



GIJS VAN TULDER

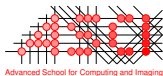
SHIFTING REPRESENTATIONS

Adventures in cross-modality
domain adaptation
for medical image analysis

Shifting representations

*Adventures in cross-modality domain
adaptation for medical image analysis*

Gijs van Tulder



This work was carried out in the ASCI graduate school.
ASCI dissertation series number 435.

Most of the research in this thesis was funded by the Netherlands Organization for Scientific Research (NWO), as part of the VIDI project “Transfer learning in biomedical image analysis” (number 639.022.010).

The publication of this thesis was supported by contributions from the ASCI graduate school and the department of Radiology and Nuclear Medicine of Erasmus MC.

Typeset by the author. Printed by ProefschriftMaken.

An electronic version of this thesis is available at www.vantulder.net.

Version: 2022-04-22

ISBN: 978-94-6423-793-1

© **Gijs van Tulder, 2022**

Except for the following chapters:

Chapter 2: © IEEE, 2016

Chapter 3: © Springer Nature, 2015

Chapter 4: © IEEE, 2019

All rights reserved. No part of this thesis may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, without written permission from the author or, when appropriate, from the publisher.

Shifting Representations

**Adventures in cross-modality domain
adaptation for medical image analysis**

Representaties veranderen

Domeinadaptatie tussen modaliteiten
voor medische beeldanalyse

Thesis

to obtain the degree of Doctor from the
Erasmus University Rotterdam
by command of the
rector magnificus

Prof. dr. A. L. Bredenoord

and in accordance with the decision of the Doctorate Board.

The public defence shall be held on
Tuesday 14 June 2022 at 15.30 hrs

by

Gijs van Tulder

born in Leiderdorp, The Netherlands

Erasmus University Rotterdam



Doctoral Committee

Promotors

prof. dr. M. de Bruijne
prof. dr. W. J. Niessen

Other members

prof. dr. J. A. Hernández Tamames
prof. dr. I. Išgum
dr. K. Kamnitsas

Contents

1	Introduction • 1
2	Combining generative and discriminative representation learning for lung CT analysis with convolutional RBMs • 7
3	Why does synthesized data improve multi-sequence classification? • 35
4	Learning cross-modality representations from multi-modal images • 49
5	Unpaired, unsupervised domain adaptation assumes your domains are already similar • 77
6	Discussion • 115
	Summary • 133
	Samenvatting • 137
	Acknowledgements • 145
	About the author • 151
	Publications • 153
	PhD portfolio • 157
	Bibliography • 161

1

Introduction

Over the last decade, machine learning has become an essential tool for medical image analysis. Improvements in methodology, computer hardware, and an increased availability of imaging data have enabled complex but effective models for a wide array of classification and segmentation tasks. Automatic image analysis is more time-efficient, and sometimes even outperforms annotation by human experts, for example, in terms of accuracy and reproducibility.

Machine learning methods learn from examples. In medical imaging, these are usually images that were annotated by medical doctors, for example with a label identifying a disease or a segmentation mask outlining some anatomical structure. Labels can also be derived from other sources, such as histopathology or clinical outcomes. The labelled examples are used to train a model that can predict the labels of new, unlabelled images from other patients. The model does this by looking for cues in pixel intensities and textures in the images, which it associates with a specific target label based on what it learned from the training examples.

The models that are produced in this way do not always generalize well to data from new sources. Because they are trained on a specific set of examples, the models learn features that work well for that specific domain. This works fine if the new images are similar to the original data, but may cause problems if the new images come from a domain with different characteristics – for example, when analyzing images from another scanner or hospital. If the model relies on features that are absent or have a different meaning in the new domain, the performance in the new domain may suffer.

This problem is called *domain shift*: a model that was trained on images from one domain (the source, e.g., scanner A) must be applied to a second domain (the target, e.g., scanner B) where the images have different characteristics. For example, Figure 1.1 shows images obtained with different MRI settings: the images show the same anatomical structures, but their appearance is different. As a result, a model that was trained on images made with one setting may not automatically work for images made with the other settings.

Domain shift is a common problem in medical imaging, because it can be time-consuming, expensive, or simply impractical to obtain and annotate data for every new domain. In machine learning research, it is often necessary to reuse existing datasets or to combine data from multiple sources to get sufficiently large datasets. In clinical applications, companies that develop medical imaging software might want to apply the same model to data from many different scanners, with different settings, and from different hospitals.

Domain shift can be addressed by *domain adaptation*, a family of machine learning methods that can adapt models that were trained on data from one domain to work well for data from another. This is usually achieved in one of two ways: by creating a single model that is domain-invariant and works reasonably well for data from both domains at the same time, or by creating a new, domain-specific model that is derived from the original model but is adapted to the characteristics of the target domain.

In this thesis, we combine domain adaptation with deep learning, a popular machine learning approach that forms the basis for most current models in medical image analysis. Deep learning methods are based on *representation learning*, using multi-layer neural networks to learn new representations from the data. For images, this is usually implemented with convolutional neural networks (CNNs). Starting from very simple representations based on simple patterns, they extract increasingly abstract features until they obtain a high-level representation that can be used to solve the prediction task.

Representation learning presents an ideal opportunity for domain adaptation: if you are learning a new representation of the data, why not learn a representation that is domain-invariant? If images from different domains are mapped to a common representation space, they can be analysed by a single prediction model that works for all domains.

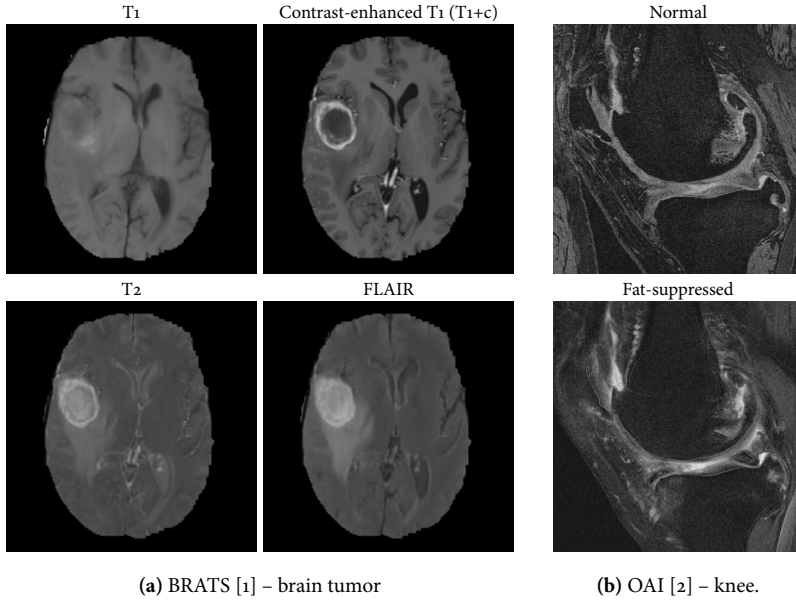


Figure 1.1: The appearance of medical images depends on the imaging settings. Left: four different MRI sequences provide radically different images of the same brain tumor. Right: two MRI images of the same knee, made with different settings. A model trained on images from one domain (e.g., T1+c) may not work well for data from another (e.g., FLAIR), because some features are absent or have a different meaning. (Image derived from Chapter 4.)

In this thesis, we focus mainly on methods that learn these *domain-invariant representations*: representations at an intermediate level in a prediction model, where images from different domains are represented in a similar way. An alternative approach is image-to-image translation, which links the domains in image space by mapping images from one domain to another. This approach is less direct, as it requires a separate model to perform the translation.

There are two ways to model domain-invariant representations in a neural network. One approach uses a shared feature encoder, extracting the same features in the same way from all domains. This works best if the domains are fairly similar. The other approach uses a separate encoding branch for each domain. This is more flexible as it can extract features in a domain-specific way that works best for each domain, but it is also more difficult to learn.

Domain adaptation methods require information to learn the relation between the source and target domains. While it is sometimes sufficient to have samples from the source domain and make assumptions about the similarities and differences between domains, most methods require some samples from the target domain. Some methods require paired samples, for example, images of the same patient scanned in two scanners. Paired samples provide a very strong link between domains, but are not always available. Other methods require unpaired but labelled target samples, which are less informative but easier to obtain. Finally, some methods require only unlabelled target samples, which are easy to get but provide relatively little information.

Models for representation-based domain adaptation must solve two problems: they must learn the main prediction task, such as classification or segmentation, and they must learn a domain-invariant representation. The main prediction task can be learned using a standard, supervised learning objective on labelled data from the source domain. The domain-invariant representation can be learned using an auxiliary domain adaptation objective that is optimised concurrently with the main prediction task. Common choices for this auxiliary objective are representation similarity, which uses paired samples to minimise the difference between representations of the same sample across domains, and feature distribution similarity, which does not require paired samples but uses methods such as domain adversarial learning to match the distributions of the representations for all samples across domains.

This thesis explores representation learning for domain adaptation in medical image analysis. We aim to learn shared representations for data from different domains, which allows us to use a single classification model that works for multiple domains. In the following chapters, we investigate how to learn useful representations and how to learn representations that work across domains, we evaluate how these representations perform in cross-domain classification, and we investigate the assumptions and limitations behind these methods.

- Chapter 2 investigates single-modality representation learning with hybrid learning objectives, using restricted Boltzmann machines (RBMs) and a combination of generative and discriminative learning to learn features for two lung CT classification tasks. By focussing on features that are relevant for classification, our models learn more informative features and achieve a

higher classification accuracy. Hybrid learning objectives are used in later chapters to combine standard generative or discriminative learning with a domain adaptation objective.

- Chapter 3 investigates cross-modality image synthesis, using autoencoders and RBMs to learn cross-modality representations and impute missing MRI sequences for multi-modal brain segmentation. Synthesizing missing images improves performance in applications with incomplete datasets, e.g., when combining data from multiple sources with different modalities. In the context of this thesis, image synthesis also provides a way to translate images between domains.
- Chapter 4 investigates modality-invariant feature learning using convolutional neural networks, which is evaluated in cross-modality classification experiments in two multi-modal MRI datasets. Each network learns a shared representation for data from different domains, which is then used as input for a shared classification model. The classification performance in cross-modality settings, training on data from one domain and evaluating on another, comes close to that of same-modality classifiers that are trained and evaluated on the same domain.
- Chapter 5 investigates the limits of representation learning in unpaired, unsupervised domain adaptation, with experiments on synthetic data and on MRI data from a multi-modal brain imaging dataset. While there are many practical examples of successful domain adaptation in medical imaging, the results are usually based on assumptions about similarities between the domains. This chapter explores what these assumptions can be and how they affect the domain adaptation results.
- Chapter 6 summarises and discusses the main findings of this thesis, looks at limitations and future work, and formulates a general conclusion.

2

Combining generative and discriminative representation learning for lung CT analysis with convolutional RBMs

The choice of features greatly influences the performance of a classification system. Despite this, many systems are built with standard, predefined filter banks that are not optimized for that particular application. Representation learning methods such as restricted Boltzmann machines may outperform these standard filter banks because they learn a feature description directly from the training data. Restricted Boltzmann machines are unsupervised and are trained with a generative learning objective; this allows them to learn representations from unlabeled data, but does not necessarily produce features that are optimal for classification. In this chapter we propose the convolutional classification restricted Boltzmann machine, which combines a generative and a discriminative learning objective. This allows it to learn filters that are good both for describing the training data and for classification. We present experiments with feature learning for lung texture classification and airway detection in CT images. Results on both tasks show that adding a discriminative objective offers consistent improvements over the purely generative, unsupervised RBM. Additionally, combining the two objectives offers improvements over just the discriminative objective in the lung tissue classification task, increasing classification accuracy by 1 to 8 percentage points. This shows that discriminative learning can help an otherwise unsupervised feature learner to learn filters that are optimized for classification.

