SEMANTIC SEGMENTATION WITH MULTI-SOURCE DOMAIN ADAPTATION FOR RADIOLOGICAL IMAGES

HUGO NEVES DE OLIVEIRA

SEMANTIC SEGMENTATION WITH MULTI-SOURCE DOMAIN ADAPTATION FOR RADIOLOGICAL IMAGES

Tese apresentada ao Programa de Pós-Graduação em Ciência da Computação do Instituto de Ciências Exatas da Universidade Federal de Minas Gerais como requisito parcial para a obtenção do grau de Doutor em Ciência da Computação.

Orientador: Prof. Dr. Jefersson Alex dos Santos
Coorientador: Prof. Dr. Arnaldo de Albuquerque Araújo

Belo Horizonte, Brazil
21 de julho de 2020

HUGO NEVES DE OLIVEIRA

SEMANTIC SEGMENTATION WITH MULTI-SOURCE DOMAIN ADAPTATION FOR RADIOLOGICAL IMAGES

Thesis presented to the Graduate Program in Computer Science of the Universidade Federal de Minas Gerais in partial fulfillment of the requirements for the degree of Doctor in Computer Science.

ADVISOR: PROF. DR. JEFERSSON ALEX DOS SANTOS
CO-ADVISOR: PROF. DR. ARNALDO DE ALBUQUERQUE ARAÚJO

Belo Horizonte, Brazil
July 21, 2020

Ficha catalográfica elaborada pela bibliotecária Belkiz Inez Rezende Costa
CRB 6ª Região nº 1510

Oliveira, Hugo Neves de.

O48s Semantic segmentation with multi-source domain adaptation for radiological images / Hugo Neves de Oliveira — Belo Horizonte, 2020.
xxviii,120 f. il.; 29 cm.

Tese (doutorado) - Universidade Federal de Minas Gerais – Departamento de Ciência da Computação;
Orientador: Jefersson Alex dos Santos.
Coorientador: Arnaldo de Albuquerque Araújo

1. Computação – Teses. 2.Aprendizado profundo. 3. Adaptação de domínio. 4. Imagens médicas. 5. Segmentação de imagens. I. Orientador. II. Coorientador.III Título.

CDU 519.6*84(043)

UNIVERSIDADE FEDERAL DE MINAS GERAIS
INSTITUTO DE CIÊNCIAS EXATAS
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

FOLHA DE APROVAÇÃO

Semantic Segmentation with Multi-Source Domain Adaptation for Radiological Images

HUGO NEVES DE OLIVEIRA

Tese defendida e aprovada pela banca examinadora constituída pelos Senhores:

Jefersson A. Santos

Prof. Jefersson Alex dos Santos - Orientador
Departamento de Ciência da Computação - UFMG

Prof. Arnaldo de Albuquerque Araújo - Coorientador
Departamento de Ciência da Computação - UFMG

Prof. Mario Fernando Montenegro Campos
Departamento de Ciência da Computação - UFMG

Prof. Anisio Mendes Lacerda
Departamento de Ciência da Computação - UFMG

Prof. Moacir Antonelli Ponti
Instituto de Ciências Matemáticas e de Computação - USP

Al. Alexandre X. Falcão

Prof. Alexandre Xavier Falcão
Departamento de Sistemas de Informação - UNICAMP

Belo Horizonte, 21 de Julho de 2020.

I dedicate this work to the love of my life, my daily inspiration, Mabel Abreu.

Acknowledgments

I would sincerely like to thank everyone who was involved in this thesis, both directly via academic knowledge transfer and indirectly through emotional support and incentives.

At first I would like to thank my advisors: Prof. Jefersson dos Santos and Prof. Arnaldo Araújo. Thank you very much for always providing a healthy and proliferous academic environment not only to me, but also to all other students under your umbrella. I also thank Prof. Alexei Machado and MD. Cláudio Saliba de Avelar for their support in the medical side of this research.

For all days and nights of patience, support, love, care and listening, I thank the love of my life: Mabel Abreu. You are responsible for a huge percentage of my motivation all throughout these years of hard work and has been a vital part of my life ever since our first days.

An indispensable part of this journey were my beloved friends, whose undeniable cooperation and continuous support proved to be fundamental to this work. I would like to specifically cite the following names that helped me go through my PhD: Virgínia Mota, Camila Laranjeira, Alan Deivite, Rafael Baeta, Edemir Ferreira, Jéssica Sena, Alex Gabriel, Lucas Lacerda and Mariana Maia.

I would also like to thank all colleagues and friends from PATREO, NPDI and SSIG who helped me to grow academically and humanely during this process. It is also imperative to recognize all my professors and colleagues from UFPB and UFMG that were an integral part of my BSc., MSc. and PhD. courses.

At last, I would like to thank NVIDIA for the GPUs granted to our laboratory that allowed the development of this research. I also thank FAPEMIG, CAPES and CNPq for their financial support to this research project.

"I'm fascinated by the idea that genetics is digital. A gene is a long sequence of coded letters, like computer information. Modern biology is becoming very much a branch of information technology."

(Richard Dawkins)

Abstract

Distinct digitization techniques for biomedical images yield different visual patterns in samples from many radiological exams. These differences may hamper the use of data-driven Machine Learning approaches for inference over these images, such as Deep Learning. Another difficulty in this field is the lack of labeled data, even though in many cases there is an abundance of unlabeled data available. Therefore an important step in improving the generalization capabilities of these methods is to perform Unsupervised and Semi-Supervised Domain Adaptation between different datasets of biomedical images. In order to tackle this problem, in this work, we propose an Unsupervised and Semi-Supervised Domain Adaptation method for dense labeling tasks in biomedical images using Generative Adversarial Networks for Unsupervised Image-to-Image Translation. We merge these generative models with well-known supervised deep semantic segmentation architectures in order to create two semi-supervised methods capable of learning from both unlabeled and labeled data, whenever labeling is available. The first Domain-to-Domain method, similarly to most other Image Translation methods in the literature, is limited to a pair of domains: one source and one target. The second proposed methodology takes advantage of conditional dataset training to encourage Domain Generalization from several data sources from the same domain. From this conditional dataset encoding, we also devise a fully novel pipeline for rib segmentation in X-Ray images that does not require any label to be computed. We compare our method using a myriad of domains, datasets, segmentation tasks and traditional baselines in the Domain Adaptation literature, such as using pretrained models both with and without fine-tuning. We perform both quantitative and qualitative analysis of the proposed method and baselines in the multitude of distinct scenarios considered in our experimental evaluation. We empirically observe the limitations of pairwise Domain Adaptation approaches to truly generalizable radiograph segmentation, evidencing the better performance of multi-source training methods in this task. The proposed Conditional Domain Adaptation method shows consistently and significantly better results than the baselines in scarce labeled data scenarios – that is, when labeled data is limited or non-existent in the target dataset – achieving Jaccard indices greater than 0.9 in most tasks. Completely Unsupervised Domain Adaptation results were observed to be close to the Fully Supervised Domain Adaptation used in the traditional procedure of fine-tuning pretrained Deep Neural Networks.

Resumo

Técnicas de digitalização distintas para imagens médicas resultam em diferentes padrões visuais nas amostras de vários tipos de exames radiológicos. Essas diferenças podem dificultar o uso de técnicas de Aprendizado de Máquina baseadas em dados para inferência sobre essas imagens, como métodos de Aprendizado Profundo. Outra considerável dificuldade neste ramo de pesquisa é a falta de amostras rotuladas, embora haja em muitos casos uma abundância de dados não-rotulados. Portanto um importante passo para melhorar a capacidade de generalização desses métodos é a aprimoração de técnicas de Adaptação de Domínio Não-Supervisionada e Semi-Supervisionada entre diferentes bancos de dados de imagens médicas. Visando resolver esse problema, neste trabalho são propostos dois métodos de Adaptação de Domínio Não-Supervisionada e Semi-Supervisionada para tarefas de rotulação densa em imagens médicas que usa Redes Generativas Adversariais para Tradução Não-Supervisionada de Imagens. Os métodos mesclam esses modelos generativos com arquiteturas conhecidas para segmentação semântica visando criar métodos semi-supervisionados capazes de aprender tanto de dados não rotulados quanto de dados rotulados, sempre que rotulação estiver disponível. O primeiro método proposto, de forma similar à maioria dos outros trabalhos na literatura de Tradução de Imagens, é limitado a um par de domínios: um domínio fonte e um domínio alvo. O segundo método proposto nesse trabalho utiliza treinamento condicional de conjuntos de dados para encorajar Generalização de Domínio entre várias fontes de dados do mesmo tipo. A partir desse segundo método baseado em condicionamento de conjunto de dados, também se propõe uma nova metodologia para segmentação de costelas em imagens de raio-x que não necessita de nenhum tipo de rotulação. Os métodos propostos foram comparados usando vários domínios, bancos de dados, tarefas de segmentação e *baselines* tradicionais da área de Adaptação de Domínio, tal qual o uso de modelos pré-treinados com e sem *fine-tuning*. Foram feitas análises quantitativas e qualitativas do método proposto e dos *baselines* nos vários cenários considerados na avaliação experimental deste trabalho. São observadas as limitações dos métodos Adaptação de Domínio entre apenas um par de conjuntos de dados na construção de métodos generalizáveis de segmentação de radiografias, o que evidencia a melhor eficácia de métodos que se utilizam de várias fontes de dados nessa tarefa. O método baseado em treinamento condicional demonstrou superioridade consistente e significante nos cenários de rotulação escassa – ou seja, quando a rotulação é limitada ou não-existente no conjunto de dados alvo – conseguindo valores de Jaccard maiores que 0.9 na maioria das tarefas. Resultados de Adaptação de Domínio Não-Supervisionada foram observados como próximos dos casos supervisionados usados no procedimento padrão de usar modelos pré-treinados de Redes Neurais Profundas com *fine-tuning*.

List of Figures

List of Tables

Contents

Acronym List

AdaIN	Adaptive Instance Normalization
AE	AutoEncoder
AIP	Average Intensity Projection
AUC	Area Under Curve
BEGAN	Boundary Equilibrium Generative Adversarial Network
BI	Biomedical Image
CAD	Computer-Aided Detection/Diagnosis
CGAN	Conditional Generative Adversarial Network
CNN	Convolutional Neural Network
CoDAGAN	Conditional Domain Adaptation Generative Adversarial Network
CGAN	Conditional Generative Adversarial Network
CoGAN	Coupled Generative Adversarial Network
CT	Computed Tomography
CV	Computer Vision
CyCADA	Cycle-Consistent Adversarial Domain Adaptation
CXR	Chest X-Ray
D2D	Domain-to-Domain
DA	Domain Adaptation
DCAN	Dual Channel-wise Alignment Network
DCGAN	Deep Convolutional Generative Adversarial Network
DNN	Deep Neural Network
DRAGAN	Deep Regret Analytic Generative Adversarial Network
DRIT	Diverse Image-to-Image Translation via Disentangled Representations

DRR	Digitally Reconstructed Radiograph
DXR	Dental X-Ray
EBGAN	Energy-based Generative Adversarial Network
FCN	Fully Convolutional Network
FFDM	Full Field Digital Mammography
FSDA	Fully-Supervised Domain Adaptation
GAN	Generative Adversarial Network
GMM	Gaussian Mixture Model
GPU	Graphics Processing Unit
I2I	Image-to-Image
ICPC	Iterated Contextual Pixel Classification
IoU	Intersection over Union
LSGAN	Least Squares Generative Adversarial Network
MIP	Maximum Intensity Projection
MLE	Maximum Likelihood Estimation
MLP	Multi-Layer Perceptron
MMD	Maximum Mean Discrepancy
MMGAN	MiniMax Generative Adversarial Network
MRI	Magnetic Resonance Imaging
MSE	Mean Squared Error
MSGAN	Mode Seeking Generative Adversarial Network
MUNIT	Multimodal Unsupervised Image-to-Image Translation
MXR	Mammographic X-Ray
NSGAN	Non-Saturating Generative Adversarial Network

PA	Posterior-Anterior
PC	Pixel Classification
PCA	Principal Component Analysis
PReLU	Parametric Rectified Linear Unit
RBF	Radial Basis Function
ReLU	Rectified Linear Unit
RF	Random Forest
RGB	Red, Green and Blue
ROC	Receiver Operating Characteristic
RoI	Region of Interest
SGD	Stochastic Gradient Descent
SSDA	Semi-Supervised Domain Adaptation
SELU	Scaled Exponential Linear Unit
SVM	Support Vector Machine
t-SNE	t-Distributed Stochastic Neighbor Embedding
UDA	Unsupervised Domain Adaptation
UNIT	Unsupervised Image-to-Image Translation
VAE	Variational AutoEncoder
WGAN	Wasserstein Generative Adversarial Network
WGAN-GP	Wasserstein Generative Adversarial Network with Gradient Penalty
WRN	Wide ResNet

Chapter 1

Introduction

Radiology has been a useful tool for assessing health conditions since the last decades of the 19^{th} century, when X-Rays were first used for medical purposes. Since then, it has become an essential tool for detecting, diagnosing and treating medical issues. More recently, algorithms have been coupled with radiology imaging techniques and other medical information in order to provide second opinions to physicians via Computer-Aided Detection/Diagnosis (CAD) systems. In this context, segmentation is a very important task [Elnakib et al., 2011; Masood et al., 2015]. Most common segmentation tools are typically used for delineating nodules, bones or other kinds of tissues in an unsupervised way but it is also very common the employment of interactive segmentation.

In recent decades, Machine Learning algorithms were incorporated into the body of knowledge of CAD systems for biomedical image analysis, providing automatic methodologies for finding patterns in big data scenarios, improving the capabilities of human physicians for diagnosing illnesses. Until the early 2010's, both computer vision and biomedical image analysis literatures were dominated by shallow techniques for both feature extraction and inference, as can be seen in Figure 1.1. Low-level image processing techniques – such as Wavelet denoising [Unser and Aldroubi, 1996], edge detection filters, active contours, splines, etc – were used ever since physicians started to have access to digital versions of radiology exams. With the advent of more robust computers capable of running more powerful supervised inference algorithms – such as Support Vector Machines (SVMs) [Cortes and Vapnik, 1995], Random Forests (RFs) [Ho, 1995], etc – Machine Learning became more popular for biomedical tasks. Shallow feature extractors based on wavelets [Unser and Aldroubi, 1996], co-occurrence matrices [Haralick et al., 1973] and histogram of gradients [Lowe, 2004; Dalal and Triggs, 2005] were common practices since the mid

1990s and were often paired with image filtering techniques.

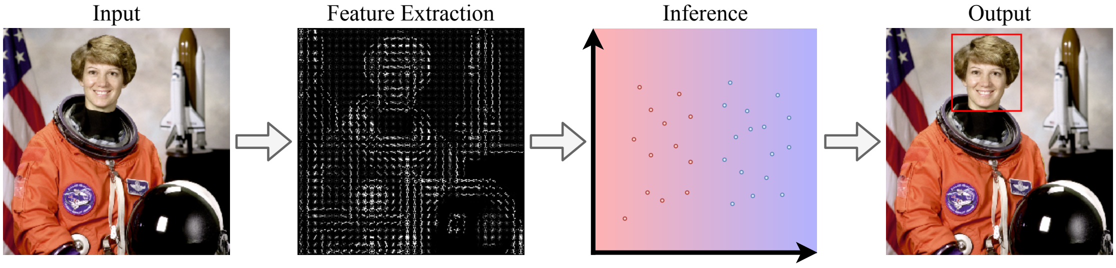


Figure 1.1. Example of a Shallow Learning pipeline for a visual recognition task.

During the last half decade, traditional Machine Learning pipelines have been losing ground to integrated Deep Neural Networks (DNNs) that can be trained from end-to-end [Litjens et al., 2017]. DNNs are powerful overcomplete models that can learn to extract features from and infer over unstructured data such as images, sounds or texts. DNNs can integrate the steps of feature extraction and statistical inference over unstructured data, such as images. While shallow methods for feature extraction and inference struggle both in computational complexity and task performance to deal with the high dimensionality and strong spatial/temporal correlation of these kinds of data, DNNs excel at it due to clever architectural designs and GPU parallelization. DNNs can be understood as ensembles of perceptrons organized in stacked layers with increasingly more semantic representation, being able to extract features and perform inference conjointly, as depicted in Figure 1.2. In general, deeper models are able to recognize information with a higher semantic level, while shallower models can only optimize for acquiring low-level semantic information, as, for instance, edge or color detection in images.

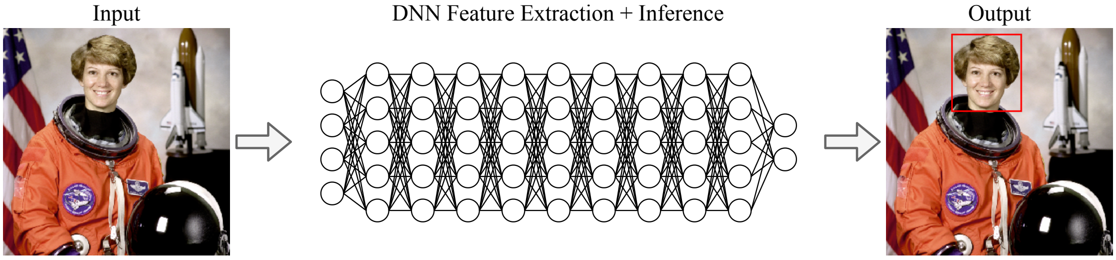


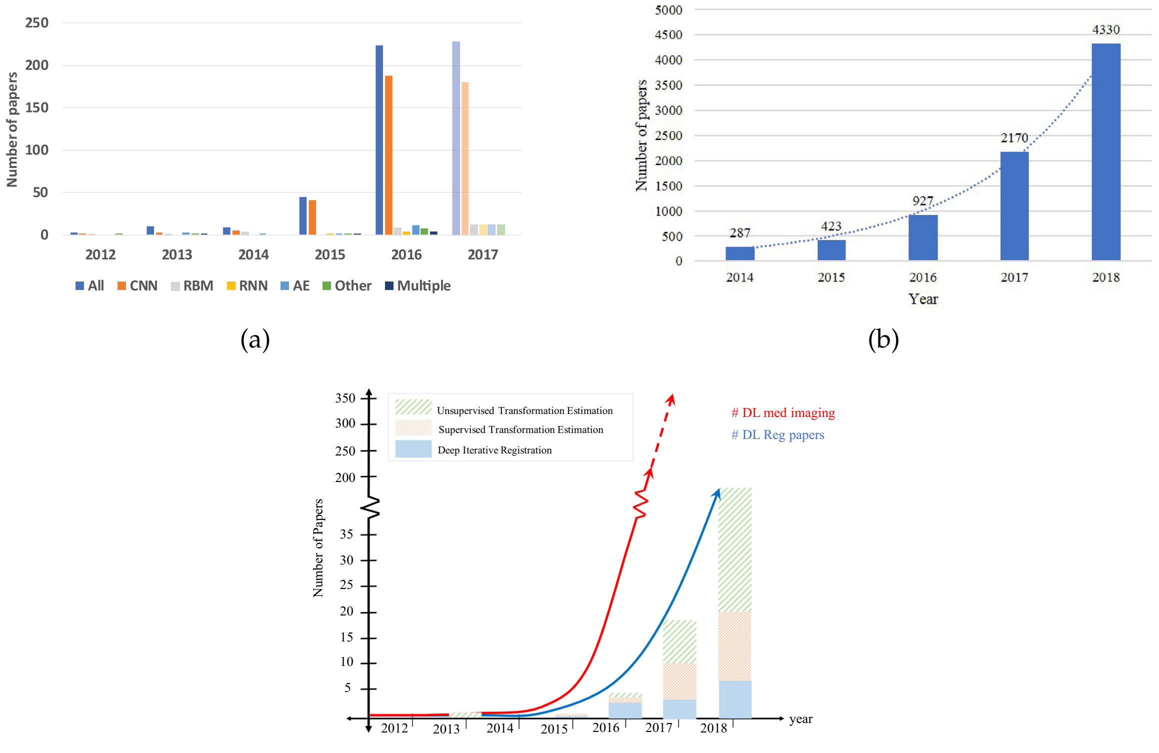
Figure 1.2. Example of a Deep Learning pipeline for a visual recognition task.

Deep Learning models for images usually are built upon some form of trainable convolutional operation, which is the basic kind of layer of Convolutional Neural Networks (CNNs) [Krizhevsky et al., 2012]. CNNs are the most popular architectures for image classification in both computer vision and biomedical imaging. Variations of CNN architectures can be found in object detection [Girshick et al., 2014; Ren et al., 2015; Redmon and Farhadi, 2018], semantic segmentation [Long et al., 2015; Ronneberger et al., 2015; Badrinarayanan et al., 2017], instance segmentation [He et al., 2017], image captioning [Karpathy and Fei-Fei, 2015; Xu et al., 2015] and video understanding [Tran et al., 2018; Wehrmann et al., 2018; Mota et al., 2020] settings. The semantic representation limits of these networks – as well as a brief timeline of the most important methods proposed for dealing with these limits – are further discussed in Section 2.

1.1 Motivations

Several surveys in Biomedical Images [Litjens et al., 2017; Zhou et al., 2019; Haskins et al., 2020] show the rapid dissemination of Deep Learning on the automated analysis of biomedical imaging over the last years. As evidenced by Figure 1.3, between 2015 and 2018 it is observed the exponential growth of works employing DNNs in biomedical image analysis.

One great limitation for Deep Learning models is the amount of data available for feeding these models, as generalizing useful patterns over unstructured data can be an exceptionally hard task. Big data is often seen as one of the culprits for the success of Deep Learning in Computer Vision problems, as these algorithms may have hundreds of millions of parameters, which requires a large number of samples to optimize. There are three basic scenarios for labeling machine learning tasks, which are depicted in Figure 1.4. The first scenario in Figure 1.4(a) covers data with no supervision at all, while the second scenario (Figure 1.4(b)) shows a dataset with both labeled and unlabeled data. At last, Figure 1.4(c) presents a fully labeled dataset. In real-world scenarios, labeled data is often limited and there are large amounts of unlabeled datasets in the medical community that can be used for unsupervised learning. To make matters worse, the generalization of DNNs is normally limited to the variability of the training data, which is a major hamper, as different digitization techniques and devices used to acquire different datasets tend to produce biomedical images with distinct visual features. Therefore the study for methods that can use both labeled and unlabeled data – that is, semi-supervised learning algorithms – is an active research area in both Computer Vision and Biomedical Image Processing. Domain Adaptation (DA) [Zhang et al., 2017] methods are often used to improve the generalization of DNNs over biomedical images in an unsupervised or



(a)

(b)

(c)

Figure 1.3. Increasing number of papers by year in (a) medical imaging (source: Litjens et al. [2017]), (b) medical image registration (source: Haskins et al. [2020]) and (c) biomedical image segmentation (source: Zhou et al. [2019]).

semi-supervised manner.

Definitions of Transfer Learning, Knowledge Transfer and Domain Adaptation in the Pattern Recognition literature [Patel et al., 2015; Shao et al., 2015; Zhang et al., 2017; Wang and Deng, 2018] are often inconsistent and hard definitions of these concepts are either nonexistent or unclear even in surveys. For the sake of simplicity, in this work we will treat these three concepts as equals and use the definition of a well known survey in Visual Domain Adaptation [Wang and Deng, 2018] to cement this part of our theoretical background. Wang and Deng [2018] define Domain Adaptation as the use of labeled data from one or more relevant source domains to execute new tasks in a target domain. In other words, Domain Adaptation reuses labeled data from one or more (source) domains and transfers this supervision to another (target) domain, lessening the annotation requirements for novel datasets, data domains or even related tasks. Even though we chose the definition of [Wang and Deng, 2018], we find the taxonomy presented by another survey ([Zhang et al., 2017]) more useful, as it captures most nuances found throughout the

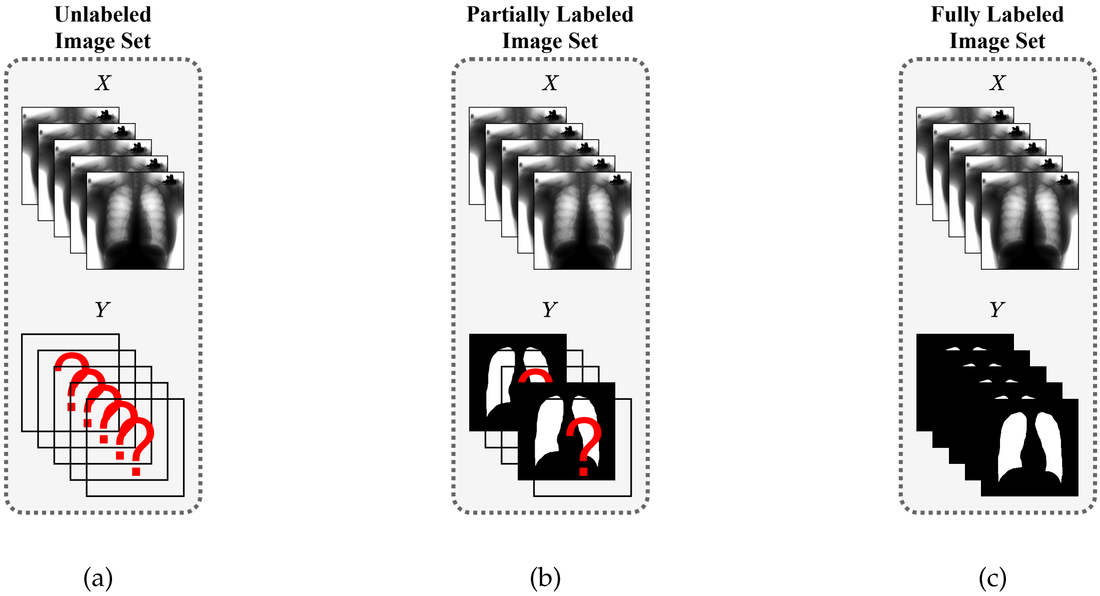


Figure 1.4. Three possible labeling scenarios for biomedical datasets. (a) Fully unlabeled data. (b) Partially labeled datasets. (c) Fully labeled data.

Transfer Learning literature. DA and some of its subareas will be further explored in Section 2.4.

The most popular method for deep DA is Transfer Learning via Fine-Tuning pretrained neural networks from larger datasets, such as ImageNet[1] [Deng et al., 2009]. However, Fine-Tuning only learns from labeled data, ignoring the larger amounts of unlabeled data available in most real-world scenarios. During the last years, several approaches have been proposed for Unsupervised Domain Adaptation (UDA) [Cao et al., 2018; Zhang et al., 2018a], Semi-Supervised Domain Adaptation (SSDA) [Yamada et al., 2014; Wu and Ji, 2016] and Fully-Supervised Domain Adaptation (FSDA) [Koniusz et al., 2017]. Depictions of generic UDA, SSDA and FSDA scenarios in dense labeling tasks can be seen in Figure 1.5. In all scenarios presented in Figure 1.5, there is a reasonably large source dataset $\mathcal{S}$ which will provide labels for the training of a supervised task in the target dataset $\mathcal{T}$.

There is a considerable gap in the literature for methods of UDA and SSDA for dense labeling tasks. The Pattern Recognition literature started to bridge this gap using modern deep approaches as Image Translation and Adversarial Training over the last years, even though UDA and SSDA in dense labeling tasks still cannot be considered resolved issues. In addition, there is also a substantial demand for methods that work over unsupervised data in areas wherein the digitization pa-

[1] www.image-net.org/

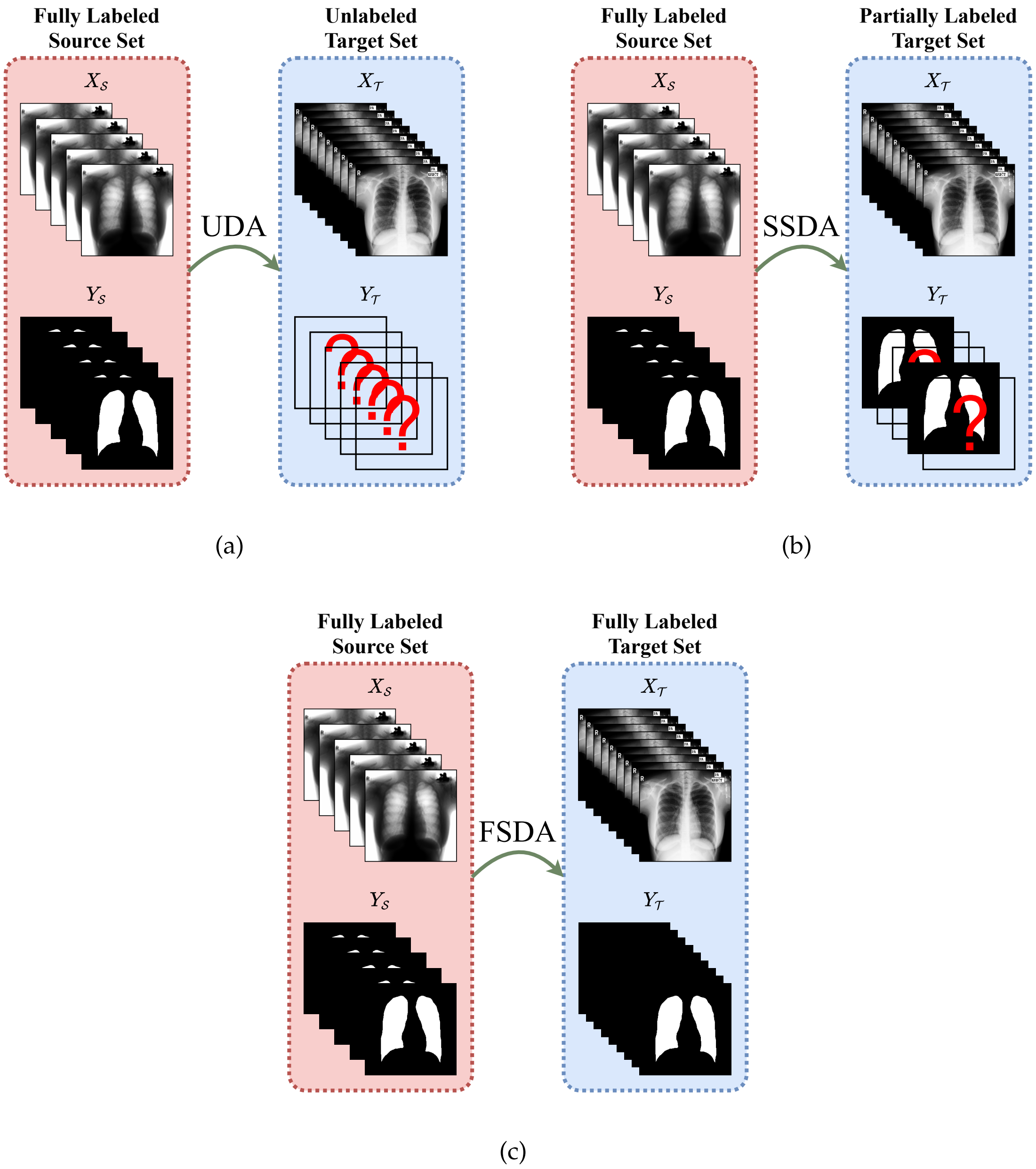


Figure 1.5. Depiction of UDA (a), SSDA (b) and FSDA (c) scenarios between a pair of source ($\mathcal{S}$) and a target ($\mathcal{T}$) sets in a biomedical image segmentation task.

rameters and equipment considerably alter the visual characteristics of the samples, as biomedical images and remote sensing.

1.2 Hypotheses

Taking into account the gap in the literature regarding UDA and SSDA for dense labeling tasks, the main hypothesis for this thesis is:

$\mathcal{H}_1$: Unsupervised Image-to-Image Translation can be used as a basis for DA in semantic segmentation tasks, providing alternatives to the traditional Transfer Learning based on fine-tuning of DNNs for dense labeling. $\mathcal{H}_1$ is further discussed and analyzed in Sections 3.1 and 5.1.

From this main hypothesis, some secondary hypotheses arose concerning the adaptability of Image Translation DNNs to multi-source and multi-target DA. Hypotheses $\mathcal{H}_2$, $\mathcal{H}_3$ and $\mathcal{H}_4$ presented above are addressed by the proposed method from Section 3.2. Experiments described in Section 4.3 assess the validity of all three hypotheses, while results presented in Section 5.2 shed light into $\mathcal{H}_2$ and $\mathcal{H}_3$. $\mathcal{H}_4$ is validated via the time consumption analysis shown in Section 5.2.5.

$\mathcal{H}_2$: In scenarios wherein labeled data is restricted or nonexistent, Conditional DA performs consistently better than its two main baselines: Pretrained DNNs and Domain-to-Domain pairwise image translation approach for UDA. If this hypothesis holds true, it is specially interesting to biomedical image analysis, as there often is a lot of unlabeled data available for a novel domain (i.e. new digitization techniques, scanners or radiological image modalities), while labeled data is often unavailable. Domain Generalization via Conditional DA could be employed whenever researchers or physicians encounter novel data using the labels from the well-known datasets in the literature.

$\mathcal{H}_3$: The addition of new datasets as labeled or unlabeled data sources can help DA methods to better perform Domain Generalization [Zhang et al., 2017]. This hypothesis is based on the premise that, if there are enough distinct datasets in the training procedure – even if most of them are fully unlabeled – the probability that at least one of them is closely related to a novel data distribution from the same domain increases.

$\mathcal{H}_4$: Conditional training is a more scalable alternative to the traditional pairwise training of standard Image-to-Image Translation techniques [Isola et al., 2017; Zhu et al., 2017a,b; Liu et al., 2017; Huang et al., 2018]. Conditional training should allow these methods to perform multi-source and multi-target DA while constraining their memory requirements in order to improve scalability to an arbitrarily large number of datasets.

Several other algorithms in the literature of Image Translation for DA [Hoffman et al., 2018; Wu et al., 2018; Murez et al., 2018] rely on synthetic data from 3D modelings of the real world [Richter et al., 2016; Ros et al., 2016]. Therefore, as Conditional DAs proved to be a highly effective method for transferring knowledge between pairs of radiological datasets, there was the hypothesis that it could be used in order to infer from a synthetic label source. This supposition is summarized in $\mathcal{H}_5$.

$\mathcal{H}_5$: It is possible to use synthetic data – as most of the image translation methods for Computer Vision applications do (see Section 2.4.3) – to acquire unsupervised knowledge from biomedical images. Synthetic data has been proved to be useful in applications as medical equipment calibration (i.e. breast [Mou et al., 2008], spine [Nord and Miller, 2001] and other phantom organs [Aoyama et al., 2002]), data augmentation for small-data scenarios [Frid-Adar et al., 2018b,a; Pelka et al., 2018] and even for acquiring useful 2D knowledge from 3D data [Candemir et al., 2016; Zhang et al., 2018b]. Hence, we hypothesize that the larger amount of information in unlabeled 3D data can be leveraged for performing Domain Generalization in segmentation tasks in 2D data.

1.3 Contributions

Aiming to answer the research questions revealed by our hypotheses, we organize this manuscript according to this work's novelties (Chapter 3), experiments (Chapter 4), results/discussion (Chapter 5) and conclusions (Chapter 6).

This project started with the proposal of an image translation approach for pairwise Domain-to-Domain (D2D) UDA and SSDA in dense labeling tasks, which was a novelty at the time. This approach was published in SIBGRAPI 2018 [Oliveira and dos Santos, 2018] and is described further in Section 3.1, while experiments and results are detailed in Sections 4.2 and 5.1. This technique was developed concurrently with several other closely related methods in the literature [Wu et al., 2018; Murez et al., 2018; Hoffman et al., 2018] and presented convergence problems, while not properly solving the problem of Domain Generalization due to its pairwise nature.

The limitations of the pairwise approach became noticeable and we started to work on Conditional DA for improving the stability, scalability and generalization of the algorithm. We also aimed at training a method that could leverage the labeled data from distinct sources and transfer this knowledge to several unlabeled ones, therefore it should be multi-source and multi-target from the start. This novel architecture for Domain Generalization became the central piece of this work and can be leveraged for the whole spectrum of Unsupervised to Fully-Supervised Cross-Dataset Transfer Learning, being able to learn from both labeled and unlabeled data. Convergence problems from D2D were solved with the proposal of the current iteration of this work, which was published in IEEE Access [Oliveira et al., 2020].

As can be seen in Figure 1.6, apart from stability, a major novelty of our Conditional DA method (henceforth known as Conditional Domain Adaptation Generative Adversarial Network – CoDAGAN) is allowing for multiple datasets to be used conjointly in the training procedure. This is contrary to most other works in the literature of Image Translation for DA, which are limited to pairwise training. CoDAGANs learn to perform inference over an isomorphic representation of multiple domains, effectively being able to take into account several sources of samples with distinct distributions drawn from the joint domain distribution in order to build more general models. CoDAGANs are described in Sections 3.2 and validated/evaluated in Sections 4.3 and 5.2.

Based on the CoDAGAN framework, we noticed that well-known techniques for extracting labels from unlabeled data sources could be generalized to other kinds of data. More specifically, we used CoDAGANs to devise a pipeline for rib segmentation presented in Section 3.3 and evaluated in Sections 4.4 and 5.3. This methodology is currently being reviewed for a special issue in Pattern Recognition Letters.

At last, collateral results from this research for deep semantic segmentation of anatomical structures were published and presented in both SIBGRAPI 2018 [Oliveira and dos Santos, 2018], CIARP 2018 [Oliveira et al., 2018].

1.4 Structure of the Text

The following chapters in this thesis are organized as follows. Chapter 2 presents the previous works that paved the way for the proposal of our proposed methods and gives an overview of both Deep Learning and DA in several distinct scenarios (i.e. supervised vs. unsupervised, sparse vs. dense labeling, discriminative vs. generative modeling, etc). Chapter 3 describes the original D2D approach from the first iteration of this work, CoDAGANs and the novel pipeline for rib segmentation based on Conditional DA. We detail components, architecture, training procedure

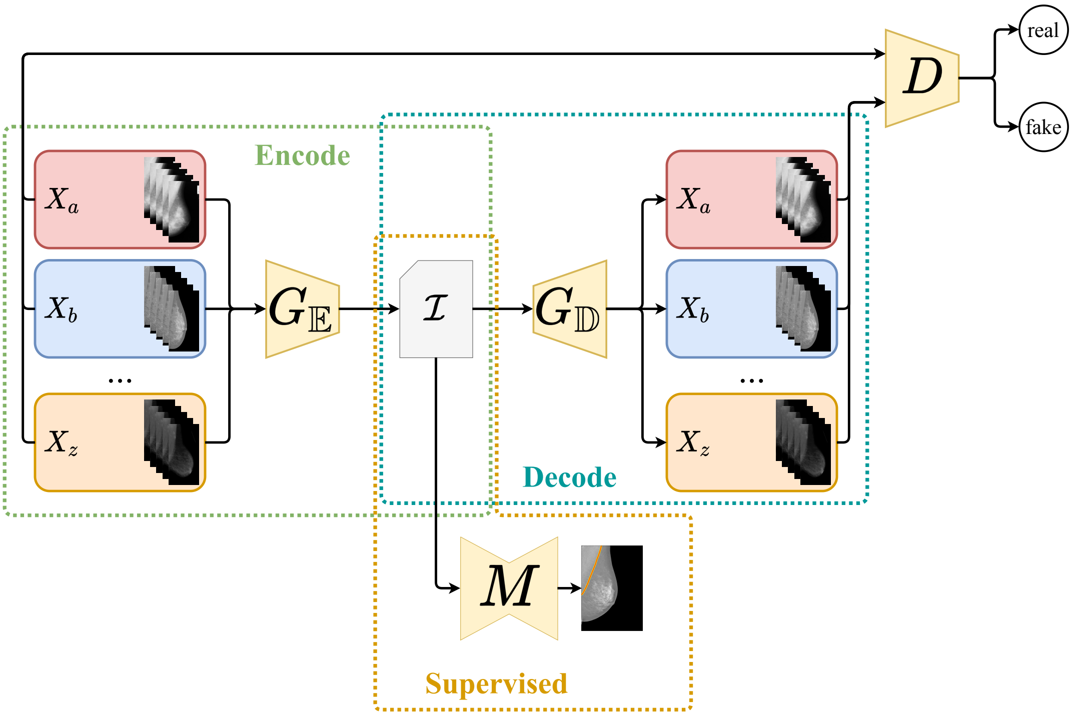


Figure 1.6. CoDAGAN scheme for Cross-Dataset Transfer Learning. A single G network divided into encoder ($G_{\mathbb{E}}$) and decoder ($G_{\mathbb{D}}$) layers performs translations conditionally between the datasets. The discriminator D evaluates if the fake images generated according to the style of the target dataset are likely samples to have been drawn from the target distribution. A single generalizable model M is trained using the isomorphic representation $\mathcal{I}$ generated by G.

and semi-supervised loss. Chapter 4 shows the experimental setup used in this work, including datasets, hyperparameters, the experimental protocol and baselines. Chapter 5 introduces and discusses the results found during the exploratory tests of CoDAGANs for UDA, SSDA and FSDA in quantitative and qualitative manners. At last, Chapter 6 finalizes this work with our final remarks and conclusions regarding the methods and experiments shown in this work, while presenting future improvements that are already being studied for CoDAGANs.

Chapter 2

Theoretical Background and Related Works

CNNs [Krizhevsky et al., 2012; Simonyan and Zisserman, 2014; Szegedy et al., 2015; He et al., 2016b; Huang et al., 2017] are the current state-of-the-art architecture in Computer Vision. Convolutions can be expressed mathematically as a function with two inputs (the signal s and kernel k), so that $f(s,k) = s * k$, where $*$ represents the convolution operation. In the discrete case used in practice, each output index n is computed independently according to $conv(s,k)[n] = \sum_{m=-\infty}^{\infty} s[m]k[n-m]$. CNNs use sets of trainable convolution kernels (also called a filter bank) $\mathcal{K} = k^{(1)}, k^{(2)}, ..., k^{(\mathcal{K})}$ in the earlier layers that highlight useful information in unstructured spatially correlated data (i.e. images, video, etc).

Convolutions are stacked into blocks together with max-poolings, given by $pool(x) = max_{k \times k}(x)$; and/or batch-normalizations, expressed as $bn_{\gamma,\beta}(x) = \gamma(\frac{x-\mu_x}{\sqrt{\sigma_x^2+\epsilon}}) + \beta$, where μ_x and σ_x^2 are the mean and variance of input batch x, respectively; while γ and β are learnable parameters. At last, the Rectified Linear Unit (ReLU) activation proposed by Nair and Hinton [Nair and Hinton, 2010] is the usual non-linear function used in deeper DNNs, including modern CNNs. ReLU is composed of a simple maximum operation $a(x) = max(0,x)$ for an input x and presents a set of desirable properties, as being a non-saturating function, which prevents vanishing gradients; having simple and fast forward and backward computation; and sparsity, leading to less overfitting. These operations are organized in blocks in the earlier layers of CNNs, which are then followed by fully-connected layers for inference at the end of the network. A convolutional block L is, therefore, a composite function $a_L = f(x, \mathcal{K}_L) = pool(relu(bn(conv(x, \mathcal{K}_L))))$, with the output called the activations of layer L (a_L), albeit with the order and number of operations possibly altered depending on the architecture. One can see the first layers of CNNs as deep feature extractors, while the fully-connected layers work similarly to traditional Multi-Layer Perceptrons (MLPs), combining the features into predictions, as shown in the upper half of Figure 2.1.

