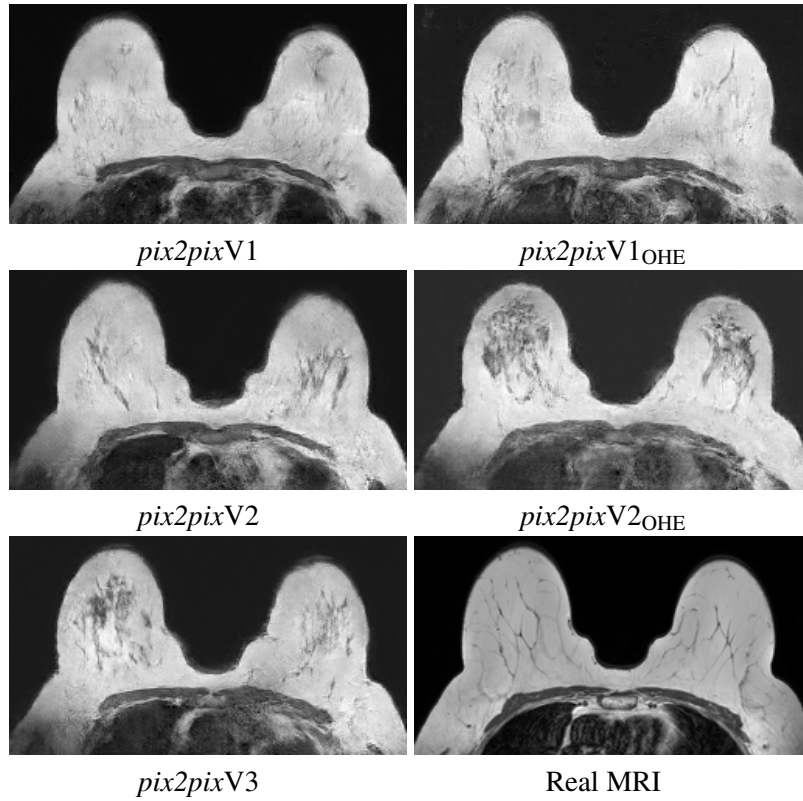


Figure 5.15: Synthetic images produced by *pix2pixV1*, *pix2pixV1_{OHE}*, *pix2pixV2*, *pix2pixV2_{OHE}* and *pix2pixV3* for the same input annotation. On the bottom right, the corresponding real MRI image.

5.4.3.2 *Pix2pix_{HD}* versions study

Pix2pix_{HD}V1 and *pix2pix_{HD}V2* allow the study of the impact of the feature matching and perceptual loss components proposed in [WLZ⁺18]. Table 5.4 shows that *pix2pix_{HD}V2* outperforms *pix2pix_{HD}V1* in every metric except the FID (244.169 versus 210.650). In fact, of all models,

$pix2pix_{HD}V2$ has the worst FID value, which is in agreement with the visual assessment of the results - the images produced with $pix2pix_{HD}V2$ are always blurry and lack definition, as can be seen in Figure 5.16. Inspecting Tables 5.5 and 5.6 reveals that, in relation to $pix2pix_{HD}V2$, $pix2pix_{HD}V1$ estimates better the content of the smaller labeled structures - nipple, pectoral muscle, clavicle and sternum - besides also segmenting more accurately the first three structures. Bearing this analysis in mind, we conclude that the perceptual and feature matching losses are essential to produce realistic images in the $pix2pix_{HD}$ framework.

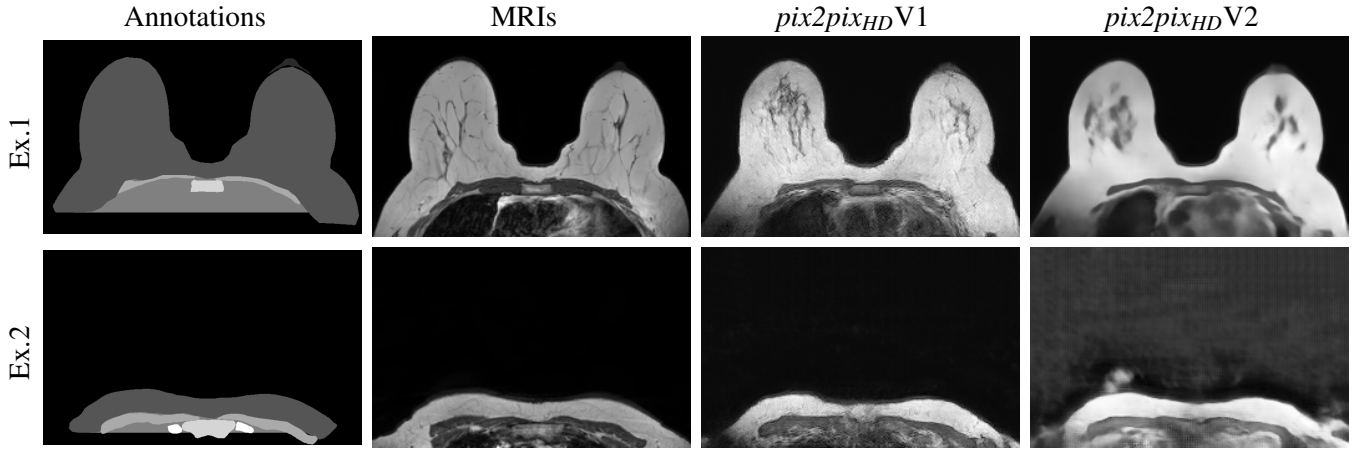


Figure 5.16: Synthetic images produced by $pix2pix_{HD}V1$ and $pix2pix_{HD}V2$ for two different input annotations.

5.4.3.3 CycleGAN study

Implementing the CycleGAN allowed us to address two challenges at once: annotation to MRI translation and the reverse task. The CycleGAN was the model that produced synthetic MRI images with the best FID score (186.320), which is noticeably lower than the FID scores for all other models. However, no other metric corroborates this result. Visually, the synthetic MRI images produced by the CycleGAN are of similar quality to the ones produced by $pix2pixV1$, $pix2pixV2$, $pix2pixV3$ and $pix2pix_{HD}V1$. As so, the information gathered is insufficient to conclude if the cycle consistency and identity loss components used during the training of this framework have a positive impact on the synthesis process. Concerning the MRI to annotation translation task, due to the training cycle implemented, the generator learned to perform regression and not classification, which resulted in poor results, as Figure 5.17 shows. The training cycle and generator network architecture should be altered in similarity with multi-class semantic segmentation tasks in order to produce more accurate annotations.

5.4.3.4 Individual structure synthesis study

The synthesis quality of each labeled structure can be studied through Tables 5.5 and 5.6. Shape-wise, the breasts are the most successfully generated structure (minimum Dice=0.787, maximum

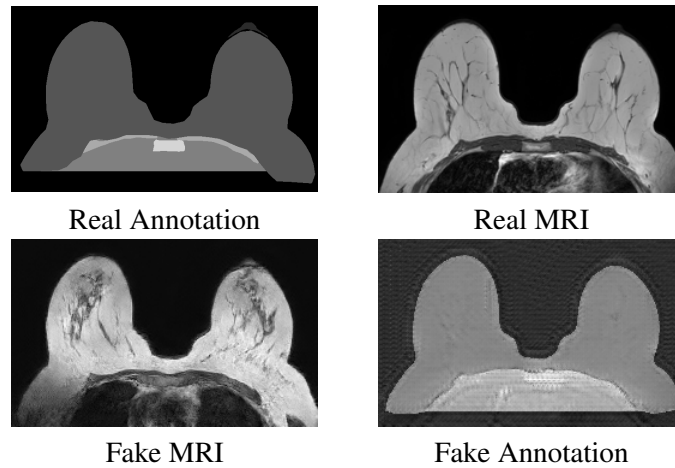


Figure 5.17: Synthetic images produced by the CycleGAN.

Dice=0.955), followed by the thoracic cavity (minimum Dice=0.737, maximum Dice=0.791). Despite having the most accurate shapes in the synthetic images, content-wise, both structures are poorly generated, presenting low NMIs. Both the labels of the breasts and thoracic cavity occupy large areas in the images, with great intensity variation, which justifies why the NMI values are poor - none of the models learned to represent accurately regions with so much intensity variation. In the case of the breasts, the fibroglandular and vascular components of this structure are generated without any guidance, as can be visually verified in Figure 5.18. Most times these components occupy areas in the fake images that do not overlap with the areas occupied in the corresponding real images. This, together with the fact that the fat tissue of the breasts has a much higher intensity than the fibroglandular and vascular tissues, contributes to the poor NMI results of the breasts, which were already expected.

Conversely, the remaining structures (pectoral muscle, sternum, nipple and clavicle), which have lower Dice scores than the breasts and thoracic cavity, have higher NMI values. This happens because their labels occupy smaller areas with low intensity variation in the images, which are simpler for all models to learn to mimic. The nipple and clavicle are very poorly segmented, having the lowest Dice scores (respectively 0.000 and 0.004 using the *pix2pix_{HD}*V2), which was already expected due to their low representation in the dataset. Despite having high NMI values, visually, both structures always have diffuse and non-defined appearances, as can be observed in the several examples presented in Figure 5.18, as well as in Figure B.1 (for the clavicle) and B.2 (for the nipple), in Appendix B. The same appendix also shows cases of poorly generated sternums in Figure B.3.