Chapter based on

G. van Tulder and M. de Bruijne, “Combining Generative and Discriminative Representation Learning for Lung CT Analysis With Convolutional Restricted Boltzmann Machines,” *IEEE Transactions on Medical Imaging*, vol. 35, no. 5, pp. 1262–1272, 2016. DOI: 10.1109/TMI.2016.2526687.

2.1 *Introduction*

Most methods for automated image classification do not work directly with image data, but first extract a higher-level description of useful features from the image. The choice of features determines a large part of the classification performance. Which features work well depends on the nature of the classification problem: for example, some problems require features that preserve and extract scale differences, whereas other problems require features that are invariant to those properties. Often, feature representations are based on standard filter banks of common feature descriptors, such as Gaussian derivatives that detect edges in the image. These predefined filter banks are not specifically optimized for a particular problem or dataset.

As an alternative to such predefined feature sets, representation learning or feature learning methods [3] learn a high-level representation directly from the training data. Because this representation is learned from the training data, it can be optimized to give a better description of the data. Using this learned representation as the input for a classification system might give a better classification performance than using a generic set of features.

In this chapter, we focus on unsupervised models that are trained with unlabeled data. While this can be an advantage because it makes it easier to create a large training set, it can also lead to suboptimal results for classification, because the features that these methods learn are not necessarily useful to discriminate between classes. Unsupervised feature learning tends to learn features that model the strongest variations in the data, while classifiers need features that discriminate between classes. If the variation between samples from the same class is much stronger than the variation between classes, feature learning is likely to produce features that capture primarily within-class variation. If those features do not represent enough between-class variation, they might give a lower classification performance if they are used as the input for a classification model.

This issue of within-class variation is relevant for many applications, including those in medical image analysis. For example, in disease classification, the differences between patients are often greater than the subtle differences between disease patterns. As a result, representation learners might learn features that model these between-patient differences, rather than features that improve classification performance.

In this chapter we study the restricted Boltzmann machine (RBM), a popular representation learning model, as a way to learn features that are optimized for classification. The standard RBM does not include labels and is trained with an unsupervised, generative learning objective. The classification RBM [4], an extension of the standard RBM, does include label information and can also be trained with a discriminative learning objective. This discriminative learning objective optimizes the classification performance of the classification RBM. The generative and discriminative objectives can be combined to learn discriminative features that represent the data and are useful for classification.

We propose the convolutional classification RBM, which combines the classification RBM with the convolutional RBM, another extension of the standard RBM. The convolutional RBM [5–8] uses the convolutional weight-sharing pattern from convolutional networks to learn small filters that are applied to every position in a larger image. This weight sharing makes learning more efficient and allows the RBM to model small features that occur in multiple areas of an image, which is useful for describing textures.

The ability to combine generative and discriminative learning objectives distinguishes the classification RBM from many other representation learning methods. Unsupervised models such as the standard RBM are usually trained with only a generative learning objective, whereas supervised representation learning methods, such as convolutional neural networks [9], are usually trained with only a discriminative learning objective. The classification RBM can be trained with a generative objective, a discriminative objective, or both.

We present experiments on lung tissue classification and airway detection. For the lung tissue classification experiments we used a dataset on interstitial lung diseases (ILD) [10] with CT images of 73 patients. Previous tissue classification work on this dataset used wavelets [11–14], local binary patterns [15, 16], bag-of-visual-words [17, 18], filter banks derived from the discrete Fourier transform [19], RBMs [20, 21], and convolutional neural networks [22].

We used RBMs to learn features for lung tissue classification. From the images, we first extracted 2D patches that we used to train RBMs with different mixtures of discriminative and generative learning. Using the RBM-learned representations, we trained and evaluated classifiers that classify each patch in one of the five tissue classes. We compared those results with those of two standard filter banks.

We expected the effect of discriminative learning to become less important for larger representations (more hidden nodes in the RBM), because larger representations are more likely to contain sufficient discriminative features even without explicit discriminative learning. To study this effect, we performed airway detection experiments on lung CT images from the Danish Lung Cancer Screening Trial (DLCST) [23]. We used non-convolutional classification RBMs with different mixtures of discriminative and generative learning to learn features for this dataset. The non-convolutional RBMs allowed us to experiment with larger numbers of hidden nodes.

This chapter is organized as follows. Section 2.2 gives an overview of other relevant representation learning work. Section 2.3 describes the RBM and its learning algorithm. Section 2.4 introduces the datasets and the experiments. Section 2.5 describes the results. Sections 2.6 and 2.7 conclude the chapter.

2.2 *Related work*

Representation learning methods have been used for tissue classification in lung CT before. In experiments similar to those presented in this chapter and using the same ILD dataset, Li et al. [20] used RBMs to extract features. Whereas we use classification RBMs with convolution to learn small filters, Li et al. trained standard (non-convolutional) RBMs on small subpatches extracted from the patch that is to be classified. In later work [21] on the same dataset, Li et al. reported that convolutional neural networks gave a slightly better performance than standard RBMs. Gao et al. [22] used convolutional neural networks to classify full slices from the ILD dataset, without requiring manually annotated ROIs. Schlegl et al. [24] also used convolutional neural networks to classify lung tissue in a different lung CT dataset.

Convolutional neural networks have also been used in other applications of lung CT, such as the detection of lung nodules and lymph nodes. In an early application of convolutional neural networks, Lo et al. [25, 26] trained a network to reject or confirm potential lung nodules selected in a preprocessing step. More recently, Shen et al. [27] used multi-scale convolutional networks to compute features for lung nodule classification. Kumar et al. [28] used multi-layer autoencoders to extract features for the classification of lung nodules. Roth et al. [29] proposed a so-called 2.5D convolutional neural network that samples orthogonal 2D views to detect lymph nodes in lung CT images.

To our knowledge, classification RBMs have not been applied to lung CT images before, and there are only a few applications in other types of medical image analysis. Shin et al. [30] used classification RBMs to detect micro-calcifications in digitized mammograms. Berry and Fasel [31] used translational deep Boltzmann machines, which are related to classification RBMs, to analyze ultrasound images of the tongue. Schmäh et al. [32] analyzed fMRI data with RBMs with generative and discriminative learning.

2.3 Restricted Boltzmann machines

2.3.1 Standard RBM

A restricted Boltzmann machine is a probabilistic neural network that learns the probability distribution of its inputs \mathbf{v} and a hidden representation \mathbf{h} . The visible nodes \mathbf{v} represent the voxels of an input patch. To model the patches from our lung CT images, we use Gaussian visible nodes \mathbf{v} and binary hidden nodes \mathbf{h} (see [33] for a description of these node types). Each visible node v_i has an undirected connection with weight $W_{ij} \in \mathbb{R}$ to each hidden node h_j . The model also includes a bias $b_i \in \mathbb{R}$ for each visible node v_i and a bias $c_j \in \mathbb{R}$ for each hidden node h_j . Together, the weights and biases define the energy function of the RBM:

$$E(\mathbf{v}, \mathbf{h}) = \sum_i \frac{(v_i - b_i)^2}{2\sigma_i^2} - \sum_{i,j} \frac{v_i}{\sigma_i} W_{ij} h_j - \sum_j c_j h_j, \quad (2.1)$$

where σ_i is the standard deviation of the Gaussian noise of visible node i . We normalize the training patches such that $\sigma_i = 1$. The joint distribution of the input \mathbf{v} and hidden representation \mathbf{h} is defined as

$$P(\mathbf{v}, \mathbf{h}) = \frac{\exp(-E(\mathbf{v}, \mathbf{h}))}{Z}, \quad (2.2)$$

where Z is a normalization constant. The conditional probabilities for the hidden nodes given the visible nodes and vice versa are

$$P(h_j | \mathbf{v}) = \text{sigm}(\sum_i W_{ij} v_i + c_j) \quad (2.3)$$

$$P(v_i | \mathbf{h}) = \mathcal{N}(v_i | \sum_j W_{ij} h_j + b_i, \sigma_i^2), \quad (2.4)$$

where $\text{sigm}(x) = \frac{1}{1+\exp(-x)}$ is the logistic sigmoid function and $\mathcal{N}(x | \mu, \sigma^2)$ is a Gaussian probability density function with mean μ and variance σ^2 .

2.3.2 Classification RBM

The standard RBM is an unsupervised model. The classification RBM [4] extends the standard RBM by adding a set of label nodes to the visible layer (Figure 2.1). This allows the RBM to learn the joint probability of the input, the hidden representation, and the label. The label nodes use a one-hot coding, where there is one node y_k per class such that $y_k = 1$ if the sample belongs to class k and $y_k = 0$ otherwise. The label nodes have a bias $d_k \in \mathbb{R}$ and are connected to the hidden nodes, with a connection with weight $U_{kj} \in \mathbb{R}$ between label node y_k and hidden node h_j . The energy function of a classification RBM with Gaussian visible nodes is

$$E(\mathbf{v}, \mathbf{h}, \mathbf{y}) = \sum_i \frac{(v_i - b_i)^2}{2\sigma_i^2} - \sum_{i,j} \frac{v_i}{\sigma_i} W_{ij} h_j - \sum_j c_j h_j - \sum_{k,j} y_k U_{kj} h_j - \sum_k d_k y_k. \quad (2.5)$$

The energy function defines the distribution

$$P(\mathbf{v}, \mathbf{h}, \mathbf{y}) = \frac{\exp(-E(\mathbf{v}, \mathbf{h}, \mathbf{y}))}{Z} \quad (2.6)$$

and the conditional probabilities

$$P(h_j | \mathbf{v}, \mathbf{y}) = \text{sigm}(\sum_i W_{ij} v_i + \sum_k U_{kj} y_k + c_j) \quad \text{and} \quad (2.7)$$

$$P(y_k | \mathbf{h}) = \text{sigm}(\sum_j U_{kj} h_j + d_k). \quad (2.8)$$

The visible nodes and the label nodes are not connected, so the expression for $P(v_i | \mathbf{h})$ is unchanged from the standard RBM. The posterior probability for classification is

$$P(y | \mathbf{v}) = \frac{\exp(d_y + \sum_j \text{softplus}(c_j + U_{jy} + \sum_i W_{ij} v_i))}{\sum_{y^*} \exp(d_{y^*} + \sum_j \text{softplus}(c_j + U_{jy^*} + \sum_i W_{ij} v_i))}, \quad (2.9)$$

where $\text{softplus}(x) = \log(1 + \exp(x))$. This definition only works for RBMs with binary hidden nodes: it implicitly sums over all possible states of the hidden layer, which can be done efficiently if each hidden node can take one of only two values [4].