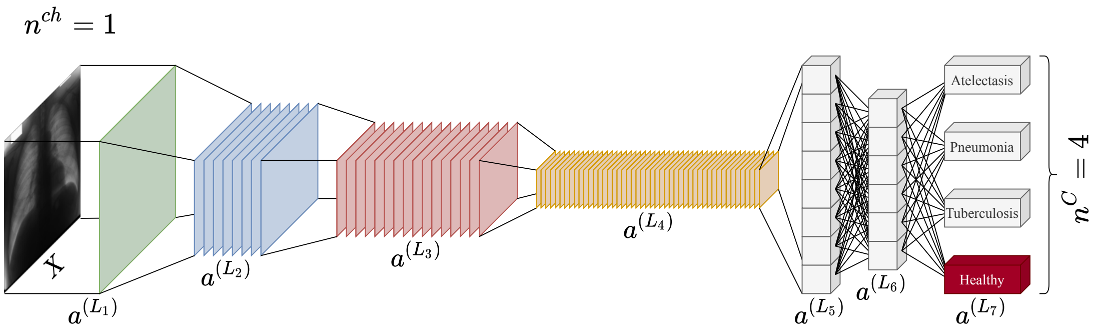


Figure 2.1. Typical architecture of a CNN for image classification with 5 convolutional layers and 3 fully connected layers.

AlexNet [Krizhevsky et al., 2012] reintroduced feature learning in visual recognition tasks, allowing for better scalability than the first CNNs (i.e. LeCun et al. [1998]) in order to perform inference over harder tasks (i.e. ImageNet [Deng et al., 2009] and CIFAR [Krizhevsky et al., 2009]). AlexNet took advantage of larger convolutional kernels in the earlier layers and contained a total of eight layers, between convolutional and fully-connected ones. VGG [Simonyan and Zisserman, 2014] simplified CNN architectures by using the same parameters in all convolutions (3 × 3 with stride 2 and padding equal to 1) and intermittent max-poolings (2 × 2 with stride 2). This architecture was based on the premise that larger kernels can be emulated by smaller sequential convolutions. In contrast to VGG, the GoogleNet architecture [Szegedy et al., 2015] – also known as Inception – studied use a diverse set of kernel sizes to enforce disentanglement in activations. Inception modules mix combinations of 1 × 1, 3 × 3 and 5 × 5 convolutions and 2 × 2 poolings in parallel, which generate a diverse set of activations that are concatenated before serving as input for the next block. Inception v3 [Szegedy et al., 2016] is the latest version of the architecture, also exploiting non-square kernel sizes as 1 × 3 and 3 × 1 mixed with square convolutions. Both VGG and Inception allow for deeper networks with smaller convolutional kernels in each module, which proved to be more efficient than shallower networks with larger convolutions, at least up to around 20 layers.

As CNNs grew larger, so did the vanishing gradient problem, as deepening the networks progressively degraded the backward propagation that allows training of

DNNs. It was observed that adding layers beyond a total of 20 was detrimental to the training of CNNs, as the gradients did not reach the earlier layers, effectively preventing their training. Residual Networks (ResNets) [He et al., 2016b] based on residual identity functions $a_{l_{i+1}} = a_{l_i} + f(a_{l_i})$ were then introduced. The gradients of certain convolutional blocks are given by an addition of the outputs of this convolutional blocks with their inputs. This identity has the effect of creating shortcuts for the backpropagations, enabling them to train the first layers, while also enforcing diversity in the features learned by each module. ResNets with between 18 and 151 convolutional blocks were investigated by He et al. [2016b], with little benefit being observed beyond that. Soon after the standard ResNets some improvements to the architectures were observed to increase their recognition performance. The most famous ones were Wide ResNets (WRNs) [Zagoruyko and Komodakis, 2016] and ResNeXt [Xie et al., 2017]. However, residual blocks were observed to be highly inefficient, as the activations of most convolutions all throughout a ResNet could be dropped with little-to-no effect on classification performance [Huang et al., 2016]. Densely Connected Convolutional Networks (DenseNets) [Huang et al., 2017] improved on the parameter efficiency of ResNets by replacing the identity function by concatenation. Huang et al. [2017] tested DenseNet with between 121 and 264 layers, observing them to be more efficient than ResNets in both parameter and flops, when compared the similar errors in the validation set. DenseNets also presented alternatives for further efficiency improvements, as bottleneck layers and compressing output activations in transition layers between densely connected modules.

2.1 Deep Semantic Segmentation

Since the resurgence of Neural Network technology as Deep Learning in the early 2010's, these networks have been adapted to perform dense labeling (i.e. segmentation tasks). Semantic segmentation has been an active research topic in the area of biomedical image analysis for decades, as it is a rather common preprocessing and evaluation tool for several medical applications. Traditionally this field of research uses several active contour, clustering, atlas and interactive methods. More recently, with the advent of DNNs, semantic segmentation in Computer Vision has become dominated by deep-based methods. Therefore, several algorithms comprising the state-of-the-art of deep semantic segmentation were used in our experimental setup. Most of these architectures are discriminative models based on improvements over CNNs and Fully Convolutional Networks (FCNs) [Long et al., 2015] – as shown in

Sections 2.1.1 and 2.1.2.

2.1.1 Fully Convolutional Networks

The most basic architectures are the FCNs [Long et al., 2015], which are often based on CNN models like AlexNet [Krizhevsky et al., 2012] and VGG [Simonyan and Zisserman, 2014] adapted to dense prediction (Figure 2.2). An FCN can be understood as a patchwise approach, wherein each pixel in an image is a sample. Whole image fully convolutional training is identical to patchwise training where each batch consists of all the pixels in an image or set of images. Replacing fully connected layers in a CNN by convolutional layers and adding a spatial loss produces an efficient machine for end-to-end dense learning [Long et al., 2015]. While the same effect could be produced by training a regular CNN for patch classification, FCNs are several times more efficient than patchwise CNNs.

One should notice that FCNs use the same loss functions as CNNs for image classification, as dense labeling can be seen as a collection of sparse labels for each pixel in an image. Therefore, Cross Entropy is the most common loss for supervised semantic segmentation and it can be expressed by:

$$\mathcal{L}_{sup}(Y,\hat{y}) = -Y\log(\hat{y}) - (1-Y)\log(1-\hat{y}), \tag{2.1}$$

where Y represents the pixelwise semantic map and $\hat{y}$ the probabilities for each class for a given sample.

2.1.2 Encoder-Decoder Architectures

Ever since FCNs, several attempts to mitigate the vanishing gradient problem have been proposed, most relying in alternative paths for information flow [Srivastava et al., 2015; Larsson et al., 2016; He et al., 2016b; Huang et al., 2017]. Skip connections are the most common way to create these alternative paths, serving as highways for backpropagation to reach earlier layers in the network without passing through all the layers in front of them. U-Nets [Ronneberger et al., 2015] take advantage of skip connections to map higher-level contextual information to low-level pixel information. These networks are Encoder-Decoder architectures wherein the downsampling half (Encoder) is symmetrical to the upsampling half (Decoder), as shown in Figure 2.3. Encoder-Decoder DNNs are based on the transposed convolution operation, which is usually implemented quite similarly to a traditional convolution with trainable kernels. Transposed convolutions present one crucial distinction to

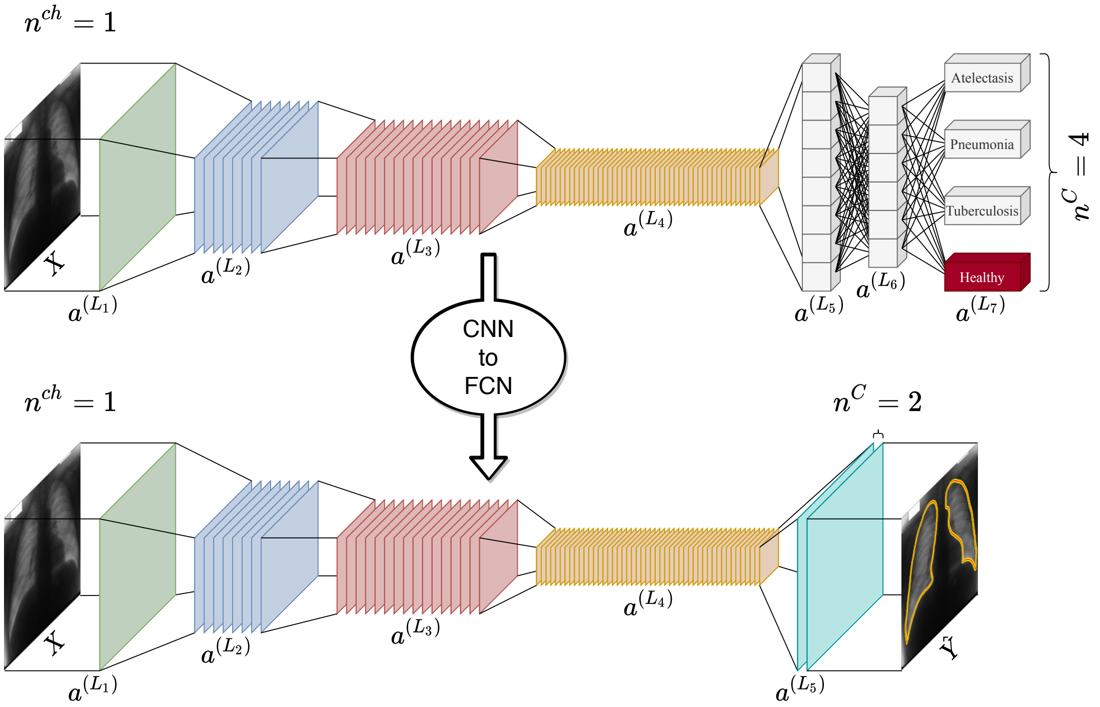


Figure 2.2. Architecture example of a CNN for image classification and its equivalent FCN architecture with the same backbone for semantic segmentation. Activations from layer l are depicted as $a^{(l)}$ for all layers in the network (L_1 through L_7 for the CNN and L_1 through L_5 for the FCN). One should notice that in both architectures the input layer $a^{(L_1)}$ has the number of channels n^{ch} depending on the input data's number of channels (in the case of RGB images, $n^{ch} = 3$). In radiology, typically the images are grayscale representations of a single x-ray band, thus, $n^{ch} = 1$ for basically all other examples and applications in this work. In the CNN, the number of neurons in the output layer must match the number of classes (n^C) in the data. Equivalently, in the FCN, the number of channels in the output layer n^C, as suggested by the notation, depends on the number of classes of the dataset.

normal convolutions: instead of decreasing (or maintaining) the spatial resolution of the input, they perform learnable spatial upsampling, thus allowing for symmetrical architectures. The presence of trainable transposed convolutions stands in contrast to the bilinear interpolation used to recover the spatial resolution of FCNs, which has no trainable parameters and, thus, cannot learn upsampling kernels specifically designed to the task.

SegNets [Badrinarayanan et al., 2017], like U-Nets, are Encoder-Decoder architectures for segmentation with an architecture composed of symmetric layers. The Encoder half of the network is usually composed of the first 13 convolutional layers in the VGG-16 network [Simonyan and Zisserman, 2014], even allowing for

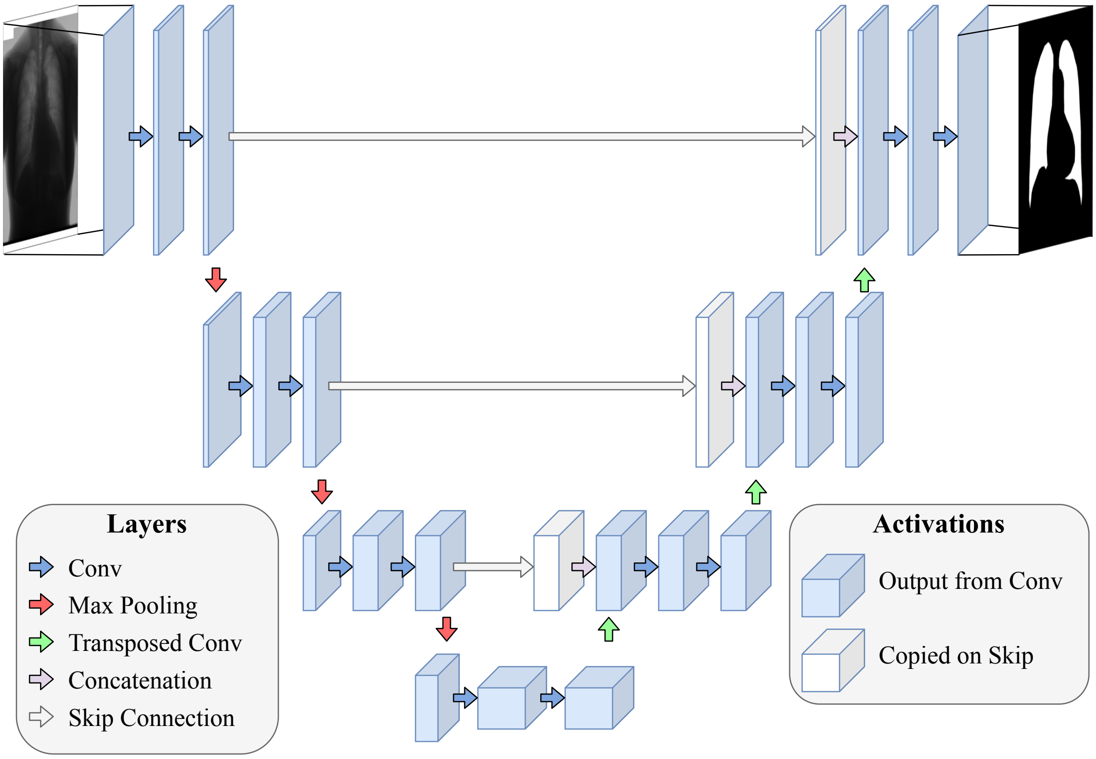


Figure 2.3. U-Net architecture. Each conv2d box corresponds to multi-channel trainable convolutional kernels followed by downsampling or upsampling. White arrows denote the skip connections between symmetric layers. Adapted from Ronneberger et al. [2015].

the pretraining of these layers in computer vision tasks. The construction of the Decoder network is accomplished by simply mirroring the Encoder layers and replacing the pooling layers for transposed convolutions, which work as upsampling layers, as can be seen in Figure 2.4. One main advantage of SegNet compared to other segmentation architectures is the use of the pooling indices in the Decoder layers. SegNet uses these indices to concatenate only the activations selected by the pooling on the Encoder, resulting in sparse activation maps on the skip connections.

Similarly to FCNs, both U-Nets and SegNets for semantic segmentation adapt supervised classification losses (i.e. Cross Entropy in Eq. 2.1) for training.

2.2 Generative Adversarial Networks

Generative Adversarial Networks (GANs) [Goodfellow et al., 2014] have been an active and proliferous subject of research during the last years, being arguably the

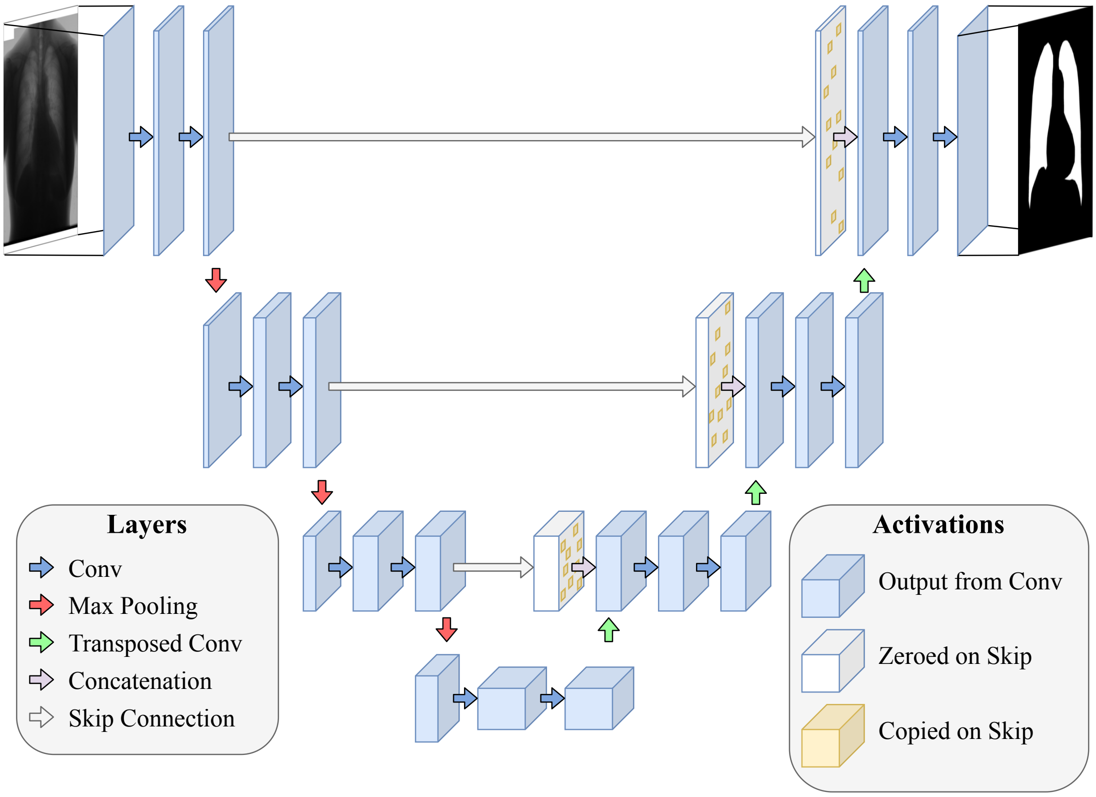


Figure 2.4. An illustration of the SegNet architecture. Each Conv2d box corresponds to multi-channel convolutions followed by downsampling or upsampling. Arrows denote the passage of pooling indices and their respective activations to later layers. Adapted from Badrinarayanan et al. [2017].

main go-to solution to deep generative modeling. Traditional GANs are composed of two networks trained conjointly: a generator (G) and a discriminator (D), as can be seen in Figure 2.5.

D is trained to correctly classify real samples $x \sim p_{data}$ drawn from the training dataset from fake samples $G(z) \sim p_{fake}$ created by the generator according to a random vector z drawn from a noise distribution p_Z. G is trained to fool D by approximating p_{fake} from the true data distribution p_{data}. Loss functions for optimization for D ($\mathcal{L}^{D}_{adv}$) and G ($\mathcal{L}^{G}_{adv}$) are given by Equations 2.2 and 2.3, respectively:

$$\mathcal{L}^{D}_{adv\_mm} = -\mathbb{E}_{x\sim p_{data}}\left[\log\left(D(x)\right)\right] - \mathbb{E}_{z\sim p_Z}\left[\log\left(1 - D(G(z))\right)\right], \tag{2.2}$$

$$\mathcal{L}^{G}_{adv\_mm} = -\mathcal{L}^{D}_{adv\_mm}. \tag{2.3}$$

As objectives for the generator and discriminator are exact opposites from each

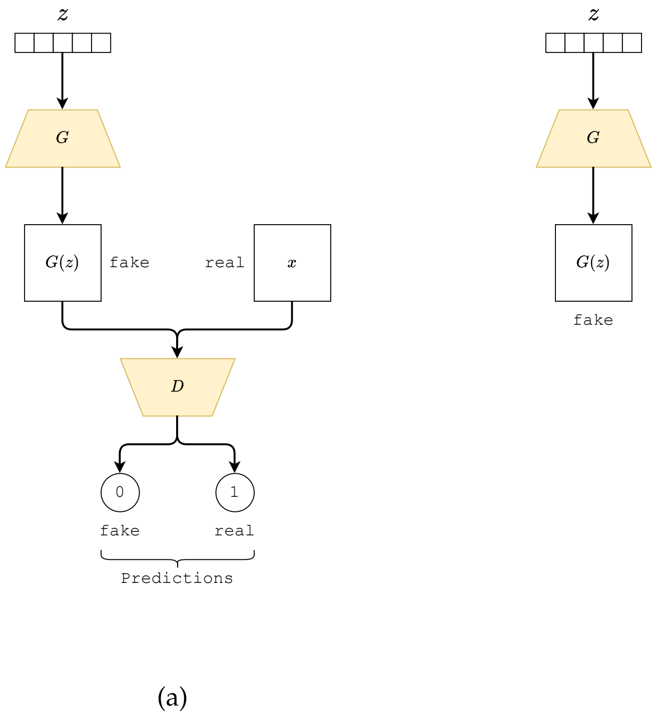


Figure 2.5. Traditional architecture for a GAN. (a) Training (distribution fitting). (b) Testing (image synthesis). The composite architecture of a GAN is traditionally composed of two distinct DNNs: one generator G and one discriminator D that are trained conjointly, while during testing only the generator is used to synthesize novel samples.

other, the networks can converge together when G and D are trained intermittently. As detailed by Goodfellow et al. [2014], this scheme is equivalent to a two-player MiniMax game. Since this first proposal for adversarial generative learning – usually known as MiniMax Generative Adversarial Network (MMGAN) – several advances have been made regarding training stability [Goodfellow et al., 2014; Mao et al., 2017; Gulrajani et al., 2017; Arjovsky et al., 2017; Lucic et al., 2018; Karras et al., 2018; Brock et al., 2018; Karras et al., 2020], some of which will be further detailed in the following paragraphs.

MMGAN is especially convenient for theoretical analysis in the sense that both loss components can be expressed by the same equation, with D minimizing the Cross Entropy and G maximizing it. However, in practice MMGANs suffers from convergence problems when D converges at its task at a faster pace than G. As shown in Figure 2.5(a), all the gradients used to train G flow from the Cross Entropy computed at the end of the classification process from D. Thus, if D is able to correctly classify all samples as real or fake ones at some point in the training procedure, no gradients are fed to G because the loss is null. This is known as the problem of cost saturation in GANs. This and other limitations of the simpler MMGAN encouraged the proposal of novel non-saturating loss functions for GANs with distinct regularizations and/or losses that ease the training [Goodfellow et al., 2014; Mao et al., 2017; Arjovsky et al., 2017; Gulrajani et al., 2017].

As pointed by Goodfellow [2016], instead of flipping the sign of G, it is possible to fool the discriminator by feeding synthetic data from the generator with the inverse label – that is, as if they were real samples – and backpropagating the loss to G. Using a more formal description, this means optimizing for the following couple of Non-saturating losses for G and D respectively:

$$\mathcal{L}^{G}_{adv\_ns} = -\mathbb{E}_{z\sim p_Z}\left[\log\left(D(G(z)\right]\right., \tag{2.4}$$

$$\mathcal{L}^{D}_{adv\_ns} = -\mathbb{E}_{x\sim p_{data}}\left[\log\left(D(x)\right)\right] - \mathbb{E}_{z\sim p_Z}\left[\log\left(1 - D(G(z))\right)\right]. \tag{2.5}$$

These equations form the basis of the first Non-Saturating Generative Adversarial Networks (NSGANs), proposed in the same manuscript as MMGANs, but with fewer convergence problems due to saturation.

Least Squares Generative Adversarial Networks (LSGANs) were introduced by Mao et al. [2017] as an alternative to NSGANs with less vanishing gradient problems, better stability during training and greater visual quality in image synthesis tasks. These GANs use the Least Squares loss function instead of the traditional Cross Entropy in D, which unbounds loss values from the interval $[0, 1]$. The loss functions for generators and discriminators of LSGANs, respectively, can be seen in the following equations:

$$\mathcal{L}^{G}_{adv\_ls} = -\mathbb{E}_{z\sim p_Z}\left[(D(G(z)) - c)^2\right], \tag{2.6}$$

$$\mathcal{L}^{D}_{adv\_ls} = -\mathbb{E}_{x\sim p_{data}}\left[(D(x) - b)^2\right] - \mathbb{E}_{z\sim p_Z}\left[(D(G(z)) - a)^2\right], \tag{2.7}$$

where a is the label for fake data, b is the label for real data and c is the value that the generator G expects the discriminator D to believe to come from the real data. The hyperparameters a, b and c are usually set to be 0, 1 and 1, respectively, as this is shown by Mao et al. [2017] to minimize the Pearson χ^2 divergence between the distributions p_{data} and p_{fake}.

Further developments in GAN training were explored in the form of new loss functions that are able to achieve unsupervised adversarial training, such as Wasserstein Generative Adversarial Networks (WGANs) [Arjovsky et al., 2017] and Energy-based Generative Adversarial Networks (EBGANs) [Zhao et al., 2016]. Another research branch focused on different kinds of regularization terms to well-known adversarial losses, as in infoGAN [Chen et al., 2016], Wasserstein Generative Adversarial Network with Gradient Penalty (WGAN-GP) [Gulrajani et al., 2017], Boundary Equilibrium Generative Adversarial Network (BEGAN) [Berthelot et al., 2017] and Deep Regret Analytic Generative Adversarial Network (DRAGAN) [Kodali et al., 2017] and BigGANs [Brock et al., 2018]. Starting from Deep Convolutional Generative Adversarial Networks (DCGANs) [Radford et al., 2015], GAN architectures have also been explored to enable the training of deeper and more stable generative models for images. Radford et al. [2015] proposed a set of architectural guidelines for both D and G that allowed for deeper convolutional GANs. Recent developments in the literature have observed the effect of distinct architectural choices in generative adversarial training, such as progressively increasing the depth of GANs [Karras et al., 2018], separating the architecture into style blocks with internal Adaptive Instance Normalization (AdaIN) layers [Karras et al., 2019] and the decoupling of biases and demodulation between style blocks [Karras et al., 2020].

If trained properly, G is able to receive new random vectors $z^1, z^2, ..., z^N$ and generate fake samples $G(z)^1, G(z)^2, ..., G(z)^N$ drawn from the approximate distribution p_{fake}. Later iterations on the research in generative modelling proposed changes in the architectures, input data and losses in order to adapt GANs to tasks such as deep convolutional architectures [Radford et al., 2015], conditional training [Mirza and Osindero, 2014], unsupervised mapping of latent variables [Chen et al., 2016] and modeling joint distributions using only marginal samples – as in Coupled GANs [Liu and Tuzel, 2016]. Current state-of-the-art GANs are able to perform tasks as diverse as: 1) generating high resolution images with visual quality reasonably close to real ones [Karras et al., 2018; Brock et al., 2018; Karras et al., 2020]; 2) single-image superresolution [Yuan12 et al., 2018]; 3) image-to-image translation between different domains [Isola et al., 2017; Zhu et al., 2017a; Liu et al., 2017; Huang et al., 2018; Liu et al., 2019]; 4) one- and few-shot generative modeling [Liu et al., 2019; Shaham et al., 2019]; and 5) knowledge transfer [Hoffman et al., 2016, 2018; Wu et al., 2018; Zou et al., 2018]. Some of these tasks have been more recently tackled by Variational AutoEncoders (VAEs) [Kingma and Welling, 2013] architectures as well, both alone and conjointly with GANs [Liu et al., 2017; Zhu et al., 2017b; Huang et al., 2018].

For simplicity, from here on out adversarial losses as a whole (be it Non-Saturating, Least Squares, Wasserstein, etc) will be referred to using the notation $\mathcal{L}_{adv}(X)$ for a GAN that fits the distribution p_X.

2.2.1 Coupled Generative Adversarial Networks

Coupled Generative Adversarial Networks (CoGANs) [Liu and Tuzel, 2016] are generative networks composed of two generators and two discriminators, which are trained in two related but distinct image domains A and B simultaneously. CoGANs aim to model a joint distribution p_{X_A,X_B} using samples X_A and X_B drawn from the marginal distributions p_{X_A} and p_{X_B}, but no samples $X_{A,B}$ from the joint distribution. In other words, these DNNs model the correlations between the two image domains without correspondence supervision between the samples of these domains. Joint convergence across the domains is enforced in CoGANs by weight sharing in both the pair of generators and the pair of discriminators, as shown in Figure 2.6. In test phase, these networks are able to produce paired image samples from the two domains and perform tasks related to the joint probability distribution generated by the coupled networks.

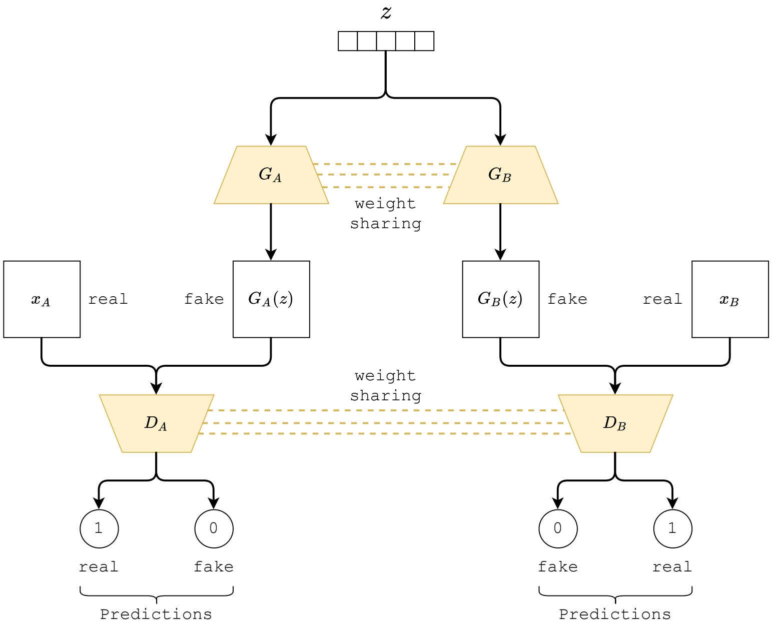


Figure 2.6. CoGAN architecture composed of two generators (G_A and G_B) and two discriminators (D_A and D_B). Weight sharing is represented by dashed lines.

Apart from qualitative evaluations on the samples from the related domains, Liu and Tuzel [2016] also reports results using CoGANs in tasks such as UDA and Cross-Domain Image Transformation. The latter will be henceforth referred to as Image-to-Image Translation.

2.3 Image-to-Image Translation

Image-to-Image (I2I) Translation Networks are GANs [Goodfellow et al., 2014] capable of transforming samples from one image domain into images from another. Access to paired images from the two domains simplifies the learning process considerably, as losses can be devised using only pixel-level or patch-level comparisons between the original and translated images [Isola et al., 2017]. Paired I2I Translation can be achieved, therefore, by Conditional Generative Adversarial Networks (CGANs) [Mirza and Osindero, 2014] coupled with simple regression models [Chen and Koltun, 2017]. In order to achieve image translation, the adversarial components presented described in Section 2.2 are added to a paired regression loss $\mathcal{L}_{pair}(X_A^{(i)}, X_B^{(i)})$ between a pair of samples of index i for datasets X_A and X_B from domains A and B:

$$\mathcal{L}_{pair}(X_A^{(i)}, X_B^{(i)}) \; = \; \mathbb{E} \left\| X_A^{(i)} - G(X_B^{(i)}) \right\|. \tag{2.8}$$

This regression loss is usually the L1 loss, as it tends to produce less blurry results than the Mean Squared Error (MSE) loss [Isola et al., 2017].

Isola et al. [2017] introduced the first DNN architecture for domain-agnostic I2I Translation. Before this contribution, image translation tasks (i.e. image coloring, mapping remote sensing images to semantic maps, creating photorealistic images from sketches, etc) were tackled with separate, special-purpose methodologies [Buades et al., 2005; Chen et al., 2009; Efros and Freeman, 2001; Eigen and Fergus, 2015; Zhang et al., 2016], but the problem remains the same in all these settings: mapping pixels to pixels. A lot of effort went into designing task-specific losses for all these restricted domain I2I translation methods. The main contribution of Isola et al. [2017] was, therefore, to provide a general architecture – henceforth referred to as pix2pix[1] – and loss for this kind of task.

Before pix2pix, the literature had already discovered that naive approaches for image translation losses – such as using Euclidean Distance – tended to produce blurry results, as the network tries to minimize the mean of the samples [Zhang

[1] https://phillipi.github.io/pix2pix/

et al., 2016; Pathak et al., 2016]. One can see this blurry effect on Figure 2.7. The solution to this problem was to introduce an adversarial loss to the pipeline by using a GAN [Goodfellow et al., 2014] architecture. Adversarial losses tend to produce more photorealistic images than traditional losses, as the discriminator is able to identify blurry images and to force the generator to produce images with sharper edges. A graphical representation of the pix2pix architecture can be seen in Figure 2.8.

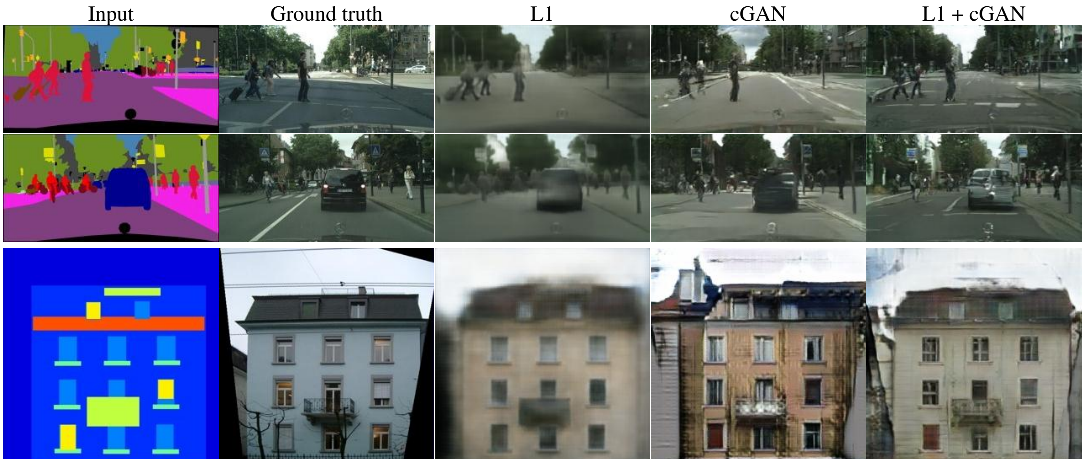


Figure 2.7. Different losses induce different quality of results. Each column shows results trained under a different loss. Source: Isola et al. [2017].

The generator network is usually either an Encoder-Decoder network such as U-Net [Ronneberger et al., 2015] or a mixture of downscaling block and residual ones (i.e. He et al. [2016b]), with both architectures receiving images in the source domain and translating them to the target domain. The discriminator network is a traditional architecture for image classification, such as a CNN [Krizhevsky et al., 2012; Simonyan and Zisserman, 2014]. The discriminator has the job of determining if the image is a natural sample from the specific domain or if it is a translated sample originally from another domain. In other words, the discriminator is a CNN for binary image classification between the classes real and fake.

Samples are fed to the network during the training phase in a supervised manner and, therefore, pix2pix requires paired images in the source and target domains, as shown in Figure 2.9. Access to paired images from the two domains (X_A and X_B) considerably simplifies the learning process, as losses can be devised using only pixel-level or patch-level comparisons between the original and translated images [Isola et al., 2017]. One can see in Figure 2.9(b) that in the beginning of the training process for pix2pix the translation of sample $X_A^{(i)}$ to $X_B^{(i)}$ may produce a warped translated version ($X_{A\rightarrow B}^{(i)}$) of the source sample due to a poorly fit translation func-

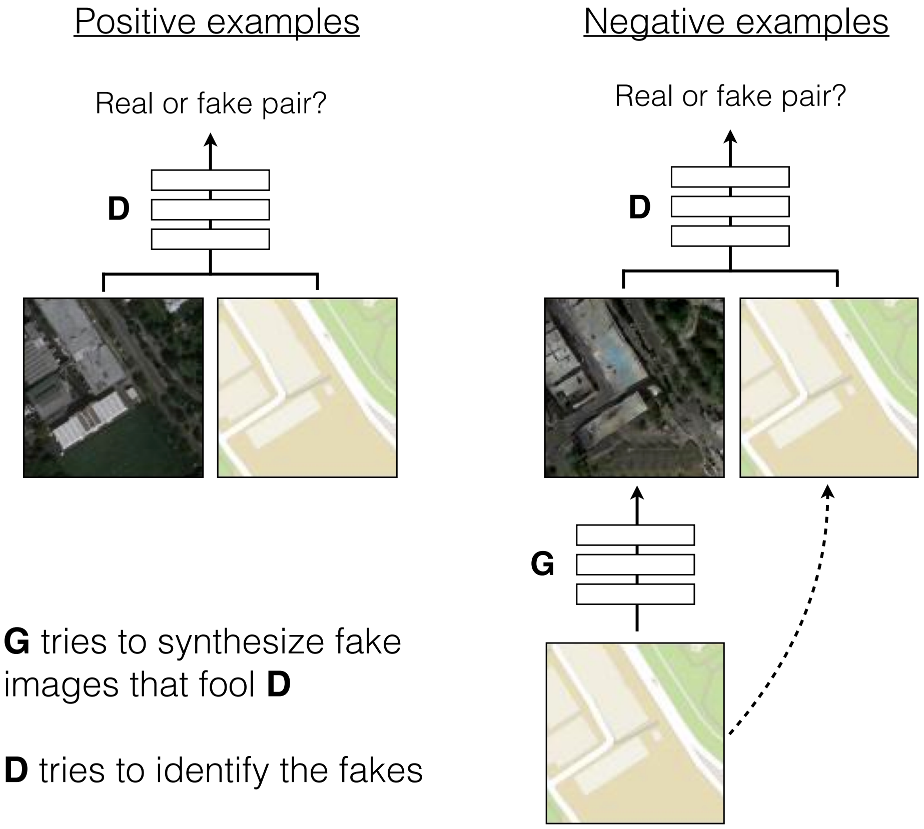

Positive examples
Negative examples
Real or fake pair?
Real or fake pair?
D
D
G
G tries to synthesize fake images that fool **D**
D tries to identify the fakes

Figure 2.8. Training a conditional GAN to predict aerial photos from maps. The discriminator, **D**, learns to classify between real and synthesized pairs. The generator **G** learns to fool the discriminator. Unlike an unconditional GAN, both the generator and discriminator observe an input image. Source: Isola et al. [2017].

tion $G_{A\to B}$. When paired samples from the two domains are accessible, one can simply compare the real $X_B^{(i)}$ sample with the synthetic sample $X_{A\to B}^{(i)}$ to devise an objective cost function based on the premise that $X_B^{(i)} \approx X_{A\to B}^{(i)}$ in order to optimize $G_{A\to B}$. An example of a distance-based cost function that could be used in such a scenario is represented in Figure 2.9(c). If convergence is met and the aforementioned cost is compensated properly, translations $A \to B$ become possible, as shown in Figure 2.9(d). During testing, pix2pix can perform translations $A \to B$ between generic samples from A to B, as in Figure 2.9(e). The composite loss function for pix2pix couples this distance-based objective loss with an adversarial loss in order to enforce photorealistic image translation.

One example of DNN architecture that can perform Paired I2I Translation can be seen in Figure 2.10. There are two generators ($G_{A\to B}$ and $G_{B\to A}$) and two discriminators (D_A and D_B) in this architecture, that is, one pair $\{G, D\}$ for each do-

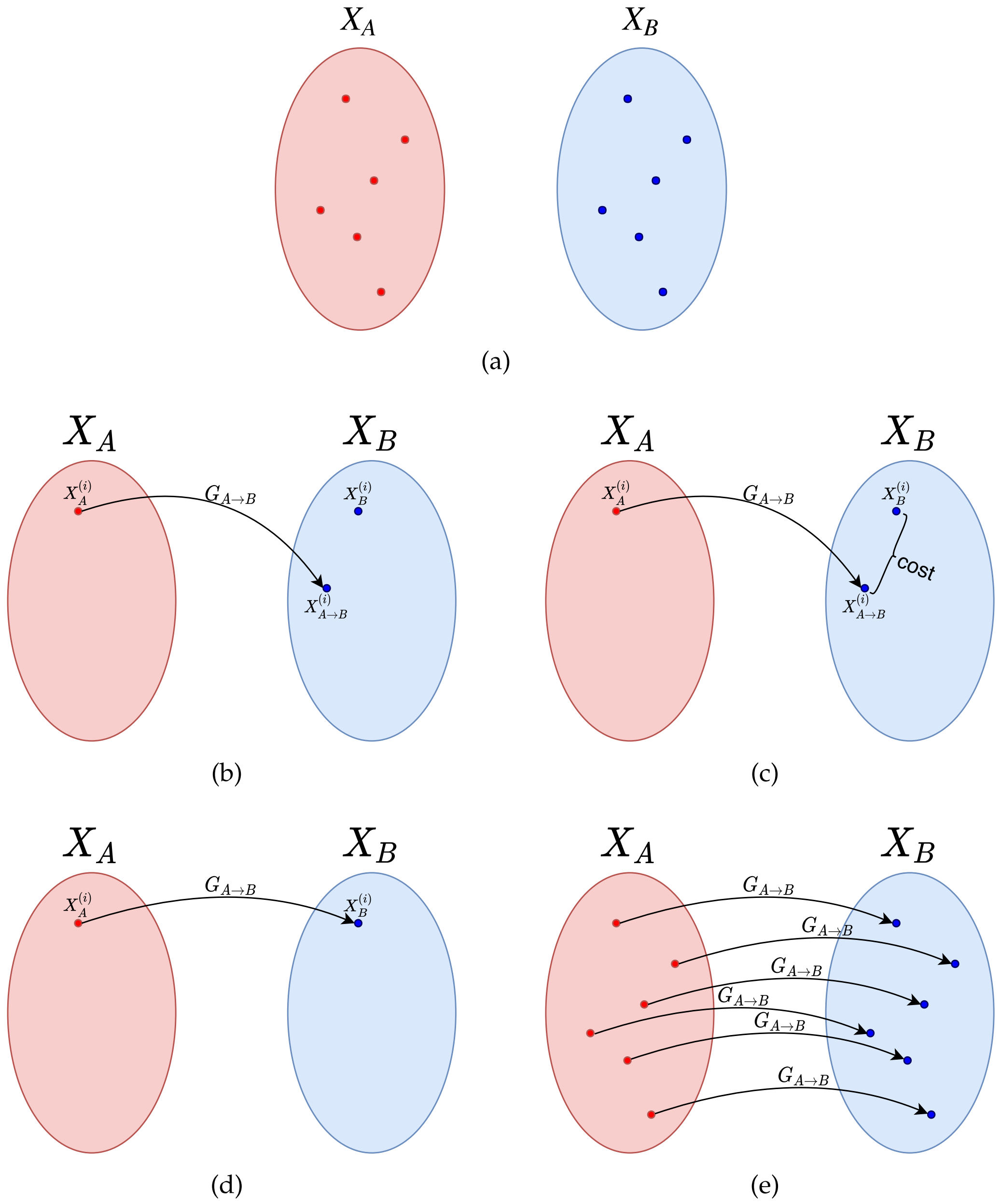
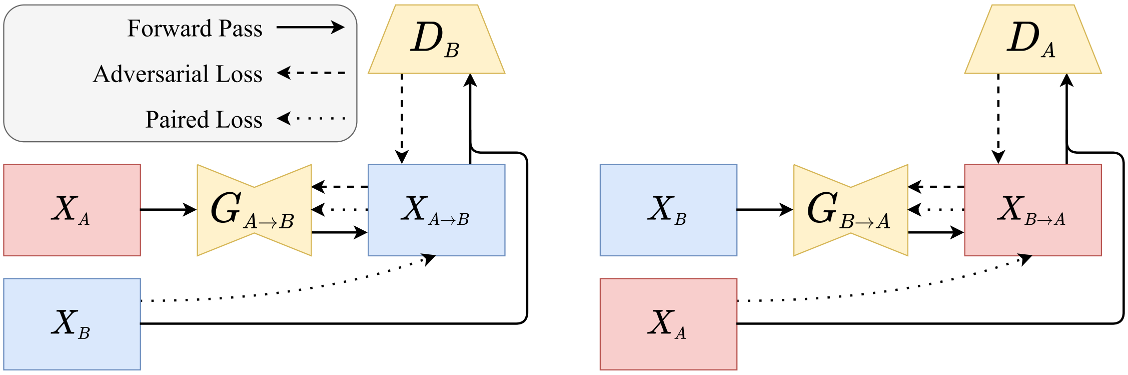


Figure 2.9. Stages in the training procedure for translations between two paired sample sets X_A and X_B from domains A and B on pix2pix. We only show the process for a translation $A \rightarrow B$, but translations $B \rightarrow A$ are analogous. (a) Samples from the domains. (b) Beginning of the training procedure with large translation errors. (c) Distance-based cost function for paired translation. (d) Correct translation from sample $X_A^{(i)}$ to sample $X_B^{(i)}$ after training. (e) Translation between samples from A and B.

main. For a pair of images $\{X_A, X_B\}$ from domains A and B, one pair of synthetic images $\{X_{A\rightarrow B}, X_{B\rightarrow A}\}$ is produced by $G_{A\rightarrow B}$ and $G_{B\rightarrow A}$. These synthetic images are forwarded through D_A and D_B in order for the discriminators to try to discern between real and fake samples. As in standard GANs [Goodfellow et al., 2014], backpropagation flows from the discriminators to the generators.