Concerning the content generation, we cannot assert one of the models as being the best since the best NIM score for each structure is not produced by the same model, and all models produce similar NIMs. Regarding the shape generation, *pix2pix_{HD}*V1, *pix2pix_{HD}*V2 and the CycleGAN are more efficient than any *pix2pix* version. The *pix2pix_{HD}*V1 is the most efficient model in synthesizing the nipple and clavicle which are, as mentioned above, the lesser represented structures in the dataset. Despite producing the most visually blurry images, *pix2pix_{HD}*V2 is the most accu-

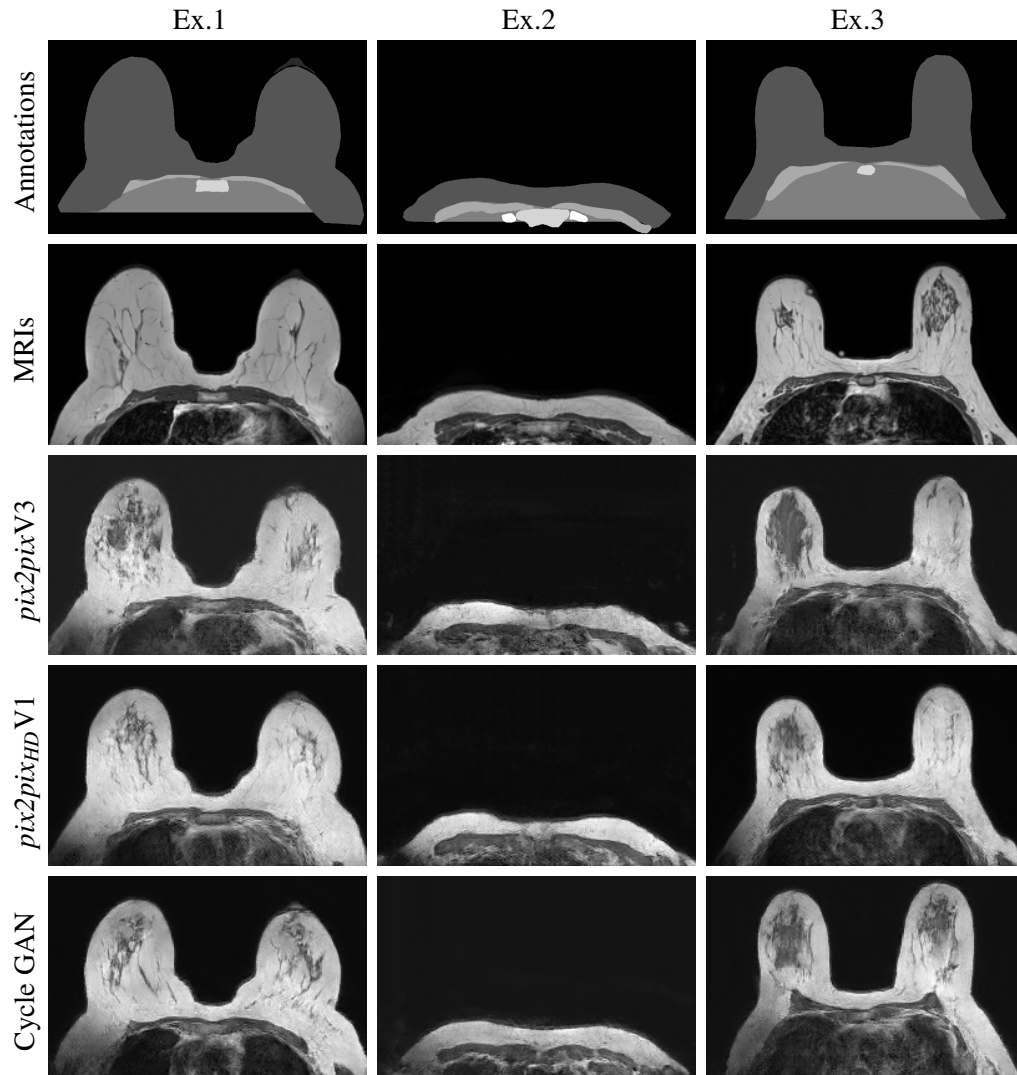


Figure 5.18: Synthetic images produced by the top 3 best performing models according to the FID: the *pix2pixV3*, the *pix2pixHD V1* and the CycleGAN.

rate model in generating the shape of the breasts and sternum. The shape of the pectoral muscle is generated with the most quality by the CycleGAN.

5.4.3.5 Baselines comparison

We resorted to the FID values to select the best variant of each baseline studied, which are the *pix2pixV3*, *pix2pixHD V1* and CycleGAN. Figure 5.18 shows the results produced by those three networks for three different input annotations. Visually, all models seem to produce images with similar quality. Examining the results produced by the second input annotation, in the middle column of Figure 5.18, we can observe that all models fail in generating the sternum and clavicle with clarity. While the CycleGAN presents the best FID value, the *pix2pixHD V1* has the best global SSIM and NMI. Neither the NMI or the Dice scores per structure consistently point to one

of the models as being more successful than the other ones. As so, we cannot draw any further conclusions.

5.4.3.6 Comparison with the Fréchet Inception Distance study

Comparing the reference FID scores of Table 5.2 with the FID scores obtained for the synthetic images displayed in Table 5.4, we can observe that most of the FID values for the fake images lie between 200 and 300, as also do the reference FID values between the MRI image and that same image tampered with the three types of blur applied - Gaussian blur (T_7), median blur (T_8) and downsampling with aliasing (T_9). This leads us to conclude that all synthetic images from all models have a blurry quality in relation to real MRI images, which is detectable by the Inception network during the FID calculation.

5.4.3.7 Hyper-parameter study

All baseline works studied applied the same training hyper-parameters, namely the number of epochs and the learning rates of the generator and discriminator, as described in Chaptre 5.3. From now on, we'll refer to this original hyper-parameter combination as HV_1 . We explored this topic further by training the *pix2pixV1*, *pix2pixV2* and *pix2pixV3* models with two other different combinations of hyper-parameters, described below:

- Hyper-parameter variation 2 (HV_2) - training during 300 epochs, maintaining the learning rate of both networks constant during the first 150 epochs, linearly decaying until 0 during the remaining epochs. The starting learning rate for both networks is 0.0002;
- Hyper-parameter variation 3 (HV_3) - training during 200 epochs, maintaining the learning rate of both networks constant during the first 100 epochs, linearly decaying until 0 during the remaining epochs. This time, the starting learning rate for the generator network is 0.002, which is superior to the starting learning rate of the discriminator which is, once again, 0.0002.

Our goal is to assess if any of the two proposed hyper-parameter combinations outperforms HV_1 . HV_2 will assess if a higher number of epochs enables better convergence, while HV_3 will indicate if the higher generator learning rate allows it to perform better in relation to the discriminator.

Figure 5.19 shows three plots which describe, for *pix2pixV1*, the generator and discriminator average losses for the training set in each training epoch, using the hyper-parameter sets HV_1 (Figure 5.19a), HV_2 (Figure 5.19b) and HV_3 (Figure 5.19c). Analyzing the plots shows that HV_1 and HV_2 reach similar minimum loss values for the generator (respectively 0.659 and 0.653), HV_2 reaching an inferior value to HV_1 . Both versions reach the minimum loss in the first epoch, after which we can observe the generator's loss increasing and then decreasing once again from around epoch 170 (for HV_1) and 270 (for HV_2). At the beginning of the training, the increase in the generator's loss is accompanied by a parallel decrease in the discriminator's loss, which

shows that, at the start, the discriminator outperforms the generator. This initial shape of the plot is coherent with the concept of competition training: for one of the networks to improve, the other must underperform. For all cases, the discriminator is improved and converges earlier in the training process than the generator.

Despite reaching a lower generator training loss, *pix2pixV1* with the weights produced by the HV_2 training produces images with a higher FID value (249.686) than the ones from HV_1 (242.473). HV_3 produces an even higher FID value (261.714) despite reaching the lowest loss value from all three cases (0.585 at epoch 4).

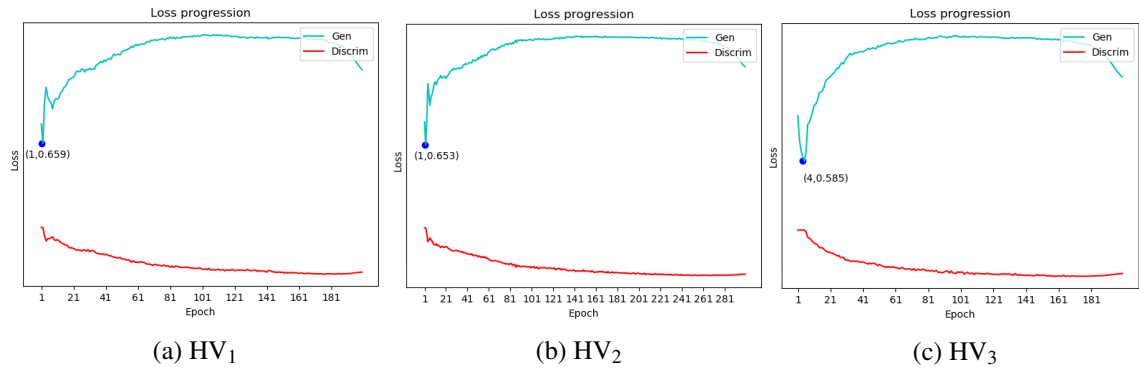


Figure 5.19: Graphics indicating the loss progression for the generator and discriminator for the *pix2pixV1* model with the the hyper-parameter sets HV_1 , HV_2 and HV_3 . The light blue and red lines refer, respectively, to the generators' and discriminators' loss, while the dark blue point refers to the minimum loss value for the generator.

Concerning *pix2pixV2*, Figure 5.20 shows the training loss plots for the parameter sets HV_1 (Figure 5.20a), HV_2 (Figure 5.20b) and HV_3 (Figure 5.20c). Once again, the minimum loss values for HV_1 and HV_2 are reached early on in the training process, respectively in epochs 3 and 5. However, HV_1 provides a lower minimum loss and FID than HV_2 , which has a minimum loss of 0.964 and FID of 247.686. In this case, for both HV_1 and HV_2 , the generator's loss starts decreasing much earlier in the training than for *pix2pixV1*, which shows that this model starts converging earlier in the training process. For HV_3 , the minimum generator training loss and FID value are worse than for HV_1 (respectively 0.881 and 273.334). However, Figure 5.20c shows that with HV_3 , the generator converges faster to a stable value than in any of the previous cases.

Lastly, for *pix2pixV3*, Figure 5.21 shows the training loss plots for the parameter sets HV_1 (Figure 5.21a), HV_2 (Figure 5.21b) and HV_3 (Figure 5.21c). Equally to the previous case, the minimum generator loss value is achieved by HV_1 , followed by HV_2 and HV_3 . Both HV_2 and HV_3 produce images with higher FID values than HV_1 (respectively 253.962 and 291.807). Once again, analyzing Figure 5.21c shows that with the hyper-parameter set HV_3 , the generator converges and stabilizes much faster than for HV_1 and HV_2 . Bearing this analysis in mind, we can conclude that the original hyper-parameter set, HV_1 , is the one that produces the best results.

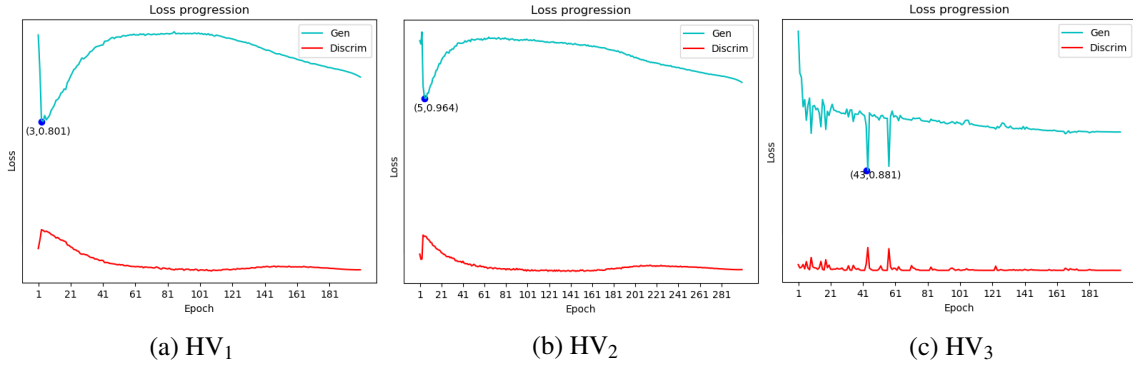


Figure 5.20: Graphics indicating the loss progression for the generator and discriminator for the *pix2pixV2* model with the hyper-parameter sets HV₁, HV₂ and HV₃. The light blue and red lines refer, respectively, to the generators' and discriminators' loss, while the dark blue point refers to the minimum loss value for the generator.