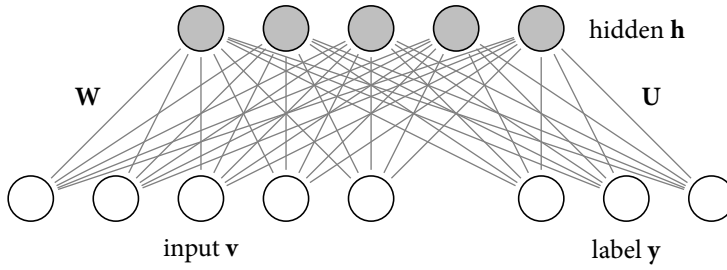


Figure 2.1: Schematic view of the classification RBM, which adds a set of label nodes to the visible layer of the standard RBM. The label nodes are connected to the input nodes through the hidden layer.

2.3.3 Generating samples and classifying with RBMs

RBMs are probabilistic models that define the activation probability for each node given all other nodes. In practice, computing the probability of a particular state \mathbf{v}, \mathbf{h} is impossible, because the normalization constant or partition function Z in the energy function is infeasible to compute for any but the smallest models. However, since it is possible to compute the conditional probabilities, we can still use Gibbs sampling to sample from the model. Gibbs sampling alternately samples from the hidden and visible layers. Given a random initialization of the visible and label nodes, the new state of the hidden nodes can be sampled using the distribution $p(\mathbf{h}_t | \mathbf{v}_t, y_t)$. Then, keeping the hidden nodes fixed, the new activation of the visible and label nodes can be sampled from $p(\mathbf{v}_t, y_t | \mathbf{h}_t)$. This can be repeated for several iterations, until the model converges to a stable state. For simplicity, we used a fixed number of iterations in our experiments.

Classifying a patch using the classification RBM is more straightforward. We input the patch values in the visible layer \mathbf{v} and use Equation (2.9) to compute the posterior probability $P(y | \mathbf{v})$ for each class. We assign the label of the class with the highest posterior probability.

2.3.4 Learning objectives

At training time, the weights and biases of the standard RBM are chosen to optimize the generative learning objective $\log P(\mathbf{v}_t)$, the probability distribution of each input image t . The classification RBM can be trained with the

generative learning objective $\log P(\mathbf{v}_t, y_t)$, which optimizes the joint probability distribution of the input image and the label. A classification RBM can also be trained with the discriminative objective $\log P(y_t | \mathbf{v}_t)$, which only optimizes the classification and does not try to optimize the likelihood of the input image. Larochelle et al. [4] suggest a hybrid objective

$$\beta \log P(\mathbf{v}_t, y_t) + (1 - \beta) \log P(y_t | \mathbf{v}_t), \quad (2.10)$$

where $\beta \in [0, 1]$ is the proportion of generative learning. We will use this objective with different values for β in our feature learning experiments.

The normalization constant or partition function Z makes it unfeasible to compute the gradient of the generative learning objective. Instead, we use Gibbs sampling and contrastive divergence [33] to estimate the stochastic gradient descent updates for our RBMs. Contrastive divergence provides an efficient approximation for the gradient-based updates to the weights and biases of the model.

Classification RBMs are slightly more computationally expensive than unsupervised RBMs, because they use an additional discriminative learning objective and include extra weights to connect the label nodes. In practice however, we find that the classification RBMs are not much slower than the unsupervised RBMs, because the additional complexity from the discriminative components is small compared with the other parts of the RBM. The number of labels and the number of associated weights is usually much smaller than the number of connections between the visible and hidden layers, and the discriminative learning objective can be computed much faster than the generative objective, which requires contrastive divergence and Gibbs sampling.

2.3.5 Convolutional RBM

Designed to model complete images instead of small patches, convolutional RBMs [5–8] use the weight-sharing approach from convolutional neural networks. Unlike convolutional neural networks, convolutional RBMs are generative models and can be trained in the same way as standard RBMs. In a convolutional RBM, the connections share weights in a pattern that resembles convolution, with M convolutional filters \mathbf{W}_m that connect hidden nodes arranged in M feature maps \mathbf{h}_m (Figure 2.2). The connections between the visible nodes and the hidden nodes in map m use the weights from convolution

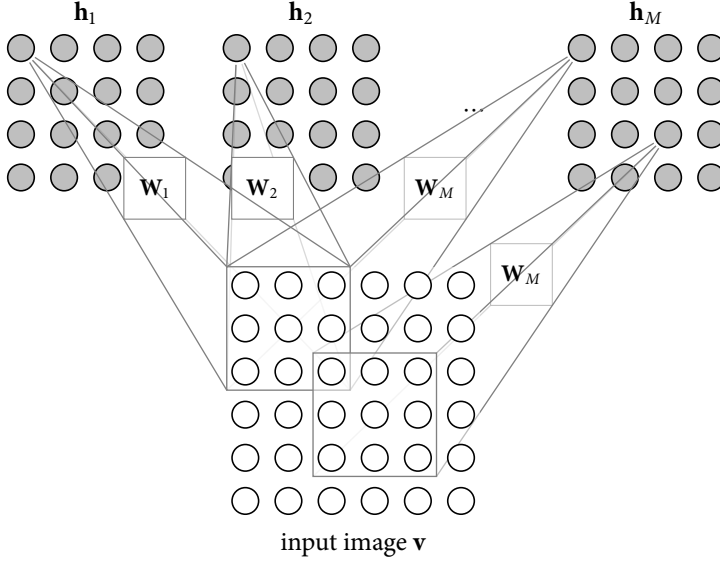


Figure 2.2: Schematic view of the convolutional RBM, which uses a convolutional weight-sharing arrangement to reduce the number of connection weights.

filter \mathbf{W}_m , such that each hidden node is connected to the visible nodes in its receptive field. The visible nodes share one bias b ; all hidden nodes in map m share the bias c_m . With the convolution operator $*$ we define the probabilities

$$P(h_{ij}^m | \mathbf{v}) = \text{sigm}((\tilde{\mathbf{W}}_m * \mathbf{v})_{ij} + c_m) \text{ and} \quad (2.11)$$

$$P(v_{ij} | \mathbf{h}) = \mathcal{N}(v_{ij} | (\sum_m \mathbf{W}_m * \mathbf{h}_m)_{ij} + b, 1), \quad (2.12)$$

where $\tilde{\mathbf{W}}_m$ is the horizontally and vertically flipped filter \mathbf{W}_m , and \cdot_{ij} denotes the voxel on location (i, j) .

Convolutional RBMs can produce unwanted border effects when reconstructing the visible layer, because the visible nodes near the borders are only connected to a few hidden nodes. We pad our patches with voxels from neighboring patches, and keep the padding voxels fixed during the iterations of Gibbs sampling.

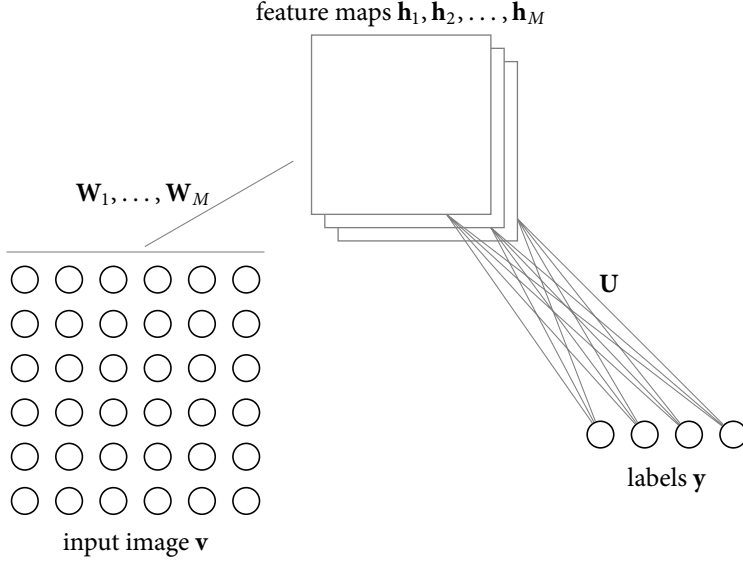


Figure 2.3: Schematic view of the convolutional classification RBM. The connection weights U are shared between all nodes in a feature map.

2.3.6 Convolutional classification RBM

We introduce a convolutional classification RBM that includes visible, hidden and label nodes (Figure 2.3) and can be trained in a discriminative way. The visible nodes are connected to the hidden nodes using convolutional weight-sharing, as in the convolutional RBM, and the hidden nodes are connected to the label nodes, as in the classification RBM. In our patch-based texture classification problem, the exact location of a feature inside the patch is not relevant, so we use shared weights to connect the hidden nodes and the label nodes. All connections from a label node y_k to a hidden node h_{ij}^m in map m share the weight U_{km} . The activation probabilities are

$$P(y_k | \mathbf{h}) = \text{sigm}\left(\sum_m U_{ym} \sum_{i,j} h_{ij}^m + d_k\right) \text{ and} \quad (2.13)$$

$$P(h_{ij}^m | \mathbf{y}) = \text{sigm}\left((\tilde{\mathbf{W}}_m * \mathbf{v})_{ij} + \sum_k U_{km} y_k + c_m\right). \quad (2.14)$$

Since the label nodes are not connected to the visible nodes, the probability for the visible nodes is unchanged from the convolutional RBM.

2.4 Experiments

We present experiments on lung CT images for two applications and datasets: lung tissue classification and airway centerline detection. On the lung tissue dataset, we studied the effects of combining generative and discriminative learning objectives. On the airway dataset, we explored how these effects change if the representation is larger (more hidden nodes).

2.4.1 Dataset 1: Lung tissue classification

Purpose. This set of experiments studied the effect of combining generative and discriminative learning objectives. We trained RBMs with purely discriminative ($\beta = 0$), with purely generative ($\beta = 1$), and with mixed learning objectives. We then used the RBM-learned filters to compute feature vectors and train a classifier. The classification accuracy gives an indication of the quality of the learned representations.

Data. We used a publicly available dataset on interstitial lung diseases (see [10] for a description). In this texture classification problem with 73 scans from different patients, we classify patches of five types of lung tissue. The in-plane voxel size varies between 0.4 – 1 mm, with a slice thickness of 1 – 2 mm and inter-slice spacing of 10 – 15 mm. The dataset provides hand-drawn 2D ROIs with labels for a subset of slices in each scan (Figure 2.4). Following other work on this dataset (e.g., [13]), we extracted patches of 32×32 voxels along a grid with a 16-voxel overlap. We include a patch if at least 75% of the voxels belong to the same class. We classify patches from the five most common tissue types in the dataset (healthy tissue: 22%, emphysema: 3%, ground glass: 16%, fibrosis: 15%, micronodules: 44% of the patches).

Experiments. We used the convolutional RBM, with and without labels, to learn filters from the patches in the lung tissue dataset. We then used these filters in a convolution to get feature maps for each of the patches in the dataset. For each feature map, we computed a histogram of the feature activations, using adaptive binning [34] over all patches in the training set. The concatenated histograms form the feature vector for each patch. We trained random forest classifiers to classify each patch in one of the five tissue classes.

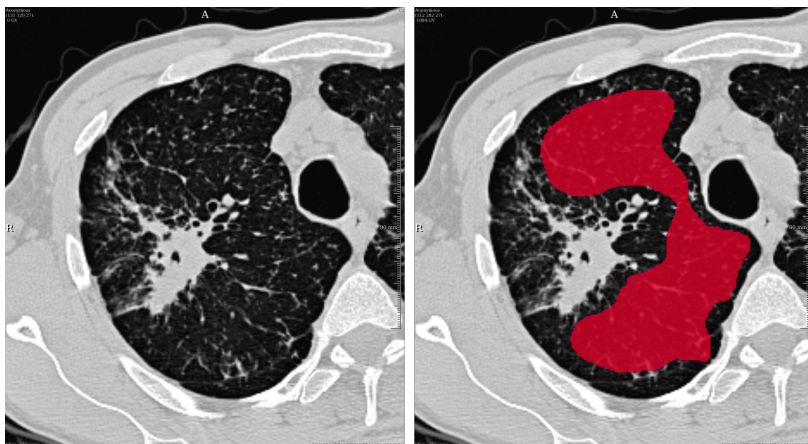


Figure 2.4: First dataset. Example from the interstitial lung disease scans. The annotation (right) shows an ROI (red) marked as micronodules.