(a) (b)

Figure 2.10. Example of a standard architecture that performs Paired Image-to-Image Translation between a pair of domains A and B. In contrast to traditional GAN architectures (Figure 2.5), these architectures are usually composed of two generators ($G_{A\to B}$ and $G_{B\to A}$) responsible for performing $A \leftrightarrow B$ translations and two discriminators (D_A and D_B), responsible for trying to discern between real and synthetic samples from each domain.

The need for paired samples represents a serious hampering for many real world applications of pix2pix, including biomedical ones. Sample pairing is not required by Unsupervised I2I Translation methods [Zhu et al., 2017a; Liu et al., 2017; Huang et al., 2018], further detailed in Section 2.3.1.

2.3.1 Unsupervised Image-to-Image Translation

Requiring paired samples reduces the applicability of I2I translation to a very small and limited subset of image domains where there is the possibility of generating paired datasets. This limitation motivated the creation of Unsupervised Image-to-Image Translation methods [Zhu et al., 2017a; Liu et al., 2017; Huang et al., 2018]. On paired networks as pix2pix, one can simply compare images from the source domain to images from the target domain, but this strategy does not work for unpaired samples, thus the need for a new distance-based loss function. As pointed by Zhu et al. [2017a], in the language domain, verifying and improving translations via "back translation and reconsiliation" is a technique used by human translators [Brislin, 1970], as well as by machines [He et al., 2016a]. Unsupervised Image-to-Image Translation networks are based on the concept of Cycle-Consistency, which models the translation process between two image domain as an invertible process represented by a cycle, as can be seen in Figure 2.11. This cyclic structure allows for Cycle-Consistent losses to be used together with the adversarial loss components of traditional GANs.

A Cycle-Consistent loss can be formulated as follows: let A and B be two image domains containing unpaired image sample sets X_A and X_B. Consider then two functions $G_{A\rightarrow B}$ and $G_{B\rightarrow A}$ that perform the translations $A \rightarrow B$ and $B \rightarrow A$ respectively. Then a loss $\mathcal{L}_{cyc}$ can be devised by comparing the pairs of images $\{X_A^{(i)}, X_{A\rightarrow B\rightarrow A}^{(i)}\}$ and $\{X_B^{(i)}, X_{B\rightarrow A\rightarrow B}^{(i)}\}$. In other words, the relations $X_A^{(i)} \approx G_{B\rightarrow A}(G_{A\rightarrow B}(X_A^{(i)}))$ and $X_B^{(i)} \approx G_{A\rightarrow B}(G_{B\rightarrow A}(X_B^{(i)}))$ should be maintained in the translation process. The counterparts of the generative networks in GANs are discriminative networks, which are trained to identify if an image is natural from the domain or translated samples originally from other domains. D_A and D_B are referred to as the discriminative networks for datasets A and B, respectively. Discriminative networks are normally traditional supervised networks, such as CNNs [Krizhevsky et al., 2012; Simonyan and Zisserman, 2014], which are trained in the classification task of distinguishing real images from fake images generated by the generators.

In practice, the loss $\mathcal{L}_{cyc}$ is usually the same L1 regression loss $\mathcal{L}_{pair}$ used in Paired I2I Translation, but due to the lack of paired $X_A^{(i)}$ and $X_B^{(i)}$ samples, the regression is computed instead using the original sample $X_A^{(i)}$ and its reconstruction $X_{A\rightarrow B\rightarrow A}^{(i)}$ as follows:

$$\mathcal{L}_{cyc}(X_A^{(i)}, X_{A\rightarrow B\rightarrow A}^{(i)}) = \mathbb{E}\left\|X_A^{(i)} - X_{A\rightarrow B\rightarrow A}^{(i)}\right\|. \tag{2.9}$$

The case for translations $B \rightarrow A \rightarrow B$ is analogous to the case of $A \rightarrow B \rightarrow A$. Cycle-Consistency can be implemented in a DNN by using an architecture such as the one presented in Figure 2.12.

The first proposal for such an architecture was the CycleGAN [Zhu et al., 2017a] and several improvements have been presented in the subject ever since [Li and Wand, 2016; Liu and Tuzel, 2016; Liu et al., 2017; Huang et al., 2018; Liu et al., 2019]. Modern Unsupervised I2I Networks [Liu et al., 2017; Huang et al., 2018; Liu et al., 2019] are built upon the basic architecture of CoGANs [Liu and Tuzel, 2016].

Newer architectures as Unsupervised Image-to-Image Translation (UNIT) [Liu et al., 2017], Multimodal Unsupervised Image-to-Image Translation (MUNIT) [Huang et al., 2018], Diverse Image-to-Image Translation via Disentangled Representations (DRIT) [Lee et al., 2018], DRIT++ [Lee et al., 2020] and Mode Seeking Gen-

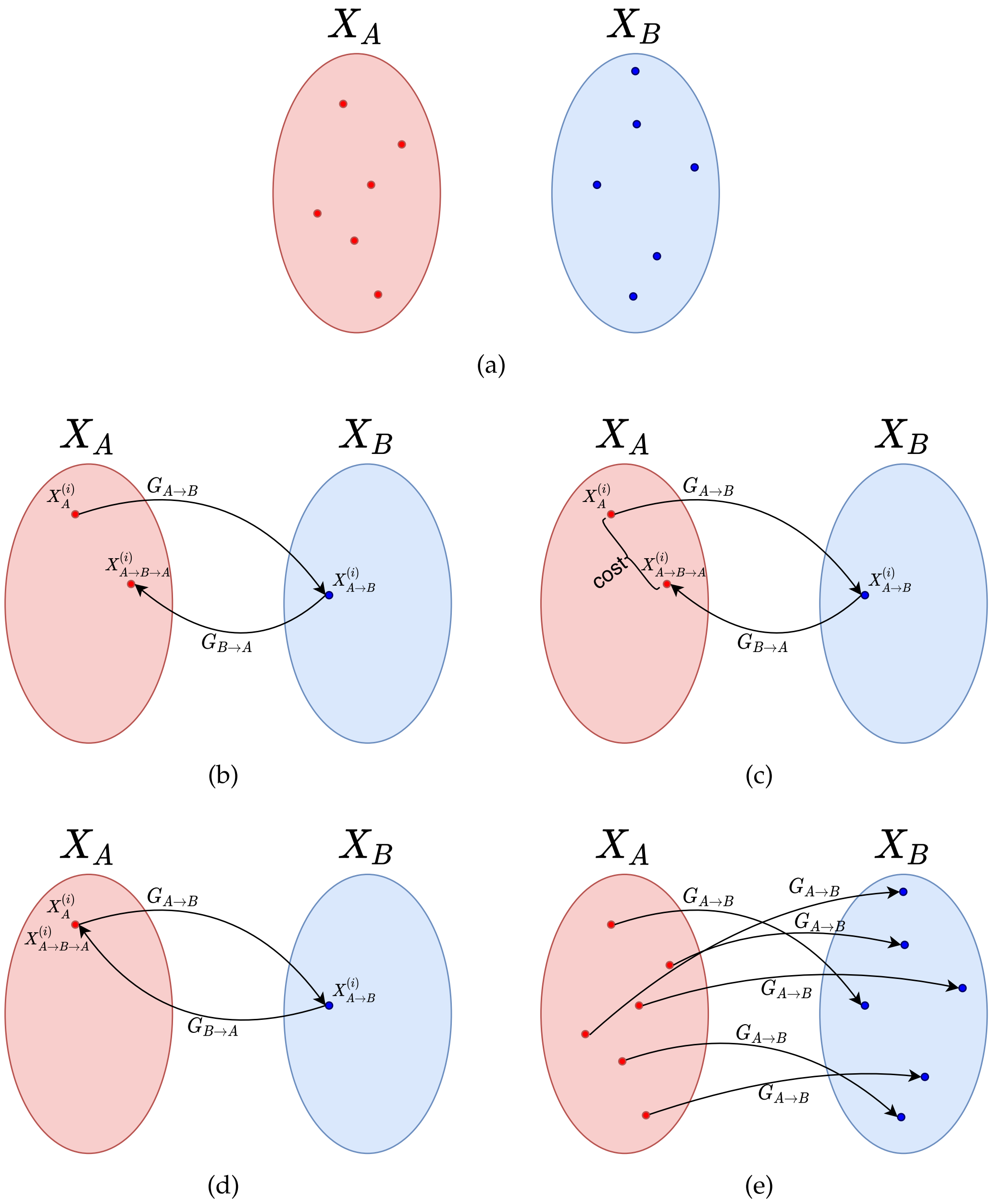
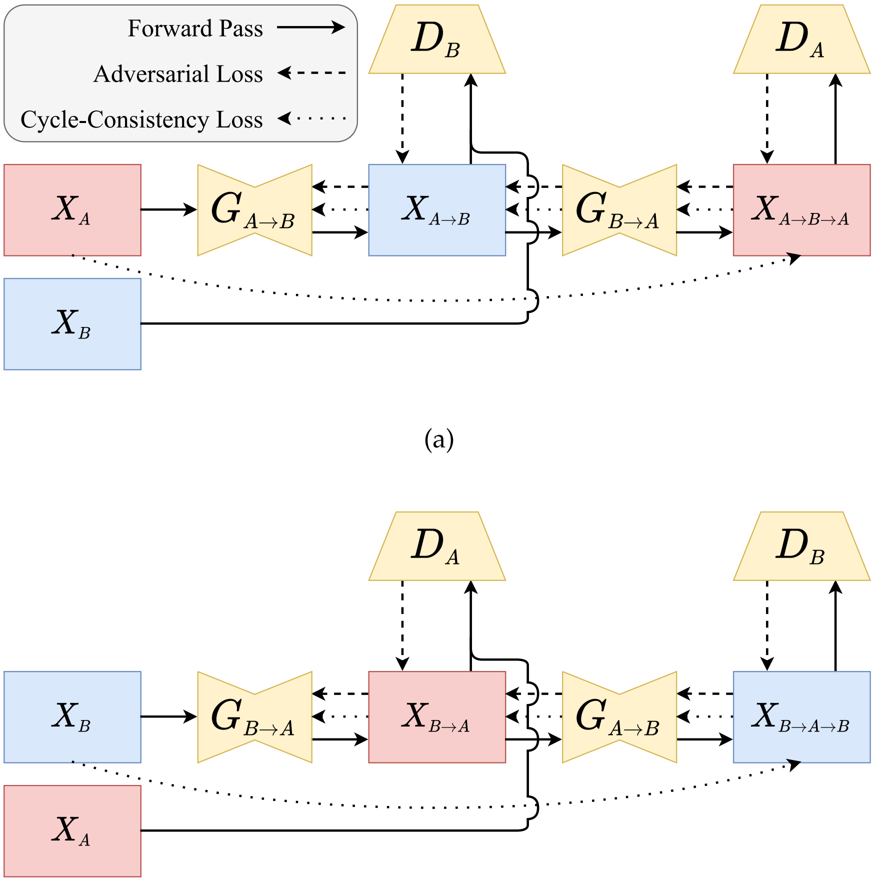


Figure 2.11. Stages in the training procedure for translations between two unpaired sample sets X_A and X_B from domains A and B. We only show the process for a translation $A \to B \to A$, but translations $B \to A \to B$ are analogous. (a) Samples from both domains. (b) Beginning of the training procedure with a large reconstruction error. (c) Distance-based cost function for unpaired translation. (d) Correct reconstruction to domain A for sample $X_A^{(i)}$ after correcting for the previous error. (e) Translation for $A \to B$ samples in X_A.

erative Adversarial Network (MSGAN) [Mao et al., 2019] achieve state-of-the-art realism in image translation by optimizing for cycle-consistency in the bottlenecks of the generators. In other words, these DNNs are trained by not only by minimiz-



(a)

(b)

Figure 2.12. Example of GAN architecture based on Cycle-Consistency. Traditionally, two generators ($G_{A\to B}$ and $G_{B\to A}$) and two discriminators (D_A and D_B) are trained in order to achieve unsupervised image translation and an objective loss can be devised by comparing the pairs $\{X_A, X_{A\to B\to A}\}$ and $\{X_B, X_{B\to A\to B}\}$.

ing the expectations $\mathbb{E}\left\|X_A^{(i)} - X_{A\to B\to A}^{(i)}\right\|$ and $\mathbb{E}\left\|X_B^{(i)} - X_{B\to A\to B}^{(i)}\right\|$, but also in the bottleneck activations of $G_{A\to B}$ and $G_{B\to A}$. One important property of this representation is that it forms an isomorphism between A and B, which is explored by CoDAGANs, as further explained in Section 3.

Generative networks for Unsupervised I2I Translation – specifically UNIT [Liu et al., 2017], MUNIT [Huang et al., 2018], DRIT/DRIT++ [Lee et al., 2018, 2020] and MSGAN [Mao et al., 2019] – closely resemble the architecture from Figure 2.13. This scheme represents the generator $G_{A\to B}$ performing a translation between a sample X_A to a synthetic version of this sample in domain B ($X_{A\to B}$), with translations $B \to A$ being analogous. While the number of each kind of block is a distinct hyperparameter to be tuned in these networks, Figure 2.13 depicts an Encoder-Decoder composed of two downsampling blocks ($G_1^{(\downarrow)}$ and $G_2^{(\downarrow)}$), two residual blocks ($G^{(\mathcal{R})_1}$ and $G^{(\mathcal{R})_2}$) and, finally, two upsampling blocks ($G_1^{(\uparrow)}$ and $G_2^{(\uparrow)}$). In this example, both the input X_A and output $X_{A \to B}$ of the generator are tensors with dimensions $n_{ch} \times 256 \times 256$. Downsampling layers halve the spatial resolution of the input and increase the channel depth of the input, while their upsampling counterparts do the exact opposite, doubling the spatial resolution at each step and decreasing the number of channels back to the original n_{ch} value. Residual layers [He et al., 2016b], do not alter the resolution of the input tensor, instead providing more trainable parameters to the DNN – further increasing its representation capabilities – while still providing shortcuts to the backpropagation gradients, as discussed in the beginning of Section 2. This architectural design is shown to be highly effective for Unsupervised I2I Translation, serving as a basis for several state-of-the-art networks that perform this task [Liu et al., 2017; Huang et al., 2018], while most other unsupervised translation methods rely on slight variations of this model [Li and Wand, 2016; Zhu et al., 2017a; Lee et al., 2018; Liu et al., 2019; Mao et al., 2019; Lee et al., 2020].

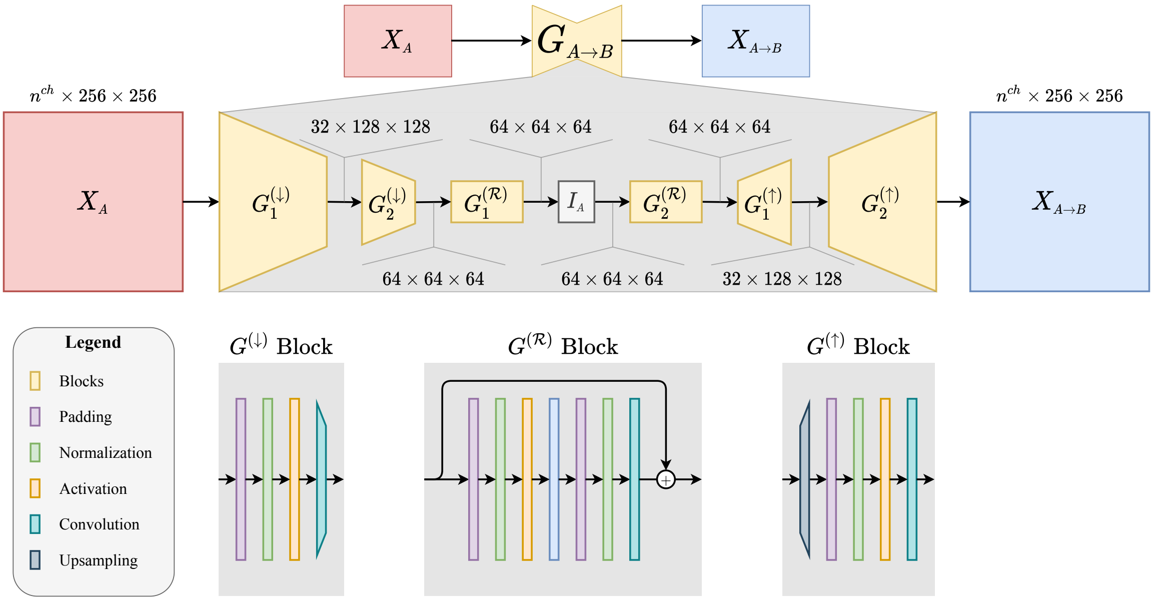


Figure 2.13. Detailed schematics of a traditional Encoder-Decoder architecture employed on Unsupervised I2I Translation. The generator $G_{A \to B}$ is broken down into its downscaling ($G^{(\downarrow)}$), residual ($G^{(\mathcal{R})}$) and upscaling ($G^{(\uparrow)}$) blocks. These blocks are subsequently broken down into their unitary components: Padding (i.e. Reflection, Replication or Zero Padding), Normalization (i.e. Instance Normalization, Batch Normalization, Adaptive Instance Normalization (AdaIN), etc), Activation (i.e. ReLU, LeakyReLU, Parametric Rectified Linear Unit (PReLU), Scaled Exponential Linear Unit (SELU), etc), Conv2d layers and upsampling.

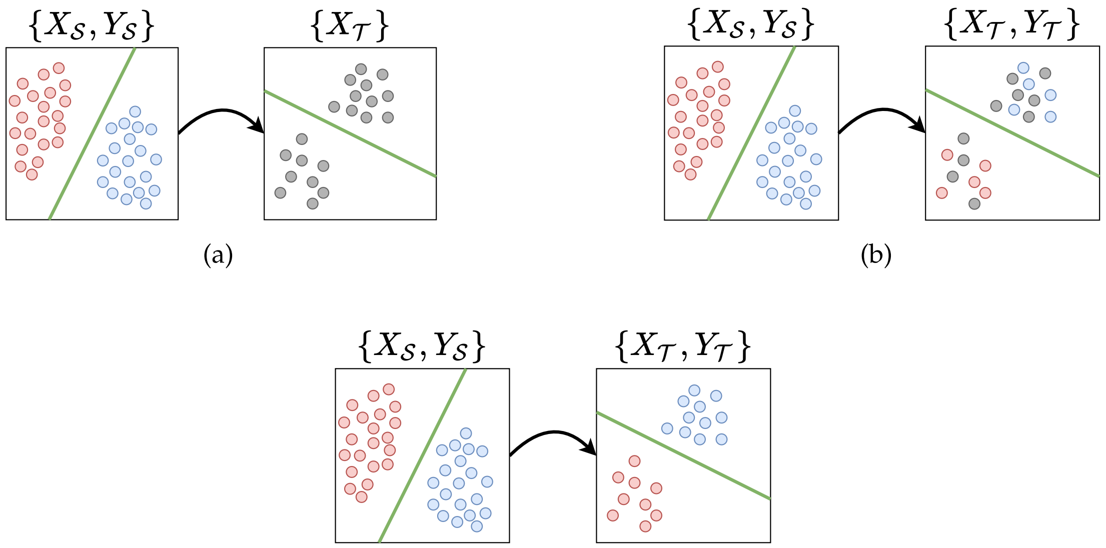
Some efforts have been spent in proposing Unpaired Image Translation GANs for multi-domain scenarios, as the case of StarGANs [Choi et al., 2018, 2020], but these networks do not explicitly present isomorphic representations of the data, as UNIT and MUNIT architectures do. Other advantages of UNIT and MUNIT over StarGANs is that they also compute reconstruction losses on the isomorphic representations, beside the traditional Cycle-Consistency between real and reconstructed images. CoDAGANs were built to be agnostic to the image translation network used as basis for the implementation, being able to transform any Image Translation GAN that has an isomorphic representation of the data into a multi-domain architecture with only minor changes to the generator and discriminator networks.

2.4 Domain Adaptation

DNNs often require a large amount of labeled training data in order to converge properly for performing supervised tasks in visual domains, such as classification [Krizhevsky et al., 2012], detection [Ren et al., 2015] and segmentation [Long et al., 2015; Ronneberger et al., 2015; Badrinarayanan et al., 2017]. Due to this hunger for data, Transfer Learning has become a common procedure and received unprecedented attention in the realm of Deep Learning research, mainly using fine-tuning for adapting DNNs pretrained in larger datasets to perform similar tasks in smaller datasets. The larger set is usually a massive database, such as ImageNet [Deng et al., 2009] and is called the source dataset, while the smaller set is called the target dataset, being composed of the samples from the domain upon which inference will be performed.

Examples of UDA, SSDA and FSDA can be seen in Figure 2.14. These scenarios only show cases of fully labeled source domains composed of data ($X_{\mathcal{S}}$) and labels ($Y_{\mathcal{S}}$). In UDA scenarios, no labels $Y_{\mathcal{T}}$ are available for the target set, while SSDA tasks have both labeled and unlabeled samples on the target domain. FSDA contains only labeled data in the target domain and it is the most common practice nowadays among deep DA methods due to the simplicity of fine-tuning pretrained DNNs to perform new tasks. Computer Vision-related domains have a lot to benefit from fine-tuning, as most off-the-shelf large labeled datasets are from competitions for traditional Computer Vision tasks [Deng et al., 2009; Everingham et al., 2015; Lin et al., 2014].

DA from Computer Vision datasets [Deng et al., 2009; Everingham et al., 2015; Lin et al., 2014] to biomedical image datasets can lead to a phenomenon known



$\{X_\mathcal{S}, Y_\mathcal{S}\}$
$\{X_\mathcal{T}\}$
$\{X_\mathcal{S}, Y_\mathcal{S}\}$
$\{X_\mathcal{T}, Y_\mathcal{T}\}$
$\{X_\mathcal{S}, Y_\mathcal{S}\}$
$\{X_\mathcal{T}, Y_\mathcal{T}\}$

(a)

(b)

(c)

Figure 2.14. Examples of UDA (a), SSDA (b) and FSDA (c) in a classification scenario. $X_\mathcal{S}$ and $Y_\mathcal{S}$ are, respectively, the source dataset data and labels, while $X_\mathcal{T}$ and $Y_\mathcal{T}$ represent the target dataset data and labels. The green line represents a possible decision boundary between the classes.

as Negative Transfer [Kuzborskij and Orabona, 2013], wherein, instead of helping the training of a model on the target set, the knowledge from the source set makes it harder to perform inference on the target. Poor knowledge transfer is known to happen mainly between domains or tasks that have large domain shifts. The semantic proximity between domains can also be explored in the opposite way to discover pairs of tasks that are likely to result in better knowledge transfer. A more precise definition for the domain shift between domains has been tackled both in a theoretical sense by Kuzborskij and Orabona [2013] and in empirical studies [Ferreira et al., 2018; Zamir et al., 2018].

Zhang et al. [2017] describes a taxonomy for DA tasks comprising most of the spectrum of deep and shallow knowledge transfer techniques. This taxonomy comprises several classes of problems with variations in feature and label spaces between source and target domains, data labeling, balanced/unbalanced data and sequential/non-sequential data.

CoDAGANs cannot be put in one single category in the taxonomy proposed by Zhang et al. [2017], as they allow for a dataset to be source and target at the same time and are adapted for UDA, SSDA and FSDA, being able to learn from both unsupervised and supervised data. CoDAGANs can also be seen as a form of

Generalization Learning, as using several data sources leads to more generalizable models, as discussed in Section 5.2.4.

2.4.1 Domain Generalization

Traditional DA techniques perform knowledge transfer between a single pair of datasets: a source S and a target T datasets. In many cases it is advantageous to acquire as much data as possible from multiple sources, mainly when there is a lack of labels. Multi-source methods [Sun et al., 2011; Gong et al., 2013; Caseiro et al., 2015; Ming Harry Hsu et al., 2015; Fang et al., 2013] try to infer a joint probability distribution $p_{X_1,X_2,...,X_N}$ from a multitude of source data $X_1, X_2, ..., X_N$, each one with its own marginal probability distribution $p_{X_1}, p_{X_2}, ..., p_{X_N}$. These methods must infer joint distributions for the domains based only on the marginal distributions of the source data.

CoDAGANs can be classified as a multi-source and multi-target DA method with the caveat that the distinction between source and target data is not clear in these DNNs, as translations and knowledge transfer are performed across all pairs of domains. As pointed by Csurka [2017]; Zhang et al. [2017], Domain Generalization is closely related to multi-source DA, as the objective is often to average the knowledge obtained from related source domains in order to make the model more robust to novel unseen data that it not available during training. Most Domain Generalization methods in the literature are based on this premise [Ghifary et al., 2015; Gan et al., 2016; Ding and Fu, 2017; Li et al., 2019; Carlucci et al., 2019], including CoDAGANs. Image-to-Image Translation for DA is further discussed in the Section 2.4.3.

2.4.2 Dense Visual Domain Adaptation

As described by Patel et al. [2015], algorithms for segmentation, reconstruction, and tracking are awaiting mechanisms that do not yet exist to be adapted toward emerging new domains. Due to the lack of methods for adaptation of tasks other than classification, another survey [Shao et al., 2015] that compiled the advances of DA in Computer Vision did not even mention segmentation tasks. Even though these studies are considerably outdated due to DNNs becoming ubiquitous in Computer Vision during the last years, this context did not change since these surveys, and, therefore, methods for deep DA in non-classification scenarios are still noticeably scarce.

Wang and Deng [2018] compiled so far the only up-to-date survey on visual DA containing methods specifically designed for semantic segmentation tasks. As argued by Wang and Deng [2018], only a few works address adaptation beyond classification and recognition, such as object detection, face recognition, semantic segmentation and person re-identification. How to achieve these tasks with no or a very limited amount of data is probably one of the main challenges that should be addressed by deep DA in the next few years.

Another recent survey [Csurka, 2017] compiles some approaches for performing visual DA in dense tasks. As far as the authors are aware, the few proposed approaches for deep DA in dense labeling tasks have been tackled mainly using synthetic data for specific problems, such as outdoor scene segmentation [Ros et al., 2016; Richter et al., 2016], depth estimation [Eigen et al., 2014; Bousmalis et al., 2017] and indoor scene understanding [Papon and Schoeler, 2015; Handa et al., 2016]. Although useful in some scenarios, these are not universal schemes for either UDA or SSDA in dense labeling tasks, as the application of these techniques depends on the availability of synthetic data corresponding to the real-world data of the target task.

2.4.3 I2I Translation for Domain Adaptation

Since the introduction of Image-to-Image Translation GANs, several works [Liu and Tuzel, 2016; Bousmalis et al., 2017; Hoffman et al., 2018; Murez et al., 2018; Wu et al., 2018; Zou et al., 2018] have used these architectures to perform DA between image domains. In the following paragraphs, when available, we will mainly focus on the experiments of the literature in dense labeling tasks.

As far as the authors are aware, the first use of I2I Translation specifically for Domain Adaptation was shown by CoGANs [Liu and Tuzel, 2016]. This work showed UDA for digit classification between the MNIST [LeCun et al., 1998] and USPS [Hull, 1994] datasets. While MNIST contains well-behaved, preprocessed and high-contrast handwritten digit samples in grayscale, USPS mimics a real-world scenario for digit classification using RGB images on noisy and highly varied backgrounds. Thus, being able to adapt a digit classifier from MNIST to USPS without using labels from the target set is a challenging problem. One should notice that CoGANs still did not present UDA results in dense labeling tasks.

With time, other works focused on using I2Is for dense labeling scenarios in either Computer Vision [Hoffman et al., 2018; Murez et al., 2018; Wu et al., 2018; Zou et al., 2018] or Biomedical Imaging [Cohen et al., 2018; Tang et al., 2019b,a; Yang et al., 2019] applications, most of them borrowing from the theoretical framework of CycleGANs [Zhu et al., 2017a]. Most of these methods were proposed concurrently to one of the main contributions of this work and, therefore, will be further discussed in Section 3.1.

Methods for DA using Cycle-Consistency [Hoffman et al., 2018; Murez et al., 2018; Wu et al., 2018; Zou et al., 2018] usually attach some fully convolutional architecture the end of a CycleGAN's generator (or other Unsupervised I2I architectures), as shown in Figure 3.1, limiting them to adapting between a pair of source and target domains $\{\mathcal{S}, \mathcal{T}\}$. One should notice in this base architecture that in the case of total lack of target labels Y_T – that is, in a UDA scenario – semantic consistency gradients are successfully fed to $G_{S \to T}$ due to its proximity to M_T, but very small gradient intensities flow from M_T to $G_{T \to S}$ in $S \to T \to S$ translations (Figure 3.1(a)). This represents an imbalance in the training of $G_{S \to T}$ and $G_{T \to S}$, which is not desirable for DA.

Cycle-Consistent Adversarial Domain Adaptation (CyCADA) [Hoffman et al., 2018] was built upon CycleGANs to perform UDA in dense labeling tasks – more specifically semantic segmentation. As most other papers in the area, CyCADA relies on synthetic data from realistic 3D simulations such as third person games to acquire labeled data for outdoor scene classification. It is much less time-consuming to annotate synthetic images from these simulations in an automated or semi-automated manner than to label entire datasets from scratch with pixel-level annotations, such as Pascal VOC [Everingham et al., 2015].

Similarly to the basic scheme presented in Figure 3.1, CyCADA uses a pair of generators ($G_{S \to T}$ and $G_{T \to S}$), a pair of discriminators (D_S and D_T) and a supervised model M_S trained on the source distribution S for performing UDA in dense labeling tasks. From an architectural point of view, the main distinctions between Figure 3.1 and CyCADA are twofold:

1. a couple of supervised models (M_S and M_T) are trained instead of only M_T, with M_S encouraging semantic consistency between $M_S(X_S)$ and $M_S(X_{S \to T})$, forcing the translations to be semantically consistent, while M_T in fact transfers the knowledge between S and T;

2. the addition of another discriminator D_{feat} for enforcing consistency between the segmentation predictions obtained from the real target image ($M_T(X_T)$) and the synthetic translated source image ($M_T(X_{S \to T})$).

The loss for CyCADA ($\mathcal{L}_{CyCADA}$) is a combination of three supervised losses $\mathcal{L}_{sup}$ (Equation 2.1); three adversarial losses (Equations 2.5 and 2.4) for enforcing that $\mathbb{E}(X_T) \approx \mathbb{E}(X_{S\rightarrow T})$, $\mathbb{E}(X_S) \approx \mathbb{E}(X_{T\rightarrow S})$ and $\mathbb{E}(f_T(X_T)) \approx \mathbb{E}(f_T(X_{S\rightarrow T}))$; and two Cycle-Consistent loss $\mathcal{L}_{cyc}$ (Equation 2.9) for ensuring that $X_S \approx X_{S\rightarrow T\rightarrow S}$ and $X_T \approx X_{T\rightarrow S\rightarrow T}$. $\mathcal{L}_{CyCADA}$ is given by the following equation:

$$\begin{aligned}\mathcal{L}_{CyCADA} = {} & \mathcal{L}_{sup^{(1)}}(M_T(X_{S\rightarrow T}), Y_S) \\ & + \mathcal{L}_{sup^{(2)}}(M_S(X_{S\rightarrow T}), M_S(X_T)) \\ & + \mathcal{L}_{sup^{(3)}}(M_S(X_{T\rightarrow S}), M_S(X_S)) \\ & + \mathcal{L}_{adv^{(1)}}(G_{S\rightarrow T}, X_T) \\ & + \mathcal{L}_{adv^{(2)}}(G_{T\rightarrow S}, X_S) \\ & + \mathcal{L}_{adv^{(3)}}(f_T(X_{S\rightarrow T}), f_T(X_T)) \\ & + \mathcal{L}_{cyc^{(1)}}(X_T, X_{T\rightarrow S\rightarrow T}) \\ & + \mathcal{L}_{cyc^{(2)}}(X_S, X_{S\rightarrow T\rightarrow S}). \end{aligned} \tag{2.10}$$

CyCADA reports successful UDA results between the synthetic GTA5 [Richter et al., 2016] dataset and the real-world CityScapes dataset [Cordts et al., 2016]. CyCADA reports mIoU results of 35.4%, frequency weighted Intersection over Union (fwIoU) of 73.8% and Pixel Accuracy of 83.6% in translations between GTA5→CityScapes. Several works improved on CyCADA by plugging a semantic segmentation DNN at one end of an Unpaired I2I Translation network [Murez et al., 2018; Wu et al., 2018], achieving comparable results on Computer Vision datasets.

Similarly to CyCADA and the architecture presented in Figure 3.1, I2IAdapt [Murez et al., 2018] uses CycleGANs coupled with segmentation architectures to perform UDA for dense labeling tasks. Again the GTA5 and CityScapes datasets are used as source and target data in I2IAdapt, comparing the results with simply testing the pretrained DNN in the target domain, yielding considerable improvements. Their best configuration with a DenseNet [Huang et al., 2017] backbone achieves 35.7% of mIoU on CityScapes.

The Dual Channel-wise Alignment Network (DCAN) [Wu et al., 2018] also follows close architectural choices to CyCADA and I2IAdapt, attaching a segmentation architecture to the target end of a translation architecture. DCAN was trained on two synthetic datasets (GTA5 and SYNTHIA [Ros et al., 2016]) and in one real-world dataset (CityScapes). Wu et al. [2018] report mIoU values of 38.9% for GTA5→CityScapes and 41.7% for SYNTHIA→CityScapes, surpassing the baselines by between 8% and 9% and other similar methods by a small percentage.

Table 2.1. Comparison between D2D, CoDAGAN and the main baselines in the literature. Methods are clustered into 4 distinct categories: Image Pairing (Paired or Unpaired), Pairwise (Domain-to-Domain) or Variable Training, Domain (Computer Vision – CV or Biomedical Images – BI) and DA Task Labeling (Sparse, Dense or Not Applicable).

Method	Pairing	Training	Domain	DA Labeling
pix2pix [Isola et al., 2017]	Paired	Pairwise	CV	–
CycleGAN [Zhu et al., 2017a]	Unpaired	Pairwise	CV	–
CoGAN [Liu and Tuzel, 2016]	Unpaired	Pairwise	CV	Sparse
UNIT [Liu et al., 2017]	Unpaired	Pairwise	CV	–
MUNIT [Huang et al., 2018]	Unpaired	Pairwise	CV	–
StarGAN [Choi et al., 2018]	Unpaired	**Variable**	CV	–
StarGAN v2 [Choi et al., 2020]	Unpaired	**Variable**	CV	–
I2IAdapt [Murez et al., 2018]	Unpaired	Pairwise	CV	Dense
DCAN [Wu et al., 2018]	Unpaired	Pairwise	CV	Dense
CyCADA [Hoffman et al., 2018]	Unpaired	Pairwise	CV	Dense
Zhang et al. [2018b]	Unpaired	Pairwise	BI	Dense
XLSor [Tang et al., 2019b]	Unpaired	Pairwise	BI	Dense
TUNA-Net [Tang et al., 2019a]	Unpaired	Pairwise	BI	Dense
Yang et al. [2019]	Unpaired	Pairwise	BI	Dense
Ours (D2D)	Unpaired	Pairwise	BI	Dense
Ours (CoDAGAN)	Unpaired	**Variable**	BI	Dense

2.5 I2I Literature and Proposed Methods

According to the background knowledge presented in the current chapter, Chapter 3 presents the proposed methods for DA in biomedical dense labeling. Section 3.1 describes the first studies regarding pairwise DAs for dense labeling in radiology (D2Ds), Section 3.2 expĺains our proposal for Domain Generalization in biomedical image segmentation tasks (CoDAGANs) and, finally, Section 3.3 uses CoDAGANs to gather unlabeled volumetric tomographic data to transfer useful knowledge to 2D images in a step towards cross-modality DA. A concise comparison between the literature of I2I for DA, D2D and CoDAGANs can be seen in Table 2.1.

Chapter 3

Proposed Methods

This chapter discusses the three main contributions of this work: 1) a D2D approach (Section 3.1) used in the earlier tests of this work in order to encounter satisfactory architectures and loss components; 2) Section 3.2 describes CoDAGAN, a fully novel and label efficient framework that allows for Domain Generalization between a myriad of datasets from the same radiological modality; and 3) a novel pipeline for Domain Generalization in the task of rib segmentation in CXRs based on Conditional DA, described in Section 3.3.

3.1 I2I for Domain Adaptation

Even though there are several architectural differences among the distinct methods of Unpaired I2I Translation, the core of the idea of Cycle-Consistency is kept across most Unpaired Translation networks [Liu and Tuzel, 2016; Zhu et al., 2017a; Liu et al., 2017; Huang et al., 2018; Lee et al., 2018; Mao et al., 2019; Lee et al., 2020]. Specific architectures of $G_{S \to T}$, $G_{T \to S}$, D_S and D_T, as well as customly designed losses can grant different translation methods special characteristics such as different encodings for style and content in an image [Huang et al., 2018] and unsupervised multimodal translations [Mao et al., 2019].

With only simple modifications to the traditional Unsupervised I2I Translation pipeline (Figure 2.12), one can adapt the a DNNs as CycleGAN [Zhu et al., 2017a], UNIT [Liu et al., 2017] or MUNIT [Huang et al., 2018] in order to perform Cross-Dataset Transfer Learning. Let S be a labeled source dataset and T be a partially labeled or fully unlabeled target dataset. We propose the architecture shown in Figure 3.1 for transferring knowledge from S to T. The unsupervised part is simply an Unsupervised I2I Network, such as Zhu et al. [2017a]. The supervised section uses a model M_S pretrained on domain S to enforce discriminative translations by $G_{S \to T}$ and $G_{T \to S}$ – that is, translations from T to S that preserve the visual features important for the class discrimination in M_S. As shown in Figure 3.1, if there are any labels for the dataset T, they are also taken into account by the architecture, allowing for a better training of $G_{S \to T}$.

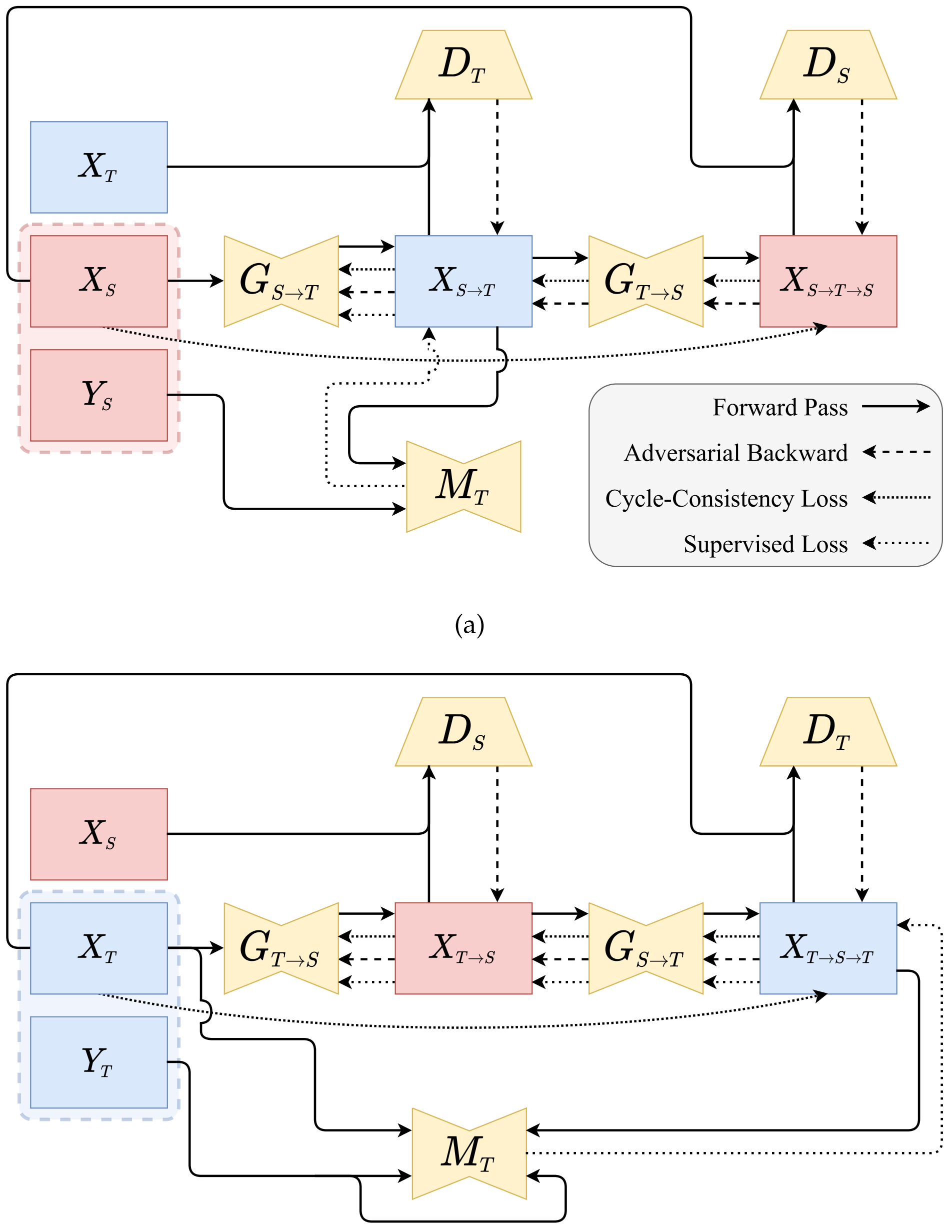
As shown, D2D simply combines the supervised learning from an FCN or an Encoder-Decoder architecture with a supervised or unsupervised image translation architecture to perform UDA or SSDA, attaching the pretrained supervised segmentation architecture at one end of the image translation.