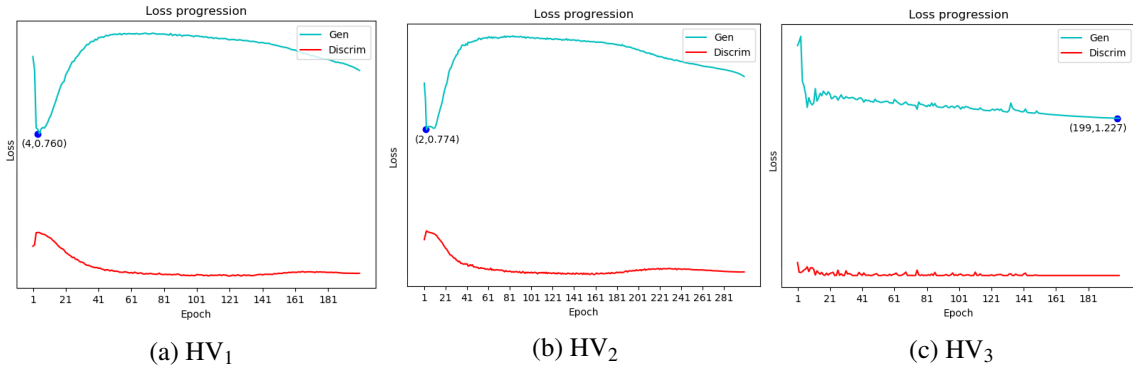


Figure 5.21: Graphics indicating the loss progression for the generator and discriminator for the *pix2pixV3* model with the hyper-parameter sets HV₁, HV₂ and HV₃. The light blue and red lines refer, respectively, to the generators' and discriminators' loss, while the dark blue point refers to the minimum loss value for the generator.

5.4.3.8 Ground truth manipulation

As a last experiment, we altered some annotations of our test set, generating new ones, and used them to synthesize corresponding MRI images. To do so, we applied our most successful model according to the FID, which was the CycleGAN. Figure 5.22 shows both the manipulated annotations and corresponding synthetic MRI images.

From the four examples displayed in Figure 5.22, it is possible to observe that all annotations, except Ex.2, produce MRIs with similar visual quality to the ones produce using real annotations from the test set. The visual quality of Ex.2 is worse, especially in the breasts and thoracic cavity regions, which may have occurred due to these labels occupying larger areas than in the other annotations.

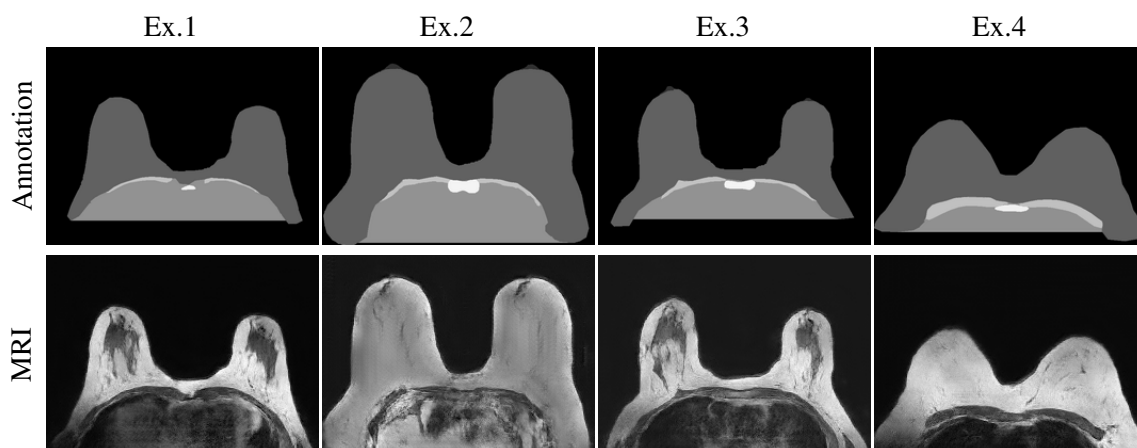


Figure 5.22: Manipulated annotations and corresponding synthetic MRI images produced by the CycleGAN.

5.5 Summary

At the beginning of this chapter, we reviewed the methods applied to both synthesize and evaluate artificial thoracic MRI images. All synthesis methods draw from the *pix2pix* and its variants, introduced in Chapter 4.1.3. We tried to directly improve the *pix2pix* framework by manipulating the generator’s architecture, and proposed testing the role of the feature matching and perceptual losses introduced in the *pix2pix_{HD}* work by training this framework with the loss function used in *pix2pix*. Additionally, we also implemented the CycleGAN with the goal of assessing if the cycle consistency loss has a positive impact on annotation to MRI translation and if this strategy is useful to segment thoracic MRI.

Concerning the segmentation task, as expected, objects with bigger areas and amply represented within the dataset (breasts and thoracic cavity) were more successfully segmented than smaller and less common ones (clavicle and nipple).

As expected, quantifying the realism of synthetic images was a challenging task. The analysis of the results conducted revealed that *pix2pix_{HD}*V1 and the CycleGAN are the top performing models, since they produce the best global SSIM, NMI and FID values (if we exclude the *pix2pix_{HD}*V2 due to its poor visual results), and since the NIM and Dice per-structure scores fail in providing further clarification. The visual analysis of our results was often inconclusive since we lack the expertise to analyze more accurately MRI images.

Our results also showed that the metrics chosen often provide contradicting information. The NMI offers a pixel-wise analysis of intensities and fails in conveying information regarding the relation between neighbouring pixels, providing the most inconsistent results from all metrics used. Regarding the FID, perhaps fine-tuning the Inception network for a classification task using our data could allow the network to extract more relevant features from our data and produce better scores.

Lastly, we tested several hyper-parameter combinations, namely a higher amount of training epochs and higher generator learning rate, for the training of *pix2pixV1*, *pix2pixV2*, and *pix2pixV3*. We proved that the standard hyper-parameter set used in all works studied outperformed any of the variations tested, despite the generator loss not converging.

Chapter 6

Conclusion

This chapter summarizes the research performed throughout this dissertation, its main conclusions, and contributions. Lastly, we list some ideas to improve our work in the future.

6.1 Overview and contributions

As stated at the beginning of this dissertation, the goal of this research was to implement an efficient method to synthesize artificial thoracic MRI exams from manually drawn annotations. With this, we intend to equip medical image analysis researchers with more comprehensive and balanced datasets, allowing them to develop more robust methods for the automatic detection of breast lesions. Due to the nature of our dataset, we started by reviewing the anatomy of the thorax, as well as the principles and motivations behind the use of thoracic MRI for the study of breast-related pathologies. We then reviewed the state of the art for image synthesis, focusing on generative models and applications for medical image synthesis. This provided a strong background to achieve the goal we had proposed ourselves and showed that no equal work was ever published. An overview of several contrasting strategies for the evaluation of generative models, which is currently a topic of vast debate, was also conducted.

Our main contributions were methods for the synthesis and evaluation of artificial thoracic MRI exams. Concerning the synthesis, the methods implemented were the original image-to-image translation framework, commonly denominated *pix2pix*, as well as two subsequent frameworks, the *pix2pix_{HD}* and the CycleGAN. For the *pix2pix*, we conceived several variations with different generator architecture, while for the *pix2pix_{HD}* we developed a variant with a different loss function than the one originally proposed. These variations were developed to try to improve both frameworks. Overall, the original *pix2pix_{HD}* and the CycleGAN were the most successful models, both through quantitative and visual assessment. The most successful *pix2pix* version used the *pix2pix_{HD}* generator architecture.

Regarding the evaluation, the strategies proposed were based both in classical concepts - the normalized mutual information - and in more recent ones - the semantic segmentation of the synthetic images with the GCN network. C

6.2 Future work

This dissertation has shown that, although we were able to generate thoracic MRI images from annotations, the synthetic exams have visually detectable artifacts and lack the definition of real exams. Below, we list some possible improvements that we propose to implement in the future.

- Extend the **hyper-parameters study** to the *pix2pix_{HD}* and CycleGAN, and test further hyper-parameter combinations;
- Explore other **evaluation metrics** to overcome the contradictions detected by the ones used for this research. For instance, we believe that performing Visual Turing Tests with radiologists would provide valuable insights regarding the perceived differences between real and fake MRIs, which we cannot detect due to lack of expertise. Moreover, other segmentation architectures can be tested, namely the PSP network [ZSQ⁺17], used for evaluation in [WLZ⁺18];
- Conduct a more extensive study regarding the Normalized Mutual Information metric;
- Acquire **annotations for more structures**, namely within structures that occupy large areas of the images, such as the breasts and thoracic cavity - for instance, for the breasts, annotations of the fibroglandular and vascular structures would avoid this structures to be generated at random;
- Synthesize images with **greater resolution** than the one chosen, matching the size of real MRI exams;
- Implement **other frameworks** for the synthesis task, such as the MedGAN, proposed in [AJF⁺18], or strategies based on Adversarial Autoencoders;
- Implement further **strategies to improve GAN training**, such as experience replay, historical averaging or mini-batch discrimination, as introduced in Chapter 4.1.1;
- For *pix2pix_{HD}*, implement instance level feature embedding;
- For the **CycleGAN**, change the training cycle and architecture for the MRI to annotation generator, making it more similar with the ones used for standard semantic segmentation networks.

Appendix A

Tables

Table A.1: Pixel-based texture synthesis methods

Paper	Year	Key aspects of the method
[EL99]	1999	Non-parametric, inside-out synthesis using a searching algorithm with fixed window size.
[WL00]	2000	Non-parametric, multiresolution synthesis in raster scan order using a searching algorithm with fixed window size.
[Wei02]	2001	Non-parametric, multiresolution synthesis in raster scan order using a searching algorithm with fixed window size. Extended applications.
[KEBK05]	2005	Synthesis by minimization of the texture energy, optimization performed using the Expectation Maximization algorithm.