Normalization. We trained the RBMs on normalized patches, with each patch normalized to zero mean intensity and unit standard deviation. We used unnormalized patches to compute the feature maps and histograms, to preserve the intensity differences between patches.

Baselines. We compare the results of the RBMs with those of several other methods. First, we show the performance of using random filters, using the same convolutional architecture but without optimizing the filter weights (see Figure 2.5 for an example). The results of random filters help to separate the contribution of feature learning from that of the convolutional architecture [35]. We also compare the RBM-learned filters with two of the frequently-used standard filter banks discussed by Varma and Zisserman [36]: the Leung-Malik and Schmid filter banks (Figure 2.5). The filter bank of Leung and Malik [37] is a set of Gaussian filters and derivatives, with 48 filters of 32×32 voxels. The filter bank of Schmid [38] has 13 filters of 31×31 voxels with rotation-invariant Gabor-like patterns.

Implementation and parameters. We implemented the RBMs in Python using the Theano library [39] and used the random forest implementation from Scikit-learn [40]. To optimize the learning parameters for the RBMs and

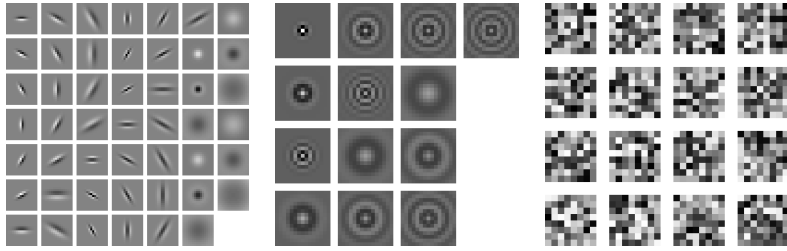


Figure 2.5: Two filter banks: Leung-Malik (left) and Schmid (middle), generated with code from <https://www.robots.ox.ac.uk/~vgg/research/texclass/filters.html>. An example of random filters (16 filters of 8×8 voxels) is shown right.

random forests, we performed a grid search using nested cross-validation with patches from the same scan grouped in the same fold. We tried various learning rates for the RBM (10^{-3} to 10^{-9}). For the random filters, we chose the best filter set out of five random initializations. We used 2 to 8 bins in the adaptive binning step. For the random forests, we varied the number of trees (10 to 200) and the maximum number of features (1 to 256), and used Scikit-learn’s default parameters for the other settings.

The initial values for the connection weights \mathbf{W} of the RBM were sampled from a normal distribution with mean 0 and standard deviation 10^{-6} . The initial values for the connection weights \mathbf{U} of the classification RBMs were sampled from a uniform distribution $[-10^{-6}, 10^{-6}]$. All biases had the initial value 0. During stochastic gradient descent we used a minibatch size of 5, with one Gibbs sampling step for contrastive divergence.

Cross-validation. Almost all scans have manually-drawn ROIs for only one tissue type. We organized the scans in five folds, of 15 or 14 scans each, while trying to create a similar class distribution in each fold. We present the cross-validation accuracy over all five folds. In each cross-validation step we used one fold for testing and the remaining four folds for classifier training and parameter tuning. For each fold, we computed the mean accuracy over all patches. Within each cross-validation step, we optimized the RBM and random forest parameters using nested cross-validation with one validation and three training folds. We used the parameters that gave the best accuracy over the four folds to train a classifier on the full training set, which we then used to classify the patches from the scans in the test fold.

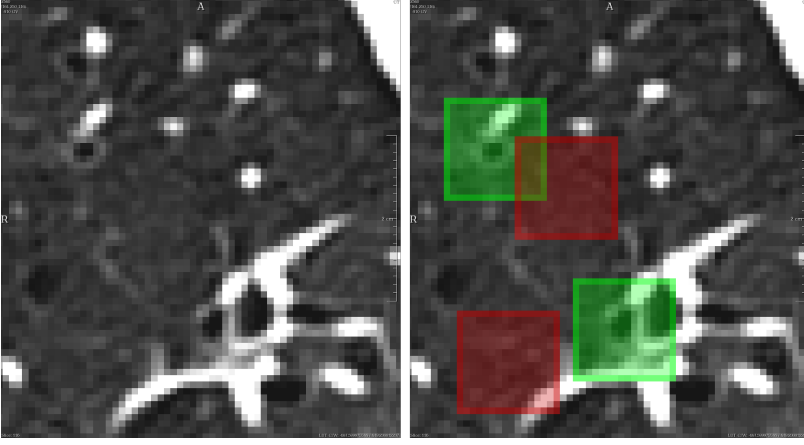


Figure 2.6: Second dataset. In the airway dataset, we extract patches at the airway centerline (green) and non-airway samples (red) close to the airway.

We report the mean classification accuracy over all five folds in the cross-validation. We used the Wilcoxon signed-rank test to test for significant differences between methods ($p < 0.05$). In these tests we compared the classification accuracy per scan (73 measurements per method).

2.4.2 Dataset 2: Airway centerlines

In our second set of experiments we explored the influence of the size of the representation – the number of hidden nodes in the RBM. Since it was computationally unfeasible to train the convolutional RBM with a very large number of filters, we performed these experiments on a different problem with a classification RBM without convolution. We used 40 lung CT scans from 20 participants of the Danish Lung Cancer Screening Trial (DLCST) [23]. The voxel size is approximately $0.78 \times 0.78 \times 1$ mm. Using the output of an existing segmentation algorithm [41] to find the airways (Figure 2.6), we extracted patches of 16×16 voxels at the center point of airways with a diameter of 16 voxels or less. For each airway patch, we created a non-airway sample by extracting a patch at a random point just outside the outer airway wall. We selected a random subset of 500 patches per scan. We used 15 subjects (30 scans, 15 000 patches) as our training set and 5 subjects (10 scans, 5 000 patches) for testing.

The implementation and parameters were similar to those for the tissue classification dataset, with a few differences. Because the airway in this dataset is always in the center of the patch, we could use RBMs without convolution to learn a representation. We used between 1 to 256 nodes in the hidden layer. We used the scans from the training set to train classification RBMs and standard RBMs. Using the representation in the hidden layer of the RBM to create the feature vectors, we trained random forests to classify airway and non-airway voxels. We optimized the parameters of the random forests using cross-validation on the training set. We report the classification accuracy of the classification RBMs and of the random forests on the test set.

2.5 *Results*

2.5.1 *Filters*

Figure 2.7 shows filters learned by the RBM from the lung tissue classification dataset, for various mixtures of generative and discriminative learning. Because of the random initialization, each set of filters looks different, but we observed no consistent visual difference between filters learned with discriminative or generative learning. The filters are useful for the classification models, but there are no recognizable structures. With the non-convolutional RBM, which we used for the airway dataset, the filters show more recognizable structures (Figure 2.8). The filters show circular structures that resemble the airways in the training set: a centered, dark circle to represent the airway, and white blobs that could represent the vessel that is often next to the airways. With a small number of filters, the RBM learned more general filters, whereas an RBM with more filters learned filters that can represent more specific structures.

2.5.2 *Random forest classification*

Figure 2.9 shows the random forest classification results comparing RBM-learned filters with different filter banks. The classification accuracy achieved using the RBM-learned filters with the best β was better than that using random filters or one of the predefined filter banks. Random filters and, in most cases, the Schmid filters performed significantly worse than the RBM-learned filters. The difference with the Leung-Malik filter bank was often not significant. The best performance was achieved using 16 filters of 5×5 voxels.

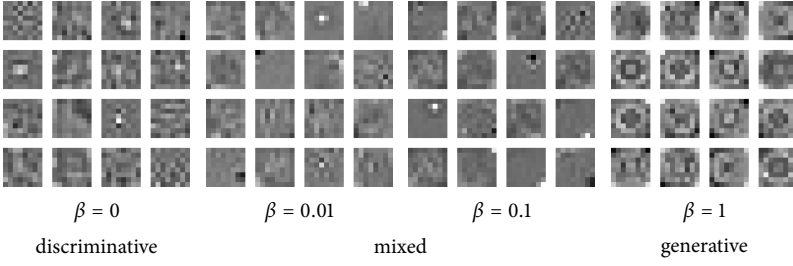


Figure 2.7: Example filters learned from the ILD dataset, with different mixtures of generative and discriminative learning (16 filters of 8×8 voxels).

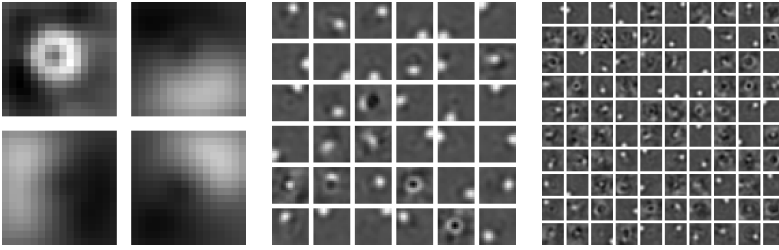


Figure 2.8: Filter sets learned from the airway data: 4, 36 or 100 filters of 16×16 voxels, learned with a mix of discriminative and generative learning ($\beta = 0.01$).

Pure generative or discriminative learning usually performed worse than a mixture of learning objectives. The effects of using different values for β were most visible with the larger filters. At most filter sizes, except for very small or very few filters, using a combination of generative and discriminative learning seems to give better results than using purely generative or discriminative learning. The classification accuracy increases as β decreases, until it decreases again when there is too much discriminative learning, which increases the risk of overfitting.