Discriminative and generative models in GANs are trained intermittently. At first, the generators are frozen while both discriminators are trained simultaneously using backpropagation. Later the inverse occurs: the discriminative networks are frozen and both generators are trained at the same time. Our method adds a third optimization procedure to this pipeline, wherein M_A is fine-tuned and backpropagates the training errors to $G_{A \to B}$ and $G_{B \to A}$, while D_A and D_B are frozen. These training steps will be henceforth called generative, discriminative and supervised steps.

If convergence is met, it is possible to forward an image $x_b \sim X_B$ to $G_{B \to A}$, get its counterpart in $x_{b \to a}$ and forward it to M_A, as $G_{B \to A}$ was enforced to preserve the visual features important for the segmentation. If Y_B is nonexistent – indicating that X_B is fully unlabeled, only the Y_A labels are used in the supervised part of the network. Therefore, this architecture can use Y_A labels to train a model for X_B samples in a completely unsupervised setting. Contrary to fine-tuning, our method uses the whole X_B dataset to transfer the knowledge, not only the labeled samples in X_B.

As the proposed DA architecture is built on top of a generic Unpaired Image Translation architecture (Figure 2.12), it is agnostic to the choice of Cycle-Consistency network. That is, one could easily shift between implementations of CycleGANs [Zhu et al., 2017a], UNIT [Liu et al., 2017] or MUNIT [Huang et al., 2018]. D2D simply added a supervised loss to the already existing unsupervised loss components for I2I in the original architectures. The method for Conditional DAs discussed in Section 3.2 directly altered the loss terms in order to mitigate stability concerns in D2D and enforce dataset agnosticism in the intermediate $\mathcal{I}$ representation, wherein supervision is applied in that scheme.

The D2D architecture was the basis for most of the exploratory tests that resulted in the migration to Conditional DA.



(a)

(b)

Figure 3.1. Simplified scheme for D2D in translations $S \to T \to S$ (a) and $T \to S \to T$ (b). A supervised model M performs the supervised learning by using the labels Y_S in the source domain S and, in the case of FSDA or SSDA, also using the target domain labels Y_T, when available.

3.1.1 Limitations of Pairwise I2I for DA

Pairwise Image Translation for DAs presented limitations that would prevent the method to maximize its label efficiency. For instance, in the task of lung segmentation in toracic radiographs there are four labeled large scale datasets that could have labels used for training. However, D2D only allows for one of the datasets to be used as source and another unlabeled dataset to serve as target to the translation. This was the main motivation to the development of conditional dataset encoding (Section 3.2.1) in the more recent iteration of this work: CoDAGANs.

CoDAGANs (further discussed in Section 3.2) apply a similar framework to D2D in order to perform UDA, SSDA and FSDA, mixing the unsupervised learning of Cycle-Consistent GANs with the supervised pixelwise learning of an Encoder-Decoder architecture. However, two crucial distinctions between D2D and CoDAGANs must be addressed, though: 1) only one Encoder, one Decoder and one Discriminator are used in the image translation process, as different domains are recognized by G and D via conditional encoding, allowing for multi-target domain adaptation; 2) supervised learning is performed only on the bottleneck of G, not in end of the translation process, allowing all domains to share a single isomorphic space $\mathcal{I}$. These differences allow for drawing supervised and unsupervised knowledge from several distinct datasets, depending on their label availability.

3.2 Deep Conditional DA

CoDAGANs combine unsupervised and supervised learning to perform UDA, SSDA or FSDA between two or more image sets. These architectures are based on adaptations of preexisting Unsupervised I2I Translation networks [Zhu et al., 2017a; Liu et al., 2017; Huang et al., 2018], adding supervision to the process in order to perform Transfer Learning. The generator networks (G) in Image Translation GANs are implemented usually using Encoder-Decoder architectures as U-Nets [Ronneberger et al., 2015]. At the end of the Encoder ($G_{\mathbb{E}}$) there is a middle-level representation $\mathcal{I}$ that can be trained to be isomorphic in these architectures. $\mathcal{I}$ serves as input of the Decoder ($G_{\mathbb{D}}$). Isomorphism allows for learning a supervised model M based on $\mathcal{I}$ that is capable of inferring over several datasets. This unsupervised translation process followed by a supervised learning model can be seen in Figure 3.2.

For this work we employed the UNIT and MUNIT architectures as a basis for the generation of $\mathcal{I}$. On top of that, we added the supervised model M – which is based on a U-Net [Ronneberger et al., 2015] – and made some considerable changes to the translation approaches, mainly regarding the architecture and conditional distribution modelling of the original GANs, as discussed in Section 3.2.1. The exact architecture for G depends on the basis translation network chosen for the adaptation. In our case, both UNIT and MUNIT use VAE-like architectures [Kingma and

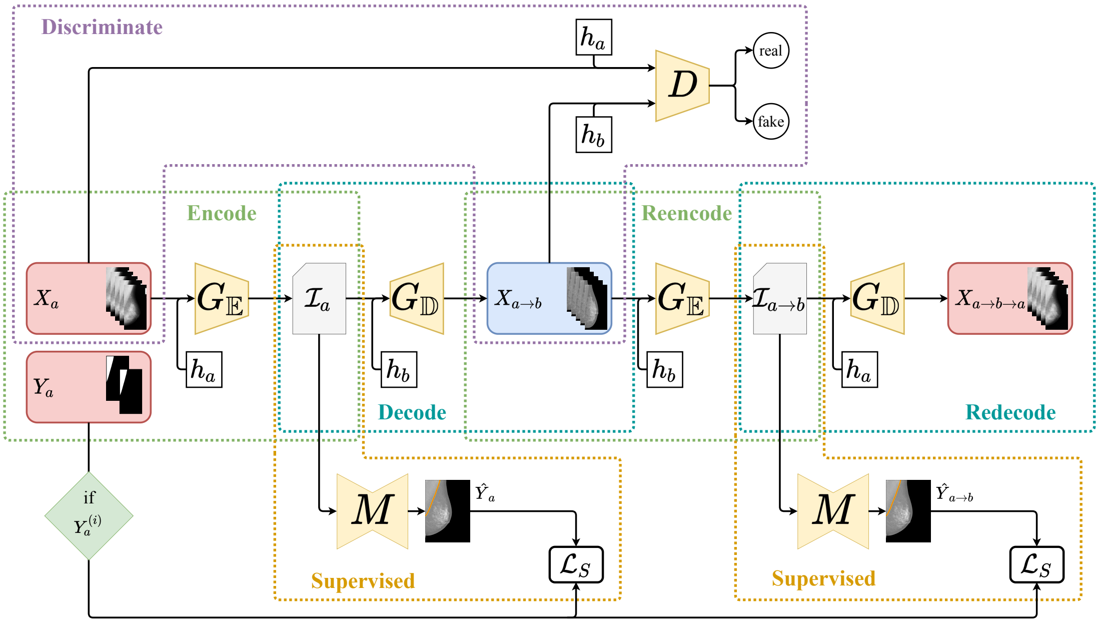


Figure 3.2. Training procedure for CoDAGANs. This figure exemplifies a translation $a \to b \to a$, but the translation $b \to a \to b$ is analogous. Notice that the reconstruction losses are omitted from this view of our architecture for simplification. The *Encode* routine transforms the real images in the mini-batch X_a into the isomorphic representation $\mathcal{I}_a$ between the datasets (through $G_{\mathcal{E}}$), followed by the *Decode* subroutine, which builds (using $G_{\mathcal{D}}$) a corresponding fake mini-batch $X_{a\to b}$ according to $\mathcal{I}$. The *Reencode* procedure reconstructs the isomorphic representation $\mathcal{I}$ according to $X_{a\to b}$. At last, the *Redecode* subroutine reconstructs the image $X_{a\to b\to a}$ according to $\mathcal{I}_{a\to b}$. The *Discriminate* subroutine tries to discern between real (X_a) and synthetic ($X_{a\to b}$) samples from the datasets. If there is a ground truth $Y_a^{(i)}$ for the sample i in the mini-batch, the model M compares the predicted segmentation $\hat{Y}_a$ with the ground truth Y_a generated by the two encoding subroutines.

Welling, 2013] for G, containing downsampling ($G_{\mathbb{E}}$), upsampling ($G_{\mathbb{D}}$) and residual layers.

The shape of $\mathcal{I}$ depends on the architecture choice for G. UNIT, for example, assumes a single latent space between the image domains, while MUNIT separates the content of an image from its style. CoDAGANs feeds the whole latent space to the supervised model when it is based on UNIT and only content information when it is built upon MUNIT, as the style vector has no spatial resolution and as we intend to ignore style and preserve content.

A training iteration on a CoDAGAN follows the sequence presented in Figure 3.2. The generator network G – similarly to U-Nets [Ronneberger et al., 2015] and AEs [Kingma and Welling, 2013] – is an Encoder-Decoder architecture. However, instead of mapping the input image into itself or into a semantic map as its Encoder-Decoder counterparts, it is capable of translating samples from one image dataset into synthetic samples from another dataset. The encoding half of this architecture ($G_{\mathbb{E}}$) receives images from the various datasets and creates an isomorphic representation somewhere between the image domains in a high dimensional space. This code will be henceforth described as $\mathcal{I}$ and is expected to correlate important features in the domains in an unsupervised manner [Liu and Tuzel, 2016]. Decoders ($G_{\mathbb{D}}$) in CoDAGAN generators are able to read $\mathcal{I}$ and produce synthetic images from the same domain or from other domains used in the learning process. This isomorphic representation is an integral part of both UNIT [Liu et al., 2017] and MUNIT [Huang et al., 2018] translations, as they also enforce good reconstructions for $\mathcal{I}$ in the learning process. It also plays an essential role in CoDAGANs, as all supervised learning is performed on $\mathcal{I}$.

As shown in Figure 3.2, CoDAGANs include five unsupervised subroutines: a) Encode, b) Decode, c) Reencode, d) Redecode and e) Discriminate; and two f) Supervision subroutines, which are the only labeled ones. These subroutines will be detailed further in the following paragraphs.

Encode: First, a dataset pair $\{a \sim \mathcal{D}, b \sim \mathcal{D}\}$ are sampled from the dataset distribution $p_{\mathcal{D}}$. A minibatch X_a of images from a is then appended to a code h_a generated by a One-Hot-Encoding scheme, aiming to inform the encoder $G_{\mathbb{E}}$ of the samples' source dataset. The 2-uple $\{X_a, h_a\}$ is passed to the encoder $G_{\mathbb{E}}$, producing an intermediate isomorphic representation $\mathcal{I}_a$ for the input X_a according to the marginal distributions computed by $G_{\mathbb{E}}$ for dataset a.

Decode: The information flow is then split into two distinct branches: 1) $\mathcal{I}_a$ is fed to the supervised model M; 2) $\mathcal{I}_a$ is appended to a code h_b and passed through the decoder $G_{\mathbb{D}}$ conditioned to dataset b. The function $G_{\mathbb{D}}(\mathcal{I}_a, h_b)$ produces $X_{a \to b}$, which is a translation of images in the minibatch X_a with the style of dataset b.

Reencode: The Reencode procedure performs the same operation of generating an isomorphic representation as the Encode subroutine, but receiving as input the synthetic image $X_{a \to b}$. More specifically, the reencoded isomorphic representation $\mathcal{I}_{a \to b}$ is generated by $G_{\mathbb{E}}(X_{a \to b}, h_b)$.

Redecode: Once again the architecture splits into two branches: 1) $\mathcal{I}_{a \to b}$ is passed to M in order to produce the prediction $\hat{Y}_{a \to b}$; 2) the isomorphic representation is decoded as in $G_{\mathbb{D}}(\mathcal{I}_{a \to b}, h_b)$, producing the reconstruction $X_{a \to b \to a}$, which can be compared to X_a via a Cycle-Consistency loss $\mathcal{L}_{cyc}$ (Equation 2.9).

Discriminate: At the end of Decode, the synthetic image $X_{a \to b}$ is produced. The original samples X_a and the translated images $X_{a \to b}$ are merged in a single batch and passed to D, which uses the adversarial loss component $\mathcal{L}_{adv}^{D}$ (Equation 2.7) in order to classify between real and synthetic samples. In Routines when the generators are being updated instead of the discriminators, the adversarial loss $\mathcal{L}_{adv}^{G}$ (Equation 2.6) is computed instead.

Supervision: At the end of Encode and Reencode subroutines, for each sample $X_a^{(i)}$ which has a corresponding label $Y_a^{(i)}$, the isomorphisms $\mathcal{I}_a^{(i)}$ and $\mathcal{I}_{a\rightarrow b}^{(i)}$ are both fed to the same supervised model M. The model M perform the desired supervised task, generating the predictions $\hat{Y}_a^{(i)}$ and $\hat{Y}_{a\rightarrow b}(i)$. Both these predictions can be compared in a supervised manner to $Y_a^{(i)}$ by using $\mathcal{L}_S$ (Equation 2.1), if there are labels for the image i in this minibatch. As there are always at least some labeled samples in this scenario, M is trained to perform inference on isomorphic encodings of both originally labeled data ($M(\mathcal{I}_a) = \hat{Y}_a \approx Y_a$) and data translated by the CoDAGAN for the style of other datasets ($M(\mathcal{I}_{a\rightarrow b}) = \hat{Y}_{a\rightarrow b} \approx Y_a$).

If domain shift is computed and adjusted properly during the training procedure, the properties $X_a \approx X_{a\rightarrow b\rightarrow a}$ and $\mathcal{I}_a \approx \mathcal{I}_{a\rightarrow b}$ are achieved, satisfying Cycle-Consistency and Isomorphism, respectively. After training, it does not matter which input dataset among the training ones is conditionally fed to $G_{\mathbb{E}}$ to the generation of isomorphism $\mathcal{I}$, as samples from all datasets should all belong to the same joint distribution in $\mathcal{I}$-space. Therefore any learning performed on $\mathcal{I}_a$ and $\mathcal{I}_{a\rightarrow b}$ is universal to all datasets used in the training procedure. Instead of performing only the translation $a \rightarrow b \rightarrow a$ for the randomly chosen datasets a and b, all mentioned subroutines are run simultaneously for both $a \rightarrow b \rightarrow a$ and $b \rightarrow a \rightarrow b$, as in UNIT [Liu et al., 2017] and MUNIT [Huang et al., 2018]. Translations $b \rightarrow a \rightarrow b$ are analogous to the $a \rightarrow b \rightarrow a$ case described previously.

One should notice that $G_{\mathbb{E}}$ performs spatial downsample, while $G_{\mathbb{D}}$ performs upsample, consequently the model M should take into account the amount of downsampling layers in $G_{\mathbb{E}}$. More specifically, we removed the first two layers of U-Net [Ronneberger et al., 2015] when using them as the model M, resulting in an asymmetrical U-Net to compensate for $G_{\mathbb{E}}$ downsamplings. The amount of input channels of M must also be compatible with the amount of output channels in $G_{\mathbb{E}}$. Another constraint for the architecture of the pair $\{G_{\mathbb{E}}, G_{\mathbb{D}}\}$ is that the upsampling performed by $G_{\mathbb{D}}$ should always compensate the downsampling factor of $G_{\mathbb{E}}$, characterizing G as a whole as a symmetric Encoder-Decoder network.

The discriminator D for CoDAGANs is basically the same as the discriminator from the original Cycle-Consistency network, that is, a basic CNN that classifies between real and fake samples. The only addition to D is conditional training in order for the discriminator to know the domain the sample is supposed to belong to. This allows D to use its marginal distribution for each dataset for determining the likelihood of veracity for the sample. It is important to notice that our model is agnostic to the choice of Unsupervised Image-to-Image Translation architecture, therefore future advances in this area based on Cycle-Consistency should be equally portable to perform DA and further benefit CoDAGAN's performance.

3.2.1 Conditional Dataset Encoding

Conditional dataset training allows CoDAGANs to process data and perform transfer from several distinct source/target datasets. Fully or partially labeled datasets act as source datasets for the method, while unlabeled data is used both to enforce isomorphism in $\mathcal{I}$ and to yield adequate image translations between domains. Partially labeled and unlabeled data are, therefore, the target datasets for in this architecture.

While D2D approaches use a coupled architecture composed of 2 encoders ($G_{\mathbb{E}_a}$ and $G_{\mathbb{E}_b}$) and 2 decoders ($G_{\mathbb{D}_a}$ and $G_{\mathbb{D}_b}$) for learning a joint distribution over datasets a and b, CoDAGANs use only one generator G composed of one encoder and one decoder ($G_{\mathbb{E}}$ and $G_{\mathbb{D}}$). Additionally to the data X_k from some dataset k, $G_{\mathbb{E}}$ is conditionally fed a One-Hot-Encoding h_k, as in $\mathcal{I} = G_{\mathbb{E}}(X_k, h_k)$. The addition of the data in X_k to the code h_k is achieved by simple concatenation, as shown in Figure 3.3. The code h_k forces the generator to encode the data according to the marginal distribution optimized for dataset k, conditioning the method to the visual style of these data, as exemplified in Figures 3.2 and 3.4. The code h_l for a second dataset l is received by the decoder, as in $\hat{X}_{k\to l} = G_{\mathbb{D}}(\mathcal{I}, h_l)$, in order to produce the translation $\hat{X}_{k\to l}$ to dataset l.

3.2.2 Training Routines in CoDAGANs

In each iteration of a traditional GAN there are two routines for training the networks: 1) freezing the discriminator and updating the generator (*Gen Update*); and 2) freezing the generator and updating the discriminator (*Dis Update*). Performing these routines intermittently allows the networks to converge together in unsupervised settings. CoDAGANs add a new supervised routine to this scheme in order to perform UDA, SSDA and FSDA: *Model Update*. The subroutines described in Section 3.2 that compose the three routines of CoDAGANs are presented in Table 3.1

Since the first proposal of GANs [Goodfellow et al., 2014], stability has been considered a major problem in GAN training. Adversarial training is known to be

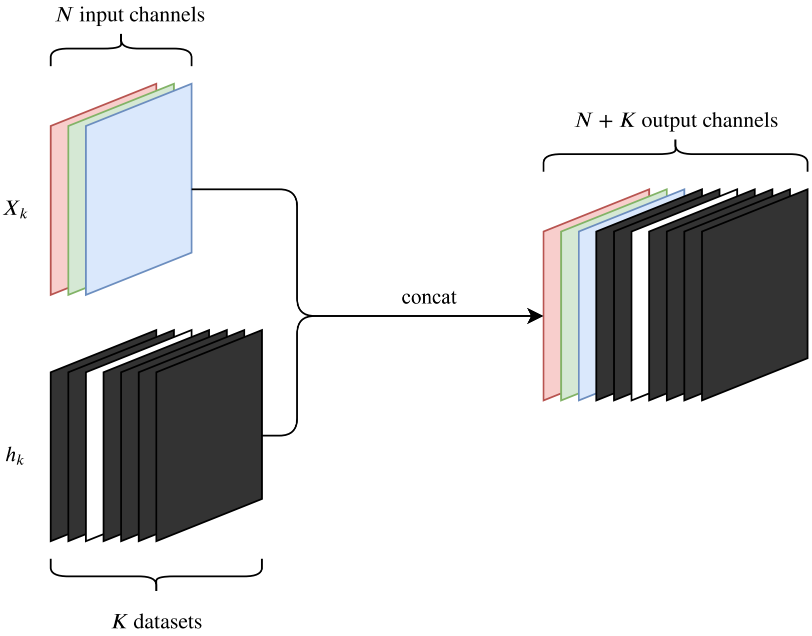


Figure 3.3. Illustration of One-Hot-Encoding on image channels in order to encode dataset information.

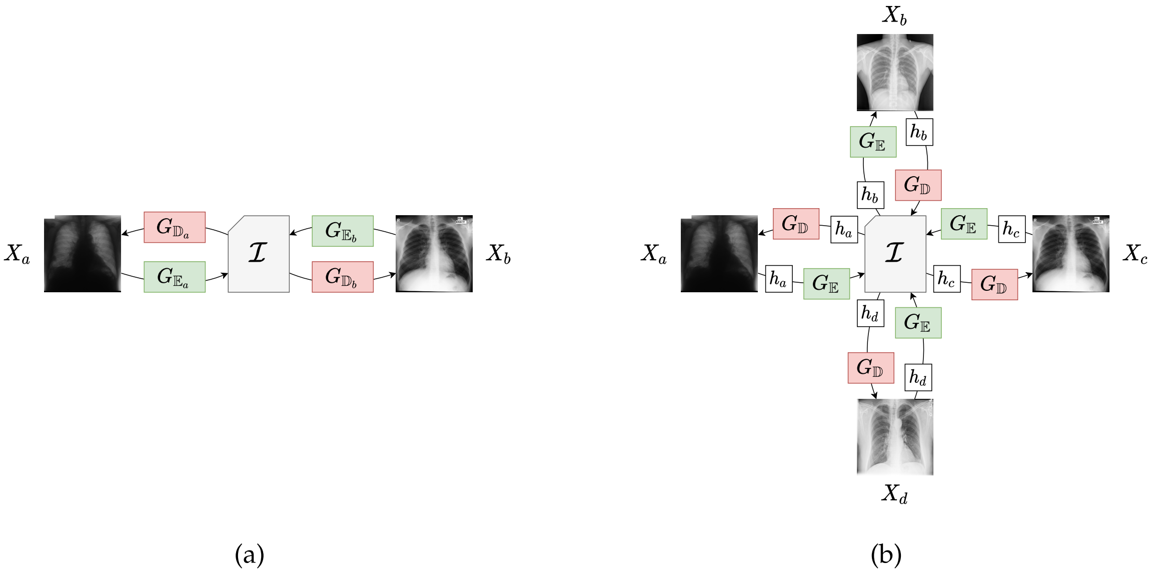


Figure 3.4. Comparison between D2D architectures and CoDAGANs regarding architectural choices for computing the isomorphic representation. While D2D use an Encoder/Decoder pair for each domain, CoDAGANs use One-Hot-Encoding in order to allow training with more than two domains without scalability hurdles.

more susceptible to convergence problems [Goodfellow et al., 2014; Salimans et al., 2016] than traditional training procedures for DNNs due to problems as: more com-

Table 3.1. Subroutines for each routine of CoDAGANs.

Routine / **Subroutine**	***G* Update**	***D* Update**	***M* Update**
Encode	✓	✓	✓
Decode	✓	✓	✓
Reencode	✓	X	✓
Redecode	✓	X	✓
Discriminate	X	✓	X
Supervision	X	X	✓

plex objectives composed of two or more (often contradictory) terms, discrepancies between the capacities of G and D, mode collapse etc. Therefore, in order to achieve more stable results, we split the training procedure of CoDAGANs into two phases: a) *Full Training* and b) *Supervision Tuning*; which will be explained on the following paragraphs.

Full Training During the first 75% of the epochs in a CoDAGAN training procedure, *Full Training* is performed. This training phase is composed of the procedures *Dis Update*, *Gen Update* and *Model Update*, executed in this order. That is, for each iteration in an epoch of the *Full Training* phase, first the discriminator D is optimized, followed by an update of G and finishing with the update of the supervised model. During this phase adversarial training enforces the creation of good isomorphic representations by G and translations between the domains. At the same time, the model M uses the existing (and potentially scarce) label information in order to improve the translations performed by G by adding semantic meaning to the translated visual features in the samples.

Supervision Tuning The last 25% of the network epochs are trained in the *Supervision Tuning* setting. This phase removes the unstable adversarial training by freezing G and performing only the Model Update procedure, effectively tuning the supervised model to a stationary isomorphic representation. Freezing G has the effect of removing the instability generated by the adversarial training in the translation process, as it is harder for M to converge properly while the isomorphic input $\mathcal{I}$ is constantly changing its visual properties due to changes in the weights of G.

3.2.3 CoDAGAN Loss

Both UNIT [Liu et al., 2017] and MUNIT [Huang et al., 2018] optimize conjointly GAN-like adversarial loss components and Cycle-Consistency reconstruction losses. Cycle-Consistency losses ($\mathcal{L}_{cyc}$) are used in order to provide unsupervised training capabilities to these translation methods, allowing for the use of unpaired image datasets, as paired samples from distinct domains are often hard or impossible to create. Cycle-Consistency is often achieved via Variational inference, which tries to find an upper bound to the Maximum Likelihood Estimation (MLE) of high dimensional data [Kingma and Welling, 2013]. Variational losses allow VAEs to generate new samples learnt from an approximation to the original data distribution as well as reconstruct images from these distributions. Optimizing an upper bound to the MLE allows VAEs to produce samples with high likelihood regarding the original data distribution, but still possessing low visual quality.

Adversarial losses ($\mathcal{L}_{adv}$) are often complementarily used with reconstruction losses in order to yield high visual quality and detailed images, as GANs are widely observed to take bigger risks in generating samples than simple regression losses [Isola et al., 2017]. Simpler approaches to image generation tend to average the possible outcomes of new samples, producing low quality images, therefore GANs produce less blurry and more realistic images than non-adversarial approaches in most settings. Unsupervised I2I Translation architectures normally use a weighted sum of these previously discussed losses as their total loss function ($\mathcal{L}_{tot}$), as in:

$$\begin{aligned} \mathcal{L}_{tot} =& \lambda_{cyc} \left[\mathcal{L}_{cyc}(X_a, X_{a \to b \to a}) + \mathcal{L}_{cyc}(X_b, X_{b \to a \to b}) \right] + \\ & \lambda_{adv} \left[\mathcal{L}_{adv}(X_b, X_{a \to b}) + \mathcal{L}_{adv}(X_a, X_{b \to a}) \right]. \end{aligned} \tag{3.1}$$

More details on UNIT and MUNIT loss components can be found in their respective original papers [Liu et al., 2017; Huang et al., 2018]. One should notice that we only presented the architecture-agnostic routines and loss components for CoDAGANs in the previous subsections, as the choice of Unsupervised I2I Translation basis network might introduce new objective terms and/or architectural changes. MUNIT, for instance, computes reconstruction losses to both the pair of images $\{X_a, X_{a \to b \to a}\}$ and the pair of isomorphic representations $\{\mathcal{I}_a, \mathcal{I}_{a \to b}\}$, which are separated into style and content components in this architecture.

CoDAGANs add a new supervised component $\mathcal{L}_{sup}$ to the completely unsupervised loss $\mathcal{L}_{tot}$ of Unsupervised Image-to-Image Translation methods. The supervised component for CoDAGANs is the default cost function for supervised classification/segmentation tasks, the Cross Entropy loss (Equation 2.1). The full objective

$\mathcal{L}_{CoDA}$ for CoDAGANs is, therefore, defined by:

$$\begin{aligned}\mathcal{L}_{CoDA} = \lambda_{cyc}[&\mathcal{L}_{cyc}(X_a, X_{a\rightarrow b\rightarrow a}) + \mathcal{L}_{cyc}(X_b, X_{b\rightarrow a\rightarrow b})] + \\ &\lambda_{adv}[\mathcal{L}_{adv}(X_b, X_{a\rightarrow b}) + \mathcal{L}_{adv}(X_a, X_{b\rightarrow a})] + \\ &\lambda_{sup}[\mathcal{L}_{sup}(Y_a, M(\mathcal{I}_a)) + \mathcal{L}_{sup}(Y_b, M(\mathcal{I}_b)) + \\ &\quad \mathcal{L}_{sup}(Y_a, M(\mathcal{I}_{a\rightarrow b})) + \mathcal{L}_{sup}(Y_b, M(\mathcal{I}_{b\rightarrow a}))], \end{aligned} \tag{3.2}$$

with the values for λ_{cyc}, λ_{adv} and λ_{sup} empirically set to 10, 1 and 1, respectively.

3.3 Rib Segmentation from Synthetic Data

Previous works have explored the larger amount of information encoded into 3D volumetric radiographs, however they either relied on low-level image processing techniques [Candemir et al., 2016] or 3D labels from the original data [Zhang et al., 2018b]. These schemes are prone to limitations in their generalization capabilities, as they require either a large manual fine-tuning scheme for the methodologies' hyperparameters for each new dataset or are limited to the variability in the training data. Based on the CoDAGAN framework, we devised a novel pipeline for rib segmentation using labels from CT-scans that is able to perform UDA for novel unlabeled data and achieve Domain Generalization in the task of rib segmentation in 2D radiology samples.

In order to extract useful unsupervised knowledge for 2D images such as CXRs from volumetric CT-scans, the proposed pipeline for rib segmentation begins with two operations for flattening 3D volumes into 2D planes: Average Intensity Projection (Average Intensity Projection) and Maximum Intensity Projection (MIP) in the PA axis of CT images; resulting in the images X_A and Y_A^{Bone} respectively. AIP is done by averaging all pixels in a certain location across all the PA axis, while MIP applies the max operation to this same pixel column. These flattening procedures were previously observed by the literature [Candemir et al., 2016; Zhang et al., 2018b] to generate useful 2D representations that could be compiled into knowledge for CXRs, especially for delineating anatomical structures such as bones and organs. Our pipeline explicitly enforces Domain Generalization and performs semantic segmentation using DNNs, according to the scheme presented in Figure 3.5. The generation of DRR samples by AIP of the PA axis is delineated in green in the pipeline.

Bone masks generated by the max operation on the CT-scans (Y_A^{Bone}) yield an

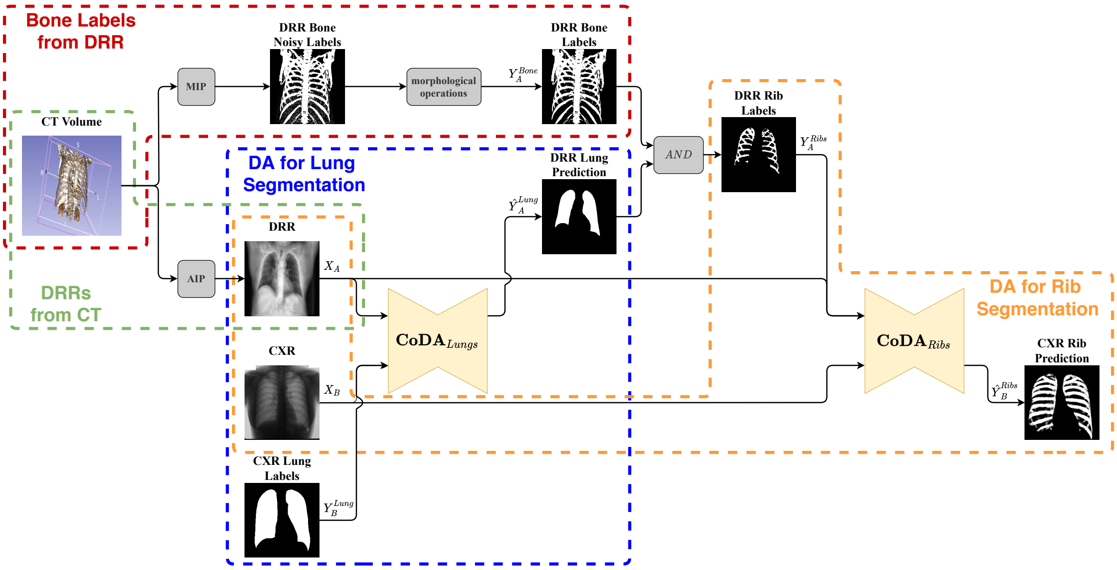


Figure 3.5. Proposed rib segmentation pipeline. There are four submodules highlighted in the image: 1) the procedure for acquiring bone labels from CT-scans in red; 2) average flattening in the Posterior-Anterior (PA) axis to produce highlighted in green; 3) Conditional DA for DRR lung segmentation from CXR labels delineated in blue; and 4) CoDAGAN for segmenting CXR ribs in orange.

acceptable yet noisy segmentation of the bones in the resulting DRR. Simple morphological filtering was observed to fix the noise introduced by the max operation in the labels. The computation of bone labels from the max operations in CTs volumes can be seen delineated in red in Figure 3.5. As bones and other natural/artificial structures prominently appear in DRRs when flattening is done using the max operation, undesirable objects as scapula and humerus bones, other anatomical features, and even pacemakers are often present as False Positives in the label maps acquired for the DRRs. These artifacts are often located outside of the lung field area in the 2D projection of the DRRs, implying that an efficient lung segmentation could remove most of them from the training label set. Thus, in order to filter all these undesirable artifacts from the labels, we first used CoDAGANs to perform UDA for lung field segmentation from labeled and unlabeled CXR datasets to DRRs, as highlighted in blue in Figure 3.5. These networks yielded semantic prediction maps for the lung pixels and allowed us to filter the noisy labels acquired from the noisy max operation on the PA axis of CT-scans. The resulting masks Y_A^{Ribs} are, therefore, computed according to:

$$Y_A^{Ribs} = Y_A^{Bone} \ \& \ \hat{Y}_A^{Lung} \quad , \tag{3.3}$$

where the & operator represents the pixelwise AND operation.

Similar approaches could be developed using the same framework as ours not only for thoracic images, but by leveraging CT scans or Magnetic Resonance Imaging (MRI) samples from other parts of the body and transferring it to 2D data. A task that could easily benefit from this knowledge transfer would be bone fracture detection, as there are no densely labeled X-Ray datasets for bone segmentation in the literature, as far as the authors are aware. In this example, even a small CT dataset of a few hundred images could serve as a source for bone segmentation for 2D X-Rays.

One should notice that in this pipeline we make two distinct uses of CoDAGANs:

1. **CoDA**$_{Lungs}$ for acquiring lung segmentation predictions $\hat{Y}_A^{Lungs}$ for DRRs from labeled CXR source datasets (X_B and Y_B) – highlighted in blue in Figure 3.5;

2. **CoDA**$_{Ribs}$ for translating the knowledge from the filtered rib segmentation masks Y_A^{Ribs} for DRRs in order to use them in CXR data (X_B), resulting in the prediction $\hat{Y}_B^{Ribs}$ – as delineated in orange in Figure 3.5.

In our experiments these two architecture were trained separately, as they had distinct objectives and the experimental procedure was conceived to be relatively lightweight in terms of GPU memory.

3.4 Proposed Methods and Hypotheses Validation

The current chapter described in detail one method for pairwise DA, which is linked to the first set of experiments and results described in Sections 4.2 and 5.1. These experiments were used as a basis for validating the main hypothesis ($\mathcal{H}_1$), which guided the other developments of this work.

Section 3.2 described the first proposal for multi-source DA – that is, Domain Generalization – in biomedical dense labeling. This approach was used for validating the secondary hypotheses ($\mathcal{H}_2$, $\mathcal{H}_3$ and $\mathcal{H}_4$) that guided the later stages of this research and allowed for truly generalizable radiological image segmentation, as shown in the experimental setup and results in Sections 4.3 and 5.2.

At last, this chapter used the framework of Conditional DA to devise a pipeline that is fed unlabeled CT-scan and CXR datasets and achieves generalization in rib segmentation even from noisy labels, further reinforcing $\mathcal{H}_2$, $\mathcal{H}_3$ and $\mathcal{H}_4$. The pipeline that makes this possible is detailed in Section 3.3 and is used as a proof of concept for hypothesis $\mathcal{H}_5$, which conjectures that synthetic data can be leveraged to acquire useful knowledge in biomedical data. Rib segmentation experiments for the proposed pipeline and both deep and shallow baseline methods can be seen in Section 4.4, while Section 5.3 presents and discusses the results from these experiments.

Chapter 4

Experimental Setup

All code was implemented using the PyTorch[1] Deep Learning framework. We used the MUNIT/UNIT implementation from Huang et al. [2018][2] as a basis. UNIT [Li et al., 2017] and MUNIT [Huang et al., 2018] were chosen in favor of other Cycle-Consistent GANs – such as CycleGANs [Zhu et al., 2017a] and CoGANs [Liu and Tuzel, 2016] – due to the fact that these DNNs have explicit representations for $\mathcal{I}$ in their architectures, while also presenting better visual results in image translation tasks. All tests were conducted on NVIDIA Titan X Pascal GPUs with 12GB of memory. Code, supplementary results and materials from this work can be found in the PATREO website[3].

Section 4.1 describes the basic training procedure and hyperparameters for CoDAGANs. These parameters were found empirically and are used in all experiments described in this thesis. Sections 4.3 and 5.3 describe the main DA experiments in various distinct radiological domains and a rib segmentation pipeline that uses data acquired from volumetric tomography data for segmenting Chest X-Rays, respectively.

4.1 Hyperparameters

Architectural choices and hyperparameters can be further analysed according to the codes and configuration files in the project's website, but the main ones are described in the following paragraphs and are used in all sets of experiments in this work (Sections 4.2, Sections 4.3 and 4.4). Both D2D and CoDAGANs were trained

[1] `https://pytorch.org/`

[2] `https://github.com/nvlabs/MUNIT`

[3] `http://www.patreo.dcc.ufmg.br/`

for 400 epochs, as this was empirically found to be a good stopping point for convergence in these networks in all the settings analyzed in our experiments. Learning rate was set to 1×10^{-4} with L2 normalization by weight decay with value 1×10^{-5}. G_E is composed of two downsampling layers followed by two residual layers for both UNIT [Liu et al., 2017] and MUNIT [Huang et al., 2018] based implementations, as these configurations were observed to simultaneously yield satisfactory results and have small GPU memory requirements. The first downsampling layer contains 32 convolutional filters, doubling this number for each subsequent layer. D was implemented using a LSGAN [Mao et al., 2017] objective with only two layers, although differently from MUNIT, we do not employ multiscale discriminators due to GPU memory constraints. Also distinctly from MUNIT and UNIT, we do not employ the VGG-based [Simonyan and Zisserman, 2014] perceptual loss – further detailed by Huang et al. [2018] – due to the dissimilarities between the domains wherein these networks were pretrained and the biomedical images used in our work, which could lead to negative transfer.

We chose the state-of-the-art Adam solver [Kingma and Ba, 2014] to optimize CoDAGANs, as it mitigates several optimization problems of the traditional Stochastic Gradient Descent (SGD), which helps to counterweight the inherent difficulties in training GANs [Salimans et al., 2016].

4.2 D2D Exploratory Experiments

The most well-known datasets for organ segmentation in CXRs are the Japanese Society of Radiological Technology (JSRT) [Shiraishi et al., 2000][4] and the Montgomery dataset [Jaeger et al., 2014][5]. Exploratory tests on the pairwise D2D approach were performed only on this dataset pair and aimed primarily to find a good configuration to run D2D on so that it could be extended to perform multi-source DA.

JSRT contains 247 PA chest radiographs, while Montgomery is composed of 138 cases. JSRT has pixel-level labels for lung field segmentation tasks as well as heart and clavicle ground truths, while Montgomery only contains ground truths for lungs. Semantic maps from other CXR datasets – as the ones presented in Section 4.3 – are fairly recent and were not available at the time for D2D. Therefore our quantitative experimental procedure only took into account the JSRT and Montgomery datasets, as they were the only ones with pixel-level annotations. Quantitative experiments for the more recent annotations of other datasets, more specifically

[4] http://db.jsrt.or.jp/eng.php

[5] https://ceb.nlm.nih.gov/repositories/tuberculosis-chest-x-ray-image-data-sets/

OpenIST, Shenzhen and Chest X-Ray8, using D2D are described in Section 4.3 and discussed in Section 5.2. Qualitative assessments of heart and clavicle UDAs results using Pretrained DNNs in JSRT are presented for Montgomery samples, which does not have ground truths for these tasks.

The exploratory tests for D2D were performed using MUNIT [Huang et al., 2018]. This architecture was chosen for the first test because it is designed to split the encoding of content and style information in the images. This allows D2D to encode images from the unlabeled/semi-labeled dataset B (Montgomery) with the style of the labeled dataset A (JSRT), while still preserving the content – that is, the basic shape and texture characteristics – of the original image $b \in B$.

In order to prevent the vanishing gradients problem in $G_{B \to A}$ and $G_{A \to B}$, we chose for M_A a segmentation architecture with Skip Connections: a U-Net [Ronneberger et al., 2015]. The Pretrained U-Net was fit on a training fold comprised of 60% of the samples in the JSRT dataset using an optimizer with the relatively standard configuration of Stochastic Gradient Descent (SGD), learning rate of $1e^{-4}$ and momentum 0.9. The other 40% of the JSRT images were used as validation (20%) and test sets (20%) for the DNN. The Montgomery dataset was also divided in a 60%/20%/20% configuration. The knowledge acquired by the Pretrained U-Net was then transferred to Montgomery using both Fine-tuning and D2D method.

We noticed that trying to transfer the knowledge in M_A since the first training epoch was detrimental to the convergence of the translation model, probably due to competing supervised and unsupervised objectives. Therefore, we first trained the generators and discriminators for 20 epochs and made sure they converged via visual assessment. Only then we started training the supervised part of the model coupled with the unsupervised translation method. This strategy also allowed us to train one single translation model in a completely unsupervised fashion for the first 20 epochs, only then starting the supervised training steps for different tasks using the same pretrained $G_{A \to B}$, $G_{B \to A}$, D_A and D_B (see Figure 3.1). Readers should notice that this scheme requires manual confirmation that the image translation converged correctly, which was not always the case. This further reinforced the high instability of the early D2D experiments.