Table A.2: Patch-based texture synthesis methods

Paper	Year	Key aspects of the method
[PFH00]	2000	Patch synthesis over surface, guided by a user defined vector field. New patches overlap older ones.
[EF01]	2001	Patch stitching by finding the minimum cost path on the overlapped region.
[LLX ⁺ 01]	2001	Patch sampling formulated as search for approximate nearest neighbors. Uses an original data structure - the quadtree pyramid - and principal components analysis.
[SCA02]	2002	Multi-scale algorithm that maps a texture onto an arbitrary surface. It progressively covers it with texture patches of various sizes and shapes, avoiding generating and storing a new texture beforehand.
[OWEB03]	2003	Extension of the algorithm proposed in [EF01], with improved path cutting and multiresolution synthesis.
[KSE03]	2003	Patch synthesis without a fixed patch size: the best patch is found through a graph cuts strategy.
[WY04]	2004	Performs feature matching and alignment by measuring structural similarity and using curvilinear features and their deformation. Extracts a feature map from the sample texture and creates both a new feature map and texture map.

Table A.3: Surface texture synthesis methods

Paper	Year	Key aspects of the method
[Tur91]	1991	Procedural method that generates a texture over a mesh based on reaction-diffusion principles. The mesh is generated by evenly distributing points over a model and determining adjacent points by constructing Voronoi regions. Textures are rendered directly from the mesh using a weighted sum of mesh values to compute surface color at a given position.
[PFH00]	2000	Patch synthesis over surface, guided by a user defined vector field. New patches overlap older ones.
[Tur01]	2001	A hierarchy of meshes is created over the surface, resembling the scheme of image pyramids. Each point in the mesh hierarchy (coarse to fine order) is given a color, by examining the color of neighboring points and finding the best match to a similar pixel neighborhood in a texture sample. The compilation of colored points produces the final texture. The synthesis is guided by a user-defined vector field.
[WL01]	2001	Generalization of the method proposed in [Tur01] to arbitrary manifold surfaces.
[YHBZ01]	2001	Extensions of the methods proposed in [Tur01] and [Ash01].
[SCA02]	2002	Multi-scale algorithm that maps a texture onto an arbitrary surface. It progressively covers it with texture patches of various sizes and shapes, avoiding generating and storing a new texture beforehand.
[ZZV03]	2003	Decoration of arbitrary surfaces with progressively-variant textures taking as input an homogeneous texture sample. The user has control over the scale, orientation, color and shape of the texture primitives at different levels. The synthesis conjugates a texton mask, an orientation field and the target texture.
[ZG03] &	2003/	Accelerated texture synthesis using jump maps, allowing real time synthesis. A jump map stores, for each pixel of a sample
[Zel04]	2004	texture, a small set of similar pixels, weighted according to similarity.
[LHZ ⁺ 04]	2004	A texture sample is modeled using a bidirectional texture function and rendered upon an arbitrary surface through a combination of geometry maps and point appearance functions.
[MW05]	2005	Firstly, a control mesh is defined and subdivided with a resolution compatible for the synthesis procedure. Synthesis occurs patch by patch for every face of the polygonal model. Patch boundaries are blended together to avoid visible seams.
[LH06]	2006	A texture sample is converted from the usual space of pixel colors to a space of appearance vectors, where the synthesis occurs. A high-dimensional appearance vector, incorporating non-local information (feature and radiance-transfer data) is computed for every pixel. Principal Component Analysis is applied to reduce dimensionality. Synthesis is performed similarly to [YHBZ01] and the obtained texture is mapped back to the traditional space of pixel colors.
[HZW ⁺ 06]	2006	Combination of the texture optimization strategy proposed in [KEBK05] and texture synthesis over surfaces methods proposed in [Tur01] and [WL01].
[ZHW ⁺ 06]	2006	Mapping several texture images onto the surface of a 3D model. The user establishes some correspondences between the textures and the 3D model, which are used to partition both the model (creating an initial mesh) and the images. A hierarchical optimization procedure is run to compute most of the mapping. The parts of the mesh which are not textured in the previous process are filled using inpainting.

Table A.4: Solid texture synthesis methods

Paper	Year	Key aspects of the method
[HB95]	1995	Pyramid decomposition of a 3D block of noise and a 2D sample image and histogram matching between the correspondent levels of the two pyramids.
[GD95]	1995	A basis function corresponding to the sum of cosines that match the texture sample's frequency properties is built and combined with a perturbation function to create a new 3D texture.
[GD96]	1996	A 3D discrete model is built applying 2D spectral analysis of two to three 2D models representing different perpendicular views of a natural 3D texture. The new 3D texture is synthesized using 3D spectral analysis of the 3D discrete model. An antialiasing method is applied to improve results.
[DGF98]	1998	Combination of histogram matching and spectral analysis.
[ONOI04]	2004	Interactive system for creating and browsing 3D models with internal textures. The user assigns internal textures to a surface mesh by cutting the mesh and providing guiding information to specify the correspondence between the cross-section and a reference 2D image. The guiding information and the geometry are stored and used for the synthesis of cross-sectional textures.
[JDR04]	2004	Synthesis of particle aggregates (concrete, asphalt, terrazzo, sponges and foams), which distribution and shape is acquired from a 2D input image. The synthesis draws from traditional stereology techniques.
[QY05]	2005	A textures sample is modeled using independent Basic Gray Level Aura Matrices, which are then used to iteratively modify a noise image and synthesize a new texture.
[POB ⁺ 07]	2007	Applies histogram matching, combined with a non-parametric texture optimization approach.
[QY07]	2007	Creates the Aura matrix ([QY05]) representation of a set of input samples, and generates a new solid texture by sampling them with constraints upon the view directions. An arbitrary object can be textured by the solid texture using a shader.
[KFCO ⁺ 07]	2007	Combines non-parametric texture synthesis with global histogram matching. Optimization by minimization of a global texture energy function.
[DLTD08]	2008	Several 3D candidates are pre-computed using three 2D example images by interleaving three 2D neighborhoods. The algorithm performs multi-resolution synthesis in a sparse volume pyramid and uses the candidates for neighborhood matching, only synthesizing the subset of the voxels necessary.
[TOI08]	2008	Synthesis of solids with spatially varying, oriented textures. 3D texture patches are placed inside a 3D tetrahedral mesh. The user must provide the mesh (geometry model) and an exemplar solid texture data – a cubic array of RGB colors. The cubic exemplar will belong to one of 7 categories, with different anisotropy and variation behavior. The output will be a lapped solid textured model, a 3D variant of the method proposed in [PFH00]

Table A.5: Deep-Learning methods for texture synthesis

Paper	Year	Key aspects of the method
[GEB15]	2016	CNN and feature map matching using Gram matrices.
[LGX16]	2016	CNN with Fourier spectrum and the statistical features constraints.
[ULVL16]	2016	Compact feed-forward CNN.
[UBGB16]	2016	Shallow CNN with random filters.
[UVL17]	2017	Compact feed-forward CNN.
[Sne17]	2017	CNN and feature map matching using Gram matrices and pyramid decomposition.

Appendix B

Visual results

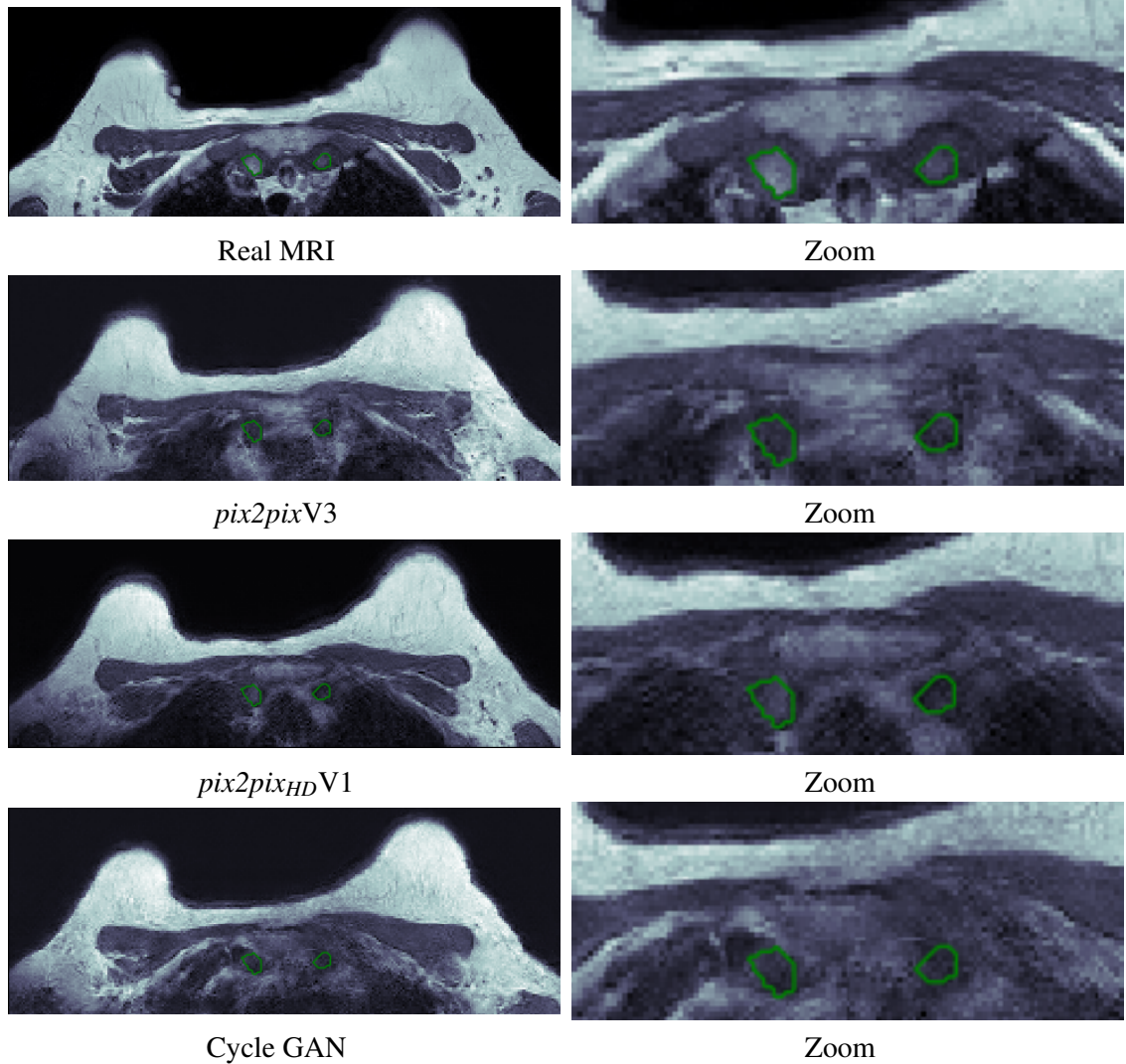


Figure B.1: Less successful cases in the synthesis of the clavicle.