2.5.3 RBM classification

We also evaluated the classification performance of the RBM itself, using Equation (2.9) to compute the posterior probability for each class. The accuracy of the RBM was always lower than that of the random forests (Figure 2.10). With only a generative learning objective, the classification accuracy of the RBM was poor, presumably because this model optimized only for representation and

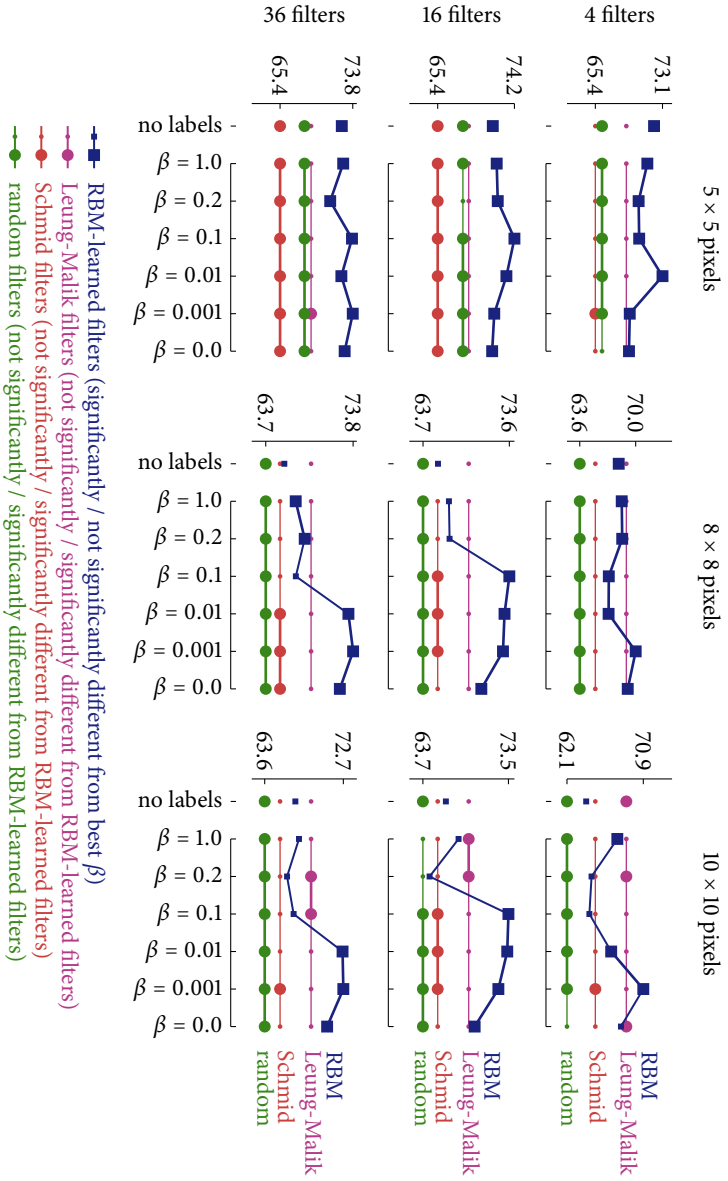


Figure 2.9: Random forest classification accuracy on the lung tissue classification dataset, for different feature representations. Large squares indicate RBM results that are not significantly different ($p < 0.05$) from the best RBM result for that network configuration. Large circles indicate results that are significantly different from the RBM result at that β . (All significance values were computed using Wilcoxon signed-rank tests comparing the per-scan classification accuracies.)

not for classification. Using the discriminative learning objective improved the accuracy, but it was still significantly lower than that of a random forest trained on the RBM hidden layer. One reason may be that the classification model of the RBM is much simpler than that of the random forests. The RBM has a linear decision function (given the state of the hidden layer) and does not compute histograms of the feature activations. In addition, the RBM optimization may be complicated by the fact that the RBM optimizes two things at the same time (representation and classification).

2.5.4 *Influence of the size of the representation*

We explored the effects of the filter size and the number of filters on the classification performance. We expected that discriminative learning would become less important as the number of filters increases, because a larger representation is more likely to include discriminative features even with generative learning.

Figure 2.9 shows the results for multiple network configurations with different filter sizes and numbers of filters on the tissue classification problem. There seems to be a connection between the number of filters and the point at which the accuracy increases. With more filters, more discriminative learning (a smaller β) is needed. This could be a consequence of the implementation of the gradients of the energy function: in an RBM with many filters, the values in the energy function (and the corresponding gradients) might be larger than when the number of filters is smaller. The number of filters influences the gradient for the generative learning objective, but not the discriminative objective. To achieve the right balance between discriminative and generative learning, the β should be smaller for smaller number of filters to compensate for the larger gradients. Note that the number of filters is relatively small (up to 36), which may make generative learning less effective.

For a closer look at the effect of the representation size, we performed additional experiments with non-convolutional RBMs on the airway dataset (Figure 2.11) and a larger number of filters. On this dataset, using only or mostly discriminative learning generally gave the best results. The performance of generative learning depended on the number of hidden nodes. With only a few hidden nodes, generative learning performed worse than discriminative learning. As we increased the size of the representation, the gap between gen-

erative and discriminative learning almost disappeared. This seems to agree with our hypothesis that at the smaller representations, the discriminative objective helps to learn discriminative features, whereas the generative objective produces features that are useful for representation but are less discriminative. As we increased the number of hidden nodes, generative learning produced enough features to also include some of the discriminative features.

2.6 Discussion

We have shown how the classification RBM can be used to learn useful features for medical image analysis, achieving a mean classification accuracy that was better than or close to that achieved using a predefined set of features. To get good classification results in feature learning, it is important to use the right learning objective. We found that adding label information and discriminative learning to the standard RBM helps to produce filters that improve performance. In some cases pure discriminative learning worked best, but in most cases a mixture with generative learning gave better results. The results show that RBM-learned filters have an advantage over random filters and two standard filter banks.

Random filters performed quite well in our experiments, although they generally performed worse than the filter banks and RBM-learned filters. The surprisingly good performance of random filters has already been noted in the literature [35]. When the number of filters is large enough, convolution with random filters can provide useful features to train a classifier. The performance of random filters is a useful baseline because it allows us to separate the contribution of the convolutional architecture from that of the feature learning algorithm. The performance difference between learned and random filters indicates that the improvement is not just an effect of using a convolution operator with a number of arbitrary filters.

2.6.1 Results on the ILD dataset

The ILD dataset [10] was also used in other papers. We will give a brief overview of the techniques and the results before comparing them with our own.

Depeursinge et al., the providers of the dataset, used wavelet transforms and intensity and gradient features [11–13] to classify tissue patches. They also used this tissue classification system as a component of a larger image retrieval

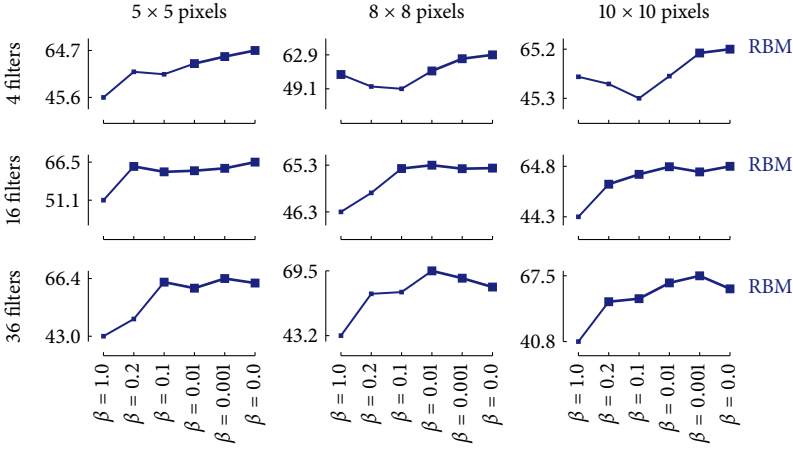


Figure 2.10: The RBM classification accuracy on the lung tissue classification dataset, for different feature representation methods. Large squares indicate results that are not significantly different from the best result for that network configuration. (All significance values were computed using Wilcoxon signed-rank tests comparing the per-scan classification accuracies.)

system [14]. From the same group, Foncubierta-Rodríguez et al. [17] proposed a retrieval system based on visual words. The five-class classification accuracy reported in these papers ranges between 76.1% and 80.8%.

Song et al. used texture, intensity and gradient features, combined with features based on rotation-invariant Gabor-local binary patterns and histogram of oriented gradients. Song et al. [15] first used a dictionary to approximate the test patch using training patches and then used the approximation error to classify the patch. They combined this approach with a large-margin local estimate method to cluster example patches [16], with a reported classification accuracy of 86.1%. A related method [42], also based on clustering, provided a 85.8% classification accuracy. Earlier, the same authors also used local binary patterns [43] and boosting [44].

Asherov et al. [18] used bags of visual words to classify patches, reporting an accuracy of 79%. Anthimopoulos et al. [19] used filter banks derived from a discrete cosine transform, which performed better than Leung-Malik, Schmid, Gabor and MR8 filters. Dash et al. [45] presented segmentation methods using Markov random fields, Gaussian mixture models and mean-shift algorithms.

Several papers applied representation learning methods to the ILD dataset. Li et al. presented experiments using RBMs to extract features [20], which gave a classification accuracy of 77%. In a later comparison, Li et al. reported that convolutional neural networks gave a slightly better performance [21] (no accuracy given). Gao et al. [22] used convolutional neural networks to classify full slices, without requiring manually annotated ROIs. Their patch-based classification showed a classification accuracy of 87.9%.

It is difficult to compare the results of our experiments with those in the previously published studies. Although many papers use a similar approach to extract patches, there are differences in cross-validation procedures and in the number of patches. Overall, our classification results seem to be in the same range, but worse than the state-of-the-art results [16]. Part of this may be due to a difference in training set size – the papers with better results use leave-one-patient-out cross-validation (e.g., [16]), whereas we used five-fold cross-validation for computational reasons. Other differences may also be important, such as the number of features (we used a relatively small number of filters, also for computational reasons) and the amount of post-processing.

2.6.2 *How much discriminative learning is required?*

There is no single optimal mixture of discriminative and generative learning. The optimal choice for β depends on the number and size of the filters, on the application, and on the dimensions of the data. The results from our lung tissue classification experiments (Figure 2.9) show that the influence of β is strongest for RBMs with larger filters, with lower β (more discriminative learning) giving a better classification accuracy. The effect of the number of filters or the number of hidden nodes is more easily visible in the results of the airway centerline experiments (Figure 2.11), which show that discriminative learning becomes less important for models with more hidden nodes. Some of these trends will be a result of the definition of the generative learning objective, which is derived from an energy function that tends to be larger for RBMs with many connections (more or larger filters). The remainder of the effect may be explained by the difficulty of finding a set of discriminative features. This difficulty is influenced by two factors: the number and the size of the filters. A model with only a few filters may require more discriminative learning than a model with many filters: a large set of filters is more likely to

contain some that are useful for classification even if the filters are learned with generative learning, but with a small set of filters it is necessary to be selective. Similarly, a model with large filters may require more discriminative learning than a model with small filters, because the model with larger filters has a larger search space: a model with larger filters can find more different filters, which makes it more important to be selective.

The optimal β also depends on the application and dataset. If it is difficult for the RBM to learn the classification rule, such as in our lung tissue classification experiments, a mixture with generative learning proved to work better than purely discriminative learning. On a somewhat easier problem such as our airway centerlines, purely discriminative learning often also gave good results.

Finally, the optimal mixture depends on the dimensions of the input data. In this work, we chose to do feature learning and classification in 2D, because the lung tissue data that we used in our experiments is highly anisotropic and has only 2D annotations. However, given the right training data, the methods discussed in this work can be extended to 3D. Having 3D inputs increases computational complexity, which is sometimes a reason to use pseudo-3D, as in [29] where 3D data is modeled with a set of orthogonal 2D planes. If real 3D is used, it is important to limit the number of filters. At the same time, a 3D model will require more filters to model the training patches effectively. In those cases a mixture of generative and discriminative learning could help to learn fewer but better filters.