4.3 Conditional DA Experiments

4.3.1 Datasets

We tested our methodology in a total of 16 datasets, 8 of them being Chest X-Ray (CXR) datasets, 6 of them being Mammographic X-Ray (MXR) datasets and 2 of them being composed of Dental X-Ray (DXR) images. The chosen CXR datasets are the Japanese Society of Radiological Technology (JSRT) [Shiraishi et al., 2000], OpenIST[6], Shenzhen and Montgomery sets [Jaeger et al., 2014], Chest X-Ray 8 [Wang et al., 2017][7], PadChest [Bustos et al., 2019][8], NLMCXR [Demner-Fushman et al., 2015][9] and the Optical Coherence Tomography and Chest X-Ray Images (OCT CXR) [Kermany et al., 2018][10] dataset. Heart, clavicle and rib label sets for OpenIST were obtained from the *testerv11* repository [11], while Chest X-Ray 8 lung ground truths were acquired via the XLSor [Tang et al., 2019b] project[12]. The MXR datasets used in this work are INbreast [Moreira et al., 2012][13], the Mammographic Image Analysis Society (MIAS) dataset [Suckling et al., 2015][14], the Digital Database for Screening Mammography (DDSM) [Heath et al., 2000][15], the Breast Cancer Digital Repository (BCDR) [Lopez et al., 2012][16], and LAPIMO [Matheus and Schiabel, 2011][17]. DDSM was split into two groups: 1) samples A, and 2) samples B/C; as these groups were acquired and digitized with different equipments, yielding considerably distinct visual patterns. A random subset of samples from DDSM B/C and DDSM A were manually labeled for the task of pectoral muscle segmentation. These samples were not used during training, but instead for computing objective evaluation metrics during test and can be downloaded for reproducibility in this project's webpage. The only two DXR datasets we used in our experiments are the IvisionLab [Silva et al., 2018][18] and the Panoramic X-Ray [Abdi et al., 2015][19] datasets.

[6] https://github.com/pi-null-mezon/OpenIST

[7] https://nihcc.app.box.com/v/ChestXray-NIHCC/folder/37178474737

[8] http://bimcv.cipf.es/bimcv-projects/padchest/

[9] https://openi.nlm.nih.gov/

[10] https://data.mendeley.com/datasets/rscbjbr9sj/3

[11] https://www.kaggle.com/viktorivanovio/testerv11

[12] https://github.com/rsummers11/CADLab/tree/master/Lung_Segmentation_XLSor

[13] http://medicalresearch.inescporto.pt/breastresearch/index.php/Get_INbreast_Database

[14] https://www.repository.cam.ac.uk/handle/1810/250394

[15] http://marathon.csee.usf.edu/Mammography/Database.html

[16] https://bcdr.eu/patient/list#

[17] http://lapimo.sel.eesc.usp.br/bancoweb/english/

[18] https://github.com/IvisionLab/deep-dental-image

[19] https://data.mendeley.com/datasets/hxt48yk462/1

A total of 7 distinct segmentation tasks are compared in our experiments: 1) Pectoral muscle, 2) Breast region in MXRs; 3) Lungs, 4) Heart, 5) Clavicles in CXRs; 6) Mandible and 7) Teeth in DXRs. The number of training and testing samples from each domain, dataset and task is available at this project's webpage.

Datasets were randomly split into training and test sets according to an 80%/20% division when possible. However, some datasets (i.e. Chest X-Ray 8 Wang et al. [2017]) contain a smaller number of labeled samples than those 20%, which required a flexibilization of our procedure to the divisions presented in Tables 4.1, 4.2 and 4.3 for CXRs, MXRs and DXRs, respectively.

As CoDAGANs are trained with samples from two datasets at each iteration and all datasets in this study have different numbers of samples, we performed random undersampling in larger datasets in order to fit the smaller sample sizes of other image sets.

4.3.2 Experimental Protocol

Aiming to mimic real-world scenarios wherein the lack of labels is a considerable problem, we did not keep samples for validation purposes. We evaluate results from epochs 360, 370, 380, 390 and 400 for computing the mean and standard deviation values presented in Section 5 in order to consider the statistical variability of the methods during the *Supervision Tuning* routine (Section 3.2.2), when training is more stable.

For quantitative assessment we used the Jaccard (Intersection over Union – IoU) metric, which is a common choice in segmentation and detection tasks and is widely used in all tested domains [Rampun et al., 2017; Van Ginneken et al., 2006; Silva et al., 2018]. Jaccard ($\mathcal{J}$) for a binary classification task is given by the following equation:

$$\mathcal{J} = \frac{TP}{TP + FN + FP} \quad , \tag{4.1}$$

where TP, FN and FP refer to True Positives, False Negatives and False Positives, respectively. Jaccard values range between 0 and 1, however we present these metrics as percentages by multiplying them by a factor of 100 in Section 5.

4.3.3 Data Augmentation

Due to the small amount of labeled data in the datasets often found as sources for DA tasks in biomedical settings, Data Augmentation strategies are oftentimes used for synthetically enlarging the quantity of information used for supervision in ma-

Table 4.1. Sample distribution in CXR datasets according to task and labels.

Lungs				
Dataset	**Unlabeled**		**Labeled**	
	train	**test**	**train**	**test**
JSRT	–	–	197	50
OpenIST	–	–	260	15
Shenzhen	–	–	452	114
Montgomery	–	–	110	28
Chest X-Ray 8	3390	–	–	100
PadChest	1590	381	–	–
NLMCXR	2204	854	–	–
OCT CXR	295	205	–	–
Heart				
Dataset	**Unlabeled**		**Labeled**	
	train	**test**	**train**	**test**
JSRT	–	–	197	50
OpenIST	–	–	260	15
Shenzhen	452	114	–	–
Montgomery	110	28	–	–
Chest X-Ray 8	3390	270	–	–
PadChest	1590	381	–	–
NLMCXR	2204	854	–	–
OCT CXR	295	205	–	–
Clavicles				
Dataset	**Unlabeled**		**Labeled**	
	train	**test**	**train**	**test**
JSRT	–	–	197	50
OpenIST	–	–	260	15
Shenzhen	452	114	–	–
Montgomery	110	28	–	–
Chest X-Ray 8	3390	270	–	–
PadChest	1590	381	–	–
NLMCXR	2204	854	–	–
OCT CXR	295	205	–	–

chine learning algorithms. Most datasets used in our experiments have only a few hundreds of samples, which is usually too small for training Deep Learning algorithms, therefore we employed some Data Augmentation methods usually used for images as standard procedures in CoDAGANs. Rotating, random cropping and color inversions were applied to the samples, according to the training needs of their respective domains. More details on the Data Augmentation employed in both D2D and CoDAGANs can be found in this project's oficial code available at the webpage

Table 4.2. Sample distribution in MXR datasets according to task and labels.

Pectoral				
Dataset	**Unlabeled**		**Labeled**	
	train	**test**	**train**	**test**
INbreast	–	–	160	40
MIAS	–	–	257	65
DDSM B/C	186	–	–	52
DDSM A	134	29	–	–
BCDR	344	66	–	–
LAPIMO	585	85	–	–
Breast				
Dataset	**Unlabeled**		**Labeled**	
	train	**test**	**train**	**test**
INbreast	–	–	160	40
MIAS	–	–	257	65
DDSM B/C	186	52	–	–
DDSM A	134	29	–	–
BCDR	344	66	–	–
LAPIMO	585	85	–	–

Table 4.3. Sample distribution in DXR datasets according to task and labels.

Teeth				
Dataset	**Unlabeled**		**Labeled**	
	train	**test**	**train**	**test**
IvisionLab	–	–	1340	160
Panoramic X-Ray	89	27	–	–
Mandible				
Dataset	**Unlabeled**		**Labeled**	
	train	**test**	**train**	**test**
IvisionLab	1340	160	–	–
Panoramic X-Ray	–	–	89	27

for the CoDAGAN project.

4.3.4 Baselines

Large datasets as ImageNet [Deng et al., 2009] turned Fine-tuning DNNs into a well known method for Transfer Learning in the Deep Learning literature, as most specific datasets do not possess the large amount of labeled data required for training from scratch in classification tasks. Fine-tuning was later adapted for dense labeling tasks [Long et al., 2015] and is nowadays common procedure in semantic segmentation tasks in the Computer Vision domain. As explained in Section 2.4, the large domain shifts between Computer Vision and Biomedical Imaging datasets makes pretraining on the former data hardly useful to the medical segmentation tasks in our experiments. Therefore, no external Computer Vision dataset was used for pretraining of any DNN in this work.

Readers should notice that Fine-tuning still does not work in UDA, as it necessarily requires labeled data. Therefore, we inserted the use of Pretrained DNNs on the source biomedical sets as baselines both without further training in UDA scenarios and as basis for Fine-tuning in SSDA and FSDA scenarios. Still in the field of classical approaches do Transfer Learning, we add as a baseline to our experimental procedure training a DNN From Scratch with the smaller amount of labeled data available for targets datasets in SSDA and FSDA scenarios.

Our main baseline was the D2D previously described in this manuscript, as it uses a Cycle-Consistent GAN with a similar architecture as CoDAGANs. In order to improve fairness, the version of D2D used in the Conditional DA experiments has several improvements to the original one. These include the division between Full Training and Supervision Tuning (Section 3.2.2), using both UNIT [Liu et al., 2017] and MUNIT [Huang et al., 2018] as Cycle-Consistent GAN backbones and the same architecture as its CoDAGANs counterpart.

We explicit that, even though D2D was an earlier iteration of this work that is used as baseline in the Conditional DA experiments, it can be seen as a stand-in for most other architectures based on pairwise training, such as CyCADA [Hoffman et al., 2018], I2IAdapt [Murez et al., 2018] and DCAN [Wu et al., 2018], as they perform essentially the same computation graph as our D2D. One should notice that $D2D_M$ is particularly similar to the method proposed by Yang et al. [2019] and XLSor [Tang et al., 2019b], as content-only training was further explored by Yang et al. [2019]. Similarly, $D2D_U$ is conceptually similar to TUNA-Net [Tang et al., 2019a], Zhang et al. [2018c] and TD-GANs [Zhang et al., 2018b]. Thus, both $D2D_U$ and $D2D_M$ in their current version – even with the limitations of pairwise training – can be considered state-of-the-art I2I DNNs for DA.

Both our method and the previously introduced baselines use the U-Net [Ronneberger et al., 2015] architecture as backbone for supervised semantic segmentation. This was a conscious choice based on early experiments of this work [Oliveira and dos Santos, 2018; Oliveira et al., 2018] that compared FCNs [Long et al., 2015], U-Nets [Ronneberger et al., 2015] and SegNets [Badrinarayanan et al., 2017] in similar setups and found that U-Nets and SegNets achieved the best results while FCNs generally presented subpar results compared to their Transposed Convolution-based peers. We then narrowed the search due to the larger amount of Skip Connections in this architecture, which mitigates the problem of vanishing gradients by creating backward flow bypasses that help on the training of earlier layers and previous modules with the supervised loss $\mathcal{L}_{sup}$.

Many UDA shallow methods [Huang et al., 2007; Pan et al., 2011; Sun et al., 2011; Gong et al., 2013; Geng et al., 2011] and most deep approaches for classification tasks [Ghifary et al., 2014; Long et al., 2016, 2017] rely on variations of the Maximum Mean Discrepancy (MMD) [Borgwardt et al., 2006] metric to perform knowledge transfer by matching the statistical moments of a dataset pair. MMD-based approaches are a kind of Feature Representation Learning, which only takes into account features space, ignoring label space, thus, it's fully unsupervised. It works, therefore, as an unsupervised alternative to Fine-tuning DNNs, as it is possible to derive a loss function based on this criterion. As there are no MMD methods specifically designed for dense labeling tasks, we tried to adapt a Radial Basis Function (RBF) kernel version of MMD [Li et al., 2017] for dense prediction, however early results showed little-to-no gains compared to training from scratch. Simple moment-matching might be an excessively simple approach for segmentation tasks, where neighbor samples often present a large spatial correlation in both pixel and label space. Hence, MMD metrics are not shown in Section 5 in order to simplify the presentation of the experiments.

4.4 Synthetic Data Experiments

Apart from the basic DA datasets, baselines and experiments presented previously, we also conducted experiments using images from the Lung Image Database Consortium and Image Database Resource Initiative (LIDC-IRDI) [Armato III et al., 2011][20]. These experiments use as a source artificially created 2D images from 3D volumes, which are called Digitally Reconstructed Radiographs (DRRs). The target CXR datasets used in the experiments from synthetic DRRs are the same ones presented in Section 4.3.1. These experiments were conducted in order to assess the validity of $\mathcal{H}_5$, which implies a novel use of Conditional DA from data and labels acquired synthetically.

The experiments for Domain Generalization in rib segmentation are considerably similar to the ones presented by Zhang et al. [2018b] and Candemir et al. [2016]. Distinctly, Zhang et al. [2018b] needs 3D organ segmentation labels – which are con-

[20] `https://wiki.cancerimagingarchive.net/display/Public/LIDC-IDRI`

siderably harder to acquire than 2D ones – in order to transfer the knowledge to 2D data. Candemir et al. [2016] inspired the flattening operations of AIP and MIP used in our pipeline, however, they use relatively simple image processing techniques for rib segmentation, not requiring any DA or Machine Learning to complete their goal. The method by Candemir et al. [2016] is, therefore, highly dependent on parameter tuning and unlikely to properly generalize to new data with large domain shifts.

Experiments with synthetic DRR used a distinct set of metrics as the ones presented in Sections 4.2 and 4.3, as the literature for rib segmentation considerably differs from other tasks in this manuscript. We also report Jaccard ($\mathcal{J}$) results, but include the threshold-independent Receiver Operating Characteristics (ROC) and Area Under Curve (AUC) metrics. The Dice $\mathcal{D}$ metric (also known as F1) is considerably common in the medical image segmentation literature as a whole, so we also report it in the synthetic data experiments. At last, we also report other threshold-dependent common metrics for rib segmentation: Accuracy ($\mathcal{A}$), Sensitivity ($\mathcal{S}$) and Specificity ($\ddot{\mathcal{S}}$). $\mathcal{D}$, $\mathcal{A}$, $\mathcal{S}$ and $\ddot{\mathcal{S}}$ are given by the following equations:

$$\mathcal{D} = \frac{2TP}{2TP + FP + FN} \quad , \tag{4.2}$$

$$\mathcal{A} = \frac{TP + TN}{TP + FN + FP + TN} \quad , \tag{4.3}$$

$$\mathcal{S} = \frac{TP}{TP + FN} \quad , \tag{4.4}$$

$$\ddot{\mathcal{S}} = \frac{TN}{TN + FP} \quad . \tag{4.5}$$

A more complete compilation of results from the synthetic DRR experiments can be seen in the subproject's webpage[21].

4.5 Experiments Sets and Hypotheses Validation

This chapted explored the experimental setup used throughout Chapter 5 for validating $\mathcal{H}_1$ (Section 5.1); $\mathcal{H}_2$, $\mathcal{H}_3$ and $\mathcal{H}_4$ (Section 5.2); and, at last, $\mathcal{H}_5$ (Section 5.3).

[21] `https://sites.google.com/view/virginiafernandes/datasets/lidc-idri-drr`

Chapter 5

Results and Discussion

This chapter is divided into three distinct parts, with one for validating each main contribution in this work. Section 5.1 describes the first tests with D2D for UDA, SSDA and FSDA compared to traditional baselines as Fine-tuning and From Scratch training of the networks in the target dataset. These earlier tests were exploratory in nature and would serve both as a proof of concept that I2I could be used for DAs in dense labeling tasks – which was still not clear at the time – and to solidify the knowledge necessary to the more advanced proposed method based on Conditional DA.

CoDAGANs are the main contribution of this manuscript, being the natural progression of D2D towards Domain Generalization. The Conditional DA proposed in CoDAGANs allowed for larger numbers of datasets to be fed to the network at once, leveraging all the labels available to the method and transferring this knowledge to unlabeled data. Section 5.2 describes the large number of tests performed with CoDAGANs using D2D and other common baselines from the Deep Transfer Learning literature.

At last, Section 5.3 presents the Domain Generalization results for rib segmentation using the synthetic data and labels acquired from DRRs computed from LIDC-IRDI [Armato III et al., 2011].

5.1 Image Translation DA for X-Ray Segmentation

Table 5.1 shows the results obtained by the proposed DA method compared with Fine-tuning and From Scratch training with the limited labels in the case of SSDA. Figure 5.1 shows the Confidence Intervals using $p \leq 0.05$ for the results in Table 5.1. The horizontal axis represents the amount of labels kept by the experiment, while the vertical axis denote $\mathcal{J}$ values achieved in these settings. It is clear that D2D significantly surpasses the effectiveness of fine-tuning when using between 0% and 20% of the labels from the target training set. When using 50% and 100% of the target labels, fine-tuning marginally surpassed our method, even though the difference was not statistically significant.

Table 5.1. Transfer Learning results from JSRT [Shiraishi et al., 2000] to Montgomery [Jaeger et al., 2014] in a Pretrained U-Net with (SSDA) and without (UDA) fine-tuning. Bold values indicate the best results for each line.

Label %	D2D	Fine-Tuning	From Scratch
0%	**88.20** ± **9.80**	4.30 ± 4.13	–
1.25%	**88.83** ± **9.81**	78.94 ± 13.32	54.23 ± 13.37
2.5%	**88.25** ± **10.19**	83.32 ± 12.32	56.01 ± 13.76
5%	**90.79** ± **7.05**	83.46 ± 8.60	55.10 ± 14.42
10%	**89.18** ± **9.18**	83.66 ± 9.69	87.80 ± 6.78
20%	**91.26** ± **7.20**	88.71 ± 8.73	89.50 ± 7.65
50%	92.15 ± 5.90	**93.78** ± **5.42**	89.82 ± 4.34
100%	93.18 ± 5.47	**94.81** ± **5.15**	94.16 ± 4.57

When using no labeled data in the target dataset, it can be seen that $\mathcal{J}$ drops to only 4.30%. This result renders it infeasible to interchange models between CXR datasets without labeled data in the target dataset using traditional transfer methods. Our method achieves a Jaccard of 88.20% even without labeled data in the target set, as it uses all the unlabeled target samples to perform the transfer and the source labels to ensure visual feature preservation by $G_{S \to T}$ and $G_{T \to S}$.

We highlight the difficulties in convergence of D2D, which were mitigated in the first iterations of this work by splitting 1/4 of the training set into a validation set in order to use the training epoch with the best generalization, as described in Section 4.2. This is highly unlikely to be possible in real-world scenarios, wherein datasets are either fully unlabeled or have only a small number of labeled samples.

5.1.1 Target Dataset Qualitative Assessment

As the Montgomery does not have pixel-level labels for clavicle and heart segmentation, we performed qualitative tests on our Transfer Learning method for these tasks using the unsupervised case, that is, without labeled data in the target dataset. One can see in Figure 5.2 that clavicles and heart regions were accurately recognized. In most Montgomery images the algorithm correctly identified the clavicles, with only 4 cases of inadequate segmentations in one or both clavicles among the 27 images tested for this task. One example of misidentification of the clavicle area is shown in

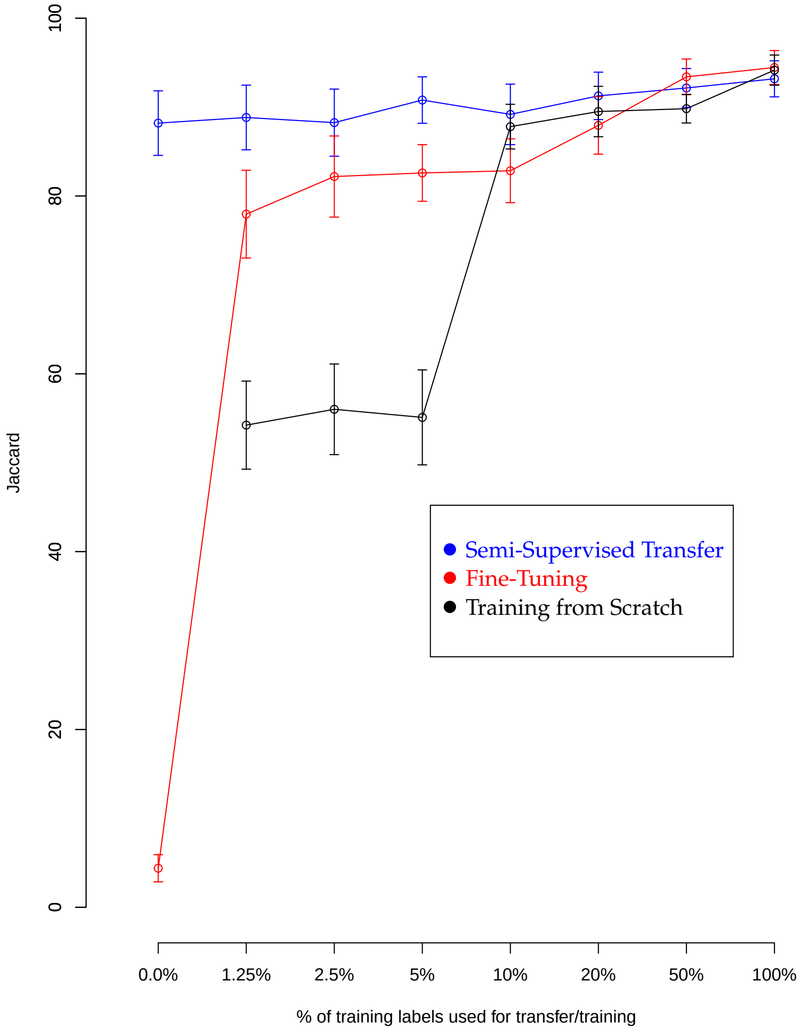


Figure 5.1. Confidence Intervals for Montgomery [Jaeger et al., 2014] in lung field segmentation using a model Pretrained U-Net from JSRT [Shiraishi et al., 2000].

Figure 5.2(d). Most heart segmentations were near perfect, but, as the Montgomery dataset contains more diverse samples, hearts with abnormal shapes were not fully identified, as can be see in Figure 5.2(h).

5.2 Conditional DA Experiments

In most scenarios D2D still behaves better than using Pretrained DNNs or Fine-tuning (in the case of SSDA), however D2D still presented considerable problems

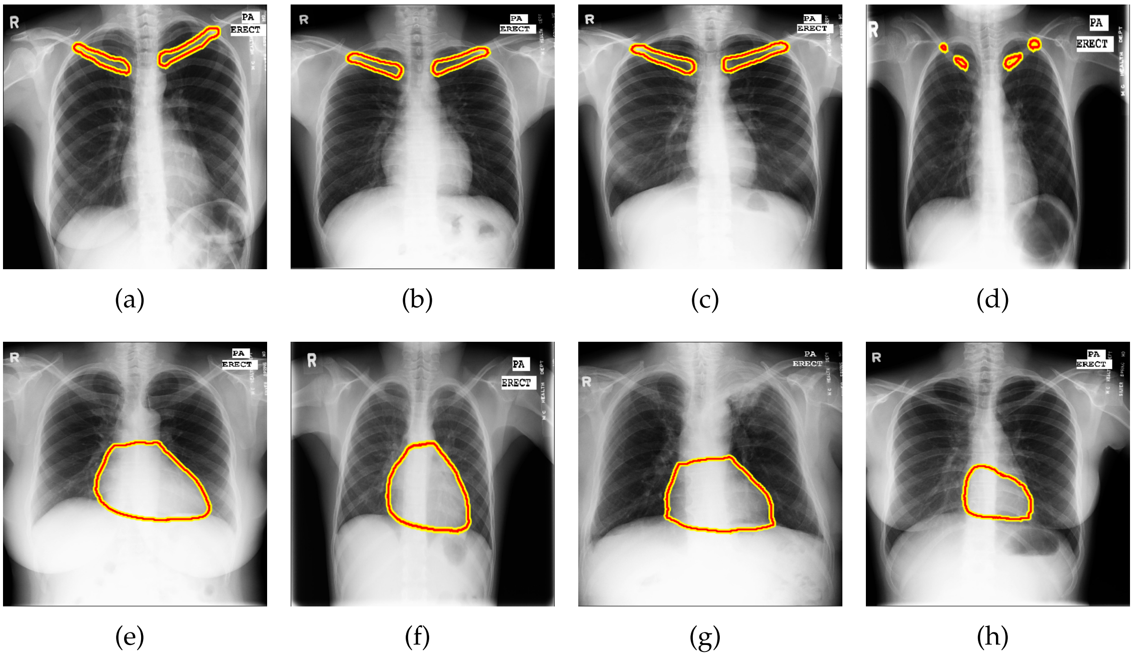


Figure 5.2. Segmentation results for the Montgomery Set [Jaeger et al., 2014] in the tasks of (a-d) clavicle and (e-h) heart segmentation from a model pretrained in JSRT [Shiraishi et al., 2000] and transferred with 0% of labeled data in the target dataset.

in convergence and could not be trained with more than two data/label sources. Hence, the following sections compare the segmentation and runtime efficiency of D2D and CoDAGANs in order to assess the validity of the hypotheses related to Conditional DA ($\mathcal{H}_2$, $\mathcal{H}_3$ and $\mathcal{H}_4$).

5.2.1 Quantitative Results for MXR Samples

Jaccard average values and standard deviations for MXR tasks are shown in Tables 5.2 and 5.3 for pectoral muscle and breast region segmentation, respectively. The first lines in the tables present the label configurations used in the experiments at each column. Results are shown separately for datasets INbreast ($\mathscr{A}$), MIAS ($\mathscr{B}$), DDSM B/C ($\mathscr{C}$), and DDSM A ($\mathscr{D}$) in Table 5.2 and for datasets INbreast ($\mathscr{A}$) and MIAS ($\mathscr{B}$) in Table 5.3. Objective results for datasets BCDR ($\mathscr{E}$) and LAPIMO ($\mathscr{F}$) for pectoral muscle and datasets ($\mathscr{C}$)-($\mathscr{F}$) in breast region segmentation are not possible due to the complete lack of labels in these tasks. We reinforce that only two CoDAGANs ($\boldsymbol{CoDA_M}$ using MUNIT [Huang et al., 2018] and $\boldsymbol{CoDA_U}$ based on UNIT [Liu et al., 2017]) were trained for all datasets in each task, as CoDAGANs allow for multi-source and multi-target DA. Thus, repeated columns indicating the results for $\boldsymbol{CoDA_M}$ and $\boldsymbol{CoDA_U}$ are simply reporting the results of the same models for different datasets. All methods beside $\boldsymbol{CoDA_M}$ and $\boldsymbol{CoDA_U}$ indicate whether the source or target data used in the training, as they are neither multi-source nor multi-target, limiting them to pairwise training.

Table 5.2. $\mathcal{J}$ results (in %) for pectoral muscle segmentation DA to and/or from six distinct MXR datasets: INbreast ($\mathscr{A}$), MIAS ($\mathscr{B}$), DDSM B/C ($\mathscr{C}$), DDSM A ($\mathscr{D}$), BCDR ($\mathscr{E}$) and LAPIMO ($\mathscr{F}$). This table shows results for CoDAGANs with backbones based on MUNIT [Huang et al., 2018] ($\boldsymbol{CoDA_M}$) and UNIT [Liu et al., 2017] ($\boldsymbol{CoDA_U}$), as well as Domain-to-Domain approaches based on these architectures ($D2D_M$ and $D2D_U$), Pretrained U-Nets [Ronneberger et al., 2015] and U-Nets trained from scratch on the limited target labels.

Experiments		$E_{0\%}$	$E_{2.5\%}$	$E_{5\%}$	$E_{10\%}$	$E_{50\%}$	$E_{100\%}$
% Labels INbreast ($\mathscr{A}$)		100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
% Labels MIAS ($\mathscr{B}$)		0.00%	2.50%	5.00%	10.00%	50.00%	100.00%
% Labels DDSM_BC ($\mathscr{C}$)		0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
% Labels DDSM_A ($\mathscr{D}$)		0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
% Labels BCDR ($\mathscr{E}$)		0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
% Labels LAPIMO ($\mathscr{F}$)		0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
($\mathscr{A}$)	$\boldsymbol{CoDA_M}$	91.95 ± 0.81	92.57 ± 0.31	**92.61 ± 0.44**	**92.00 ± 0.90**	**90.66 ± 0.53**	88.58 ± 1.76
	$\boldsymbol{CoDA_U}$	91.18 ± 0.36	90.61 ± 0.89	91.03 ± 1.43	91.23 ± 1.51	90.36 ± 0.98	89.98 ± 0.37
	$\boldsymbol{D2D_M}$ **($\mathscr{A}$)→($\mathscr{B}$)**	93.27 ± 0.51	**92.67 ± 0.64**	87.54 ± 10.00	90.58 ± 2.14	84.46 ± 3.36	**90.11 ± 1.06**
	$\boldsymbol{D2D_U}$ **($\mathscr{A}$)→($\mathscr{B}$)**	92.27 ± 0.55	83.17 ± 9.27	89.56 ± 2.21	23.82 ± 10.21	81.64 ± 6.22	86.81 ± 2.98
	$\boldsymbol{D2D_M}$ **($\mathscr{A}$)→($\mathscr{C}$)**	93.43 ± 0.24	–	–	–	–	–
	$\boldsymbol{D2D_U}$ **($\mathscr{A}$)→($\mathscr{C}$)**	93.64 ± 0.50	–	–	–	–	–
	$\boldsymbol{D2D_M}$ **($\mathscr{A}$)→($\mathscr{D}$)**	93.72 ± 0.96	–	–	–	–	–
	$\boldsymbol{D2D_U}$ **($\mathscr{A}$)→($\mathscr{D}$)**	88.06 ± 2.30	–	–	–	–	–
	$\boldsymbol{D2D_M}$ **($\mathscr{A}$)→($\mathscr{E}$)**	**93.87 ± 0.71**	–	–	–	–	–
	$\boldsymbol{D2D_U}$ **($\mathscr{A}$)→($\mathscr{E}$)**	92.55 ± 0.57	–	–	–	–	–
	$\boldsymbol{D2D_M}$ **($\mathscr{A}$)→($\mathscr{F}$)**	92.29 ± 2.06	–	–	–	–	–
	$\boldsymbol{D2D_U}$ **($\mathscr{A}$)→($\mathscr{F}$)**	91.47 ± 0.55	–	–	–	–	–
	From Scratch in ($\mathscr{A}$)	93.25 ± 0.75	–	–	–	–	**–**
($\mathscr{B}$)	$\boldsymbol{CoDA_M}$	**67.61 ± 2.07**	69.92 ± 2.42	**72.31 ± 0.65**	75.66 ± 0.98	78.24 ± 0.23	**79.08 ± 0.78**
	$\boldsymbol{CoDA_U}$	60.01 ± 2.77	61.81 ± 3.26	71.33 ± 1.81	**76.67 ± 0.15**	**78.37 ± 0.54**	78.49 ± 1.38
	$\boldsymbol{D2D_M}$ **($\mathscr{A}$)→($\mathscr{B}$)**	0.00 ± 0.00	0.00 ± 0.00	22.73 ± 17.62	17.72 ± 15.05	35.46 ± 15.24	64.46 ± 5.61
	$\boldsymbol{D2D_U}$ **($\mathscr{A}$)→($\mathscr{B}$)**	41.06 ± 19.00	36.72 ± 15.07	59.67 ± 3.59	59.63 ± 12.37	62.69 ± 9.92	75.95 ± 2.57
	Pretrained ($\mathscr{A}$)→($\mathscr{B}$)	40.49	**72.11 ± 0.16**	60.46 ± 3.76	71.89 ± 1.22	75.52 ± 0.34	78.35 ± 1.20
	From Scratch in ($\mathscr{B}$)	–	58.51 ± 5.44	51.90 ± 1.38	63.32 ± 5.62	77.79 ± 0.44	78.08 ± 0.46
($\mathscr{C}$)	$\boldsymbol{CoDA_M}$	**89.99 ± 0.80**	**90.73 ± 0.83**	**91.49 ± 0.36**	**92.34 ± 0.57**	**92.80 ± 0.40**	**92.50 ± 0.48**
	$\boldsymbol{CoDA_U}$	82.45 ± 4.01	86.21 ± 3.13	89.90 ± 2.10	90.71 ± 0.72	91.21 ± 0.63	92.24 ± 0.65
	$\boldsymbol{D2D_M}$ **($\mathscr{A}$)→($\mathscr{C}$)**	0.03 ± 0.01	–	–	–	–	–
	$\boldsymbol{D2D_U}$ **($\mathscr{A}$)→($\mathscr{C}$)**	0.64 ± 0.90	–	–	–	–	–
	Pretrained ($\mathscr{A}$)→($\mathscr{C}$)	78.22	–	–	–	–	–
($\mathscr{D}$)	$\boldsymbol{CoDA_M}$	**49.38 ± 5.21**	**50.00 ± 4.37**	**49.20 ± 1.84**	**54.37 ± 2.21**	**77.59 ± 0.75**	**69.76 ± 3.89**
	$\boldsymbol{CoDA_U}$	23.83 ± 2.15	26.13 ± 2.74	42.93 ± 4.73	35.10 ± 4.81	31.23 ± 3.76	56.46 ± 5.78
	$\boldsymbol{D2D_M}$ **($\mathscr{A}$)→($\mathscr{D}$)**	0.41 ± 0.27	–	–	–	–	–
	$\boldsymbol{D2D_U}$ **($\mathscr{A}$)→($\mathscr{D}$)**	0.74 ± 0.94	–	–	–	–	–
	Pretrained ($\mathscr{A}$)→($\mathscr{D}$)	22.20	–	–	–	–	–

Bold values in these tables indicate the best results for the corresponding dataset indicated in the first column of these tables. As there are four datasets being evaluated in Table 5.2, there are four bold values for each experiment. Analogously, Table 5.3 only has two bold values per column because only two datasets are being objectively evaluated in breast region segmentation. In both tables INbreast was

Table 5.3. $\mathcal{J}$ results (in %) for breast region segmentation DA to and/or from six distinct MXR datasets: INbreast ($\mathscr{A}$), MIAS ($\mathscr{B}$), DDSM B/C ($\mathscr{C}$), DDSM A ($\mathscr{D}$), BCDR ($\mathscr{E}$) and LAPIMO ($\mathscr{F}$). This table shows results for CoDAGANs with backbones based on MUNIT [Huang et al., 2018] (***CoDA***$_M$) and UNIT [Liu et al., 2017] (***CoDA***$_U$), as well as Domain-to-Domain approaches based on these architectures ($D2D_M$ and $D2D_U$), Pretrained U-Nets [Ronneberger et al., 2015] and U-Nets trained from scratch on the limited target labels.

Experiments		$E_{0\%}$	$E_{2.5\%}$	$E_{5\%}$	$E_{10\%}$	$E_{50\%}$	$E_{100\%}$
% Labels INbreast ($\mathscr{A}$)		100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
% Labels MIAS ($\mathscr{B}$)		0.00%	2.50%	5.00%	10.00%	50.00%	100.00%
% Labels DDSM_BC ($\mathscr{C}$)		0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
% Labels DDSM_A ($\mathscr{D}$)		0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
% Labels BCDR ($\mathscr{E}$)		0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
% Labels LAPIMO ($\mathscr{F}$)		0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
($\mathscr{A}$)	***CoDA***$_M$	98.69 ± 0.06	98.48 ± 0.15	98.59 ± 0.09	97.98 ± 0.36	**98.27 ± 0.41**	**98.11 ± 0.20**
	CoDA$_U$	98.29 ± 0.13	98.12 ± 0.10	98.37 ± 0.15	97.79 ± 0.64	97.89 ± 0.27	98.04 ± 0.18
	D2D$_M$ **($\mathscr{A}$)→($\mathscr{B}$)**	98.90 ± 0.09	**98.93 ± 0.16**	98.27 ± 0.45	98.36 ± 0.82	97.36 ± 1.60	85.89 ± 7.74
	D2D$_U$ **($\mathscr{A}$)→($\mathscr{B}$)**	98.74 ± 0.14	98.26 ± 0.31	**98.92 ± 0.19**	**98.65 ± 0.09**	97.80 ± 0.52	95.01 ± 1.51
	D2D$_M$ **($\mathscr{A}$)→($\mathscr{C}$)**	99.00 ± 0.08	–	–	–	–	–
	D2D$_U$ **($\mathscr{A}$)→($\mathscr{C}$)**	98.90 ± 0.09	–	–	–	–	–
	D2D$_M$ **($\mathscr{A}$)→($\mathscr{D}$)**	98.87 ± 0.15	–	–	–	–	–
	D2D$_U$ **($\mathscr{A}$)→($\mathscr{D}$)**	98.83 ± 0.14	–	–	–	–	–
	D2D$_M$**($\mathscr{A}$)→($\mathscr{E}$)**	**99.02 ± 0.05**	–	–	–	–	–
	D2D$_U$ **($\mathscr{A}$)→($\mathscr{E}$)**	98.90 ± 0.05	–	–	–	–	–
	D2D$_M$ **($\mathscr{A}$)→($\mathscr{F}$)**	98.11 ± 1.65	–	–	–	–	–
	D2D$_U$ **($\mathscr{A}$)→($\mathscr{F}$)**	98.69 ± 0.15	–	–	–	–	–
	From Scratch in ($\mathscr{A}$)	98.75 ± 0.11	–	–	–	–	–
($\mathscr{B}$)	***CoDA***$_M$	68.96 ± 1.57	88.13 ± 3.81	91.97 ± 0.68	93.11 ± 2.01	96.53 ± 0.45	97.19 ± 0.10
	CoDA$_U$	69.72 ± 0.30	90.12 ± 0.60	91.97 ± 3.61	95.28 ± 0.25	95.86 ± 0.50	97.21 ± 0.15
	D2D$_M$ **($\mathscr{A}$)→($\mathscr{B}$)**	5.02 ± 0.21	69.26 ± 14.74	5.91 ± 2.64	13.78 ± 8.17	50.20 ± 20.80	58.70 ± 29.26
	D2D$_U$ **($\mathscr{A}$)→($\mathscr{B}$)**	9.63 ± 2.49	15.72 ± 2.39	66.02 ± 26.18	90.70 ± 1.82	95.93 ± 0.77	80.00 ± 13.50
	Pretrained ($\mathscr{A}$)→($\mathscr{B}$)	**75.53**	**91.94 ± 0.28**	93.21 ± 0.35	94.06 ± 1.66	96.64 ± 0.13	**97.40 ± 0.08**
	From Scratch in ($\mathscr{B}$)	–	91.38 ± 0.28	**93.27 ± 0.25**	**95.69 ± 0.22**	**97.10 ± 0.25**	97.37 ± 0.15

used as source dataset, providing 100% of its labels in all experiments. MIAS was used as both source ($E_{0\%}$) and target ($E_{2.5\%}$ to $E_{100\%}$) dataset, depending on the label configuration of the experiment. As DDSM does not possess pixel-level labels, we created some ground truths only for a small subset of images from this dataset for the pectoral muscle segmentation task in order to objectively evaluate the UDA. One should notice that these ground truths were used only on the test procedure, but not in training, as all cases presented in Tables 5.2 and 5.3 show DDSM with 0% of labeled data. Thus DDSM is used only as a source dataset in our experiments. Breast region segmentation analysis on DDSM was only performed qualitatively, as there are no ground truths for this task.

5.2.1.1 Pectoral Muscle Segmentation in MXR Images

For the completely unlabeled case $E_{0\%}$ in pectoral muscle segmentation, ***CoDA***$_M$ and ***CoDA***$_U$ achieved $\mathcal{J}$ values of 67.61% and 60.01% for the MIAS target dataset, while the best baseline achieved 41.06%. SSDA and FSDA experiments ($E_{2.5\%}$ to $E_{100\%}$) regarding the MIAS dataset show that CoDAGANs achieve considerably better results than all baselines in all but one case. These results evidenced the higher instability of training pairwise translation architectures compared to conditional training. Across the training procedure, Jaccard values for D2D fluctuated by several percentage units, yielding standard deviations of one magnitude or more larger than CoDAGANs.

In the case of pectoral muscle for DDSM B/C ($\mathscr{C}$), UDA using $\boldsymbol{CoDA_M}$ and $\boldsymbol{CoDA_U}$ achieved 89.99% and 82.45%, with the D2D baseline achieving worse than random results, evidencing its lack of capability to translate between domains ($\mathscr{A}$) and ($\mathscr{C}$). The best baseline in this case was simply the use of Pretrained DNNs in ($\mathscr{A}$) and testing on ($\mathscr{C}$), which achieved 78.22%. Segmentation results for DDSM A ($\mathscr{D}$) were considerably worse for all methods and experiments, as samples from this subset of images showed an extremely lower contrast compared to the samples of DDSM B/C. Even in this suboptimal case, CoDAGANs achieved much better results than the baseline in UDA. Preprocessing using adaptive histogram equalization in DDSM A ($\mathscr{C}$) samples might improve results, although more empirical evidence is required. As there were only few samples labeled from ($\mathscr{C}$) and ($\mathscr{D}$), only UDA was possible for these datasets in D2D and pretrained baselines, as all labels were kept for testing. However, one can easily see that experiments $E_{2.5\%}$ to $E_{100\%}$ show better results in ($\mathscr{C}$) and ($\mathscr{D}$) as the number of labels from ($\mathscr{B}$) increases, achieving a $\mathcal{J}$ of 79.08% with all ($\mathscr{B}$) labels being used in training. This is due to two factors: 1) the larger number of labels achieved with the combination of ($\mathscr{A}$) and ($\mathscr{B}$); and 2) the more similar visual patterns between ($\mathscr{B}$)→($\mathscr{C}$) and ($\mathscr{B}$)→($\mathscr{D}$). Similarly to DDSM ($\mathscr{C}$)–($\mathscr{D}$), MIAS ($\mathscr{B}$) is an older originally analog that was later digitized, while INbreast ($\mathscr{A}$) is a Full Field Digital Mammography (FFDM) dataset.