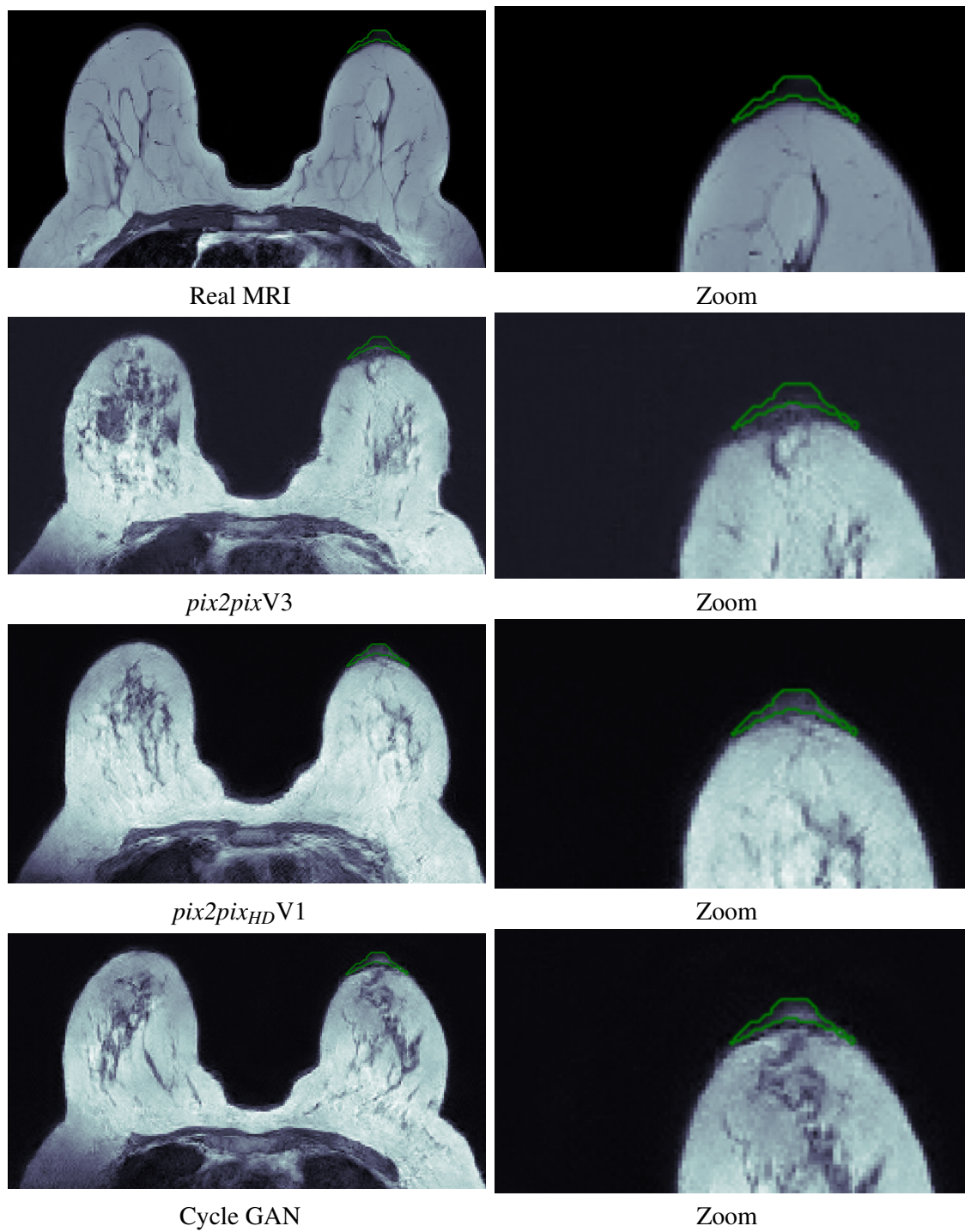


Figure B.2: Less successful cases in the synthesis of the nipple.

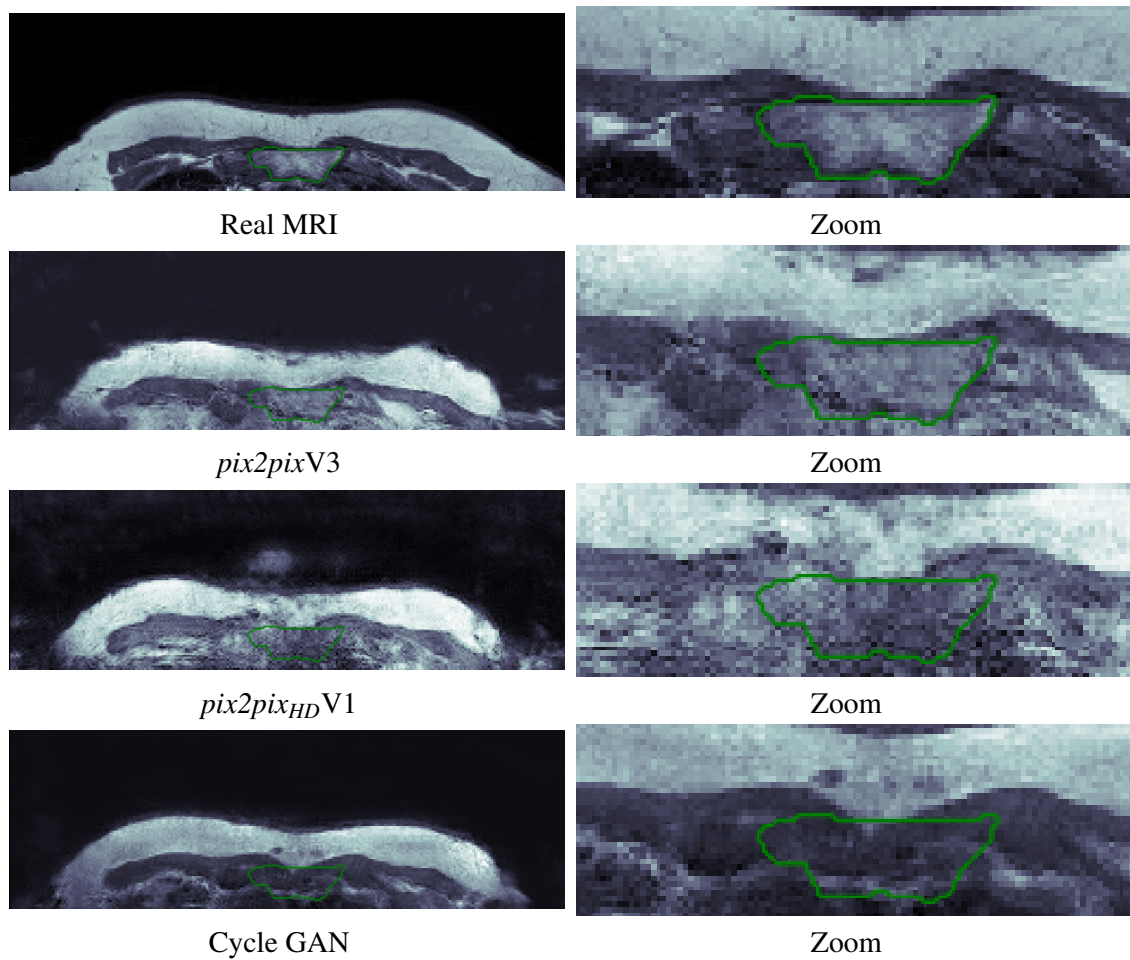


Figure B.3: Less successful cases in the synthesis of the sternum.

References

- [AATVG18] Asha Anoosheh, Eirikur Agustsson, Radu Timofte, and Luc Van Gool. ComboGAN: Unrestrained Scalability for Image Domain Translation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 783–790. IEEE, 6 2018. doi:[10.1109/CVPRW.2018.00122](https://doi.org/10.1109/CVPRW.2018.00122).
- [ACB17] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein Generative Adversarial Networks. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, pages 214–223. PMLR, 2017. URL: <http://proceedings.mlr.press/v70/arjovsky17a.html>.
- [AJF⁺18] Karim Armanious, Chenming Jiang, Marc Fischer, Thomas Küstner, Konstantin Nikolaou, Sergios Gatidis, and Bin Yang. MedGAN: Medical Image Translation using GANs. *arXiv preprint arXiv:1806.06397*, 6 2018. URL: <http://arxiv.org/abs/1806.06397>.
- [Ame17] American Cancer Society. Breast Cancer Facts & Figures 2017-2018, 2017.
- [ARS⁺18] Amjad Almahairi, Sai Rajeswar, Alessandro Sordoni, Philip Bachman, and Aaron Courville. Augmented CycleGAN: Learning Many-to-Many Mappings from Unpaired Data. *arXiv preprint arXiv:1802.10151*, 2 2018. URL: <http://arxiv.org/abs/1802.10151>.
- [Ash01] Michael Ashikhmin. Synthesizing natural textures. In *Proceedings of the 2001 symposium on Interactive 3D graphics - SI3D '01*, 2001. doi:[10.1145/364338.364405](https://doi.org/10.1145/364338.364405).
- [BAE99] F Bochud, C Abbey, and M Eckstein. Statistical texture synthesis of mammographic images with clustered lumpy backgrounds. *Optics express*, 1999. doi:[10.1364/OE.4.000033](https://doi.org/10.1364/OE.4.000033).
- [Bor19] Ali Borji. Pros and cons of GAN evaluation measures. *Computer Vision and Image Understanding*, 179:41–65, 2 2019. doi:[10.1016/j.cviu.2018.10.009](https://doi.org/10.1016/j.cviu.2018.10.009).
- [BS18] Shane Barratt and Rishi Sharma. A Note on the Inception Score. *arXiv preprint arXiv:1801.01973*, 2018. URL: <http://arxiv.org/abs/1801.01973>.
- [BSM17] David Berthelot, Thomas Schumm, and Luke Metz. BEGAN: Boundary Equilibrium Generative Adversarial Networks. *arXiv preprint arXiv:1703.10717*, 3 2017. URL: <http://arxiv.org/abs/1703.10717>.
- [BXP08] A.C. Brooks, Xiaonan Zhao, and T.N. Pappas. Structural Similarity Quality Metrics in a Coding Context: Exploring the Space of Realistic Distortions. *IEEE*