2.6.3 Further considerations

Since the mixture of generative and discriminative learning objectives can improve performance for RBMs, it might be interesting to try this combination for other representation learning methods, such as convolutional neural networks, deep belief networks or deep Boltzmann machines. However, this requires definitions for both the generative and the discriminative objective. Defining such mixed learning objectives could be difficult for many multi-layer networks. In this work we used single-layer RBMs, for which it is straightforward to combine discriminative and generative learning objectives. A similar combined objective could be defined for deep Boltzmann machines – which are similar to RBMs but have multiple layers that are trained at the same time – by adding a label component to the top layer and using a combined learning

objective to update the weights in all layers of the model. This approach only works for models in which all layers can be trained at the same time using both learning objectives. In practice, deep Boltzmann machines are often initialized with layer-wise pre-training [46], and since this initialization influences the final solution, it may be important to include a discriminative objective in this first phase as well. A similar problem applies to deep belief networks, which consist of stacked RBMs that are also trained layer-by-layer [47]. In both approaches, including a temporary label component while training the lower layers might provide a solution. In convolutional neural networks, all layers are trained at the same time, but usually only using a discriminative objective. Unsupervised generative pre-training can give good results [48] by using a generative learning objective to initialize weights that are then refined with a discriminative learning objective, but this approach separates the generative and discriminative training. This may give worse results than training with a combined objective. Classification RBMs have the advantage that they can be trained with generative and discriminative objectives simultaneously.

Although we found that learned filters could outperform the predefined filter banks in our experiments, the predefined filter banks had one obvious advantage: they did not have to be learned. Learning the filters can take some time, depending on the implementation, the hardware and the number and size of the filters (in our tissue classification experiments, training one RBM with 16 filters of 8×8 pixels took approximately 4 days using two CPU cores). The runtime of the classification RBMs was not longer than that of the standard RBMs. Once the features have been learned, however, computing features and training and applying the classifiers does not require more time than with predefined filter banks.

2.7 Conclusion

We presented experiments with convolutional classification RBMs, which we trained with generative and discriminative learning objectives. Feature learning is usually done with a purely generative learning objective, which favors a representation that gives the most faithful description of the data but is not always the representation that is best for the goal of the system. This chapter showed how the standard generative learning objective of an RBM can be combined with a discriminative learning objective. In our experiments evalu-

ating the classification accuracy of random forests using RBM-learned features, we found that a mixture of discriminative and generative learning objectives often gave a better classification accuracy than generative or discriminative learning alone. The features learned with the mixed learning objective gave better results than several standard filter banks. Our results suggest that adding discriminative learning is most useful when learning smaller representations, with fewer filters or hidden nodes.

Acknowledgments

This research is financed by the Netherlands Organization for Scientific Research (NWO). Data for the lung tissue classification experiments was provided by the University Hospitals of Geneva [10]. Data for the airway centerline experiments was provided by the Danish Lung Cancer Screening Trial [23].

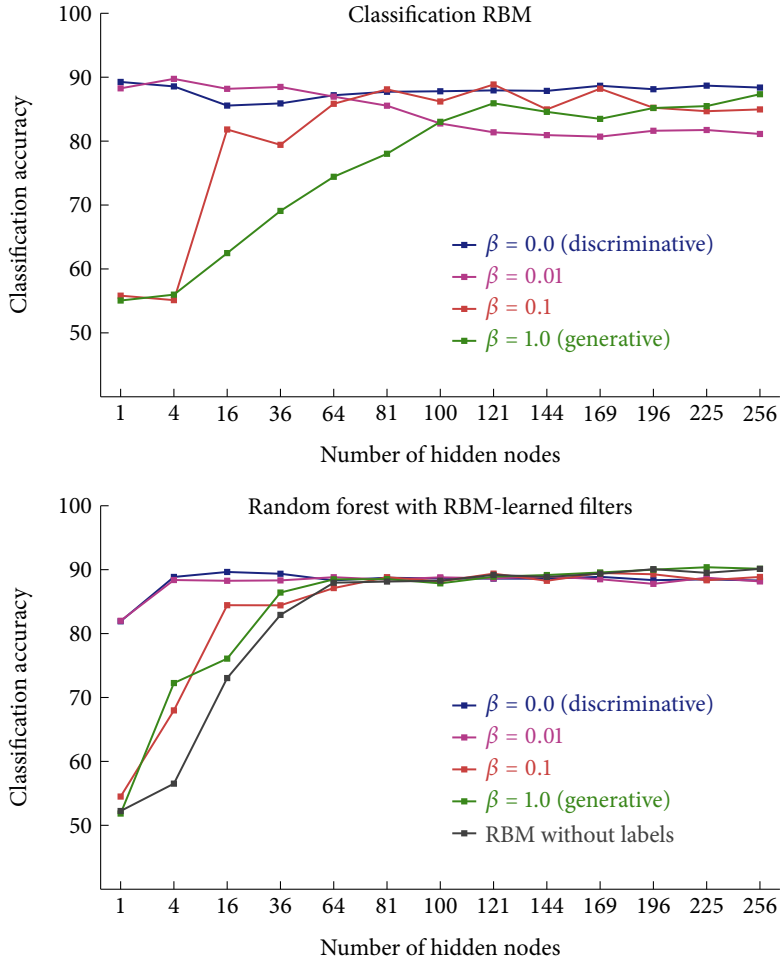


Figure 2.11: Classification accuracy on the airway dataset, showing the influence of the number of hidden nodes in the RBM representation on the classification accuracy, for different mixtures of discriminative and generative learning. The graph on the left shows the classification accuracy of the classification RBM. The graph on the right shows the classification accuracy of a random forest using the RBM-learned filters.

3

Why does synthesized data improve multi-sequence classification?

The classification and registration of incomplete multi-modal medical images, such as multi-sequence MRI with missing sequences, can sometimes be improved by replacing the missing modalities with synthetic data. This may seem counter-intuitive: synthetic data is derived from data that is already available, so it does not add new information. Why can it still improve performance? In this chapter we discuss possible explanations. If the synthesis model is more flexible than the classifier, the synthesis model can provide features that the classifier could not have extracted from the original data. In addition, using synthetic information to complete incomplete samples allows them to be used by a model that requires all modalities, increasing the effective size of the training set.

We present experiments with two classifiers, linear support vector machines (SVMs) and random forests, together with two synthesis methods that can replace missing data in an image classification problem: neural networks and restricted Boltzmann machines (RBMs). We used data from the BRATS 2013 brain tumor segmentation challenge, which includes multi-modal MRI scans with T₁, T₁ post-contrast, T₂ and FLAIR sequences. The linear SVMs appear to benefit from the complex transformations offered by the synthesis models, whereas the random forests mostly benefit from having more training data. Training on the hidden representation from the RBM brought the accuracy of the linear SVMs close to that of random forests.

Chapter based on

G. van Tulder and M. de Bruijne, “Why Does Synthesized Data Improve Multi-sequence Classification?” In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, N. Navab et al., Eds., Lecture Notes in Computer Science, Cham: Springer International Publishing, 2015, pp. 531–538. DOI: 10.1007/978-3-319-24553-9_65.

3.1 *Introduction*

Multi-sequence data can be very informative in medical imaging, but using it may cause some practical problems. Training a classifier on multi-modal data, for instance, generally requires that all modalities are available for all samples. If some modalities are missing, there is a range of methods for handling or imputing the missing values in standard statistical analysis [49]. Specifically for image analysis, there are synthesis methods that predict missing modalities. Some methods model the physical properties of the imaging process, e.g., to derive intrinsic tissue parameters from MRI scans [50] or to derive pseudo-CT from MRI in radiotherapy applications [51, 52]. But an explicit model of the imaging process is not even required, as image processing techniques can be sufficient: for example, pseudo-CT images have also been made with tissue segmentation [53, 54], with Gaussian mixture models [55] or by registering and combining CT images [56, 57].

Interestingly, data synthesis can not only generate images but also helps as an intermediate step. For example, Iglesias et al. [58] found that synthetic data improved the registration of multi-sequence brain MRI. Roy et al. [59] showed that synthetic sequences can improve segmentation consistency in datasets with multiple MRI contrasts. Li et al. [60] predicted PET patches from MRI data with convolutional neural networks, and found that including this synthetic PET data could improve classification of Alzheimer’s disease.

There is something paradoxical about these results: if the synthetic data is derived from the available data and does not add new information, how can it still improve the performance? We discuss three possible explanations. If the data synthesis is more flexible than the existing model, the synthetic data could add a useful transformation that makes the data easier to analyze. Data synthesis may also help to use the training data more efficiently, by allowing samples with different missing modalities to be combined into a single, large training set. Finally, synthesis methods that use unlabeled data, such as those discussed here, are an elegant way to add unsupervised learning to supervised models. However, most studies with synthetic data do not feature mixed training data or extra unlabeled examples, which suggests that the extra modeling power of the synthesis method could be important.

We present experiments comparing simple and complex classifiers trained with synthetic data on multi-sequence MRI data from the BRATS brain tumor

segmentation challenge [1]. We use neural networks and restricted Boltzmann machines (RBMs) to provide synthetic replacements for missing image sequences. These representation learning [3] methods aim to learn new, abstract representations from the data. We use these representations to train linear support vector machines (SVMs) and random forests. We compare the results of using data synthesis with those of simply replacing missing data with a constant value. The data synthesis models are non-linear, so we expect that they can improve the results of the linear SVM but have a smaller effect for the random forests.

3.2 Methods

Image Synthesis with Neural Networks. We use a neural network with three layers: an input layer with nodes v_i to represent the voxels from the 3D input patches, a hidden layer with nodes h_j , and a layer with nodes y_k representing the 3D patch to be predicted. In this feed-forward network the visible nodes v_i are connected with weights W_{ij} to the hidden nodes h_j , which are connected to the output nodes \hat{y}_k with weights U_{jk} . The parameters b_j and c_k are biases. The activation of the nodes given input \mathbf{v} is given by

$$h_j = \text{sigm}(\sum_i W_{ij}v_i + b_j) \quad \text{and} \quad \hat{y}_k = \sum_j U_{jk}h_j + c_k, \quad (3.1)$$

with $\text{sigm}(x) = \frac{1}{1+\exp(-x)}$. We use backpropagation to learn the weights that optimize the reconstruction error between the predicted $\hat{\mathbf{y}}$ and true values \mathbf{y} :

$$\text{err}(\mathbf{y}, \hat{\mathbf{y}}) = \sum_k |y_k - \hat{y}_k|. \quad (3.2)$$

Restricted Boltzmann Machines. A restricted Boltzmann machine (RBM) models the joint probability over a set of visible nodes \mathbf{v} and hidden nodes \mathbf{h} , with an undirected connection with weight W_{ij} between each visible node v_i and hidden node h_j . Each visible node has a bias b_i , each hidden node a bias c_j . We use noisy rectified linear units in the hidden layer and real-valued nodes with a Gaussian distribution for the visible nodes [33]. The weights and biases define the energy function

$$E(\mathbf{v}, \mathbf{h}) = \sum_i \frac{(v_i - b_i)^2}{2\sigma_i^2} - \sum_{i,j} \frac{v_i}{\sigma_i} W_{ij}h_j - \sum_j c_j h_j, \quad (3.3)$$

where σ_i is the standard deviation of the Gaussian noise of visible node i . The joint distribution of the input \mathbf{v} and hidden representation \mathbf{h} is defined as

$$P(\mathbf{v}, \mathbf{h}) = \frac{\exp(-E(\mathbf{v}, \mathbf{h}))}{Z}, \quad (3.4)$$

where Z is a normalization constant. The conditional probabilities for the hidden nodes given the visible nodes and vice versa are

$$P(h_j | \mathbf{v}) = \max(0, \sum_i W_{ij} v_i + c_j + \mathcal{N}(0, \text{sigm}(\sum_i W_{ij} v_i + c_j))), \quad (3.5)$$

$$P(v_i | \mathbf{h}) = \mathcal{N}(\sum_j W_{ij} h_j + b_i, \sigma_i), \quad (3.6)$$

$$\text{with } \text{sigm}(x) = \frac{1}{1 + \exp(-x)}. \quad (3.7)$$

We use stochastic gradient descent with persistent contrastive divergence [33, 61] to find weights \mathbf{W} and biases \mathbf{b} and \mathbf{c} that give a high probability to samples from the training distribution.