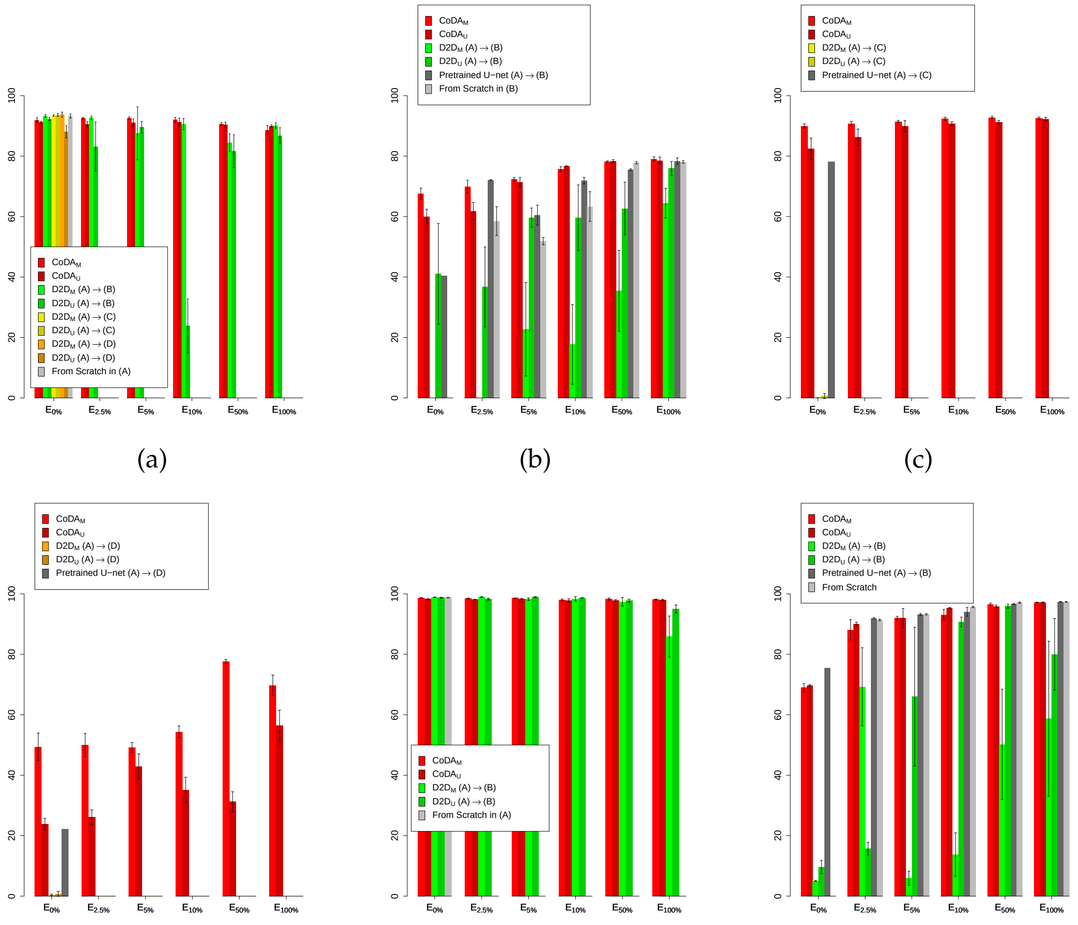
5.2.1.2 Breast Region Segmentation in MXR Images

Breast region segmentation (Table 5.3) proved to be an easier task, with most methods achieving Jaccard values higher than 90%. Pretrained DNNs and From Scratch training in SSDA scenarios achieved superior results in breast region segmentation for all experiments in the target MIAS ($\mathscr{B}$) dataset, followed closely by CoDAGANs. D2D, however, grossly underperformed in this relatively easy task for all experiments, reiterating this strategy's instability during training.

The marginally lower performance of CoDAGANs in this task can be attributed to the high transferrability of pretrained models, as can be seen in experiment $E_{0\%}$, where pretrained models with no Fine-tuning already achieved a $\mathcal{J}$ value of 75.53%. This easier DA task also benefits from the higher capability of U-Nets to segment details using skip connections between symmetric layers. As CoDAGANs remove the first layers of U-Net's Encoder to fit the smaller spatial dimensions of the isomorphic representation, the last layers of the network do not receive skip connections from the first layers, allowing for fine object details to be lost. This can be seen as a compromise between generalization capability and fine segmentation details.

5.2.1.3 MXR Segmentation Confidence Intervals

Figure 5.3 show the $\mathcal{J}$ values from Tables 5.2 and 5.3 with confidence intervals for $p \leq 0.05$ using a t-Student distribution.



(a) (b) (c)

(d) (e) (f)

Figure 5.3. Confidence Intervals for MXRs according to the values shown in Tables 5.2 and 5.3. Methods shown in these plots include CoDAGANs ($CoDA_M$ and $CoDA_U$), Domain-To-Domain translation ($D2D_M$ and $D2D_U$), pretrained U-Nets and U-Nets trained from scratch for SSDA and FSDA.

A first noticeable trait in Figures 5.3(a) and 5.3(e) is that CoDAGANs maintained their capability to perform inference on the INbreast source dataset for both pectoral muscle and breast region experiments when labels from other sources are added to the procedure. D2D tends to get more unstable when the plots get closer to FSDA ($E_{100\%}$) due to the incongruities in labeling styles from the different datasets.

Figures 5.3(b), 5.3(c) and 5.3(d) clearly show that CoDAGANs outperforms all baselines in UDA for the MIAS ($\mathscr{B}$), DDSM B/C ($\mathscr{C}$) and DDSM A ($\mathscr{D}$) datasets by a large margin for pectoral muscle segmentation. All of these discrepancies between CoDAGANs and baselines are statistically significant, showing a clear superiority of CoDAGANs in UDA scenarios in this task. Another important result is that Figure 5.3(d) shows a clear increase in the performance of $\boldsymbol{CoDA_M}$ on dataset ($\mathscr{D}$) when more labels from dataset ($\mathscr{B}$) were allowed to be used – that is, in results close to fully supervised learning with labels from ($\mathscr{B}$). Figure 5.3(f) show the UDA, SSDA and FSDA results for CoDAGANs and baselines on the target MIAS ($\mathscr{B}$) dataset in the task of breast region segmentation. CoDAGANs yield considerably higher results than D2D, even though a Pretrained U-Net surpassed all methods in this task for UDA. Domain shifts between the MIAS and INbreast datasets are probably considerably small. Pretrained U-Nets might not be universally better than CoDAGANs in UDA, though, as it is usually unable to compensate for large domain shifts. This trend was shown in Figures 5.3(b), 5.3(c) and 5.3(d) and will be further reinforced in Section 5.2.2.

5.2.2 Quantitative Results for CXR Samples

CXR results can be seen in Tables 5.4 and 5.5 for lungs, heart and clavicle segmentations. The JSRT ($\mathscr{A}$), OpenIST ($\mathscr{B}$), Shenzhen ($\mathscr{C}$), Montgomery ($\mathscr{D}$) and Chest X-Ray 8 ($\mathscr{E}$) datasets are objectively evaluated in the lung field segmentation task, as shown in Table 5.4, while PadChest ($\mathscr{F}$), NLMCXR ($\mathscr{G}$) and OCT CXR ($\mathscr{H}$) do not possess pixel-level ground truths for quantitative assessment. In heart and clavicle segmentation, apart from the source JSRT ($\mathscr{A}$) dataset, only OpenIST ($\mathscr{B}$) contains a subset of 15 labeled samples for these two task. Therefore, we reserved the labeled samples for testing and trained on the remaining samples for UDA quantitative assessment, as shown in Table 5.5. Analogously to Section 5.2.1, bold values in Tables 5.4 and 5.5 represent the best overall results in a given label configuration for a specific dataset.

Table 5.4. $\mathcal{J}$ results (in %) for lung field segmentation DA to and/or from eight distinct CXR datasets: JSRT ($\mathscr{A}$), OpenIST ($\mathscr{B}$), Shenzhen ($\mathscr{C}$), Montgomery ($\mathscr{D}$), Chest X-ray 8 ($\mathscr{E}$), PadChest ($\mathscr{F}$), NLMCXR ($\mathscr{G}$) and OCT CXR ($\mathscr{H}$). This table shows results for CoDAGANs with backbones based on MUNIT [Huang et al., 2018] ($\boldsymbol{CoDA_M}$) and UNIT [Liu et al., 2017] ($\boldsymbol{CoDA_U}$), as well as Domain-to-Domain approaches based on these architectures ($D2D_M$ and $D2D_U$), Pretrained U-Nets [Ronneberger et al., 2015] and U-Nets trained from scratch on the limited target labels.

Experiments		$E_{0\%}$	$E_{2.5\%}$	$E_{5\%}$	$E_{10\%}$	$E_{50\%}$	$E_{100\%}$
% Labels JSRT ($\mathscr{A}$)		100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
% Labels OpenIST ($\mathscr{B}$)		0.00%	2.50%	5.00%	10.00%	50.00%	100.00%
% Labels Shenzhen ($\mathscr{C}$)		0.00%	2.50%	5.00%	10.00%	50.00%	100.00%
% Labels Montgomery ($\mathscr{D}$)		0.00%	2.50%	5.00%	10.00%	50.00%	100.00%
% Labels ChestX-Ray8 ($\mathscr{E}$)		0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
% Labels PadChest ($\mathscr{F}$)		0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
% Labels NLMCXR ($\mathscr{G}$)		0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
% Labels OCT CXR ($\mathscr{H}$)		0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
($\mathscr{A}$)	**$CoDA_M$**	95.27 ± 0.07	94.08 ± 0.44	94.41 ± 0.86	94.74 ± 0.12	94.87 ± 0.39	95.31 ± 0.17
	$CoDA_U$	95.55 ± 0.07	95.16 ± 0.12	94.77 ± 0.17	94.83 ± 0.08	94.56 ± 0.99	94.97 ± 0.62
	$D2D_M$ ($\mathscr{A}$)→($\mathscr{B}$)	96.39 ± 0.06	**96.51 ± 0.07**	**96.54 ± 0.07**	**96.43 ± 0.03**	93.96 ± 4.85	**96.34 ± 0.12**
	$D2D_U$ ($\mathscr{A}$)→($\mathscr{B}$)	96.16 ± 0.35	96.41 ± 0.06	96.43 ± 0.04	96.26 ± 0.17	96.22 ± 0.29	96.20 ± 0.19
	$D2D_M$ ($\mathscr{A}$)→($\mathscr{C}$)	**96.45 ± 0.02**	96.44 ± 0.05	96.07 ± 0.61	**96.43 ± 0.10**	**96.29 ± 0.07**	96.20 ± 0.10
	$D2D_U$ ($\mathscr{A}$)→($\mathscr{C}$)	96.10 ± 0.51	96.34 ± 0.03	96.04 ± 0.35	96.33 ± 0.07	96.23 ± 0.04	96.33 ± 0.06
	$D2D_M$ ($\mathscr{A}$)→($\mathscr{D}$)	96.23 ± 0.21	96.22 ± 0.08	96.20 ± 0.09	96.24 ± 0.09	96.16 ± 0.21	95.84 ± 0.70
	$D2D_U$ ($\mathscr{A}$)→($\mathscr{D}$)	96.26 ± 0.11	96.21 ± 0.06	96.22 ± 0.13	96.21 ± 0.17	96.22 ± 0.10	96.02 ± 0.13
	$D2D_M$ ($\mathscr{A}$)→($\mathscr{E}$)	96.35 ± 0.19	–	–	–	–	–
	$D2D_U$ ($\mathscr{A}$)→($\mathscr{E}$)	96.42 ± 0.09	–	–	–	–	–
	$D2D_M$ ($\mathscr{A}$)→($\mathscr{F}$)	96.38 ± 0.15	–	–	–	–	–
	$D2D_U$ ($\mathscr{A}$)→($\mathscr{F}$)	96.30 ± 0.09	–	–	–	–	–
	$D2D_M$ ($\mathscr{A}$)→($\mathscr{G}$)	96.11 ± 0.65	–	–	–	–	–
	$D2D_U$ ($\mathscr{A}$)→($\mathscr{G}$)	96.34 ± 0.09	–	–	–	–	–
	$D2D_M$ ($\mathscr{A}$)→($\mathscr{H}$)	96.24 ± 0.56	–	–	–	–	–
	$D2D_U$ ($\mathscr{A}$)→($\mathscr{H}$)	95.91 ± 1.02	–	–	–	–	–
	From Scratch in ($\mathscr{A}$)	95.70 ± 0.06	–	–	–	–	–
($\mathscr{B}$)	**$CoDA_M$**	90.67 ± 0.80	92.58 ± 0.59	92.83 ± 1.25	**93.41 ± 0.49**	93.65 ± 1.07	94.71 ± 0.15
	$CoDA_U$	**91.03 ± 0.96**	92.08 ± 0.37	**93.50 ± 0.41**	93.32 ± 0.16	94.23 ± 0.41	94.63 ± 0.25
	$D2D_M$ ($\mathscr{A}$)→($\mathscr{B}$)	19.67 ± 28.59	**92.79 ± 1.68**	93.41 ± 0.65	92.15 ± 1.26	93.22 ± 1.79	93.46 ± 1.35
	$D2D_U$ ($\mathscr{A}$)→($\mathscr{B}$)	56.82 ± 31.88	70.11 ± 12.05	88.87 ± 5.33	61.40 ± 28.25	93.94 ± 0.82	**94.95 ± 0.50**
	Pretrained ($\mathscr{A}$)→($\mathscr{B}$)	7.48	83.91 ± 0.17	90.37 ± 0.15	92.47 ± 0.16	94.37 ± 0.17	94.88 ± 0.06
	From Scratch in ($\mathscr{B}$)	–	85.69 ± 0.33	88.94 ± 0.56	91.70 ± 0.22	**94.87 ± 0.10**	94.35 ± 0.12
($\mathscr{C}$)	**$CoDA_M$**	88.69 ± 0.46	**89.88 ± 0.36**	**90.75 ± 0.49**	90.64 ± 0.23	90.99 ± 0.89	91.84 ± 0.09
	$CoDA_U$	**88.99 ± 0.29**	89.74 ± 0.12	89.90 ± 0.61	90.04 ± 0.41	91.35 ± 0.49	91.61 ± 0.49
	$D2D_M$ ($\mathscr{A}$)→($\mathscr{C}$)	70.01 ± 8.67	89.45 ± 0.97	83.46 ± 10.87	**91.42 ± 0.57**	91.61 ± 0.68	92.03 ± 0.80
	$D2D_U$ ($\mathscr{A}$)→($\mathscr{C}$)	55.40 ± 33.75	82.72 ± 4.42	80.83 ± 11.11	89.02 ± 3.47	91.82 ± 0.31	91.99 ± 0.19
	Pretrained ($\mathscr{A}$)→($\mathscr{C}$)	17.19	88.68 ± 0.16	90.82 ± 0.07	91.60 ± 0.10	92.17 ± 0.15	**92.40 ± 0.03**
	From Scratch in ($\mathscr{C}$)	–	89.62 ± 0.20	90.74 ± 0.35	91.79 ± 0.08	**92.32 ± 0.04**	92.25 ± 0.05
($\mathscr{D}$)	**$CoDA_M$**	81.88 ± 1.35	**87.86 ± 0.88**	87.72 ± 1.60	90.48 ± 0.64	93.15 ± 0.44	94.19 ± 0.31
	$CoDA_U$	**84.58 ± 1.48**	87.12 ± 0.59	87.07 ± 0.78	87.75 ± 0.81	92.76 ± 2.27	92.95 ± 1.83
	$D2D_M$ ($\mathscr{A}$)→($\mathscr{D}$)	30.20 ± 26.08	82.60 ± 4.05	**88.34 ± 2.99**	80.48 ± 8.89	93.46 ± 0.81	93.51 ± 0.99
	$D2D_U$ ($\mathscr{A}$)→($\mathscr{D}$)	79.44 ± 5.64	64.61 ± 8.95	76.89 ± 1.52	82.33 ± 5.01	94.02 ± 0.27	94.23 ± 0.66
	Pretrained ($\mathscr{A}$)→($\mathscr{D}$)	10.79	81.47 ± 0.05	86.79 ± 0.07	89.60 ± 0.14	94.40 ± 0.07	94.82 ± 0.05
	From Scratch in ($\mathscr{D}$)	–	76.32 ± 0.27	87.12 ± 0.20	**90.91 ± 0.25**	**94.66 ± 0.09**	**95.19 ± 0.12**
($\mathscr{E}$)	**$CoDA_M$**	67.91 ± 5.34	72.98 ± 1.34	74.77 ± 3.71	72.84 ± 2.87	76.41 ± 5.34	82.53 ± 4.61
	$CoDA_U$	73.39 ± 1.45	70.82 ± 2.58	67.14 ± 3.12	67.79 ± 6.79	74.36 ± 10.64	71.50 ± 16.42
	$D2D_M$ ($\mathscr{A}$)→($\mathscr{E}$)	**75.48 ± 5.07**	–	–	–	–	–
	$D2D_U$ ($\mathscr{A}$)→($\mathscr{E}$)	75.26 ± 1.20	–	–	–	–	–
	Pretrained ($\mathscr{A}$)→($\mathscr{E}$)	19.50	–	–	–	–	–

Table 5.5. $\mathcal{J}$ results (in %) for heart and clavicle segmentation DA to and/or from eight distinct CXR datasets: JSRT ($\mathscr{A}$), OpenIST ($\mathscr{B}$), Shenzhen ($\mathscr{C}$), Montgomery ($\mathscr{D}$), Chest X-ray 8 ($\mathscr{E}$), PadChest ($\mathscr{F}$), NLMCXR ($\mathscr{G}$) and OCT CXR ($\mathscr{H}$). This table shows results for CoDAGANs with backbones based on MUNIT [Huang et al., 2018] (***CoDA***$_{\boldsymbol{M}}$) and UNIT [Liu et al., 2017] (***CoDA***$_{\boldsymbol{U}}$), as well as Domain-to-Domain approaches based on these architectures ($D2D_M$ and $D2D_U$), Pretrained U-Nets [Ronneberger et al., 2015] and U-Nets trained from scratch on the limited target labels.

Experiments		**$E_{0\%}$ (Heart)**	**$E_{0\%}$ (Clavicles)**
% Labels JSRT ($\mathscr{A}$)		100.00%	100.00%
% Labels OpenIST ($\mathscr{B}$)		0.00%	0.00%
% Labels Shenzhen ($\mathscr{C}$)		0.00%	0.00%
% Labels Montgomery ($\mathscr{D}$)		0.00%	0.00%
% Labels ChestX-Ray8 ($\mathscr{E}$)		0.00%	0.00%
% Labels PadChest ($\mathscr{F}$)		0.00%	0.00%
% Labels NLMCXR ($\mathscr{G}$)		0.00%	0.00%
% Labels OCT CXR ($\mathscr{H}$)		0.00%	0.00%
($\mathscr{A}$)	***CoDA***$_{\boldsymbol{M}}$	89.86 ± 0.29	77.31 ± 0.37
	CoDA$_{\boldsymbol{U}}$	89.89 ± 0.32	76.03 ± 1.06
	D2D$_M$ ($\mathscr{A}$)→($\mathscr{B}$)	90.68 ± 0.19	87.76 ± 0.18
	D2D$_U$ ($\mathscr{A}$)→($\mathscr{B}$)	90.97 ± 0.10	$\mathbf{87.96 \pm 0.14}$
	D2D$_M$ ($\mathscr{A}$)→($\mathscr{C}$)	91.16 ± 0.18	87.20 ± 0.22
	D2D$_U$ ($\mathscr{A}$)→($\mathscr{C}$)	90.70 ± 0.48	87.09 ± 0.61
	D2D$_M$ ($\mathscr{A}$)→($\mathscr{D}$)	90.65 ± 0.17	84.78 ± 0.53
	D2D$_U$ ($\mathscr{A}$)→($\mathscr{D}$)	90.67 ± 0.23	84.46 ± 1.24
	D2D$_M$ ($\mathscr{A}$)→($\mathscr{E}$)	90.97 ± 0.20	87.38 ± 0.21
	D2D$_U$ ($\mathscr{A}$)→($\mathscr{E}$)	91.11 ± 0.12	87.02 ± 0.63
	D2D$_M$ ($\mathscr{A}$)→($\mathscr{F}$)	89.74 ± 2.32	87.63 ± 0.55
	D2D$_U$ ($\mathscr{A}$)→($\mathscr{F}$)	90.63 ± 0.37	87.39 ± 0.47
	D2D$_M$ ($\mathscr{A}$)→($\mathscr{G}$)	91.15 ± 0.18	87.59 ± 0.06
	D2D$_U$ ($\mathscr{A}$)→($\mathscr{G}$)	90.90 ± 0.10	87.37 ± 0.59
	D2D$_M$ ($\mathscr{A}$)→($\mathscr{H}$)	$\mathbf{91.33 \pm 0.16}$	86.46 ± 0.62
	D2D$_U$ ($\mathscr{A}$)→($\mathscr{H}$)	90.22 ± 0.94	84.49 ± 0.64
	From Scratch in ($\mathscr{A}$)	88.91 ± 0.48	76.07 ± 0.41
($\mathscr{B}$)	***CoDA***$_{\boldsymbol{M}}$	$\mathbf{64.63 \pm 1.28}$	61.94 ± 1.03
	CoDA$_{\boldsymbol{U}}$	63.71 ± 0.95	57.59 ± 1.26
	D2D$_M$ ($\mathscr{A}$)→($\mathscr{B}$)	54.91 ± 2.35	67.11 ± 0.96
	D2D$_U$ ($\mathscr{A}$)→($\mathscr{B}$)	64.50 ± 0.84	$\mathbf{68.53 \pm 0.89}$
	Pretrained ($\mathscr{A}$)→($\mathscr{B}$)	0.0 ± 0.0	0.24 ± 0.40

5.2.2.1 Lung Segmentation in CXR Images

In the task of lung segmentation in CXRs (Table 5.4), baselines showed considerably poor results for target datasets ($\mathscr{B}$)-($\mathscr{D}$) in UDA experiments. Following the results from Sections 5.2.1.1 and 5.2.1.2, D2D with a small amount of target labels proved to be highly unstable, yielding worse results and considerably higher standard deviations, when compared with CoDAGANs. $\boldsymbol{CoDA_M}$ and $\boldsymbol{CoDA_U}$ achieve the best UDA results in ($\mathscr{B}$), ($\mathscr{C}$) and ($\mathscr{D}$), surpassing all baselines by a considerable margin, yielding $\mathcal{J}$ values of 91.03%, 88.99% and 84.58% for these three datasets, respectively. Pretrained U-Nets yielded worse than random results in these tasks, which can be expĺained by the high domain shift across ($\mathscr{A}$)→($\mathscr{B}$), ($\mathscr{A}$)→($\mathscr{C}$) and ($\mathscr{A}$)→($\mathscr{D}$).

CoDAGANs maintain state-of-the-art results in SSDA experiments with small amount of labels, surpassing baselines in most datasets for $E_{2.5\%}$ and $E_{5\%}$. In $E_{50\%}$ and $E_{100\%}$ state-of-the-art results are achieved mainly by From Scratch training in the target domain due to label abundance. Similarly, D2D methods are only able to achieve stable results, after $E_{10\%}$. As in MXRs, D2D underperformed in UDA settings compared to CoDAGANs, even though it presented considerably better results than Pretrained DNNs.

We also show that the source dataset presented little to no deterioration in segmentation quality when segmented by CoDAGANs compared to D2D and From Scratch training on ($\mathscr{A}$). D2D from translations ($\mathscr{A}$)→($\mathscr{B}$) to ($\mathscr{A}$)→($\mathscr{H}$) present remarkably similar results in UDA, SSDA and FSDA, achieving state-of-the-art results in all cases. It is noticeable that CoDAGANs achieved no superiority in the source domain, as it aims for generalization and does not focus in fine-grained segmentation. However, the difference of Jaccard values between CoDAGANs and baseline methods that only consider a pair of domains or even only the source domain (From Scratch) remained limited to between 1% and 2%.

5.2.2.2 Heart and Clavicle Segmentation in CXR Images

As shown in Table 5.5, heart and clavicle segmentation proved to be harder tasks than lung field segmentation. Both tasks only count with the JSRT dataset as fully labeled, with OpenIST having only 15 images with pixel-level annotations for both heart and clavicles. We therefore used these samples only for evaluating UDA in a target dataset, as the small number of samples would not allow for proper SSDA and FSDA experiments. $CoDA_M$ achieved the best results in heart segmentation on OpenIST with a $\mathcal{J}$ value of 64.63%, closely followed by $D2D_U$ with 64.50%. Clavicle segmentation topped on 68.53% for D2D and was the only task that clearly showed an underperformance of CoDAGANs compared with D2D, achieving only 61.94%. Both D2D and CoDAGANs greatly surpassed the Pretrained U-Net in both tasks for

($\mathscr{B}$), with the pretrained baseline achieving close to 0% in Jaccard.

Table 5.5 also shows the remarkable stability of D2D for the source dataset ($\mathscr{A}$), evidencing that performing DA using Image Translation does not compromise performance in the source domain. CoDAGANs closely followed the performance of D2D in the source dataset ($\mathscr{A}$) for heart segmentation, but again showed considerably worse performance in clavicle segmentation. This underperformance of CoDAGANs in clavicle segmentation for both datasets is probably explained by the higher imbalance of this task. Clavicles cover a much smaller area in a CXR than lungs or a heart and, therefore, are more susceptible to low performance in segmentation DNNs that contain fewer skip connections, as the case of the truncated asymmetrical U-Net configured to receive data from the isomorphic representation $\mathcal{I}$ in CoDAGANs.

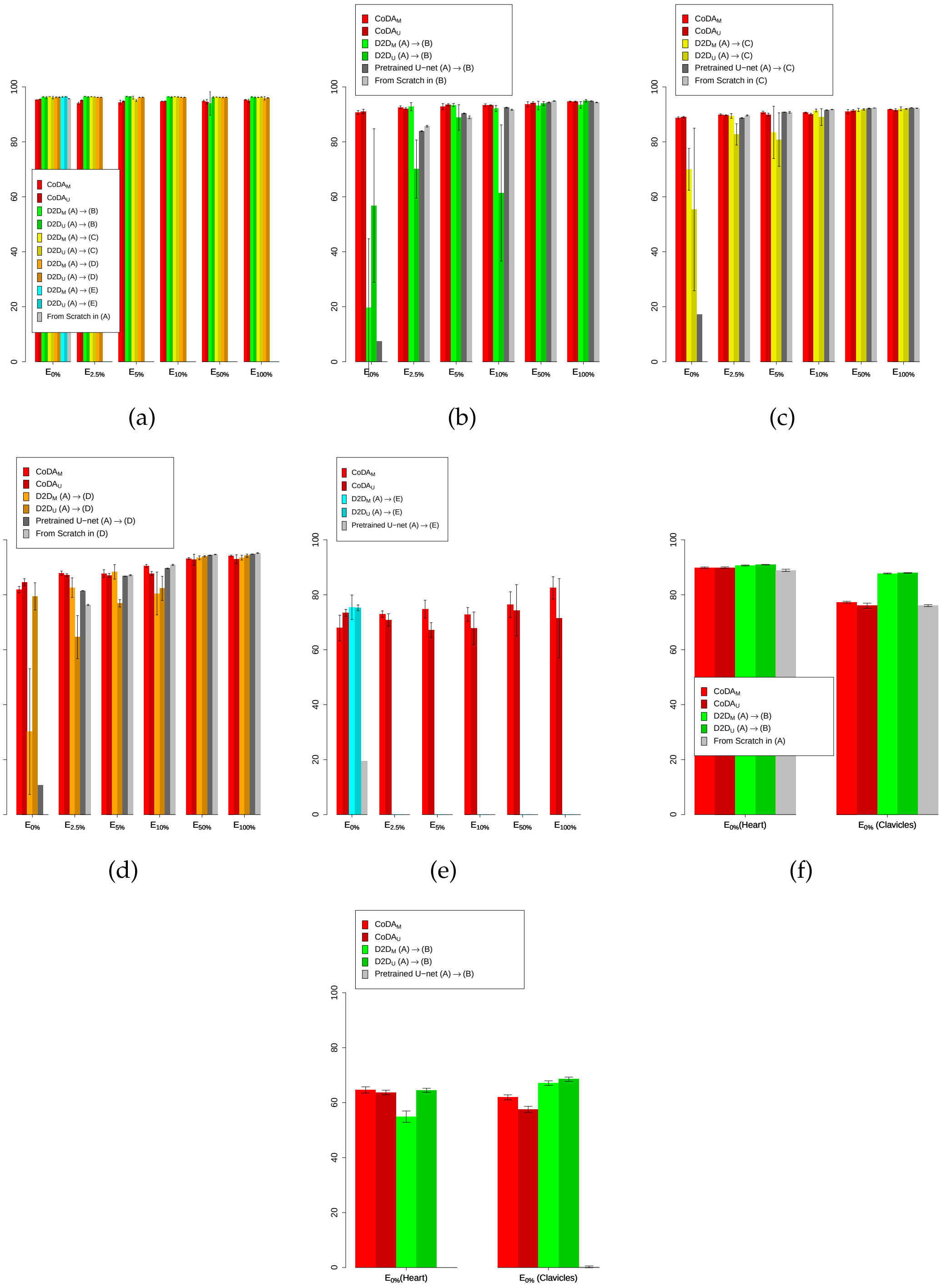
5.2.2.3 CXR Segmentation Confidence Intervals

Figure 5.4 shows the confidence intervals for $p \leq 0.05$ in lung segmentation for both the source JSRT dataset (Figure 5.4(a)) and the target image sets (Figures 5.4(b), 5.4(c), 5.4(d) and 5.4(e)). Figures 5.4(f) and 5.4(g) show the results for heart and clavicle segmentation in the source (JSRT) and target (OpenIST) datasets, respectively.

One can see by Figures 5.4(a) and 5.4(f) that segmentation in the source dataset is preserved even when labels from other datasets are introduced in the training procedure. Figures 5.4(b), 5.4(c), 5.4(d), 5.4(e) and 5.4(g) show the UDA, SSDA and FSDA efficiency of CoDAGANs in the fully or partially labeled target datasets, that is, OpenIST, Shenzhen, Montgomery and Chest X-Ray 8 for lung segmentation and only OpenIST for heart and clavicles.

Figures 5.4(b) through 5.4(e) and 5.4(g) show the progression of CoDAGAN and baseline methods in distinct target CXR datasets according to the different label configurations in our experimental procedure. While most methods converge to similar efficiencies in scenarios closer to FSDA ($E_{50\%}$ and $E_{100\%}$), baselines start considerably worse than CoDAGANs in most cases when there is scarcity of target labels ($E_{0\%}$ and $E_{2.5\%}$). $\boldsymbol{D2D_M}$ and $\boldsymbol{D2D_U}$ also yield highly unstable predictions in scenarios between these two extremities ($E_{5\%}$ and $E_{10\%}$), with much larger confidence intervals resulting from larger standard deviations than their counterparts.

Another interesting phenomenon can be seen in Figure 5.4(e), where CoDAGANs start worse in UDA than both $\boldsymbol{D2D_M}$ and $\boldsymbol{D2D_U}$ for target samples from dataset ($\mathscr{E}$), but $\boldsymbol{CoDA_M}$ improves as the experiments get closer to $E_{100\%}$. One



(a) (b) (c)

(d) (e) (f)

(g)

Figure 5.4. Confidence Intervals for CXRs according to the values shown in Tables 5.4 and 5.5. Methods shown in these plots include CoDAGANs ($CoDA_M$ and $CoDA_U$), Domain-To-Domain translation ($D2D_M$ and $D2D_U$), Pretrained U-Nets and U-Nets trained from scratch for SSDA and FSDA.

should notice that in experiments $E_{2.5\%}$ through $E_{100\%}$ no labels from ($\mathscr{E}$) are being used at any time during the training procedure of CoDAGANs, and even still the objective evaluation for dataset ($\mathscr{E}$) improves. This serves as yet another evidence that CoDAGANs are able to acquire semantic information for one dataset (Chest X-Ray 8) by using labels from others; in this case, JSRT, OpenIST, Montgomery and Shenzhen.

5.2.3 Qualitative DA Analysis

Figures 5.5, 5.6 and 5.7 show segmentation qualitative results for two tasks of MXR segmentation, two tasks of DXR segmentation and three tasks of CXR segmentation, respectively. Figure 5.5(a) presents predictions for pectoral muscle segmentation $E_{0\%}$, while Figure 5.5(b) shows breast region segmentation on experiment $E_{0\%}$. Experiment $E_{0\%}$ for lung field segmentation can be seen in Figure 5.7(a), while heart and clavicle segmentation DA experiments ($E_{0\%}$) are shown respectively on Figures 5.7(b) and 5.7(c). At last, DXR segmentations from UDA experiments can be seen in Figure 5.6 for both teeth (Figure 5.6(a)) and mandible (Figure 5.6(b)) segmentation. Columns for all figures present the original sample, the ground truth segmentation for the specific task for this sample when available and predictions from Pretrained U-Nets, D2D and CoDAGANs for visual comparison.

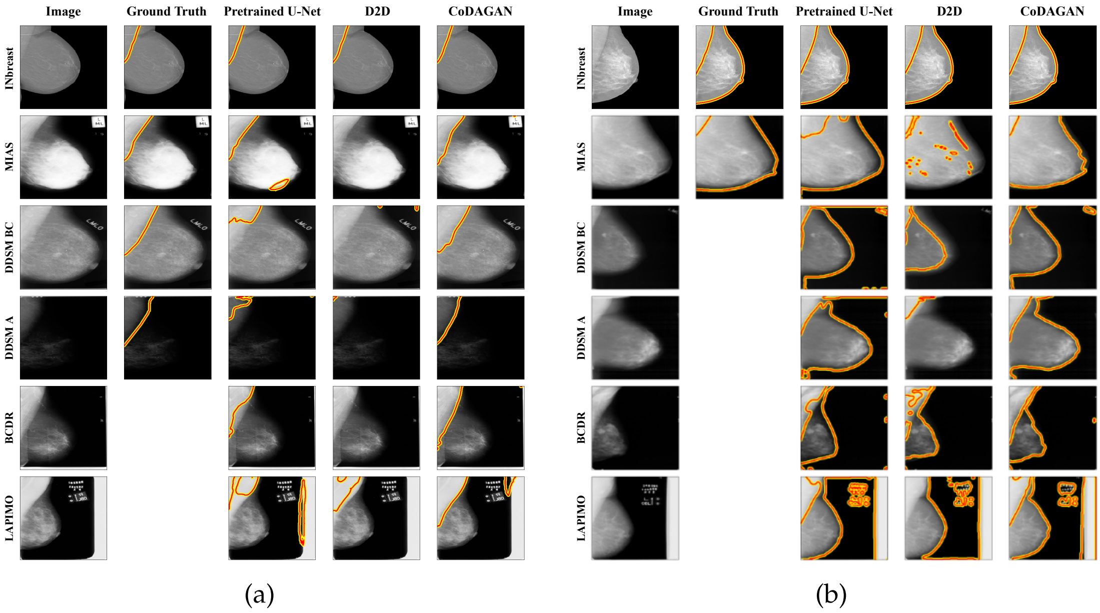
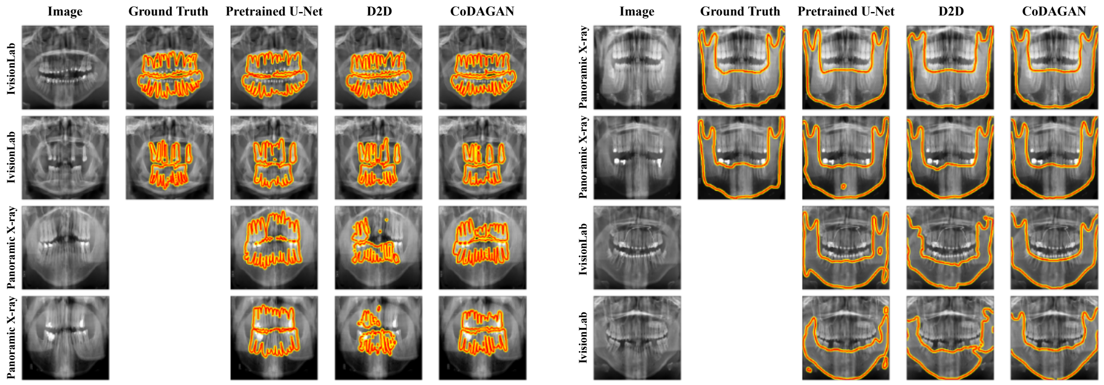


Figure 5.5. Qualitative segmentation results in MXR images for two distinct tasks: $\boldsymbol{E_{0\%}}$ pectoral (a) and $\boldsymbol{E_{0\%}}$ breast region (b).

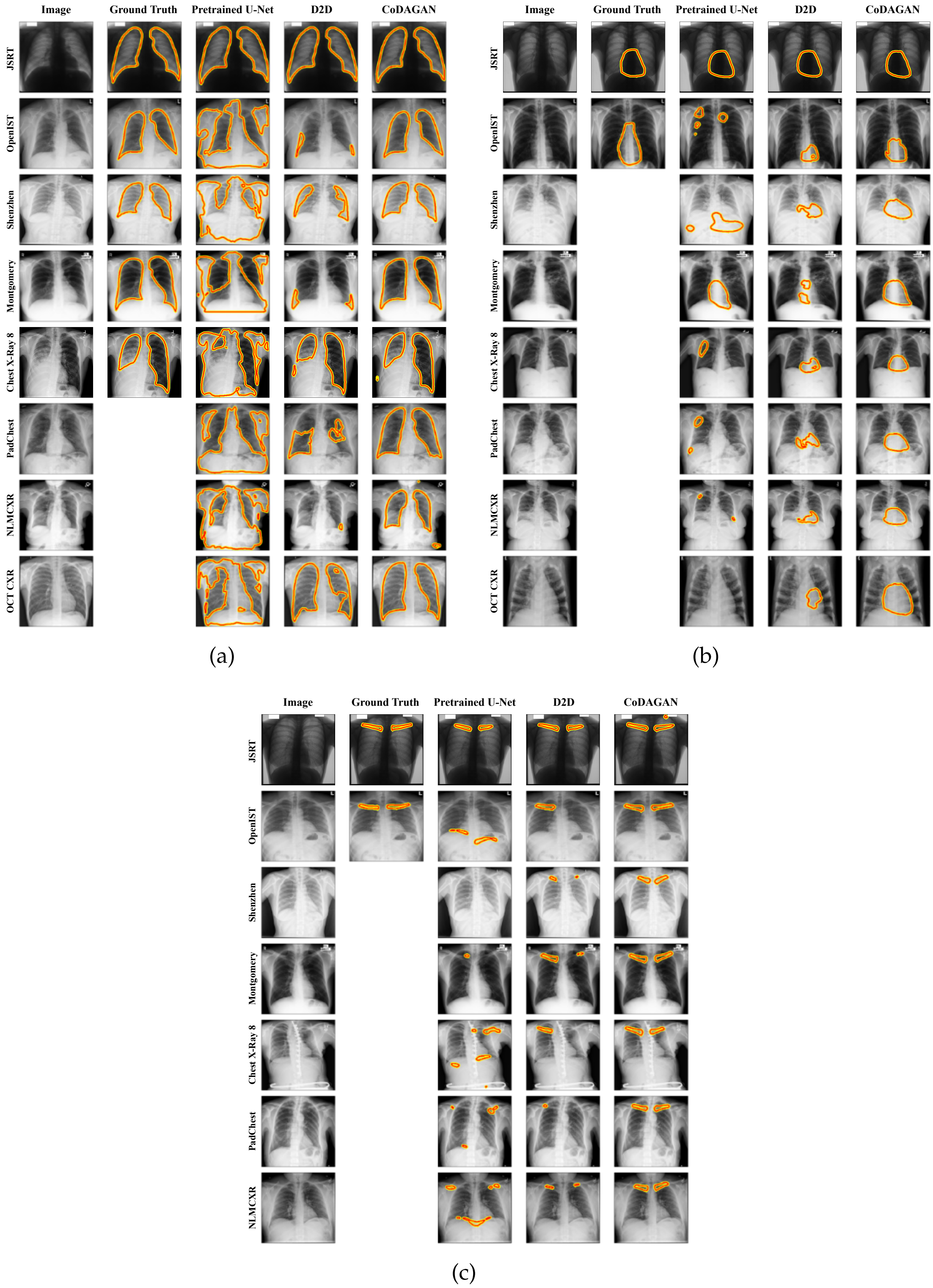


(a) (b)

Figure 5.6. Qualitative segmentation results in DXR images for two distinct tasks: $\boldsymbol{E}_{0\%}$ teeth (a) and $\boldsymbol{E}_{0\%}$ mandible (b).

Each row in Figures 5.5(a) and 5.5(b) highlights one sample from each one of the six MXR datasets used in our experiments. One can see in both figures that D2D underperformed in most cases, failing to predict any pectoral muscle pixel as positive in multiple samples from target datasets. UDA for breast region segmentation also proved to be a hard task for D2D, as in most samples it segmented either only the pectoral muscle or background. While in the pectoral muscle segmentation task most methods were able to successfully ignore the labels in the background of some digitized datasets such as DDSM, MIAS and LAPIMO, these artifacts were shown to be harder to compensate for on breast region segmentation, as all baselines and CoDAGANs wrongly and frequently segmented them as part of the breast. We observed overwhelmingly better results in our qualitative assessment from CoDAGANs, when compared to all other baselines. CoDAGAN superiority proved to be stable both in easier target datasets such as MIAS or BCDR and in more difficult ones as DDSM A and LAPIMO, which contain extremely low contrast and large digitization artifacts, respectively. At last, as the breast boundary contour is fuzzy and extremely hard to segment even for humans in non-FFDM datasets, all methods either underrepresent or overrepresent positive breast pixels in these regions in most samples and datasets.

Figure 5.6(a) shows teeth segmentation predictions for both source (IvisionLab) and target (Panoramic X-ray) datasets, while Figure 5.6(b) presents DXR mandible segmentations using Panoramic X-ray as source and IvisioLab as target. DXR results show that Pretrained U-Nets and D2D, as expected from a supervised setting, yield mostly predictions in the source dataset for both tasks. However, both methods underperform in the target datasets, missing the segmentation of several



(a)

(b)

(c)

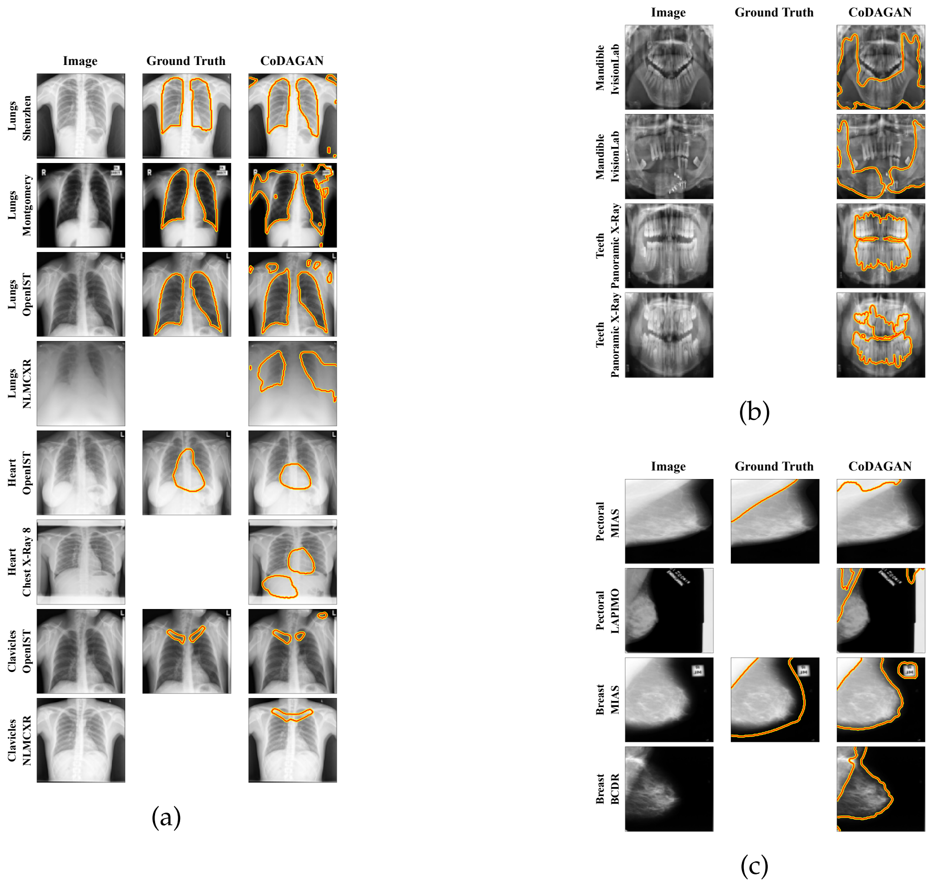
Figure 5.7. Qualitative segmentation results in CXR images for three distinct tasks: $\boldsymbol{E_{0\%}}$ lungs (a), $\boldsymbol{E_{0\%}}$ heart (b) and $\boldsymbol{E_{0\%}}$ clavicles (c).

teeth and mislabeling mandible regions as background. CoDAGANs achieve much more consistent results in the target datasets, once again evidencing the method's capabilities in UDA. However, CoDAGAN predictions were observed to be less robust for modeling sharp corners in the shapes probably due to the smaller spatial resolution of representation $\mathcal{I}$ when compared to the images themselves, which may lead to loss of small detail and slightly smoother shape contours. This issue might be fixed by passing the outputs of the encoder layers in $G_{\mathbb{E}}$ to the supervised model M in order to preserve spatial information, much like a skip connection does.