- Transactions on Image Processing*, 17(8):1261–1273, 8 2008. doi:[10.1109/TIP.2008.926161](https://doi.org/10.1109/TIP.2008.926161).
- [Cas09] Cyril Castella. Breast Texture Synthesis and Estimation of the Role of the Anatomy and Tumor Shape in the Radiological Detection Process: from Digital Mammography to Breast Tomosynthesis. Technical report, EPFL, 2009.
- [CCK⁺18] Yunje Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. StarGAN: Unified Generative Adversarial Networks for Multi-domain Image-to-Image Translation. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8789–8797. IEEE, 6 2018. doi:[10.1109/CVPR.2018.00916](https://doi.org/10.1109/CVPR.2018.00916).
- [CEMA11] Mark Clemens, Karen Evans, Samir Mardini, and Phillip Arnold. Introduction to Chest Wall Reconstruction: Anatomy and Physiology of the Chest and Indications for Chest Wall Reconstruction. *Seminars in Plastic Surgery*, 25(01):005–015, 2 2011. doi:[10.1055/s-0031-1275166](https://doi.org/10.1055/s-0031-1275166).
- [CGB06] Weijie Chen, Maryellen L. Giger, and Ulrich Bick. A fuzzy c-means (FCM)-based approach for computerized segmentation of breast lesions in dynamic contrast-enhanced MR images. *Academic Radiology*, 13(1):63–72, 2006. doi:[10.1016/j.acra.2005.08.035](https://doi.org/10.1016/j.acra.2005.08.035).
- [CGM⁺18] Pedro Costa, Adrian Galdran, Maria Ines Meyer, Meindert Niemeijer, Michael Abramoff, Ana Maria Mendonca, and Aurelio Campilho. End-to-End Adversarial Retinal Image Synthesis. *IEEE Transactions on Medical Imaging*, 37(3):781–791, 3 2018. doi:[10.1109/TMI.2017.2759102](https://doi.org/10.1109/TMI.2017.2759102).
- [CHBB18] Maria J. M. Chuquicusma, Sarfaraz Hussein, Jeremy Burt, and Ulas Bagci. How to fool radiologists with generative adversarial networks? A visual turing test for lung cancer diagnosis. In *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)*, pages 240–244. IEEE, 2018. doi:[10.1109/ISBI.2018.8363564](https://doi.org/10.1109/ISBI.2018.8363564).
- [CJGT18] Agisilaos Chartsias, Thomas Joyce, Mario Valerio Giuffrida, and Sotirios A. Tsaftaris. Multimodal MR Synthesis via Modality-Invariant Latent Representation. *IEEE Transactions on Medical Imaging*, 37(3):803–814, 3 2018. doi:[10.1109/TMI.2017.2764326](https://doi.org/10.1109/TMI.2017.2764326).
- [CMP06] Steven Conolly, Albert Macovski, and John Pauly. Chapter 12 - Magnetic Resonance Imaging. In *The Biomedical Engineering Handbook - Medical Devices and Systems*, chapter 12. CRC press, 2006.
- [COR⁺16] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The Cityscapes Dataset for Semantic Urban Scene Understanding. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3213–3223. IEEE, 6 2016. doi:[10.1109/CVPR.2016.350](https://doi.org/10.1109/CVPR.2016.350).
- [CR68] B Y F W Campbell and J G Robson. Applications of Fourier analysis to the visibility of gratings. *Journal of Physiology*, 1968. doi:[10.1113/jphysiol.1968.sp008574](https://doi.org/10.1113/jphysiol.1968.sp008574).

- [CS18] Daniel Chernoff and Paul Stark. Principles of magnetic resonance imaging. *UpToDate*, 2018.
- [CWD⁺18] Antonia Creswell, Tom White, Vincent Dumoulin, Kai Arulkumaran, Biswa Sengupta, and Anil A. Bharath. Generative Adversarial Networks: An Overview. *IEEE Signal Processing Magazine*, 35(1):53–65, 1 2018. doi:[10.1109/MSP.2017.2765202](https://doi.org/10.1109/MSP.2017.2765202).
- [DGF98] J.M. Dischler, D Ghazanfarpour, and R Freydier. Anisotropic Solid Texture Synthesis Using Orthogonal 2D Views. *Computer Graphics Forum*, 17(3), 1998. doi:[10.1111/1467-8659.00256](https://doi.org/10.1111/1467-8659.00256).
- [DLTD08] Yue Dong, Sylvain Lefebvre, Xin Tong, and George Drettakis. Lazy solid texture synthesis. *Computer Graphics Forum*, 27(4), 2008. doi:[10.1111/j.1467-8659.2008.01254.x](https://doi.org/10.1111/j.1467-8659.2008.01254.x).
- [DYK⁺19] Salman UH. Dar, Mahmut Yurt, Levent Karacan, Aykut Erdem, Erkut Erdem, and Tolga Cukur. Image Synthesis in Multi-Contrast MRI with Conditional Generative Adversarial Networks. *IEEE Transactions on Medical Imaging*, 2 2019. doi:[10.1109/TMI.2019.2901750](https://doi.org/10.1109/TMI.2019.2901750).
- [EF01] Alexei A. Efros and William T. Freeman. Image quilting for texture synthesis and transfer. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques - SIGGRAPH '01*, 2001. doi:[10.1145/383259.383296](https://doi.org/10.1145/383259.383296).
- [EL99] A.A. Efros and T.K. Leung. Texture synthesis by non-parametric sampling. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, pages 1033–1038. IEEE, 1999. URL: <http://ieeexplore.ieee.org/document/790383/>, doi:[10.1109/ICCV.1999.790383](https://doi.org/10.1109/ICCV.1999.790383).
- [FADK⁺18] Maayan Frid-Adar, Idit Diamant, Eyal Klang, Michal Amitai, Jacob Goldberger, and Hayit Greenspan. GAN-based synthetic medical image augmentation for increased CNN performance in liver lesion classification. *Neurocomputing*, 321:321–331, 12 2018. doi:[10.1016/j.neucom.2018.09.013](https://doi.org/10.1016/j.neucom.2018.09.013).
- [FAKA⁺18] Maayan Frid-Adar, Eyal Klang, Michal Amitai, Jacob Goldberger, and Hayit Greenspan. Synthetic data augmentation using GAN for improved liver lesion classification. In *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)*, pages 289–293. IEEE, 4 2018. doi:[10.1109/ISBI.2018.8363576](https://doi.org/10.1109/ISBI.2018.8363576).
- [FGL16] Chelsea Finn, Ian Goodfellow, and Sergey Levine. Unsupervised Learning for Physical Interaction through Video Prediction. *Advances in neural information processing systems*, pages 64–72, 5 2016. URL: <http://papers.nips.cc/paper/6161-unsupervised-learning-for-physical-interaction-through-video-prediction.pdf>.
- [GD95] D. Ghazanfarpour and J. M. Dischler. Spectral analysis for automatic 3-D texture generation. *Computers and Graphics*, 19, 1995. doi:[10.1016/0097-8493\(95\)00011-Z](https://doi.org/10.1016/0097-8493(95)00011-Z).

- [GD96] Djamchid Ghazanfarpour and Jean-Michel Dischler. Generation of 3D Texture Using Multiple 2D Models Analysis. *Computer Graphics Forum*, 5:311, 1996. doi:[10.1111/1467-8659.1530311](https://doi.org/10.1111/1467-8659.1530311).
- [GEB15] Leon Gatys, Alexander S Ecker, and Matthias Bethge. Texture synthesis using convolutional neural networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems* 28, pages 262–270. Curran Associates, Inc., 2015. URL: <http://papers.nips.cc/paper/5633-texture-synthesis-using-convolutional-neural-networks.pdf>.
- [GGHY15] Donald Geman, Stuart Geman, Neil Hallonquist, and Laurent Younes. Visual Turing test for computer vision systems. *Proceedings of the National Academy of Sciences*, 112(12):201422953, 3 2015. doi:[10.1073/pnas.1422953112](https://doi.org/10.1073/pnas.1422953112).
- [GL15] Preeti Gupta and Eugene C Lin. Magnetic Resonance Mammography. *Medscape*, 2015.
- [GL16] Allen Gabriel and James Neal Long. Breast Anatomy. *Medscape*, 2016.
- [Gm13] Albert Gubern-m. Segmentation of the Pectoral Muscle in Breast MRI Using Atlas-Based Approaches. *MICCAI*, 8151(October), 2013. URL: <http://link.springer.com/10.1007/978-3-642-40760-4>, doi:[10.1016/j.ajo.2018.02.015](https://doi.org/10.1016/j.ajo.2018.02.015).
- [GMKM⁺15] Albert Gubern-Mérida, Michiel Kallenberg, Ritse M. Mann, Robert Martí, and Nico Karssemeijer. Breast segmentation and density estimation in breast MRI: A fully automatic framework. *IEEE Journal of Biomedical and Health Informatics*, 19(1):349–357, 2015. doi:[10.1109/JBHI.2014.2311163](https://doi.org/10.1109/JBHI.2014.2311163).
- [Goo16] Ian Goodfellow. NIPS 2016 Tutorial: Generative Adversarial Networks. *arXiv preprint arXiv:1701.00160*, 12 2016.
- [GPAM⁺14] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Networks. *Advances in neural information processing systems*, pages 2672–2680, 6 2014. URL: <http://arxiv.org/abs/1406.2661>.
- [GSB17] Swaminathan Gurumurthy, Ravi Kiran Sarvadevabhatla, and R. Venkatesh Babu. DeLiGAN: Generative Adversarial Networks for Diverse and Limited Data. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4941–4949. IEEE, 7 2017. doi:[10.1109/CVPR.2017.525](https://doi.org/10.1109/CVPR.2017.525).
- [HB95] David J. Heeger and James R. Bergen. Pyramid-based texture analysis/synthesis. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques - SIGGRAPH '95*, 1995. doi:[10.1109/ICIP.1995.537718](https://doi.org/10.1109/ICIP.1995.537718).
- [HHR⁺18] Changhee Han, Hideaki Hayashi, Leonardo Rundo, Ryosuke Araki, Wataru Shimoda, Shinichi Muramatsu, Yujiro Furukawa, Giancarlo Mauri, and Hideki Nakayama. GAN-based synthetic brain MR image generation. In *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)*, pages 734–738. IEEE, 4 2018. doi:[10.1109/ISBI.2018.8363678](https://doi.org/10.1109/ISBI.2018.8363678).