Although the energy $E(\mathbf{v}, \mathbf{h})$ can be calculated with Equation (3.3), the normalization constant Z prohibits computing the probability $P(\mathbf{v}, \mathbf{h})$ for non-trivial models. However, we can still sample from the distribution using Gibbs sampling and the conditional probabilities $P(h_j | \mathbf{v})$ and $P(v_i | \mathbf{h})$ (Equations (3.5) and (3.6)).

The standard RBM has one set of visible nodes. To model the patches for multiple sequences we use a separate set of visible nodes \mathbf{v}^s for each sequence s , connected to a shared set of hidden nodes \mathbf{h} . There are no direct connections between visible nodes, so the interactions between sequences are modeled through the hidden nodes. We train this RBM on training samples with the same patch in every sequence to learn the joint probability distribution of the four sequences.

Image Synthesis with RBMs. In theory we could calculate the probability of one sequence given the others, $P(\mathbf{v}^s | \mathbf{v} \setminus \mathbf{v}^s)$, to predict a missing sequence, but the normalization constant Z makes this impossible. We resort to Gibbs sampling to synthesize the missing sequence. We initialize the model with the available sequences and keep these values fixed. We set the visible nodes for the missing sequence to 0, the mean value for our normalized patches.

During Gibbs sampling we alternate sampling from the visible and hidden layers. We use the final values of the visible nodes for the missing sequence as the synthesized patch.

3.3 *Data and Implementation*

We used data of 30 patients from the BRATS 2013 brain tumor segmentation challenge [1] with four MRI sequences per patient: T1, T1 post-contrast (T1+c), T2 and FLAIR. The scans of each patient are rigidly registered to the T1+c scan, which has the highest resolution, and resampled to 1 mm isotropic resolution. The dataset includes brain masks and class labels for four tumor structures.

For each patient we extracted patches of $9 \times 9 \times 9$ voxels from the same location in each sequence. For feature learning we used 10 000 patches per scan, centered at random voxels in the brain mask. For classification we used the label data to create a balanced training set with approximately $\frac{1}{5}$ th of the samples for each class (four tissue classes and the non-tumor background).

We normalized the data twice. First, each scan was normalized to zero mean and unit variance to remove large differences between scans. After extracting patches we calculated the mean intensity, standard deviation and the intensity of the center voxel for each patch, since these features may help to discriminate tissue classes. Finally, we normalized each patch before training the neural networks and RBMs, since this helps to learn the local image structures.

We trained the neural network and RBM on unlabeled patches, implemented with the Theano library [39] for Python. The neural networks had one hidden layer of 600 binary nodes; the RBMs had 600 noisy rectified linear units in the hidden layer. Using more nodes or layers did not improve the performance. We used stochastic gradient descent with a decreasing learning rate for both models, with persistent contrastive divergence to estimate the updates of the RBM.

After training the models, we synthesized missing sequences from three known sequences, using Equation (3.1) for the neural network and Gibbs sampling (20 iterations) for the RBM. As a baseline method, we replaced missing sequences with all zeros, the mean value of the normalized patches.

We trained random forest and linear SVM classifiers from Scikit-learn [40] to classify the five tissue types. The feature vectors were composed of either the normalized intensity values of observed and synthesized patches, or the values

of the hidden layer of the RBM. We also included the intensity of the center voxel and the mean intensity and standard deviation of the patch intensities.

We repeated our experiments for five train/validation/test splits, each with 20 training scans, 5 scans to validate the model parameters and 5 test scans. For each split, we used the validation set to optimize the number of trees (up to 200) in the random forest, the L2 regularization of the SVM, and the hyperparameters of the neural networks and RBMs. We report the mean accuracy on the test sets.

3.4 *Experiments*

We present two classification scenarios. In the first, all samples are missing the same sequence. As a baseline we use the classification accuracy without data synthesis, measured on the full dataset and on datasets where we removed one sequence from the training and test data. Next, we look at data synthesis to complete the missing sequences. We trained classifiers on complete samples and tested on samples with one synthetic sequence. We also give the accuracy of classifiers trained on samples with a synthetic sequence, because the synthetic data might have a different distribution than the real data. Training and testing a classifier on data with different distributions might reduce its performance. Finally, we trained classifiers on the hidden representation from the RBM directly.

The second scenario uses a mixed training set, in which every sample is still missing one sequence, but where every quarter of the training set is missing a different sequence to simulate a combination of heterogeneous datasets. Without data synthesis, a separate classifier is needed for each subset of samples with the same three sequences. We use this as a baseline for the synthesis experiments. The RBM can be trained on the mixed training set. The neural networks have a practical problem: with no training samples with four sequences, we cannot train a network that predicts one sequence from the other three. Instead, we trained networks with one (MLP 1-1) or two (MLP 2-1) input sequences to predict one output sequence. Each option yields three networks to predict one sequence for a sample with three available sequences; we used the average prediction. We used the synthesis methods to complete the training set and compare with replacing the missing values with zeros, the mean value of the normalized patches.

3.5 *Results*

Table 3.1 shows the results of removing one of the MRI sequences from the test set. When training without synthesis, removing T1+c or FLAIR reduced the accuracy more than removing T1 or T2, suggesting that T1+c and FLAIR provide information that is not in T1 or T2. (The T1+c scans also had a higher original resolution.)

Training and testing with one synthetic sequence gave an accuracy similar to that of training on the dataset without the sequence. Replacing the synthetic data with zeros also gave similar results. This fits with our hypothesis that the synthetic data might not add new information. Adding synthetic data did not make the results much worse, which is useful if the synthetic data is used to combine data from multiple datasets. Using RBM synthesis was slightly better than using a neural network or replacing the sequence with zeros. Training on synthetic data instead of on real data slightly improved the accuracy, most likely because classifiers were confused by the different distributions of the real and synthetic data. Training on the hidden representation from the RBM increased the accuracy of the linear SVM and brought it closer to that of the random forest. This suggests that although the RBM does not add new information, it can still transform the data in a way that helps the linear SVM. The RBM representation did not improve the accuracy of the more complex random forests.

Table 3.2 shows the results of training with a mixed training set with partially incomplete data. Training on subsets of complete samples (sharing the same three sequences, $\frac{1}{4}$ th of the samples) gave a lower accuracy than training on the full set. Using the synthesis methods to complete the samples, we trained a classifier on all samples, which gave a higher accuracy than training on subsets. There was little difference between the two neural network approaches and replacing the missing values by zeros. The RBM synthesis gave a lower accuracy, possibly because synthesizing the missing training sequences made it harder to optimize the model. Training directly on the hidden representation from the RBM gave the highest accuracy for the linear SVM, as in the first experiment. The results with random forests were comparable to those of training on synthesized data.

	Missing sequence				
	Full set	T ₁	T ₁ +c	T ₂	FLAIR
Train and evaluate on voxel values, without synthesis					
	68.83 73.22	67.90 72.97	58.67 61.62	68.26 72.87	59.13 69.60
Train on complete samples, evaluate with synthesized data					
with zeros		67.32 72.61	54.26 59.77	67.17 72.08	58.10 65.03
by MLP		68.32 72.95	56.21 60.00	67.48 72.52	58.33 68.53
by RBM		68.42 73.06	55.34 60.33	67.35 72.38	59.66 67.57
Train and evaluate with synthesized data					
with zeros		68.47 73.36	57.88 61.75	67.90 72.73	59.94 69.38
by MLP		67.37 73.01	58.34 61.22	66.59 72.89	60.19 69.90
by RBM		69.25 73.24	60.53 61.47	68.17 72.55	62.30 69.88
Train and evaluate on values from the RBM hidden layer					
RBM	72.89 74.16	72.18 73.47	61.68 61.51	70.78 72.93	66.33 69.52

Table 3.1: Classification accuracy (linear SVM | random forest) for different synthesis methods, with test sets in which all samples are missing the same sequence. Results in bold are significantly different from the baseline results in the top row ($p < 0.05$).

	Missing sequence in evaluation				
	Full set	T ₁	T ₁ +c	T ₂	FLAIR
Train on subsets with complete samples (three sequences, $\frac{1}{4}$ th of the full set)					
		62.30 67.99	54.92 59.48	62.71 69.03	51.06 65.51
Train on the mixed training set, with missing sequences filled-in					
with zeros	66.85 70.86	63.64 69.90	58.21 63.70	62.90 70.55	54.03 67.63
by MLP 1-1	66.99 71.44	64.28 69.99	59.27 63.95	63.50 71.17	55.82 68.73
by MLP 2-1	65.42 71.22	65.03 69.76	59.15 64.01	63.88 71.21	55.84 68.38
by RBM	57.81 70.26	54.56 69.25	51.94 63.23	56.12 70.65	50.60 68.10
Train and evaluate on values from the RBM hidden layer (all samples)					
RBM	70.17 70.79	69.80 69.60	59.72 59.78	68.57 70.30	62.90 65.90

Table 3.2: Classification accuracy (linear SVM | random forest) with partially incomplete training data, in which every scan is missing a random sequence. Boldface indicates a significant difference with the baseline ($p < 0.05$). The results for the full test set are compared with the best performing baseline (missing T₂).

3.6 *Discussion and Conclusion*

Data synthesis can improve the classification accuracy of multi-modal image analysis by providing synthetic replacements for incomplete examples. We first explored the explanation that the synthesis models may offer data transformations that are useful to the classifier. In our experiments in which the same modality was missing for all samples, we found few significant improvements from using synthetic T1, T1+c or T2. We suspect that these modalities are too similar to produce useful transformations. Synthesized FLAIR did give a small improvement. Moreover, training on the RBM hidden layer significantly improved the accuracy for both classifiers and brought the SVMs close to the random forests. This suggests that the RBM extracts features that are new to the linear SVMs, but that could already be extracted by the random forests.

We found stronger improvements from using synthetic data in our second experiment. The synthesis methods made it possible to combine samples with different missing sequences in one training set. Using this larger training set increased the accuracy of both linear SVMs and random forests. We found similar results by replacing the missing values with zeros, the mean intensity after normalization. This suggests that at least part of the improvement in accuracy might be the result of having more training data.

In these applications the RBMs have a practical advantage over neural networks, because RBMs learn a joint probability distribution that can be used to predict any missing sequence. In contrast, neural networks are explicitly trained to predict one sequence given the others, so they need a separate network for each sequence. In our experiments the neural networks had a slightly lower reconstruction error, because the RBMs optimize a different learning objective.

Both neural networks and RBMs are trained with unlabeled data, a useful property that makes it easier to train them on large datasets. This can be an elegant way to use unlabeled data to improve a supervised classifier.

In conclusion: synthetic data might help classification because it allows better use of available training data, and because it offers new transformations of the data. This second contribution depends on the difference in complexity of the synthesis model and the classifier. A simpler classifier is more likely to benefit from the additional features that the synthesis model can extract from the data, even though the synthetic data does not contain extra information.

In contrast, more complex classifiers can extract more information from the original data and are less likely to benefit from synthetic data. Whether it is better to include the extra complexity in the classifier or in a synthesis model is up for discussion.