Figure 5.7(a) shows DA results for lung field segmentation in 4 fully labeled datasets (JSRT, OpenIST, Shenzhen and Montgomery), 1 partially labeled dataset (Chest X-Ray 8) and 3 other target unlabeled datasets (PadChest, NLMCXR and OCT CXR). We reiterate that one single CoDAGAN was trained for all datasets and made all predictions contained in the last column of Figure 5.7(a). One should notice that the target datasets in this case are considerably harder than the source ones due to poor image contrast, presence of unforeseen artifacts as pacemakers, rotation and scale differences and a much wider variety of lung sizes, shapes and health conditions. Yet, the DA procedure using CoDAGANs for lung segmentation was adequate for the vast majority of images, only presenting errors in distinctly difficult images. As the source dataset (JSRT) has completely distinct visual patterns when compared to the target datasets, both Pretrained U-Nets and D2D are not able to properly compensate for domain shift in these cases, yielding grossly wrong predictions.

Heart and clavicle segmentation (Figures 5.7(b) and 5.7(c)) are harder tasks than lung segmentation due to heart boundary fuzziness and a high variability of clavicle sizes, shapes and positions. In addition, clavicle segmentation is a highly unbalanced task. Those factors, paired with the fact that the well-behaved samples from the JSRT dataset are the only source of labels to this task contributed to higher segmentation error rates mainly in clavicle segmentation. Results for heart and clavicles are presented for the same 8 datasets as lung segmentation, but only a small subset of OpenIST contains labels for clavicles and heart. Even with all these hampers, CoDAGANs still yielded consistent prediction maps for hearts and clavicles across all target datasets, while baselines are, again, unable to compensate for domain shifts.

Figure 5.8 presents a visual assessment of segmentation errors in CXR (Figure 5.8(a)), DXR (Figure 5.8(b)) and MXR (Figure 5.8(c)) tasks for some samples of target datasets in UDA scenarios. A full assessment of results and both CoDAGAN and baseline errors can be seen in this project's webpage.



(a)

(b)

(c)

Figure 5.8. Noticeable errors in CoDAGAN UDA results for unlabeled target datasets in three domains: CXRs (a), DXRs (b) and MXRs (c).

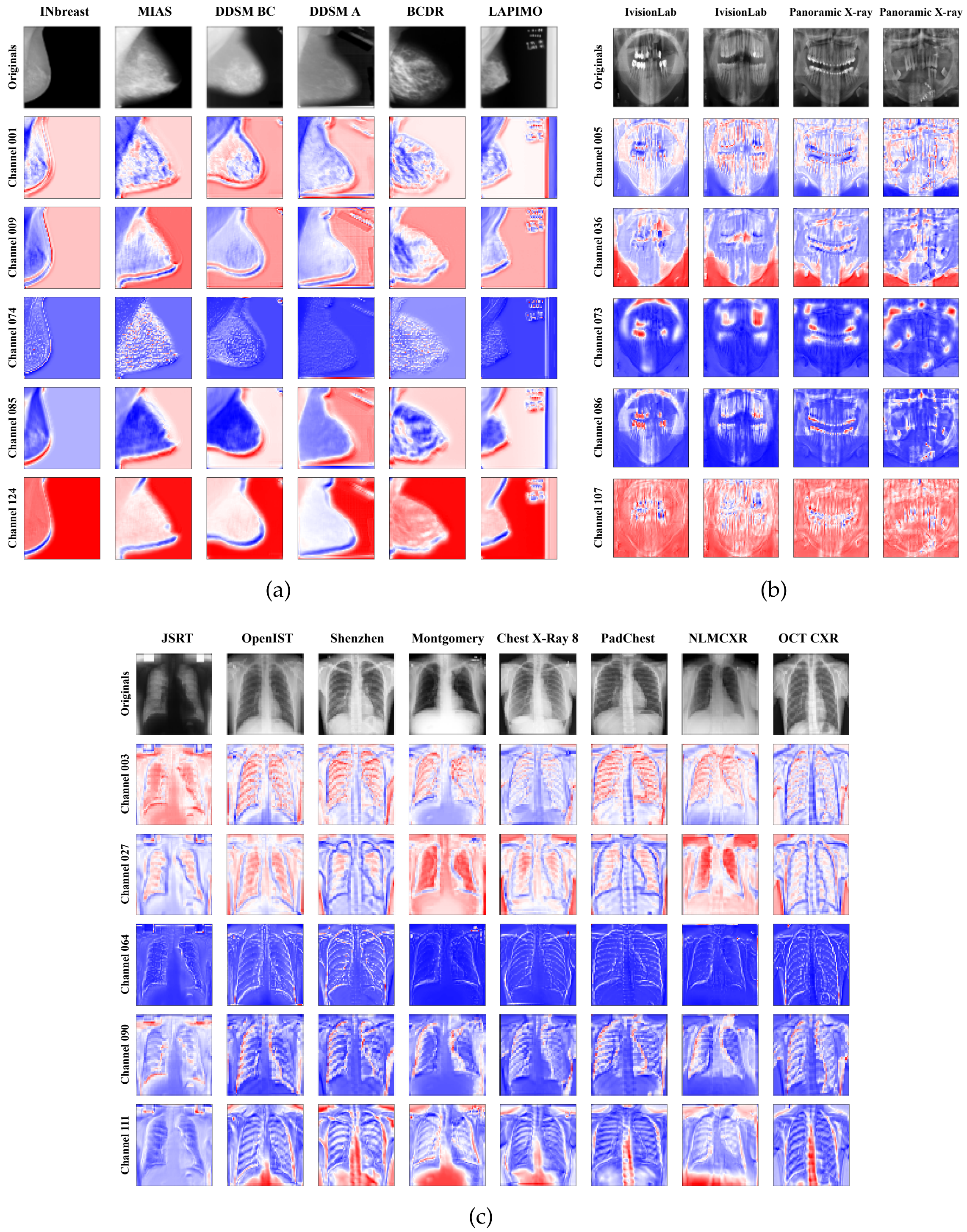
One can see that several lung predictions by CoDAGANs yielded small isles of false positives in other bony areas of CXRs (Figure 5.8(a)) as well as in the background of the images due to wrongly compensated domain shifts. While most of these errors can be corrected by simply filtering for keeping only the larger contiguous areas lung field segmentation and heart segmentation, this would be harder to implement for clavicles due to their smaller relative sizes in CXR exams. Extremely low contrast images as the NLMCXR sample presented in the fourth row of Figure 5.8(a) presented a challenge for CoDAGANs on all CXR tasks, being the most common source of missed predictions for our method.

We noticed that there were large inter-dataset labeling differences for all CXR tasks. For instance, several OpenIST heart labels contain larger heart delineations than JSRT labels, which led to a larger number of false negatives on OpenIST, as can be seen in the fifth row of Figure 5.8(a). Also, clavicle labels on JSRT delineate only pixels within lung borders, while OpenIST labels delineate the whole pair of bones both inside and outside the lung fields. We employed a binary mask between clavicles and lungs for each labeled OpenIST sample in order to fix this discrepancy in labeling characteristics.

Even though both Pretrained U-Nets and D2D yielded worse general results in DXR tasks, CoDAGANs still missed a considerable number of teeth, failed to separate the upper and lower dental arches and wrongfully split mandibles, as shown in Figure 5.8(b). At last, MXR prediction errors can be seen in Figure 5.8(c), mainly in denser breasts, which hamper the differentiation between pectoral muscle and breast tissue and due to fuzzy breast-boundary borders. Some of the non-FFDM datasets also contain digitization artifacts in the background, which were frequently misclassified as breast pixels. Therefore, there is still a lot of room for improvement in CoDAGAN's domain shift compensation capabilities.

Another important qualitative assessment to be performed in CoDAGANs is to visually assess that the same objects in distinct datasets are represented similarly in $\mathcal{I}$-space. This is shown in Figure 5.9 for five $\mathcal{I}$ activation channels in MXRs (Figure 5.9(a)), DXRs (Figure 5.9(b)) and CXRs (Figure 5.9(c)).

In Figure 5.9(a), high density tissue patterns and important object contours in the images from INbreast, MIAS, DDSM BC, DDSM A, BCDR and LAPIMO are encoded similarly by CoDAGANs. Breast boundaries are also visually similar across samples from all MXR datasets, as CoDAGANs are able to infer that these information is semantically similar despite the differences in the visual patterns of the images. Visual patterns that compose the patient's anatomical structures, such as ribs and lung contours, in Figure 5.9(c) are visibly similar in the samples from all eight CXR datasets: JSRT, OpenIST, Shenzhen, Montgomery, Chest X-Ray 8, PadChest, NLMCXR and OCT CXR. The third radiological domain used in our comparisons is composed of two different DXRs datasets: IvisionLab and Panoramic X-Ray (Figure 5.9(b)). It is easy to notice the common patterns encoded by CoDAGANs for the same semantic areas of the distinct images such as the teeth edges and mandible contours. One should notice that despite the clear visual distinctions between the original samples from the different datasets in all domains, the isomorphic representations were visually alike across samples from the domains. These results show that CoDAGANs successfully create a joint representation for high semantic-level information which encodes analogous visual patterns across datasets in a similar manner. In other words, different convolutional channels in $\mathcal{I}$ activate visual patterns with the same semantic information from the distinct datasets in a similar manner. This feature of encoding a joint distribution between domains by looking only to the marginal distributions of the samples is what allows CoDAGANs to perform UDA, SSDA and FSDA with high accuracy.



(a)

(b)

(c)

Figure 5.9. Original images and five different activation channels for samples of (a) MXRs, (b) DXRs and (c) CXRs. Readers should notice the visual distinctions between the original input samples and the visually similar encodings generated by the isomorphic representations of CoDAGANs for the same semantic content in different parts of the radiographs.

5.2.4 Low Dimensionality

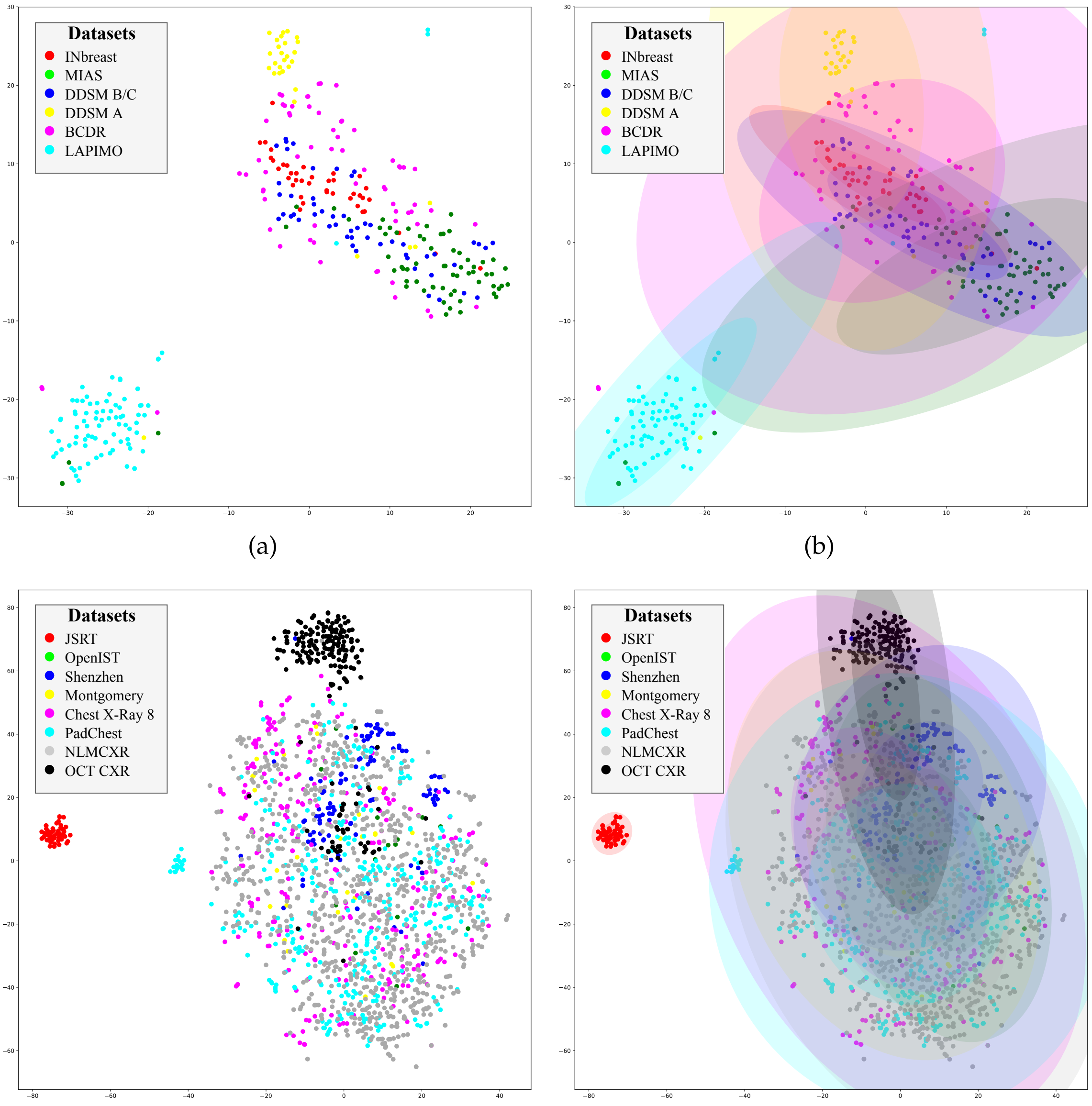
In order to view the data distributions of samples from the different datasets in the $\mathcal{I}$-space of CoDAGAN representations, we reduced the dimensionality of $\mathcal{I}$ to a 2D visualization using Principal Component Analysis (PCA) and the t-Distributed Stochastic Neighbor Embedding (t-SNE) algorithm [Maaten and Hinton, 2008]. First, in order to reduce computational requirements, we reduced the original $524,288$ dimensions of $\mathcal{I}$ to a 200-dimensional space using PCA and applied t-SNE on the remaining components. We also fit a gaussian on the data distributions for each dataset using the Gaussian Mixture Model (GMM) from sklearn[1]. The resulting 2D visualizations of the MXR and CXR datasets can be seen in Figure 5.10.

Figures 5.10(a) and 5.10(b) show respectively the original 2D representation of the $\mathcal{I}$-space from PCA/t-SNE and the GMM fit for the data on the MXR datasets. Visual analysis of Figure 5.10(b) shows the domain shifts between LAPIMO and the other MXR datasets. This is due to the fact that LAPIMO samples have a characteristic digitization artifact on one side of all samples, as can be easily seen in Figures 5.5(a) and 5.5(b). Not coincidentally, these artifacts in LAPIMO samples hampered CoDAGAN abilities to compensate for domain shift and severely hampered the segmentation quality in all baselines.

A similar pattern can be seen in Figures 5.10(c) and 5.10(d), which show respectively the 2D projections of CXR datasets and the GMM fits for these data. JSRT samples have the most standardized data among all CXR datasets, containing only high visual quality samples with fixed posture, high contrasts between anatomical structures (i.e. lungs, ribs, etc) and no major lung shape-distorting illnesses (i.e. pneumonia, tuberculosis. etc). Other datasets – such as Chest X-Ray 8, Montgomery and Shenzhen – present more real-world scenarios with a high variety of lung shapes and sizes and smaller control over patient's position during the exam, that is, higher rotation, scale and translations in these images. Thus samples from the JSRT dataset in Figure 5.10(d) are clustered in a small region in the 2D projection of $\mathcal{I}$-space, while the other datasets contain more spread samples in this projection. This result evidences that the use of distinct sources of data should better enforce satisfactory Domain Generalization for the supervised model M in CoDAGANs.

Another visibly distinct cluster in Figure 5.10(d) is formed of samples pertaining to the OCT CXR set. Samples from this dataset were noticeably harder to segment due to their smaller contrast range. OCT CXR patients also performed the exam on a distinct position with their arms pointing upward, contrary to all other

[1] https://scikit-learn.org/stable/



(a)

(b)

(c)

(d)

Figure 5.10. 2D projections of in $\mathcal{I}$-space for MXR in pectoral muscle segmentation (a, b) and CXRs in the lung segmentation task (c, d). The original 2D-projected samples after PCA/t-SNE (a, c) are shown conjointly with gaussian fits over the data (b, d) for both domains.

CXR data used in our experiments. These visual features reinforce this dataset's distinction from other CXRs in our experiments and explain its homogeneity in the 2D projections of Figure 5.10(d).

Another use for these 2D projections could be to perform inference from datasets that were never trained by the algorithm, effectively achieving Domain Generalization [Zhang et al., 2017] for new samples. This Domain Generalization

CoDAGAN could find the natural cluster closer to the new data according to a dissimilarity metric and assign the novel samples to the cluster. This approach could, therefore, personalize the One-Hot-Encoding so that it better captures the particular visual patterns of previously unseen data.

5.2.5 Time Comparison

One can see in Figures 5.11 and 5.12 the runtime comparisons for lung segmentation in CXRs and pectoral muscle segmentation in MXRs. The comparison is made between CoDAGANs, which are only trained once for all datasets in a certain domain; and D2Ds, considering all pairs of source and target datasets in an image domain. These results show that conditional multi-source training is several times faster than pairwise training for all target datasets.

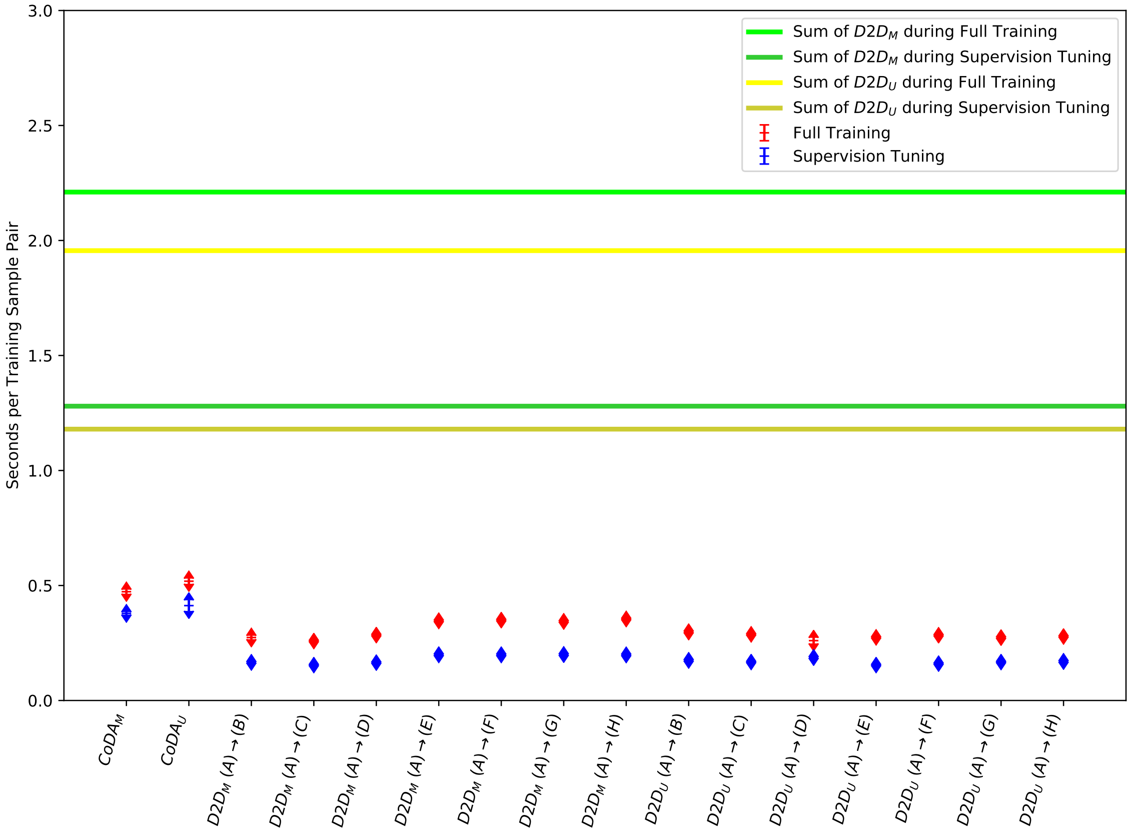


Figure 5.11. Per-sample time comparisons between CoDAGANs and D2D approaches in the segmentation of lungs in CXRs. Runtimes are given in seconds per sample.

The plot shown in Figures 5.11 and 5.12 specifically focus on the tasks of lung and pectoral muscle segmentation because they were the ones with the larger num-

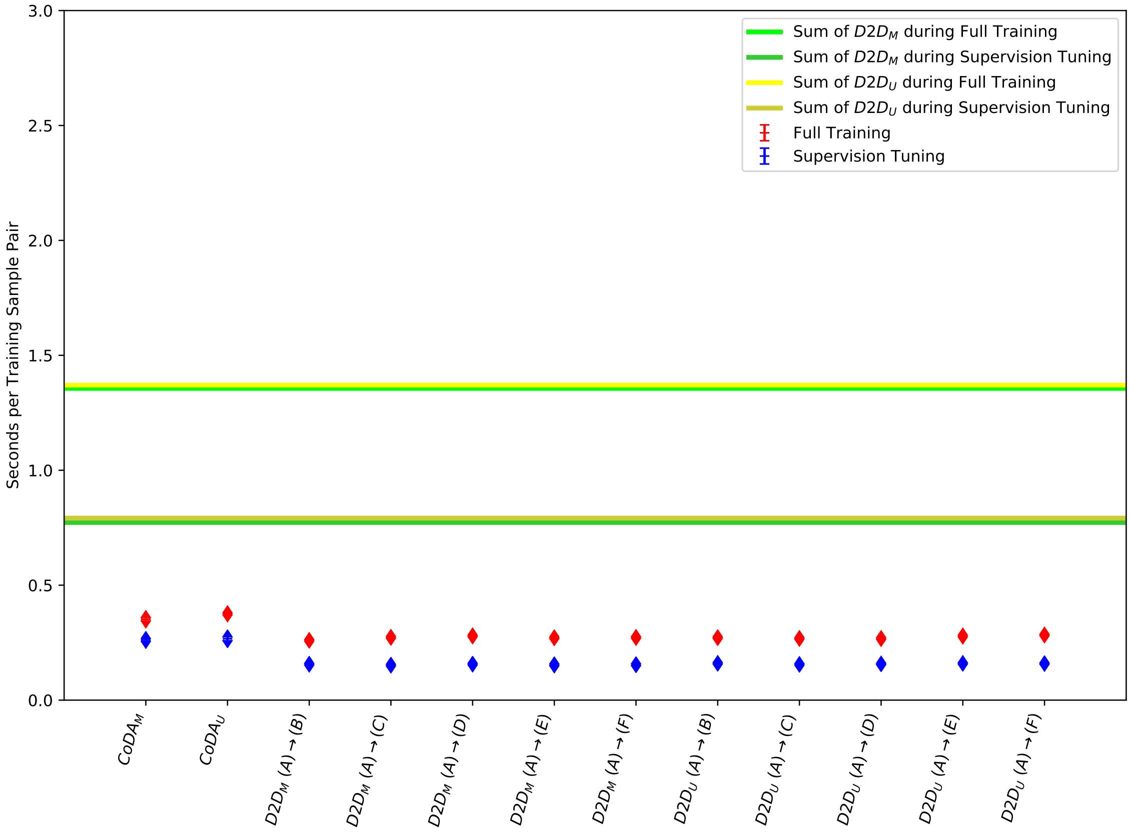


Figure 5.12. Per-sample time comparisons between CoDAGANs and D2D approaches in the segmentation of pectoral muscle in MXRs. Runtimes are given in seconds per sample.

ber of labeled samples in the CXR and MXR datasets. Also, the runtime plots for heart, clavicle and breast region segmentation followed similar trends to the ones presented in the aforementioned figures.

Per-sample training times on CoDAGAN took between 0.4 and 0.6 seconds, on average, while D2D ranged between 0.1 and 0.5 seconds. However, for multiple domains several D2D networks must be trained in order to achieve Domain Generalization from a single source dataset. The yellow and green lines represent the averages of the sums of all D2D (both $D2D_U$ and $D2D_M$) trained for a certain domain departing from one single source dataset $\mathscr{A}$. One can easily notice that CoDAGANs run between 3 and 4 times faster than the sum of D2Ds in the phase of *Supervision Tuning*, and between 4 and 6 times faster during *Full Training*

5.3 Unsupervised Segmentation from Synthetic DRRs

In this section we assess if synthetic data could be leveraged using the CoDAGAN framework in order to acquire useful information from 3D to 2D data, specifically for rib segmentation. Section 5.3.1, which presents quantitative results using the evaluation metrics described in Section 4.4 for the JSRT and OpenIST datasets. We compare CoDAGANs with both common shallow baselines in the area of rib segmentation [van Ginneken and ter Haar Romeny, 2000; Loog and Ginneken, 2006; Candemir et al., 2016] and with U-Nets [Ronneberger et al., 2015] pretrained on DRR data for CXR rib segmentation. Section 5.3.2 discusses qualitative results obtained in both labeled and unlabeled test datasets.

5.3.1 Rib Segmentation Effectiveness in CXRs

Table 5.6 shows the quantitative results according to the metrics described in Section 4.4 of the proposed methodology and common baselines in the literature for the OpenIST and JSRT datasets – which are the only ones with pixel-map labels available or computable. Bold cells highlight the method with the best results among all for their respective datasets and metrics. In the OpenIST dataset, CoDAGANs obtained similar results to simply using the Pretrained DNN for image segmentation trained in the DRR data, as this dataset presents rather similar visual features as the DRRs themselves. Pretrained DNNs showed considerably higher results mainly in Sensitivity, Dice and Jaccard, while Accuracy metrics were rather close in both Pretrained DNNs and CoDAGANs for OpenIST. Specificity, however, presented significantly better results for CoDAGANs, highlighting a larger portion of non-rib pixels being classified correctly by the method, which is backed up by qualitative analysis in Section 5.3.2. CoDAGANs also presented better AUC results, implying that this method yields a better trade-off between False Positives and False Negatives along the ROC curve, as can also be seen in Figure 5.13(a). Pretrained models presenting higher Sensitivity results and lower Specificity results than CoDAGANs implies that the former method tends to overshoot the prediction of positive pixels, while the latter has a higher certainty when classifying a pixel as pertaining to a rib.

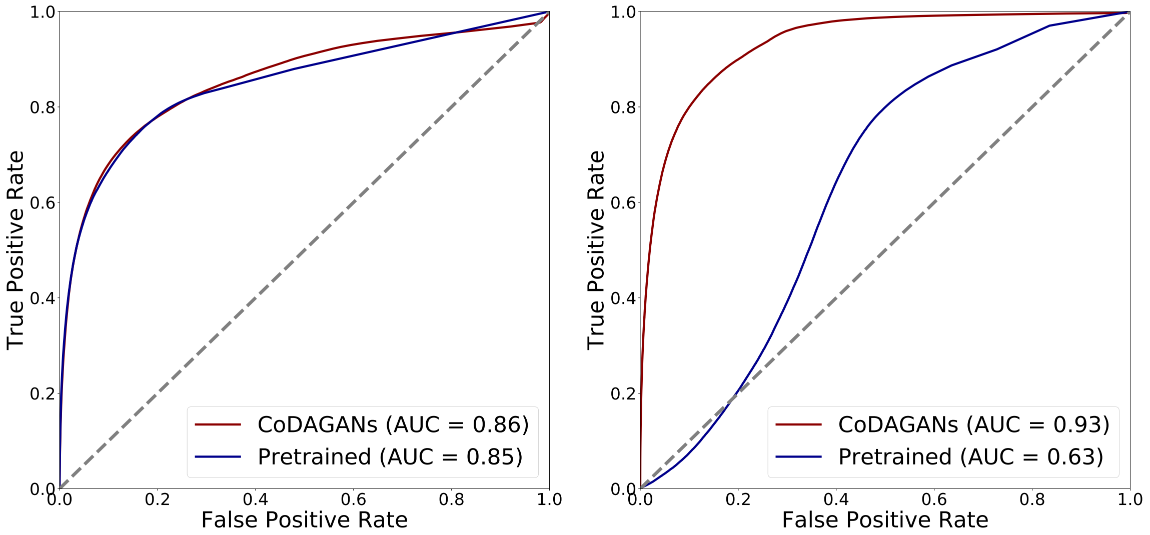
Even with competitive results in OpenIST, the real advantage of using CoDAGANs over both shallow [van Ginneken and ter Haar Romeny, 2000; Loog and Ginneken, 2006; Candemir et al., 2016] and deep [Ronneberger et al., 2015] baseline methods is seen when comparing the results in the JSRT dataset, as these data

Table 5.6. Quantitative results for rib segmentation in the JSRT and OpenIST datasets yielded from shallow (Model-based [van Ginneken and ter Haar Romeny, 2000], Pixel Classification – PC and Iterated Contextual Pixel Classification – ICPC [Loog and Ginneken, 2006] and Atlas-based [Candemir et al., 2016]) and deep (Pretrained DNNs) baselines and the proposed pipeline based on CoDAGANs. This table summarizes all metrics presented in Section 4.4, which were chosen according to the literature and/or due to their large use in segmentation tasks. Blank cells represent metrics that were not reported in the original works that proposed their respective method and, thus, could not be used in our comparisons.

Dataset	**Method**	**Metric**					
		AUC	**Accuracy**	**Sensitivity**	**Specificity**	**Dice**	**Jaccard**
OpenIST	**Pretrained DNN**	0.8487	**0.85 ± 0.02**	**0.57 ± 0.07**	0.95 ± 0.02	**0.66 ± 0.05**	**0.49 ± 0.05**
	CoDAGANs	**0.8557**	0.84 ± 0.02	0.44 ± 0.08	**0.98 ± 0.02**	0.58 ± 0.07	0.41 ± 0.07
JSRT	**Model-based**	0.9105	0.74 ± 0.05	0.71 ± 0.08	0.85 ± 0.03	-	-
	PC	-	0.79 ± 0.05	0.71 ± 0.08	0.85 ± 0.03	-	-
	ICPC	-	0.86 ± 0.06	**0.79 ± 0.09**	0.92 ± 0.04	-	-
	Atlas-based	-	0.86 ± 0.03	0.75 ± 0.06	0.92 ± 0.02	-	-
	Pretrained DNN	0.6335	0.69 ± 0.03	0.19 ± 0.03	0.81 ± 0.03	0.19 ± 0.02	0.11 ± 0.02
	CoDAGANs	**0.9341**	**0.89 ± 0.02**	0.51 ± 0.08	**0.98 ± 0.01**	**0.63 ± 0.08**	**0.47 ± 0.07**

present a much larger domain shift from the original DRRs. These distinct visual features between samples from the domains are further highlighted in Section 5.3.2. One can see in Table 5.6 that CoDAGANs are able to compensate much more efficiently for the visual differences in the two data sources than other methods, achieving state-of-the-art results for AUC, Accuracy, Specificity, Dice and Jaccard methods. The only metric wherein the literature reports better performance than CoDAGAN is Sensitivity, as most of the baseline methods tend to overestimate positive pixels, being more susceptible to present higher FP rates. This relatively higher propensity of baseline methods to predict larger amounts of False Positive rib pixels is further evidenced by CoDAGANs' near-perfect Specificity scores.

Figure 5.13 presents the overlayed ROC curves for OpenIST and JSRT of both Pretrained DNNs and Conditional DA in the task of rib segmentation. One can see that in Figure 5.13(a) both curves follow similar paths, with CoDAGANs presenting a slight edge over Pretrained DNNs for FP rates between 0.3 and 0.8, while Pretrained DNNs surpass Conditional DA in the rightmost side of the plot. However, analogously to the results presented in Table 5.6, CoDAGANs present considerably larger TP rates for any value of FP rates in JSRT, which is explained by the large domain shift between this dataset and the source DRRs.



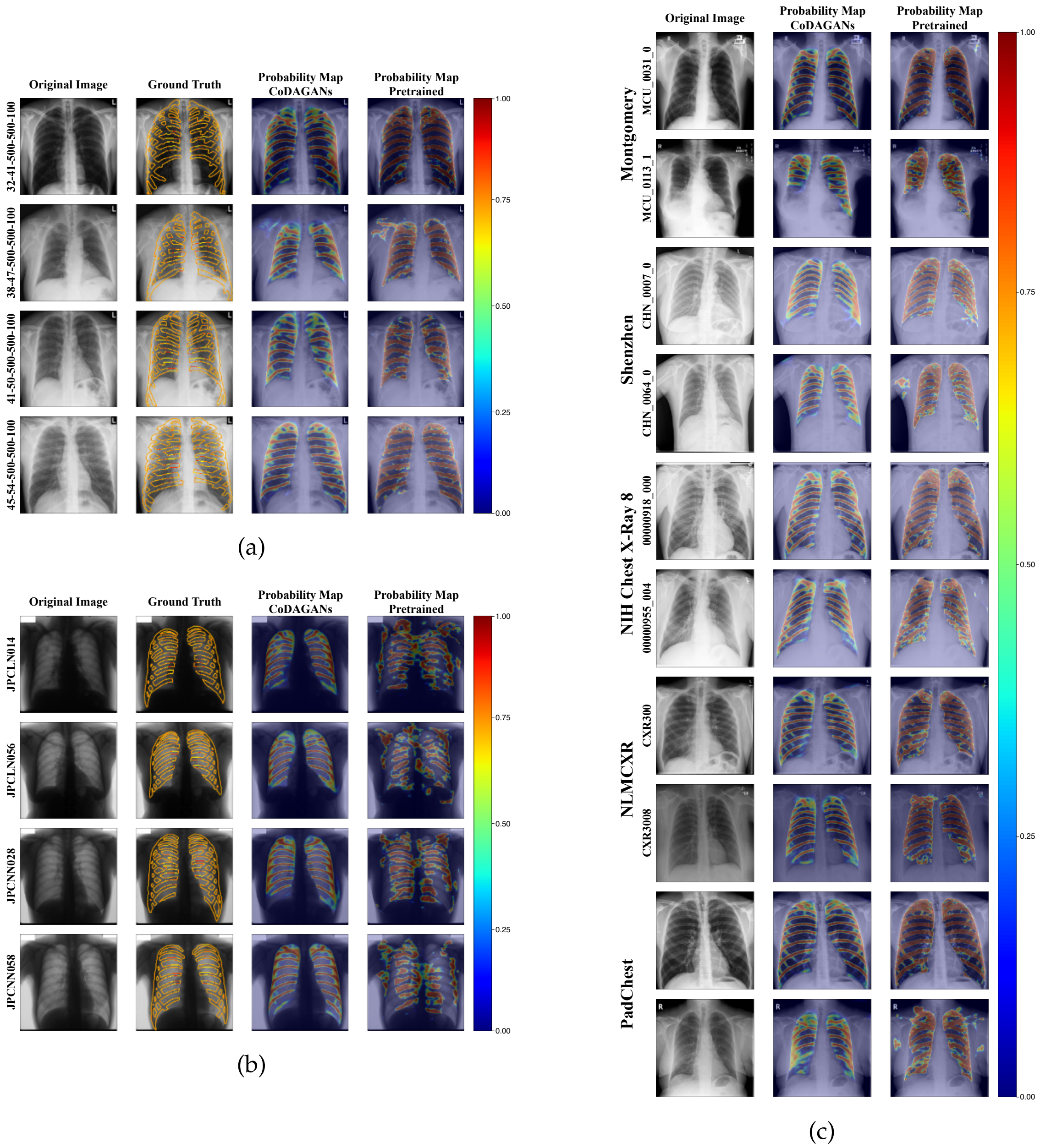
(a) (b)

Figure 5.13. Overlayed ROC curves for both Pretrained DNNs and CoDAGANs in the OpenIST (a) and JSRT (b) datasets. Pretrained DNNs present similar ROC curves to CoDAGANs on OpenIST, while the distinction between these methodologies in JSRT is much more noticeable due to the larger domain shift between this dataset and the source DRRs.

5.3.2 Generalization Analysis

The leftmost half of Figure 5.14 shows a small sample of qualitative results in rib segmentation for the JSRT (Figure 5.14(b)) and OpenIST (Figure 5.14(a)) datasets, which are the only ones that have pixel-level labels in our experiments. Visual analysis over the OpenIST samples further reinforce the previously mentioned tendency to overestimate rib pixels of DNNs pretrained in DRR synthetic samples, while CoDAGANs are more conservative in predicting rib pixel labels. It is also evident from the overlayed prediction probability map that the pretrained models have a much sharper decision boundary than CoDAGANs. That is, the deep baseline method predicts either rib or background pixels with more confidence than Conditional DA, which also results in rougher segmentation boundaries.

Also consistently with the objective results, the most evident advantage of using the proposed pipeline for DA is seen when there is a larger shift between the source and target domains, as can be seen in Figure 5.14(b) wherein segmentation results for JSRT are shown. One can easily see that the Pretrained U-Net severely missed the regions in the samples with actual ribs, presenting highly erratic predictions. CoDAGAN, however, are capable of Domain Generalization, being able to translate the knowledge from the DRR images and noisy labels to JSRT much more



(a)

(b)

(c)

Figure 5.14. Sample of qualitative results in the datasets with labeled test sets: OpenIST (a) and JSRT (b). The colormap on the right side indicates the probabilities in the predictions of the two rightmost columns.

effectively, resulting in high quality predictions for the rib semantic maps.

Additionally to the results from the datasets with labeled data, we show a small sample of CoDAGAN's predictions from unlabeled data from 5 additional datasets, as can be seen in Figure 5.14(c). Probability maps predicted by our methodology for all samples of these datasets in order to encourage reproducibility can be found in this project's webpage.

Chapter 6

Conclusion

This work described a pairwise method (D2D) and a Domain Generalization (CoDAGAN) method that cover the whole spectrum of UDA, SSDA and FSDA in dense labeling tasks, with the latter being able to learn from multiple source and target biomedical datasets. In order to have an answer to hypothesis $\mathcal{H}_1$ – presented in Section 1.2 – we performed exploratory tests on CXR images (Section 5.1) and an extensive quantitative and qualitative experimental evaluation on several distinct domains, datasets and tasks (Section 5.2), comparing the proposed methods with traditional Transfer Learning baselines in the literature. Both methods were shown to be effective DA methodologies that could learn a single model that performs dense labeling in either a pair (D2D) or several distinct datasets (CoDAGAN), even when the visual patterns of source and target data were visually distinct.

Another evidence of the generalization capabilities of D2D and CoDAGANs was the good performance in DA tasks even in highly imbalanced classes, as in clavicle segmentation, wherein the Region of Interest (RoI) in the images represents only a tiny portion of the total set of pixels. Many DA algorithms do not perform well with imbalance, requiring additional measures as random undersampling or elaborate data augmentation routines, while CoDAGANs are automatically able to compensate for class imbalance. The main hypothesis $\mathcal{H}_1$ – which states that I2I Translation can be used for visual DA – is found to be, therefore, confirmed.

The pairwise method (D2D) was observed to perform better UDA and SSDA than both From Scratch training and Fine-tuning. However, D2D variations were observed to be much more unstable even when using essentially the same settings as their Conditional DA counterparts (CoDAGANs). Thus, D2D was treated in the more thorough experimental setup of Section 5.2 as a stand-in baseline for other pairwise Image Translation DA methodologies [Cohen et al., 2018; Tang et al., 2019b,a; Yang et al., 2019] that work remarkably similar to the proposed pairwise approach.

It was observed in Sections 5.2.1 and 5.2.2 that CoDAGANs achieve results in fully unsupervised settings that are comparable to fully supervised DA methods – such as fine-tuning pretrained DNNs to new data – while the D2D baseline in UDA presented a higher instability. CoDAGANs yielded significantly better $\mathcal{J}$ values in most experiments where labeled data was scarce in the target datasets, while fine-tuning and From Scratch training was only able to achieve properly converge when labeled data from the target domain was abundant (i.e. $E_{50\%}$ and above). These results further reiterate the validity of $\mathcal{H}_1$. It is important to highlight that label scarcity – mainly for dense labeling tasks – is a major problem in real-world biomedical image tasks. As previously mentioned, Moment Matching losses for UDAs [Borgwardt et al., 2006; Li et al., 2017] were discarded from the beginning in our experiments, as they showed little-to-no improvement when compared to simply using the pretrained U-Net on the target datasets.

The proposed method was able to successfully learn from both labeled and unlabeled data, making it adaptable to a wide variety of data scarcity scenarios in SSDA due to its ability to correctly compensate for domain shift, which is evidence for reinforcing hypothesis $\mathcal{H}_2$. We performed thorough evaluations on a myriad of datasets, domains and tasks, and, in the overwhelming majority of cases, Conditional DA was superior both qualitatively and quantitatively to D2D in the unlabeled target domains and indistinguishable from D2D in source domains wherein labeled data was available.

In order to model a real-world data scenario, we specifically chose simpler "well-behaved" source datasets (JSRT for CXRs and INbreast for MXRs) and more real ones as targets (i.e. Chest X-Ray8, PadChest, MIAS, DDSM, etc). Even then Conditional DA was able to transfer knowledge via UDAs from the simpler labeled dataset to the novel real-world scenarios, evidencing the strategy's robustness.

Conceptually, as a side effect of using One-Hot-Encoding for conditional training, CoDAGANs have half as many generators and discriminators as other image translation DNNs, such as CoGANs [Liu and Tuzel, 2016], CycleGANs [Zhu et al., 2017a], UNIT [Liu et al., 2017], MUNIT [Huang et al., 2018], DRIT [Lee et al., 2018, 2020] and MSGANs [Mao et al., 2019]. At the same time, also due to conditional encoding, CoDAGANs are not limited to translations between only two domains at a time. Empirically, per-sample runtimes were gathered from the executions of D2D and CoDAGANs and were shown in Section 5.2.5. Conditional DA training with stochastic dataset sampling is shown to be, on average, between 4 and 5 times faster than pairwise training for a multitude of target domains.