- [HJO⁺01] Aaron Hertzmann, Charles E. Jacobs, Nuria Oliver, Brian Curless, and David H. Salesin. Image analogies. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques - SIGGRAPH '01*, pages 327–340, New York, New York, USA, 2001. ACM Press. doi:10.1145/383259.383295.
- [HLP⁺17] Xun Huang, Yixuan Li, Omid Poursaeed, John Hopcroft, and Serge Belongie. Stacked generative adversarial networks. In *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, pages 5077–5086, 2017. doi:10.1109/CVPR.2017.202.
- [HRU⁺17] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium. *Advances in Neural Information Processing Systems*, pages 6626–6637, 2017. URL: <http://papers.nips.cc/paper/7240-gans-trained-by-a-two-time-scale-update-rule-converge-to-a-local-nash-equilibrium.pdf>.
- [HZ10] Alain Horé and Djemel Ziou. Image quality metrics: PSNR vs. SSIM. In *Proceedings - International Conference on Pattern Recognition*, pages 2366–2369, 2010. doi:10.1109/ICPR.2010.579.
- [HZRS16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778. IEEE, 6 2016. doi:10.1109/CVPR.2016.90.
- [HZW⁺06] Jianwei Han, Kun Zhou, Li Yi Wei, Minmin Gong, Hujun Bao, Xinming Zhang, and Baining Guo. Fast example-based surface texture synthesis via discrete optimization. *Visual Computer*, 22(9-11):918–925, 2006. doi:10.1007/s00371-006-0078-3.
- [IZZE17] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-Image Translation with Conditional Adversarial Networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134. IEEE, 7 2017. doi:10.1109/CVPR.2017.632.
- [JDR04] Robert Jagnow, Julie Dorsey, and Holly E Rushmeier. Stereological Techniques for Solid Textures Real-time rendering of partially translucent objects View project Development of software for the Cultural Heritage Experts. CHER-Ob: A Tool for Shared Analysis in Cultural Heritage: View project. *ACM Transactions on Graphics*, 2004. doi:10.1145/1015706.1015724.
- [JGSF73] Bela Julesz, EN Gilbert, LA Shepp, and HL Frisch. Inability of humans to discriminate between visual textures that agree in second-order statistics—revisited. *Perception*, 2(4):391–405, 1973.
- [Jor18] Jeremy Jordan. Variational autoencoders, 2018. Date Accessed: January 17, 2019. URL: <https://www.jeremyjordan.me/variational-autoencoders/>.
- [KA19] Bodo Kaiser and Shadi Albarqouni. MRI to CT Translation with GANs. *arXiv preprint arXiv:1901.05259*, 1 2019. URL: <http://arxiv.org/abs/1901.05259>.

- [KALL17] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive Growing of GANs for Improved Quality, Stability, and Variation. *arXiv preprint arXiv:1710.10196*, 10 2017. URL: <http://arxiv.org/abs/1710.10196>.
- [KEBK05] Vivek Kwatra, Irfan Essa, Aaron Bobick, and Nipun Kwatra. Texture optimization for example-based synthesis. *ACM Transactions on Graphics*, 2005. doi:10.1145/1073204.1073263.
- [KFCO⁺07] Johannes Kopf, Chi-Wing Fu, Daniel Cohen-Or, Oliver Deussen, Dani Lischinski, and Tien-Tsin Wong. Solid texture synthesis from 2D exemplars. *ACM Transactions on Graphics*, 2007. doi:10.1145/1276377.1276380.
- [KRMW14] Diederik P. Kingma, Danilo J. Rezende, Shakir Mohamed, and Max Welling. Semi-Supervised Learning with Deep Generative Models. *Advances in Neural Information Processing Systems*, pages 3581–3589, 6 2014. URL: <http://arxiv.org/abs/1406.5298>.
- [KSE03] Vivek Kwatra, A Schödl, and Irfan Essa. Graphcut textures: image and video synthesis using graph cuts. *ACM Transactions on Graphics (ToG)*, 2003. doi:10.1145/882262.882264.
- [KW13] Diederik P Kingma and Max Welling. Auto-Encoding Variational Bayes. *arXiv preprint arXiv:1312.6114*, 12 2013. URL: <http://arxiv.org/abs/1312.6114>.
- [LGX16] Gang Liu, Yann Gousseau, and Gui Song Xia. Texture synthesis through convolutional neural networks and spectrum constraints. In *Proceedings - International Conference on Pattern Recognition*, 2016. doi:10.1109/ICPR.2016.7900133.
- [LH06] Sylvain Lefebvre and Hugues Hoppe. Appearance-space texture synthesis. In *ACM Transactions on Graphics (TOG)*, volume 25, pages 541–548. ACM, 2006. doi:10.1145/1141911.1141921.
- [LHZ⁺04] Xinguo Liu, Yaohua Hu, Jingdan Zhang, Xin Tong, Baining Guo, and Heung Yeung Shum. Synthesis and Rendering of Bidirectional Texture Functions on Arbitrary Surfaces. *IEEE Transactions on Visualization and Computer Graphics*, 10(3):278–289, 2004. doi:10.1109/TVCG.2004.1272727.
- [LKC15] William Lotter, Gabriel Kreiman, and David Cox. Unsupervised Learning of Visual Structure using Predictive Generative Networks. *arXiv preprint arXiv:1511.06380*, 11 2015. URL: <http://arxiv.org/abs/1511.06380>.
- [LLX⁺01] Lin Liang, Ce Liu, Ying-Qing Xu, Baining Guo, and Heung-Yeung Shum. Real-time texture synthesis by patch-based sampling. *ACM Transactions on Graphics*, 20(3):127–150, 2001. doi:10.1145/501786.501787.
- [LTH⁺17] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, and Wenzhe Shi. Photo-realistic single image super-resolution using a generative adversarial network. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017-Janua:105–114, 2017. doi:10.1109/CVPR.2017.19.

- [LW16] Chuan Li and Michael Wand. Precomputed Real-Time Texture Synthesis with Markovian Generative Adversarial Networks. In *European Conference on Computer Vision*, pages 702–716, 2016. URL: http://link.springer.com/10.1007/978-3-319-46487-9_43, doi:10.1007/978-3-319-46487-9_43.
- [Mis] Prakhar Mishra. Understanding Autoencoders — Unsupervised Learning Technique. Date Accessed: November 30, 2018. URL: <https://becominghuman.ai/understanding-autoencoders-unsupervised-learning-technique-82fb3fbaec2>.
- [MLX⁺17] Xudong Mao, Qing Li, Haoran Xie, Raymond Y.K. Lau, Zhen Wang, and Stephen Paul Smolley. Least Squares Generative Adversarial Networks. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2813–2821. IEEE, 10 2017. doi:10.1109/ICCV.2017.304.
- [MO14] Mehdi Mirza and Simon Osindero. Conditional Generative Adversarial Nets. *arXiv preprint arXiv:1411.1784*, 11 2014. URL: <http://arxiv.org/abs/1411.1784>.
- [MRB⁺17] O. Mattausch, E. Ren, M. Bajka, K. Vanhoey, and O. Goksel. Comparison of texture synthesis methods for content generation in ultrasound simulation for training. In *Progress in Biomedical Optics and Imaging - Proceedings of SPIE*, 2017. doi:10.1117/12.2250604.
- [MSJ⁺15] Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, and Brendan Frey. Adversarial Autoencoders. *arXiv preprint arXiv:1511.05644*, 11 2015. URL: <http://arxiv.org/abs/1511.05644>.
- [MTCA17] Emmanuel Maggiori, Yuliya Tarabalka, Guillaume Charpiat, and Pierre Alliez. Can semantic labeling methods generalize to any city? the inria aerial image labeling benchmark. In *2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pages 3226–3229. IEEE, 7 2017. doi:10.1109/IGARSS.2017.8127684.
- [MVS03] Frederik Maes, Dirk Vandermeulen, and Paul Suetens. Medical image registration using mutual information. *Proceedings of the IEEE*, 91(10):1699–1721, 2003. doi:10.1109/JPROC.2003.817864.
- [MW05] Weiyin Ma and Zhuangzhi Wu. Texture Synthesis on Subdivision Surfaces. *Computer-Aided Design and Applications*, 2:449, 2005. doi:10.1080/16864360.2005.10738394.
- [NTL⁺17] Dong Nie, Roger Trullo, Jun Lian, Caroline Petitjean, Su Ruan, Qian Wang, and Dinggang Shen. Medical Image Synthesis with Context-Aware Generative Adversarial Networks. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 417–425, 2017. doi:10.1007/978-3-319-66179-7_48.

- [NTL⁺18] Dong Nie, Roger Trullo, Jun Lian, Li Wang, Caroline Petitjean, Su Ruan, Qian Wang, and Dinggang Shen. Medical Image Synthesis with Deep Convolutional Adversarial Networks. *IEEE Transactions on Biomedical Engineering*, 65(12):2720–2730, 12 2018. URL: <https://ieeexplore.ieee.org/document/8310638/>, doi:10.1109/TBME.2018.2814538.
- [ONOI04] Shigeru Owada, Frank Nielsen, Makoto Okabe, and Takeo Igarashi. Volumetric illustration: designing 3D models with internal textures. *ACM Transactions on Graphics*, 2004. doi:10.1145/1015706.1015723.
- [OSDU18] Sahin Olut, Yusuf Huseyin Sahin, Ugur Demir, and Gozde Unal. Generative Adversarial Training for MRA Image Synthesis Using Multi-contrast MRI. In *International Workshop on Predictive Intelligence In Medicine*, pages 147–154, 4 2018. doi:10.1007/978-3-030-00320-3_18.
- [OWEB03] J. T. O’Brien, D. S. Wickramanayake, E. A. Edirisinghe, and H. E. Bez. Image quilting for texture synthesis: A revisit & a variation. *ICICS-PCM 2003 - Proceedings of the 2003 Joint Conference of the 4th International Conference on Information, Communications and Signal Processing and 4th Pacific-Rim Conference on Multimedia*, 2(January):763–767, 2003. doi:10.1109/ICICS.2003.1292559.
- [PFH00] Emil Praun, Adam Finkelstein, and Hugues Hoppe. Lapped textures. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques - SIGGRAPH '00*, 2000. doi:10.1145/344779.344987.
- [PG16] Navid Pourtaheri and Thomas R Gest. Chest Wall Anatomy. *Medscape*, 2016. URL: <https://emedicine.medscape.com/article/2151800-overview#a2>.
- [POB⁺07] Nico Pietroni, Miguel A. Otaduy, Bernd Bickel, Fabio Ganovelli, and Markus Gross. Texturing internal surfaces from a few cross sections. *Computer Graphics Forum*, 2007. doi:10.1111/j.1467-8659.2007.01087.x.
- [PV16] David Pfau and Oriol Vinyals. Connecting Generative Adversarial Networks and Actor-Critic Methods. *arXiv preprint arXiv:1610.01945*, 10 2016. URL: <http://arxiv.org/abs/1610.01945>.
- [PZY⁺17] Chao Peng, Xiangyu Zhang, Gang Yu, Guiming Luo, and Jian Sun. Large kernel matters - Improve semantic segmentation by global convolutional network. In *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, volume 2017-Janua, pages 1743–1751, 2017. doi:10.1109/CVPR.2017.189.
- [QY05] Xuejie Qin and Yee Hong Yang. Basic gray level aura matrices: Theory and its application to texture synthesis. *Proceedings of the IEEE International Conference on Computer Vision*, I(4):128–135, 2005. doi:10.1109/ICCV.2005.43.
- [QY07] Xuejie Qin and Yee Hong Yang. Aura 3D textures. *IEEE Transactions on Visualization and Computer Graphics*, 13(2), 2007. doi:10.1109/TVCG.2007.31.
- [RFB15] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *Lecture Notes in Computer Science*