Thus, the limitation of pairwise training from D2D affects segmentation performance, as many labels from other data sources end up being ignored; results in considerably larger runtimes when it is desirable to train models for a multitude of datasets; accentuate GPU memory requirements, due to the pair of generators used in D2D; and even requires more disk space, as trained models are often saved in non-volatile memory. All those pieces of evidence corroborate the validity of $\mathcal{H}_3$, as the scalability of translation architectures is not bound by the number of domains when Conditional DA is employed.

As explained in Section 2.4.1, Zhang et al. [2017]; Csurka [2017] closely relate Domain Generalization with multi-source DA, as adding marginal distributions for each dataset/domain tends to make the model more robust to novel data. As far as we are aware, CoDAGANs are the sole Domain Generalization method in the literature that is able to perform cross-dataset learning in dense labeling tasks.

CoDAGANs were observed to perform satisfactory DA to a myriad of distinct datasets even when the sole labeled source dataset was considerably simpler than the target unlabeled datasets, as presented in Sections 5.2.2, 5.2.1 and 5.3. In experiment $E_{0\%}$ for CXR lung, clavicle and heart segmentations, JSRT has images acquired in a much more controlled environment than all other datasets, while still performing UDA, SSDA and FSDA reasonably well on these more complex settings, validating $\mathcal{H}_4$. The low-dimensionality experiments presented in Section 5.2.4 were also designed specifically in order to question the validity of hypothesis $\mathcal{H}_4$. We conclude from the results presented in Figure 5.10 that "well-behaved" datasets – such as JSRT or INbreast – tend to be more clustered in a small subspace of the distributions. Thus, the inclusion of more realistic unsupervised data – such as Chest X-Ray 8 and DDSM – tends to incorporate a more realistic view of the true data distribution to the semi-supervised model, as stated in $\mathcal{H}_4$. At last, rib segmentation with noisy labeled data acquired from synthetic DRRs was generalized for multiple output datasets.

Results of DA acquired from synthetic DRRs and adapted to a myriad of CXR datasets were presented in Section 5.3 according to the pipeline from Figure 3.5. Even though there is a considerable visual distinction between the DRR samples computed from CT-scans via AIP and real CXRs, Conditional DA was able to compensate for this domain shift. This task also was fully unsupervised, as the noisy labels for rib segmentation were obtained via synthetic transformations on volumetric data instead of manual labeling. These novel results surpassed the previous literature – as shown in Table 5.6 – and allowed for the segmentation of ribs in datasets that do not possess any rib segmentation label. Thus, $\mathcal{H}_5$ holds true in the sense that synthetic radiological data can indeed be leveraged to improve and/or allow biomedical tasks via the use of Conditional DAs.

Essentially the same pipeline presented in Section 3.3 could be used to enforce Domain Generalization in the tasks of segmenting other structures in the human body, such as the spine or pelvic bones in CXRs or even single bones from 2D X-Rays of arms or legs in order to automatically detect fractures or other abnormalities. Alternatively, CoDAGANs can be applied to convert existing 3D labels from CT-scan or MRI datasets into 2D, much like the pipeline of Zhang et al. [2018b]. The advantage of employing CoDAGANs instead of the original method in Zhang et al. [2018b] would be that the labels could be generalized to multiple target datasets and even combined with existing organ segmentation labels from CXR data. We also see this interchangeability of data and labels between CXRs and tomographic samples as a step toward cross-modality DA in biomedical imaging, which is already relatively common between CT and MRI data.

6.1 Current Limitations and Future Works

The basic architecture of both D2D and CoDAGAN could be adapted to other imaging domains both within biomedical applications and in other areas as Computer Vision and Remote Sensing. Some of these variations of the proposed pipeline were tried during the course of this work with little success.

Given the success in the Domain Generalization on 2D data, CoDAGANs were adapted to volumetric images in order to perform cross-dataset and cross-modality image translation in CT-scans and MRIs. However, we found two major hampers in this application, starting with the 3D convolutional kernels, which are still prohibitively more expensive than 2D ones with mid-tier GPUs. 3D convolutions possess a much larger number of trainable parameters and generate activation and backpropagation gradient tensors with higher dimensionalities than their 2D counterparts. Thus, for even executing the first exploratory experiments we needed to set the minibatch size to one single sample per iteration and resize the volumes to the resolution $64 \times 64 \times 64$. A second problem in volumetric DA using CoDAGANs was the lack of both labeled and unlabeled data. The largest labeled datasets we could find for liver, spleen and lung segmentation in CT images had only a few tens of labeled samples, while the other unlabeled magnetic resonance and tomography available either did not align properly with the labeled ones or imaged completely distinct parts of the body (i.e. brain MRIs).

CoDAGANs did not properly converge on the previously described setting for volumetric samples, instead falling on some sort of "modal collapse" by outputting always the same organ segmentation prediction for all image inputs. These predictions were often a rough delineation in the correct shape, size and location of the organ that was to be segmented, indicating that the network learned the mode of the label distribution instead of properly generalizing to new samples.

In addition to these experiments, we tried using conditional encoding to perform multitask learning on CXRs using CoDAGANs and another One-Hot-Encoding passed to the supervised model M. These experiments also resulted in a kind of "mode collapse" in the prediction space, as the network always output the same basic outline for the organ, completely ignoring the input.

At last, we also tried to adapt CoDAGANs to the segmentation of Remote Sensing urban scenes and perform cross-dataset DA between the Vaihingen[1] and Potsdam[2] datasets. However, in contrast to all scenarios presented in Chapters 4 and 5, these Remote Sensing tasks are multiclass scenarios and the data are inherently more multimodal than the medical datasets used in our experiments. CoDAGANs were not able to converge on this scenario, even though a better hyperparameter tuning might prove effective in this conversion.

We believe most of the previously mentioned hurdles can be addressed in future works for CoDAGANs with a push in the academy for larger source labeled datasets and proper tuning of the method to the new settings.

Future experiments encompass testing the DA and Domain Generalization capabilities of CoDAGANs on volumetric radiological data, such as MRIs and CT-scans. Cross-modality DAs between magnetic resonance and tomographic data has been shown to work via D2D approaches [Yang et al., 2019], even though some artifacts have been observed on these applications after translation [Cohen et al., 2018]. The addition of multiple-source DA that is able to learn from both healthy and ill patients could alleviate these artifacts. At the same time, simply by adding novel labeled data to the training procedure might mitigate the previously discussed modal collapse problem on 3D data.

Another major future work would be to test the CoDAGAN framework on sparse labeling tasks in MXRs and CXRs. This talk would likely be the detection of diseases such as tuberculosis, pneumonia, pulmonary effusion and even symptoms of the more recent COVID-19 outbreak.

[1] http://www2.isprs.org/commissions/comm3/wg4/2d-sem-label-vaihingen.html

[2] http://www2.isprs.org/commissions/comm3/wg4/2d-sem-label-potsdam.html

As shown in Section 5.2.4, the distributions of well-behaved source data and real-world target data in $\mathcal{I}$ space are still not fully integrated. Perhaps adding a moment-matching loss term – either via MMDs or adversarial learning – could alleviate this problem and better merge the distributions for better Domain Generalization.

At last, both meta-learning [Vinyals et al., 2016; Snell et al., 2017; Finn et al., 2017, 2018] and self-supervision [Dosovitskiy et al., 2014; Noroozi and Favaro, 2016; Gidaris et al., 2018; Chen et al., 2020; Minderer et al., 2020] have recently been shown to be highly effective for zero-/one-/few-shot learning by increasing the label efficiency of DNNs. Hence, a natural follow-up to this research could be to integrate these schemes to CoDAGANs in order to improve the label efficiency in SSDA scenarios where labeled data exist, but are scarce (i.e. $E_{2.5\%}$ up to $E_{10\%}$).

Bibliography

Abdi, A. H., Kasaei, S., and Mehdizadeh, M. (2015). Automatic Segmentation of Mandible in Panoramic X-Ray. *Journal of Medical Imaging*, 2(4):044003.

Aoyama, T., Koyama, S., and Kawaura, C. (2002). An In-Phantom Dosimetry System Using Pin Silicon Photodiode Radiation Sensors for Measuring Organ Doses in X-Ray CT and Other Diagnostic Radiology. *Medical Physics*, 29(7):1504--1510.

Arjovsky, M., Chintala, S., and Bottou, L. (2017). Wasserstein GAN. *arXiv preprint arXiv:1701.07875*.

Armato III, S. G., McLennan, G., Bidaut, L., McNitt-Gray, M. F., Meyer, C. R., Reeves, A. P., Zhao, B., Aberle, D. R., Henschke, C. I., Hoffman, E. A., et al. (2011). The Lung Image Database Consortium (LIDC) and Image Database Resource Initiative (IDRI): A Completed Reference Database of Lung Nodules on CT Scans. *Medical Physics*, 38(2):915--931.

Badrinarayanan, V., Kendall, A., and Cipolla, R. (2017). SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *Transactions on Pattern Analysis and Machine Intelligence*, 39(12):2481--2495.

Berthelot, D., Schumm, T., and Metz, L. (2017). BEGAN: Boundary Equilibrium Generative Adversarial Networks. *arXiv preprint arXiv:1703.10717*.

Borgwardt, K. M., Gretton, A., Rasch, M. J., Kriegel, H.-P., Schölkopf, B., and Smola, A. J. (2006). Integrating Structured Biological Data by Kernel Maximum Mean Discrepancy. *Bioinformatics*, 22(14):e49--e57.

Bousmalis, K., Silberman, N., Dohan, D., Erhan, D., and Krishnan, D. (2017). Unsupervised Pixel-Level Domain Adaptation with Generative Adversarial Networks. In *Conference on Computer Vision and Pattern Recognition*, volume 1, page 7.

Brislin, R. W. (1970). Back-Translation for Cross-Cultural Research. *Journal of Cross-Cultural Psychology*, 1(3):185--216.

Brock, A., Donahue, J., and Simonyan, K. (2018). Large Scale GAN Training for High Fidelity Natural Image Synthesis. *arXiv preprint arXiv:1809.11096*.

Buades, A., Coll, B., and Morel, J.-M. (2005). A Non-Local Algorithm for Image Denoising. In *Conference on Computer Vision and Pattern Recognition*, volume 2, pages 60--65. IEEE.

Bustos, A., Pertusa, A., Salinas, J.-M., and de la Iglesia-Vayá, M. (2019). Padchest: A Large Chest X-ray Image Dataset with Multi-label Annotated Reports. *arXiv preprint arXiv:1901.07441*.

Candemir, S., Jaeger, S., Antani, S., Bagci, U., Folio, L. R., Xu, Z., and Thoma, G. (2016). Atlas-based Rib-Bone Detection in Chest X-Rays. *Computerized Medical Imaging and Graphics*, 51:32--39.

Cao, Z., Ma, L., Long, M., and Wang, J. (2018). Partial Adversarial Domain Adaptation. In *European Conference on Computer Vision*, pages 135--150.

Carlucci, F. M., D'Innocente, A., Bucci, S., Caputo, B., and Tommasi, T. (2019). Domain Generalization by Solving Jigsaw Puzzles. In *Conference on Computer Vision and Pattern Recognition*, pages 2229--2238.

Caseiro, R., Henriques, J. F., Martins, P., and Batista, J. (2015). Beyond the Shortest Path: Unsupervised Domain Adaptation by Sampling Subspaces Along the Spline Glow. In *Conference on Computer Vision and Pattern Recognition*, pages 3846--3854.

Chen, Q. and Koltun, V. (2017). Photographic Image Synthesis with Cascaded Refinement Networks. In *International Conference on Computer Vision*, volume 1, page 3.

Chen, T., Cheng, M.-M., Tan, P., Shamir, A., and Hu, S.-M. (2009). Sketch2Photo: Internet Image Montage. *ACM Transactions on Graphics*, 28(5):124.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. (2020). A Simple Framework for Contrastive Learning of Visual Representations. *arXiv preprint arXiv:2002.05709*.

Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., and Abbeel, P. (2016). InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets. In *Neural Information Processing Systems*, pages 2172--2180.

Choi, Y., Choi, M., Kim, M., Ha, J.-W., Kim, S., and Choo, J. (2018). StarGAN: Unified Generative Adversarial Networks for Multi-Domain Image-to-Image Translation. In *Conference on Computer Vision and Pattern Recognition*, pages 8789--8797.

Choi, Y., Uh, Y., Yoo, J., and Ha, J.-W. (2020). StarGAN v2: Diverse Image Synthesis for Multiple Domains. In *Conference on Computer Vision and Pattern Recognition*, pages 8188--8197.

Cohen, J. P., Luck, M., and Honari, S. (2018). Distribution Matching Losses Can Hallucinate Features in Medical Image Translation. In *International Conference on Medical Image Computing and Computer Assisted Intervention*, pages 529--536. Springer.

Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., and Schiele, B. (2016). The Cityscapes Dataset for Semantic Urban Scene Understanding. In *Conference on Computer Vision and Pattern Recognition*, pages 3213--3223.

Cortes, C. and Vapnik, V. (1995). Support-Vector Networks. *Machine Learning*, 20(3):273--297.

Csurka, G. (2017). Domain Adaptation for Visual Applications: A Comprehensive Survey. *arXiv preprint arXiv:1702.05374*.

Dalal, N. and Triggs, B. (2005). Histograms of Oriented Gradients for Human Detection. In *Conference on Computer Vision and Pattern Recognition*, volume 1, pages 886--893. IEEE.

Demner-Fushman, D., Kohli, M. D., Rosenman, M. B., Shooshan, S. E., Rodriguez, L., Antani, S., Thoma, G. R., and McDonald, C. J. (2015). Preparing a Collection of Radiology Examinations for Distribution and Retrieval. *Journal of the American Medical Informatics Association*, 23(2):304--310.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). ImageNet: A Large-Scale Hierarchical Image Database. In *Conference on Computer Vision and Pattern Recognition*, pages 248--255. Ieee.

Ding, Z. and Fu, Y. (2017). Deep Domain Generalization with Structured Low-Rank Constraint. *IEEE Transactions on Image Processing*, 27(1):304--313.

Dosovitskiy, A., Springenberg, J. T., Riedmiller, M., and Brox, T. (2014). Discriminative Unsupervised Feature Learning with Convolutional Neural Networks. In *Neural Information Processing Systems*, pages 766--774.

Efros, A. A. and Freeman, W. T. (2001). Image Quilting for Texture Synthesis and Transfer. In *Conference on Computer Graphics and Interactive Techniques*, pages 341--346. ACM.

Eigen, D. and Fergus, R. (2015). Predicting Depth, Surface Normals and Semantic Labels with a Common Multi-Scale Convolutional Architecture. In *International Conference on Computer Vision*, pages 2650--2658.

Eigen, D., Puhrsch, C., and Fergus, R. (2014). Depth Map Prediction from a Single Image Using a Multi-Scale Deep Network. In *Neural Information Processing Systems*, pages 2366--2374.

Elnakib, A., Gimel′farb, G., Suri, J. S., and El-Baz, A. (2011). Medical Image Segmentation: A Brief Survey. In *Multi Modality State-of-the-Art Medical Image Segmentation and Registration Methodologies*, pages 1--39. Springer.

Everingham, M., Eslami, S. A., Van Gool, L., Williams, C. K., Winn, J., and Zisserman, A. (2015). The Pascal Visual Object Classes Challenge: A Retrospective. *International Journal of Computer Vision*, 111(1):98--136.

Fang, C., Xu, Y., and Rockmore, D. N. (2013). Unbiased Metric Learning: On the Utilization of Multiple Datasets and Web Images for Softening Bias. In *International Conference on Computer Vision*, pages 1657--1664.

Ferreira, E., Oliveira, H., Alvim, M. S., and dos Santos, J. A. (2018). A Comparative Study on Unsupervised Domain Adaptation for Coffee Crop Mapping. In *Iberoamerican Congress on Pattern Recognition*, pages 72--80. Springer.

Finn, C., Abbeel, P., and Levine, S. (2017). Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. *arXiv preprint arXiv:1703.03400*.

Finn, C., Xu, K., and Levine, S. (2018). Probabilistic Model-Agnostic Meta-Learning. In *Neural Information Processing Systems*, pages 9516--9527.

Frid-Adar, M., Diamant, I., Klang, E., Amitai, M., Goldberger, J., and Greenspan, H. (2018a). GAN-based Synthetic Medical Image Augmentation for Increased CNN Performance in Liver Lesion Classification. *Neurocomputing*, 321:321--331.

Frid-Adar, M., Klang, E., Amitai, M., Goldberger, J., and Greenspan, H. (2018b). Synthetic Data Augmentation Using GAN for Improved Liver Lesion Classification. In *International Symposium on Biomedical Imaging*, pages 289--293. IEEE.

Gan, C., Yang, T., and Gong, B. (2016). Learning Attributes Equals Multi-Source Domain Generalization. In *Conference on Computer Vision and Pattern Recognition*, pages 87--97.

Geng, B., Tao, D., and Xu, C. (2011). DAML: Domain Adaptation Metric Learning. *IEEE Transactions on Image Processing*, 20(10):2980--2989.

Ghifary, M., Bastiaan Kleijn, W., Zhang, M., and Balduzzi, D. (2015). Domain Generalization for Object Recognition with Multi-Task Autoencoders. In *International Conference on Computer Vision*, pages 2551--2559.

Ghifary, M., Kleijn, W. B., and Zhang, M. (2014). Domain Adaptive Neural Networks for Object Recognition. In *Pacific Rim International Conference on Artificial Intelligence*, pages 898--904. Springer.

Gidaris, S., Singh, P., and Komodakis, N. (2018). Unsupervised Representation Learning by Predicting Image Rotations. In *Int. Conference on Learning Representations*.

Girshick, R., Donahue, J., Darrell, T., and Malik, J. (2014). Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In *Conference on Computer Vision and Pattern Recognition*, pages 580--587.

Gong, B., Grauman, K., and Sha, F. (2013). Connecting the Dots with Landmarks: Discriminatively Learning Domain-Invariant Features for Unsupervised Domain Adaptation. In *International Conference of Machine Learning*, pages 222--230.

Goodfellow, I. (2016). NIPS 2016 Tutorial: Generative Adversarial Networks. *arXiv preprint arXiv:1701.00160*.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative Adversarial Nets. In *Neural Information Processing Systems*, pages 2672--2680.

Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., and Courville, A. C. (2017). Improved Training of Wasserstein GANs. In *Neural Information Processing Systems*, pages 5767--5777.

Handa, A., Patraucean, V., Badrinarayanan, V., Stent, S., and Cipolla, R. (2016). Understanding Real World Indoor Scenes with Synthetic Data. In *Conference on Computer Vision and Pattern Recognition*, pages 4077--4085.

Haralick, R. M., Shanmugam, K., and Dinstein, I. H. (1973). Textural Features for Image Classification. *IEEE Transactions on Systems, Man and Cybernetics*, (6):610--621.

Haskins, G., Kruger, U., and Yan, P. (2020). Deep Learning in Medical Image Registration: A Survey. *Machine Vision and Applications*, 31(1):8.

He, D., Xia, Y., Qin, T., Wang, L., Yu, N., Liu, T., and Ma, W.-Y. (2016a). Dual Learning for Machine Translation. In *Neural Information Processing Systems*, pages 820--828.

He, K., Gkioxari, G., Dollár, P., and Girshick, R. (2017). Mask R-CNN. In *International Conference on Computer Vision*, pages 2961--2969.

He, K., Zhang, X., Ren, S., and Sun, J. (2016b). Deep Residual Learning for Image Recognition. In *Conference on Computer Vision and Pattern Recognition*, pages 770--778.

Heath, M., Bowyer, K., Kopans, D., Moore, R., and Kegelmeyer, P. (2000). The Digital Database for Screening Mammography. *Digital Mammography*, pages 431--434.

Ho, T. K. (1995). Random Decision Forests. In *International Conference on Document Analysis and Recognition*, volume 1, pages 278--282. IEEE.

Hoffman, J., Tzeng, E., Park, T., Zhu, J.-Y., Isola, P., Saenko, K., Efros, A., and Darrell, T. (2018). CyCADA: Cycle-Consistent Adversarial Domain Adaptation. In *International Conference of Machine Learning*, pages 1994--2003.

Hoffman, J., Wang, D., Yu, F., and Darrell, T. (2016). FCNs in the Wild: Pixel-level Adversarial and Constraint-based Adaptation. *arXiv preprint arXiv:1612.02649*.

Huang, G., Liu, Z., Weinberger, K. Q., and van der Maaten, L. (2017). Densely Connected Convolutional Networks. In *Conference on Computer Vision and Pattern Recognition*, volume 1, page 3.

Huang, G., Sun, Y., Liu, Z., Sedra, D., and Weinberger, K. Q. (2016). Deep Networks with Stochastic Depth. In *European Conference on Computer Vision*, pages 646--661. Springer.

Huang, J., Gretton, A., Borgwardt, K. M., Schölkopf, B., and Smola, A. J. (2007). Correcting Sample Selection Bias by Unlabeled Data. In *Neural Information Processing Systems*, pages 601--608.

Huang, X., Liu, M.-Y., Belongie, S., and Kautz, J. (2018). Multimodal Unsupervised Image-to-Image Translation. In *European Conference on Computer Vision*, pages 172--189.

Hull, J. J. (1994). A Database for Handwritten Text Recognition Research. *Transactions on Pattern Analysis and Machine Intelligence*, 16(5):550--554.

Isola, P., Zhu, J.-Y., Zhou, T., and Efros, A. A. (2017). Image-to-Image Translation with Conditional Adversarial Networks. In *Conference on Computer Vision and Pattern Recognition*, pages 5967--5976. IEEE.

Jaeger, S., Candemir, S., Antani, S., Wáng, Y.-X. J., Lu, P.-X., and Thoma, G. (2014). Two Public Chest X-Ray Datasets for Computer-Aided Screening of Pulmonary Diseases. *Quantitative Imaging in Medicine and Surgery*, 4(6):475.

Karpathy, A. and Fei-Fei, L. (2015). Deep Visual-Semantic Alignments for Generating Image Descriptions. In *Conference on Computer Vision and Pattern Recognition*, pages 3128--3137.

Karras, T., Aila, T., Laine, S., and Lehtinen, J. (2018). Progressive Growing of GANs for Improved Quality, Stability, and Variation. In *Int. Conference on Learning Representations*.

Karras, T., Laine, S., and Aila, T. (2019). A Style-based Generator Architecture for Generative Adversarial Networks. In *Conference on Computer Vision and Pattern Recognition*, pages 4401--4410.

Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., and Aila, T. (2020). Analyzing and Improving the Image Quality of StyleGAN. In *Conference on Computer Vision and Pattern Recognition*, pages 8110--8119.

Kermany, D. S., Goldbaum, M., Cai, W., Valentim, C. C., Liang, H., Baxter, S. L., McKeown, A., Yang, G., Wu, X., Yan, F., et al. (2018). Identifying Medical Diagnoses and Treatable Diseases by Image-based Deep Learning. *Cell*, 172(5):1122--1131.

Kingma, D. P. and Ba, J. (2014). Adam: A Method for Stochastic Optimization. *arXiv preprint arXiv:1412.6980*.

Kingma, D. P. and Welling, M. (2013). Auto-Encoding Variational Bayes. *arXiv preprint arXiv:1312.6114*.

Kodali, N., Abernethy, J., Hays, J., and Kira, Z. (2017). On Convergence and Stability of GANs. *arXiv preprint arXiv:1705.07215*.

Koniusz, P., Tas, Y., and Porikli, F. (2017). Domain Adaptation by Mixture of Alignments of Second-or Higher-Order Scatter Tensors. In *Conference on Computer Vision and Pattern Recognition*, volume 2.

Krizhevsky, A., Hinton, G., et al. (2009). Learning Multiple Layers of Features from Tiny Images.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. In Pereira, F., Burges, C. J. C., Bottou, L., and Weinberger, K. Q., editors, *Neural Information Processing Systems*, pages 1097--1105. Curran Associates, Inc.

Kuzborskij, I. and Orabona, F. (2013). Stability and Hypothesis Transfer Learning. In *International Conference of Machine Learning*, pages 942--950.

Larsson, G., Maire, M., and Shakhnarovich, G. (2016). FractalNet: Ultra-Deep Neural Networks without Residuals. *arXiv preprint arXiv:1605.07648*.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based Learning Applied to Document Recognition. *Proceedings of the IEEE*, 86(11):2278--2324.

Lee, H.-Y., Tseng, H.-Y., Huang, J.-B., Singh, M., and Yang, M.-H. (2018). Diverse Image-to-Image Translation via Disentangled Representations. In *European Conference on Computer Vision*, pages 35--51.

Lee, H.-Y., Tseng, H.-Y., Mao, Q., Huang, J.-B., Lu, Y.-D., Singh, M., and Yang, M.-H. (2020). DRIT++: Diverse Image-to-Image Translation via Disentangled Representations. *International Journal of Computer Vision*, pages 1--16.

Li, C. and Wand, M. (2016). Combining Markov Random Fields and Convolutional Neural Networks for Image Synthesis. In *Conference on Computer Vision and Pattern Recognition*, pages 2479--2486.

Li, C.-L., Chang, W.-C., Cheng, Y., Yang, Y., and Póczos, B. (2017). MMD GAN: Towards Deeper Understanding of Moment Matching Network. In *Neural Information Processing Systems*, pages 2203--2213.

Li, D., Zhang, J., Yang, Y., Liu, C., Song, Y.-Z., and Hospedales, T. M. (2019). Episodic Training for Domain Generalization. In *International Conference on Computer Vision*, pages 1446--1455.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (2014). Microsoft COCO: Common Objects in Context. In *European Conference on Computer Vision*, pages 740--755. Springer.

Litjens, G., Kooi, T., Bejnordi, B. E., Setio, A. A. A., Ciompi, F., Ghafoorian, M., van der Laak, J. A., van Ginneken, B., and Sánchez, C. I. (2017). A Survey on Deep Learning in Medical Image Analysis. *Medical Image Analysis*, 42:60--88. ISSN 1361-8415.

Liu, M.-Y., Breuel, T., and Kautz, J. (2017). Unsupervised Image-to-Image Translation Networks. In *Neural Information Processing Systems*, pages 700--708.

Liu, M.-Y., Huang, X., Mallya, A., Karras, T., Aila, T., Lehtinen, J., and Kautz, J. (2019). Few-Shot Unsupervised Image-to-Image Translation. In *International Conference on Computer Vision*, pages 10551--10560.

Liu, M.-Y. and Tuzel, O. (2016). Coupled Generative Adversarial Networks. In *Neural Information Processing Systems*, pages 469--477.

Long, J., Shelhamer, E., and Darrell, T. (2015). Fully Convolutional Networks for Semantic Segmentation. In *Conference on Computer Vision and Pattern Recognition*, pages 3431–3440. ISSN 1063-6919.

Long, M., Zhu, H., Wang, J., and Jordan, M. I. (2016). Unsupervised Domain Adaptation with Residual Transfer Networks. In *Neural Information Processing Systems*, pages 136--144.

Long, M., Zhu, H., Wang, J., and Jordan, M. I. (2017). Deep Transfer Learning with Joint Adaptation Networks. In *International Conference of Machine Learning*, pages 2208--2217.

Loog, M. and Ginneken, B. (2006). Segmentation of the Posterior Ribs in Chest Radiographs Using Iterated Contextual Pixel Classification. *IEEE Transactions on Medical Imaging*, 25(5):602--611.

Lopez, M. G., Posada, N., Moura, D. C., Pollán, R. R., Valiente, J. M. F., Ortega, C. S., Solar, M., Diaz-Herrero, G., Ramos, I., Loureiro, J., et al. (2012). BCDR: A Breast Cancer Digital Repository. In *International Conference on Experimental Mechanics*.

Lowe, D. G. (2004). Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60(2):91--110.

Lucic, M., Kurach, K., Michalski, M., Gelly, S., and Bousquet, O. (2018). Are GANs Created Equal? A Large-scale Study. In *Neural Information Processing Systems*, pages 700--709.

Maaten, L. v. d. and Hinton, G. (2008). Visualizing Data Using t-SNE. *Journal of Machine Learning Research*, 9(Nov):2579--2605.

Mao, Q., Lee, H.-Y., Tseng, H.-Y., Ma, S., and Yang, M.-H. (2019). Mode Seeking Generative Adversarial Networks for Diverse Image Synthesis. In *Conference on Computer Vision and Pattern Recognition*, pages 1429--1437.

Mao, X., Li, Q., Xie, H., Lau, R. Y., Wang, Z., and Smolley, S. P. (2017). Least Squares Generative Adversarial Networks. In *International Conference on Computer Vision*, pages 2813--2821. IEEE.

Masood, S., Sharif, M., Masood, A., Yasmin, M., and Raza, M. (2015). A Survey on Medical Image Segmentation. *Current Medical Imaging*, 11(1):3--14.

Matheus, B. R. N. and Schiabel, H. (2011). Online Mammographic Images Database for Development and Comparison of CAD Schemes. *Journal of Digital Imaging*, 24(3):500--506.

Minderer, M., Bachem, O., Houlsby, N., and Tschannen, M. (2020). Automatic Shortcut Removal for Self-Supervised Representation Learning. *arXiv preprint arXiv:2002.08822*.

Ming Harry Hsu, T., Yu Chen, W., Hou, C.-A., Hubert Tsai, Y.-H., Yeh, Y.-R., and Frank Wang, Y.-C. (2015). Unsupervised Domain Adaptation with Imbalanced Cross-Domain Data. In *International Conference on Computer Vision*, pages 4121--4129.

Mirza, M. and Osindero, S. (2014). Conditional Generative Adversarial Nets. *arXiv preprint arXiv:1411.1784*.

Moreira, I. C., Amaral, I., Domingues, I., Cardoso, A., Cardoso, M. J., and Cardoso, J. S. (2012). INbreast: Toward a Full-Field Digital Mammographic Database. *Academic Radiology*, 19(2):236--248.

Mota, V. F., de Oliveira, H. N., Scalzo, S., Dittz, D., Santos, R. J., dos Santos, J. A., and Araújo, A. d. A. (2020). From Video Pornography to Cancer Cells: A Tensor Framework for Spatiotemporal Description. *Multimedia Tools and Applications*, pages 1--31.

Mou, X., Chen, X., Sun, L., Yu, H., Ji, Z., and Zhang, L. (2008). The Impact of Calibration Phantom Errors on Dual-Energy Digital Mammography. *Physics in Medicine & Biology*, 53(22):6321.

Murez, Z., Kolouri, S., Kriegman, D., Ramamoorthi, R., and Kim, K. (2018). Image to Image Translation for Domain Adaptation. In *Conference on Computer Vision and Pattern Recognition*, pages 4500--4509.

Nair, V. and Hinton, G. E. (2010). Rectified Linear Units Improve Restricted Boltzmann Machines. In *International Conference of Machine Learning*, pages 807--814.

Nord, R. H. and Miller, C. G. (2001). Spine Phantom Simulating Cortical and Trabecular Bone for Calibration of Dual Energy X-Ray Bone Densitometers. US Patent 6,302,582.

Noroozi, M. and Favaro, P. (2016). Unsupervised Learning of Visual Representations by Solving Jigsaw Puzzles. In *European Conference on Computer Vision*, pages 69--84. Springer.

Oliveira, H. N., Avelar, C. S., Machado, A. M. C., Araujo, A. A., and dos Santos, J. A. (2018). Exploring Deep-Based Approaches for Semantic Segmentation of Mammographic Images. In *Iberoamerican Congress on Pattern Recognition*. Springer.

Oliveira, H. N. and dos Santos, J. A. (2018). Deep Transfer Learning for Segmentation of Anatomical Structures in Chest Radiographs. In *Conference on Graphics, Patterns and Images*. IEEE.

Oliveira, H. N., Ferreira, E., and Dos Santos, J. A. (2020). Truly Generalizable Radiograph Segmentation With Conditional Domain Adaptation. *IEEE Access*, 8:84037--84062.

Pan, S. J., Tsang, I. W., Kwok, J. T., and Yang, Q. (2011). Domain Adaptation via Transfer Component Analysis. *Transactions on Neural Networks*, 22(2):199--210.

Papon, J. and Schoeler, M. (2015). Semantic Pose Using Deep Networks Trained on Synthetic RGB-D. In *International Conference on Computer Vision*, pages 774--782.

Patel, V. M., Gopalan, R., Li, R., and Chellappa, R. (2015). Visual Domain Adaptation: A Survey of Recent Advances. *IEEE Signal Processing Magazine*, 32(3):53--69.

Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., and Efros, A. A. (2016). Context Encoders: Feature Learning by Inpainting. In *Conference on Computer Vision and Pattern Recognition*, pages 2536--2544.

Pelka, O., Koitka, S., Rückert, J., Nensa, F., and Friedrich, C. M. (2018). Radiology Objects in Context (ROCO): A Multimodal Image Dataset. In *Intravascular Imaging and Computer Assisted Stenting and Large-Scale Annotation of Biomedical Data and Expert Label Synthesis*, pages 180--189. Springer.

Radford, A., Metz, L., and Chintala, S. (2015). Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. *arXiv preprint arXiv:1511.06434*.

Rampun, A., Morrow, P. J., Scotney, B. W., and Winder, J. (2017). Fully Automated Breast Boundary and Pectoral Muscle Segmentation in Mammograms. *Artificial Intelligence in Medicine*, 79:28--41. ISSN 0933-3657.

Redmon, J. and Farhadi, A. (2018). YOLOv3: An Incremental Improvement. *arXiv*.

Ren, S., He, K., Girshick, R., and Sun, J. (2015). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In *Neural Information Processing Systems*, pages 91--99.

Richter, S. R., Vineet, V., Roth, S., and Koltun, V. (2016). Playing for Data: Ground Truth from Computer Games. In *European Conference on Computer Vision*, pages 102--118. Springer.

Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional Networks for Biomedical Image Segmentation. In *International Conference on Medical Image Computing and Computer Assisted Intervention*, pages 234--241. Springer.

Ros, G., Sellart, L., Materzynska, J., Vazquez, D., and Lopez, A. M. (2016). The SYNTHIA Dataset: A Large Collection of Synthetic Images for Semantic Segmentation of Urban Scenes. In *Conference on Computer Vision and Pattern Recognition*, pages 3234--3243.

Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., and Chen, X. (2016). Improved Techniques for Training GANs. In *Neural Information Processing Systems*, pages 2234--2242.

Shaham, T. R., Dekel, T., and Michaeli, T. (2019). SinGAN: Learning a Generative Model from a Single Natural Image. In *International Conference on Computer Vision*, pages 4570--4580.

Shao, L., Zhu, F., and Li, X. (2015). Transfer Learning for Visual Categorization: A Survey. *IEEE Transactions on Neural Networks and Learning Systems*, 26(5):1019--1034.

Shiraishi, J., Katsuragawa, S., Ikezoe, J., Matsumoto, T., Kobayashi, T., Komatsu, K.-i., Matsui, M., Fujita, H., Kodera, Y., and Doi, K. (2000). Development of a Digital Image Database for Chest Radiographs with and without a Lung Nodule: Receiver Operating Characteristic Analysis of Radiologists' Detection of Pulmonary Nodules. *American Journal of Roentgenology*, 174(1):71--74.

Silva, G., Oliveira, L., and Pithon, M. (2018). Automatic Segmenting Teeth in X-Ray Images: Trends, a Novel Data Set, Benchmarking and Future Perspectives. *Expert Systems with Applications*, 107:15--31.

Simonyan, K. and Zisserman, A. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv preprint arXiv:1409.1556*.

Snell, J., Swersky, K., and Zemel, R. (2017). Prototypical Networks for Few-Shot Learning. In *Neural Information Processing Systems*, pages 4077--4087.

Srivastava, R. K., Greff, K., and Schmidhuber, J. (2015). Training Very Deep Networks. In *Neural Information Processing Systems*, pages 2377--2385.

Suckling, J., Parker, J., Dance, D., Astley, S., Hutt, I., Boggis, C., Ricketts, I., Stamatakis, E., Cerneaz, N., Kok, S., et al. (2015). Mammographic Image Analysis Society (MIAS) Database v1. 21.

Sun, Q., Chattopadhyay, R., Panchanathan, S., and Ye, J. (2011). A Two-Stage Weighting Framework for Multi-Source Domain Adaptation. In *Neural Information Processing Systems*, pages 505--513.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2015). Going Deeper with Convolutions. In *Conference on Computer Vision and Pattern Recognition*, pages 1--9.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. (2016). Rethinking the Inception Architecture for Computer Vision. In *Conference on Computer Vision and Pattern Recognition*, pages 2818--2826.

Tang, Y., Tang, Y., Sandfort, V., Xiao, J., and Summers, R. M. (2019a). TUNA-Net: Task-oriented UNsupervised Adversarial Network for Disease Recognition in Cross-Domain Chest X-Rays. In *International Conference on Medical Image Computing and Computer Assisted Intervention*, pages 431--440. Springer.

Tang, Y., Tang, Y., Xiao, J., and Summers, R. M. (2019b). XLSor: A Robust and Accurate Lung Segmentor on Chest X-Rays Using Criss-Cross Attention and Customized Radiorealistic Abnormalities Generation. In *International Conference on Medical Imaging with Deep Learning*.

Tran, D., Wang, H., Torresani, L., Ray, J., LeCun, Y., and Paluri, M. (2018). A Closer Look at Spatiotemporal Convolutions for Action Recognition. In *Conference on Computer Vision and Pattern Recognition*, pages 6450--6459.

Unser, M. and Aldroubi, A. (1996). A Review of Wavelets in Biomedical Applications. *Proceedings of the IEEE*, 84(4):626--638.

Van Ginneken, B., Stegmann, M. B., and Loog, M. (2006). Segmentation of Anatomical Structures in Chest Radiographs Using Supervised Methods: A Comparative Study on a Public Database. *Medical Image Analysis*, 10(1):19--40.

van Ginneken, B. and ter Haar Romeny, B. M. (2000). Automatic Delineation of Ribs in Frontal Chest Radiographs. In *Medical Imaging 2000: Image Processing*, volume 3979, pages 825--837. International Society for Optics and Photonics.

Vinyals, O., Blundell, C., Lillicrap, T., Wierstra, D., et al. (2016). Matching Networks for One Shot Learning. In *Neural Information Processing Systems*, pages 3630--3638.

Wang, M. and Deng, W. (2018). Deep Visual Domain Adaptation: A Survey. *Neurocomputing*.

Wang, X., Peng, Y., Lu, L., Lu, Z., Bagheri, M., and Summers, R. M. (2017). ChestX-Ray8: Hospital-Scale Chest X-Ray Database and Benchmarks on Weakly-Supervised Classification and Localization of Common Thorax Diseases. In *Conference on Computer Vision and Pattern Recognition*, pages 3462--3471. IEEE.

Wehrmann, J., Simões, G. S., Barros, R. C., and Cavalcante, V. F. (2018). Adult Content Detection in Videos with Convolutional and Recurrent Neural Networks. *Neurocomputing*, 272:432--438.

Wu, Y. and Ji, Q. (2016). Constrained Deep Transfer Feature Learning and its Applications. In *Conference on Computer Vision and Pattern Recognition*, pages 5101--5109.

Wu, Z., Han, X., Lin, Y.-L., Gokhan Uzunbas, M., Goldstein, T., Nam Lim, S., and Davis, L. S. (2018). DCAN: Dual Channel-wise Alignment Networks for Unsupervised Scene Adaptation. In *European Conference on Computer Vision*, pages 518--534.

Xie, S., Girshick, R., Dollár, P., Tu, Z., and He, K. (2017). Aggregated Residual Transformations for Deep Neural Networks. In *Conference on Computer Vision and Pattern Recognition*, pages 1492--1500.

Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., Zemel, R., and Bengio, Y. (2015). Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. In *International Conference of Machine Learning*, pages 2048--2057.

Yamada, M., Sigal, L., and Chang, Y. (2014). Domain Adaptation for Structured Regression. *International Journal of Computer Vision*, 109(1-2):126--145.

Yang, J., Dvornek, N. C., Zhang, F., Chapiro, J., Lin, M., and Duncan, J. S. (2019). Unsupervised Domain Adaptation Via Disentangled Representations: Application to Cross-Modality Liver Segmentation. In *International Conference on Medical Image Computing and Computer Assisted Intervention*, pages 255--263. Springer.

Yuan12, Y., Liu134, S., Zhang, J., Zhang, Y., Dong, C., and Lin, L. (2018). Unsupervised Image Super-Resolution using Cycle-in-Cycle Generative Adversarial Networks. *Computer Vision and Pattern Recognition Workshop*, 30:32.

Zagoruyko, S. and Komodakis, N. (2016). Wide Residual Networks. *arXiv preprint arXiv:1605.07146*.

Zamir, A. R., Sax, A., Shen, W., Guibas, L., Malik, J., and Savarese, S. (2018). Taskonomy: Disentangling Task Transfer Learning. In *Conference on Computer Vision and Pattern Recognition*, pages 3712--3722.

Zhang, J., Li, W., and Ogunbona, P. (2017). Transfer Learning For Cross-Dataset Recognition: A Survey.

Zhang, R., Isola, P., and Efros, A. A. (2016). Colorful Image Colorization. In *European Conference on Computer Vision*, pages 649--666. Springer.

Zhang, W., Ouyang, W., Li, W., and Xu, D. (2018a). Collaborative and Adversarial Network for Unsupervised Domain Adaptation. In *Conference on Computer Vision and Pattern Recognition*, pages 3801--3809.

Zhang, Y., Miao, S., Mansi, T., and Liao, R. (2018b). Task Driven Generative Modeling for Unsupervised Domain Adaptation: Application to X-Ray Image Segmentation. In *International Conference on Medical Image Computing and Computer Assisted Intervention*, pages 599--607. Springer.

Zhang, Z., Yang, L., and Zheng, Y. (2018c). Translating and Segmenting Multimodal Medical Volumes with Cycle-and Shape-Consistency Generative Adversarial Network. In *Conference on Computer Vision and Pattern Recognition*, pages 9242--9251.

Zhao, J., Mathieu, M., and LeCun, Y. (2016). Energy-based Generative Adversarial Network. *arXiv preprint arXiv:1609.03126*.

Zhou, T., Ruan, S., and Canu, S. (2019). A Review: Deep Learning for Medical Image Segmentation Using Multi-Modality Fusion. *Array*, 3:100004.

Zhu, J.-Y., Park, T., Isola, P., and Efros, A. A. (2017a). Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. In *International Conference on Computer Vision*.

Zhu, J.-Y., Zhang, R., Pathak, D., Darrell, T., Efros, A. A., Wang, O., and Shechtman, E. (2017b). Toward Multimodal Image-to-Image Translation. In *Neural Information Processing Systems*, pages 465--476.

Zou, Y., Yu, Z., Vijaya Kumar, B., and Wang, J. (2018). Unsupervised Domain Adaptation for Semantic Segmentation via Class-Balanced Self-Training. In *European Conference on Computer Vision*, pages 289--305.