- (including subseries *Lecture Notes in Artificial Intelligence* and *Lecture Notes in Bioinformatics*), volume 9351, pages 234–241, 2015. doi:10.1007/978-3-319-24574-4_28.
- [RG18] Gustavo de la Roza and Harris Gellman. Histology of Bone. *Medscape*, 2018. URL: <https://emedicine.medscape.com/article/1254517-overview>.
- [RMC15] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. *arXiv preprint arXiv:1511.06434*, 11 2015. URL: <http://arxiv.org/abs/1511.06434>.
- [SBB⁺07] Debbie Saslow, Carla Boetes, Wylie Burke, Steven Harms, Martin O. Leach, Constance D. Lehman, Elizabeth Morris, Etta Pisano, Mitchell Schnall, Stephen Sener, Robert A. Smith, Ellen Warner, Martin Yaffe, Kimberly S. Andrews, and Christy A. Russell. American Cancer Society Guidelines for Breast Screening With MRI as an Adjunct to Mammography. *Obstetrical & Gynecological Survey*, 62(7):458–460, jul 2007. URL: <https://insights.ovid.com/crossref?an=00006254-200707000-00021>, doi:10.1097/01.ogx.0000269073.50925.38.
- [SCA02] Cyril Soler, Marie-Paule Cani, and Alexis Angelidis. Hierarchical pattern mapping. *ACM Transactions on Graphics*, 2002. doi:10.1145/566570.566635.
- [Sci] Scientist Cindy. Anatomical Positions. Date Accessed: June 17, 2019. URL: <https://www.scientistcindy.com/anatomical-positions.html>.
- [SGZ⁺16] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved Techniques for Training GANs. *Advances in neural information processing systems*, pages 2234–2242, 6 2016.
- [Sha] Irhum Shafkat. Intuitively Understanding Variational Autoencoders. Date Accessed: November 30, 2018. URL: <https://towardsdatascience.com/intuitively-understanding-variational-autoencoders-1bfe67eb5daf>.
- [SLD17] Evan Shelhamer, Jonathan Long, and Trevor Darrell. Fully Convolutional Networks for Semantic Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(4):640–651, 4 2017. doi:10.1109/TPAMI.2016.2572683.
- [SLJ⁺15] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2015. doi:10.1109/CVPR.2015.7298594.
- [SLY15] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning Structured Output Representation using Deep Conditional Generative Models. *Advances in Neural Information Processing Systems*, pages 3483–3491, 2015.

- [Sne17] Xavier Snelgrove. High-resolution multi-scale neural texture synthesis. *SIGGRAPH Asia 2017 Technical Briefs on - SA '17*, pages 1–4, 2017. doi:[10.1145/3145749.3149449](https://doi.org/10.1145/3145749.3149449).
- [SPS⁺17] Chao Shang, Aaron Palmer, Jiangwen Sun, Ko-Shin Chen, Jin Lu, and Jinbo Bi. VIGAN: Missing view imputation with generative adversarial networks. In *2017 IEEE International Conference on Big Data (Big Data)*, pages 766–775. IEEE, 12 2017. doi:[10.1109/BigData.2017.8257992](https://doi.org/10.1109/BigData.2017.8257992).
- [SS17] Roberta J Seidman and Erik D Schraga. *Skeletal Muscle-Structure and Histology*. Medscape, 2017.
- [SV18] Irina Sanchez and Veronica Vilaplana. Brain MRI super-resolution using 3D generative adversarial networks. *arXiv preprint arXiv:1812.11440*, 12 2018. URL: <http://arxiv.org/abs/1812.11440>.
- [SZ14] Karen Simonyan and Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv preprint arXiv:1409.1556*, 9 2014. URL: <http://arxiv.org/abs/1409.1556>.
- [TJ98] Mihran Tuceryan and Anil K Jain. The Handbook of Pattern Recognition and Computer Vision (2nd Edition), Texture Analysis. *Pattern Recognition*, 1998. doi:[10.1097/RCT.0b013e3181ec05e4](https://doi.org/10.1097/RCT.0b013e3181ec05e4).
- [TOI08] Kenshi Takayama, Makoto Okabe, and Takeo Igarashi. Lapped Solid Textures: Filling a Model with Anisotropic Textures. *ACM Transactions on Graphics (ToG)*, 27(3):1, 2008. doi:[10.1145/1360612.1360652](https://doi.org/10.1145/1360612.1360652).
- [Tur91] Greg Turk. Generating textures on arbitrary surfaces using reaction-diffusion. *ACM SIGGRAPH Computer Graphics*, 25(4):289–298, 1991. doi:[10.1145/127719.122749](https://doi.org/10.1145/127719.122749).
- [Tur01] Greg Turk. Texture synthesis on surfaces. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques - SIGGRAPH '01*, 2001. doi:[10.1145/383259.383297](https://doi.org/10.1145/383259.383297).
- [UBGB16] Ivan Ustyuzhaninov, Wieland Brendel, Leon A Gatys, and Matthias Bethge. Texture synthesis using shallow convolutional networks with random filters. *arXiv preprint arXiv:1606.00021*, 2016.
- [ULVL16] Dmitry Ulyanov, Vadim Lebedev, Andrea Vedaldi, and Victor S Lempitsky. Texture networks: Feed-forward synthesis of textures and stylized images. In *ICML*, volume 1, page 4, 2016. URL: <http://proceedings.mlr.press/v48/ulyanov16.pdf>.
- [UVL16] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance Normalization: The Missing Ingredient for Fast Stylization. *arXiv preprint arXiv:1607.08022*, 7 2016. URL: <http://arxiv.org/abs/1607.08022>.
- [UVL17] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Improved texture networks: Maximizing quality and diversity in feed-forward stylization and texture synthesis. In *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017. doi:[10.1109/CVPR.2017.437](https://doi.org/10.1109/CVPR.2017.437).

- [WBSS04] Zhou Wang, Alan Conrad Bovik, Hamid Rahim Sheikh, and Eero P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. doi:10.1109/TIP.2003.819861.
- [WDS⁺17] Jelmer M. Wolterink, Anna M. Dinkla, Mark H. F. Savenije, Peter R. Seevinck, Cornelis A. T. van den Berg, and Ivana Išgum. Deep MR to CT Synthesis Using Unpaired Data. In *International Workshop on Simulation and Synthesis in Medical Imaging*, pages 14–23, 2017. doi:10.1007/978-3-319-68127-6_2.
- [Wei02] Li-Yi Wei. *Texture Synthesis by Fixed Neighborhood Searching*. PhD thesis, Stanford University, Stanford, CA, USA, 2002. AAI3038169.
- [WL00] Li-Yi Wei and Marc Levoy. Fast texture synthesis using tree-structured vector quantization. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques - SIGGRAPH '00*, 2000. doi:10.1145/344779.345009.
- [WL01] Li-Yi Wei and Marc Levoy. Texture synthesis over arbitrary manifold surfaces. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques - SIGGRAPH '01*, 2001. doi:10.1145/383259.383298.
- [WLKT09] Li-Yi Wei, Sylvain Lefebvre, Vivek Kwatra, and Greg Turk. State of the art in example-based texture synthesis. In *Eurographics '09 State of the Art Reports (STARs)*, 2009.
- [WLZ⁺18] Ting Chun Wang, Ming Yu Liu, Jun Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-Resolution Image Synthesis and Semantic Manipulation with Conditional GANs. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 8798–8807, 2018. doi:10.1109/CVPR.2018.00917.
- [Wor19] World Health Organization. Breast cancer, 2019. Date Accessed: January 12, 2019. URL: <https://www.who.int/cancer/prevention/diagnosis-screening/breast-cancer/en/>.
- [WY04] Qing Wu and Yizhou Yu. Feature matching and deformation for texture synthesis. In *ACM SIGGRAPH 2004 Papers, SIGGRAPH '04*, pages 364–367, New York, NY, USA, 2004. ACM. doi:10.1145/1186562.1015730.
- [YHBZ01] Lexing Ying, Aaron Hertzmann, Henning Biermann, and Denis Zorin. *Texture and Shape Synthesis on Surfaces*. New York, 2001.
- [YYSL15] Xinchun Yan, Jimei Yang, Kihyuk Sohn, and Honglak Lee. Attribute2Image: Conditional Image Generation from Visual Attributes. *European Conference on Computer Vision*, pages 702–716, 12 2015. URL: <http://arxiv.org/abs/1512.00570>.
- [ZBLN97] Ciyu Zhu, Richard H Byrd, Peihuang Lu, and Jorge Nocedal. L-Bfgs-B - Fortran Subroutines for Large-Scale Bound Constrained Optimization. *ACM Transactions on Mathematical Software (TOMS)*, 23:550–560, 1997. doi:10.1145/279232.279236.

- [Zel04] Steve Zelinka. Jump map-based interactive texture synthesis. *ACM Transactions on Graphics*, 2004. doi:10.1145/1027411.1027413.
- [ZG03] Steve Zelinka and Michael Garland. Interactive texture synthesis on surfaces using jump maps. In *Proceedings of the 14th Eurographics Workshop on Rendering Techniques*, pages 90–96, 2003. URL: <http://dl.acm.org/citation.cfm?id=882404.882418>, doi:10.1016/0010-0277(82)90003-8.
- [ZHW⁺06] Kun Zhou, Xin Huang, Xi Wang, Yiying Tong, Mathieu Desbrun, Baining Guo, and Heung-Yeung Shum. Mesh quilting for geometric texture synthesis. *ACM SIGGRAPH 2006 Papers on - SIGGRAPH '06*, 2006. doi:10.1145/1141911.1141942.
- [ZML16] Junbo Zhao, Michael Mathieu, and Yann LeCun. Energy-based Generative Adversarial Network. *arXiv preprint arXiv:1609.03126*, abs/1609.0, 9 2016. URL: <http://arxiv.org/abs/1609.03126>.
- [ZPIE17] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2242–2251. IEEE, 10 2017. doi:10.1109/ICCV.2017.244.
- [ZSQ⁺17] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid Scene Parsing Network. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6230–6239. IEEE, 7 2017. doi:10.1109/CVPR.2017.660.
- [ZYM18] Yufeng Zheng, Clifford Yang, and Aleksey Merkulov. Breast cancer screening using convolutional neural network and follow-up digital mammography. *Proc. SPIE 10669, Computational Imaging III*, 1066905, 1066905(September), 2018. doi:10.1117/12.2304564.
- [ZZP⁺17] Jun-Yan Zhu, Richard Zhang, Deepak Pathak, Trevor Darrell, Alexei A. Efros, Oliver Wang, and Eli Shechtman. Toward Multimodal Image-to-Image Translation. *Advances in Neural Information Processing Systems*, pages 465–476, 11 2017.
- [ZZV03] Jingdan Zhang, K Zhou, and L Velho. Synthesis of progressively-variant textures on arbitrary surfaces. *ACM Transactions on Graphics (TOG)*, 22(3):295–302, 2003. doi:10.1145/882262.882266.