Acknowledgements

This research is financed by the Netherlands Organization for Scientific Research (NWO). Brain tumor image data was obtained from the NCI-MICCAI 2013 Challenge on Multimodal Brain Tumor Segmentation (originally at <https://martinos.org/rtim/miccai2013/>, currently at <https://www.smir.ch/BRATS/Start2013>).

4

Learning cross-modality representations from multi-modal images

Machine learning algorithms can have difficulties adapting to data from different sources, for example from different imaging modalities. We present and analyze three techniques for unsupervised cross-modality feature learning, using a shared autoencoder-like convolutional network that learns a common representation from multi-modal data. We investigate a form of feature normalization, a learning objective that minimizes cross-modality differences, and modality dropout, in which the network is trained with varying subsets of modalities. We measure the same-modality and cross-modality classification accuracies and explore whether the models learn modality-specific or shared features. This chapter presents experiments on two public datasets, with knee images from two MRI modalities, provided by the Osteoarthritis Initiative, and brain tumor segmentation on four MRI modalities from the BRATS challenge. All three approaches improved the cross-modality classification accuracy, with modality dropout and per-feature normalization giving the largest improvement. We observed that the networks tend to learn a combination of cross-modality and modality-specific features. Overall, a combination of all three methods produced the most cross-modality features and the highest cross-modality classification accuracy, while maintaining most of the same-modality accuracy.

Chapter based on

G. van Tulder and M. de Bruijne, “Learning Cross-Modality Representations From Multi-Modal Images,” *IEEE Transactions on Medical Imaging*, vol. 38, no. 2, pp. 638–648, Feb. 2019. DOI: 10.1109/TMI.2018.2868977.

4.1 Introduction

Many machine learning methods that work well on data that is similar to their training data might fail on data with different characteristics. This can lead to practical problems in medical image analysis, for example when existing models need to be applied to scans acquired with different imaging protocols or with different scanners. In these cases, transfer learning approaches can help to improve results, by allowing data from different sources to be used to train a single model that works for all sources. This chapter proposes one of these approaches, based on representation learning using convolutional neural networks (CNNs). We present and study several ways to encourage a CNN to learn a common feature representation from heterogeneous data, in order to obtain a source-independent representation that is similar for data from all sources. This common representation makes it possible to train a model on data from one source and apply it to data from another. We apply these methods in cross-modality experiments.

Neural networks for cross-modality learning, such as the model presented here, have been popular in computer vision for some years (starting with [62]) and have more recently also been applied to medical images (e.g., [63]). Similar approaches to transfer knowledge between modalities have also been used to learn from incomplete datasets with missing modalities (e.g., [64, 65]). In contrast with previous work learning a joint representation using a single transformation for all modalities (e.g., [66]), we propose cross-modality networks that learn a separate transformation for each modality. This allows the networks to model more complex transformations between modalities, such as intensity inversions, instead of merely learning modality-invariant features that are expressed in the same way in all modalities.

Cross-modality classification is a relatively unexplored topic in medical image analysis, but has received more attention in multimedia retrieval, most often in works on cross-modality classification of images and text (e.g., [67–72]). Feng et al. [70] present cross-modal retrieval experiments in cross-modal feature learning, using autoencoders and restricted Boltzmann machines to learn shared representations from images and text. They evaluate a learning objective similar to the similarity term discussed in this chapter, as well as a form of modality dropout. Srivastava and Salakhutdinov [71] use deep Boltzmann machines to learn joint representations for text and images, reporting

that multi-modal learning can improve results even if some modalities are not available at test time. Ngiam et al. [62] present cross-modality classification experiments with restricted Boltzmann machines and deep autoencoders, showing that speech classification can be improved by learning from video and audio. They train with a form of modality dropout to learn models that are robust to inputs with missing modalities. Vukotić et al. [72] present cross-modal deep networks based on deep autoencoders, aiming to learn a common hidden representation from text and images in a video hyperlinking task. In the medical domain, Moradi et al. [73] proposed a cross-modality neural network combining text and images for semi-automatic annotation of medical images, using a two-step approach that first extracts features from text and images and then learns a mapping between the two domains. In this chapter, we propose a single-step method to learn cross-domain representations from multi-modal medical images, and evaluate a number of additions to obtain representations that perform well in cross-modality classification.

Recent work using adversarial learning provides an alternative method for unsupervised domain adaptation, using an adversarial loss function. This can be done at the image level or at the feature representation level. Adversarial domain adaptation on an image level can be implemented with cycle-consistent generative adversarial networks (CycleGANs). For example, Zhang et al. [74] applied this to CT and MRI data, by training a CycleGAN to convert MRI data to CT and back. In this case, the discriminator network attempts to discriminate between CT derived from MRI data and real CT images. On a feature level, the adversarial loss can be implemented by a discriminator network that attempts to identify the source modality of a sample from its feature representation. For example, Kamnitsas et al. [75] applied this to an MRI and CT brain segmentation task, and describe how the adversarial loss helps to produce a feature representation that is more similar across modalities. Unlike the methods proposed in this chapter, the adversarial methods do not use corresponding image samples from both domains, but rely solely on the adversarial loss to learn the translation.

We present results of patch-wise cross-modality classification experiments on two multi-modal datasets: a knee cartilage segmentation dataset with two different MRI sequences, and a brain tumor segmentation dataset with four MRI sequences. Voxel classification approaches such as the deep convolutional

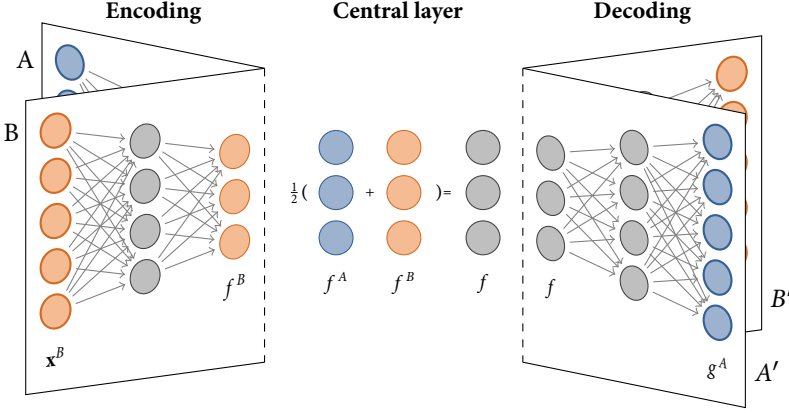


Figure 4.1: Schematic overview of the axial CNN for two modalities. For each modality m , the input \mathbf{x}^m is encoded into a representation $f^m(\mathbf{x}^m)$. The representations for all modalities are averaged into a mean representation $f(\mathbf{x})$, which is then used to compute reconstructions $g^m(f(\mathbf{x}))$ for all modalities. Each additional modality adds an extra input and output plane and is included in the average for the central layer.

networks used in this chapter have been used previously for both types of data. For example, knee cartilage segmentation has been approached with texture features (e.g., [76]) and deep neural networks [77]. Texture-based voxel classification also gave good results for the brain tumor segmentation problem (see [1] for an overview). In recent years, deep convolutional networks have also been applied to this problem (e.g., [78]).

For both datasets, we use unlabeled training data with multiple modalities per subject to train an axial CNN [63] that learns source-specific transformations that map data from each source to a single common representation. We evaluate this common representation in a transfer learning setting, training a classifier on labeled data from one source and applying it to data from another. We combine the basic cross-modality architecture with three techniques to further improve cross-modality feature learning: modality dropout [62, 65], a similarity term [63], and a normalization step. We analyze whether the models learn mostly shared features, mostly modality-specific features, or a combination of both.

In this chapter, we use an axial neural network architecture that is similar to the architecture that we used in our workshop paper [63], although we used a much simpler non-convolutional network for those experiments. The idea of using a separate network path for each input source also appears in work by Ngiam et al. [62] and Havaei et al. [65] on modality dropout, although the latter only applied it to the input side of a supervised classification network and not to reconstruction. In this chapter we combine all three methods, and provide an extensive evaluation and analysis of the feature representations learned with the different combinations.

This chapter is organized as follows. Section 4.2 outlines the basic model and the three techniques to improve cross-modality feature learning. Section 4.3 discusses the datasets. Section 4.4 gives an overview of the experiments, the results of which are presented in Section 4.5. Finally, Section 4.6 and Section 4.7 discuss the conclusions.

4.2 *Methods*

We investigate the axial convolutional neural network [63] (Figure 4.1) for cross-modality learning. This is an autoencoder-like model that learns a common representation for data from multiple modalities, which can then be used for cross-modality classification: training a classifier on data from one modality and applying it to data from another, using the shared representation as a common feature description for samples from both modalities. In this section, we describe the model and three extensions that can further improve the cross-modality similarity of the representations.

4.2.1 *Axial convolutional neural network*

We construct a multi-input autoencoder network (Figure 4.1) that has an input $\mathbf{x} = \{\mathbf{x}^1, \mathbf{x}^2, \dots, \mathbf{x}^M\}$ with corresponding input patches \mathbf{x}^m for each of the M modalities. For a modality m , given an input patch \mathbf{x}^m , the network uses a modality-specific encoding transformation f^m to compute the representation $f^m(\mathbf{x}^m)$. Because the model should produce the same representation for each of the modalities, we compute the mean representation $f(\mathbf{x}) = \frac{1}{M} \sum_{m=1}^M f^m(\mathbf{x}^m)$ and use this as the input for the modality-specific decoding transformations $g^m(f(\mathbf{x}))$.

The network is trained with an auto-encoder objective to minimize the sum of the reconstruction errors:

$$\mathcal{L}_{\text{recon}} = \sum_{m=1}^M |g^m(f(\mathbf{x})) - \mathbf{x}^m|. \quad (4.1)$$

The model is trained with paired input patches $\mathbf{x} = \{\mathbf{x}^1, \mathbf{x}^2, \dots, \mathbf{x}^M\}$. We assume that the images are registered and that there is a voxelwise correspondence between all patches \mathbf{x}^m for a given sample. Furthermore, although the network can handle incomplete training samples for which not all M modalities are available, it needs sufficient training pairs to learn the correspondences between all modalities.

We implement the encoding and decoding transformations as convolutional networks (Figure 4.2) with a sequence of convolution and batch normalization layers. The encoding part of the network uses strided, valid convolutions to avoid border effects in the central layer. The decoding part is the inverse of the encoding part, using transposed convolutions to reconstruct the original input size. All inner layers use leaky rectified linear units; the reconstruction layer is linear to allow it to reproduce the full range of input values.

Taking the mean representation over all modalities encodes our goal of learning a common representation across modalities in the structure of the network: ideally, we want the representation $f^m(\mathbf{x}^m) \approx f(\mathbf{x})$ to be the same for all modalities m . Using the average representation instead of a single shared layer makes it possible to train and test with incomplete data for which not all modalities are available: by dividing the sum by the correct number of modalities, the scale of the combined feature values becomes independent of the number of input modalities.

Averaging the representations over all modalities is not sufficient to learn cross-modality representations, because it still allows the network to learn modality-specific features. If the network is always trained with complete training samples, for which all modalities are always available, it might allocate a different part of the feature representation to each modality. This would produce a single feature vector that can be used to reconstruct all modalities, but it would not produce a true cross-modality representation, because it is still dependent on all input modalities. To obtain a true cross-modality representation, we need to change how the model is trained. The remainder of this section presents three techniques to do this.

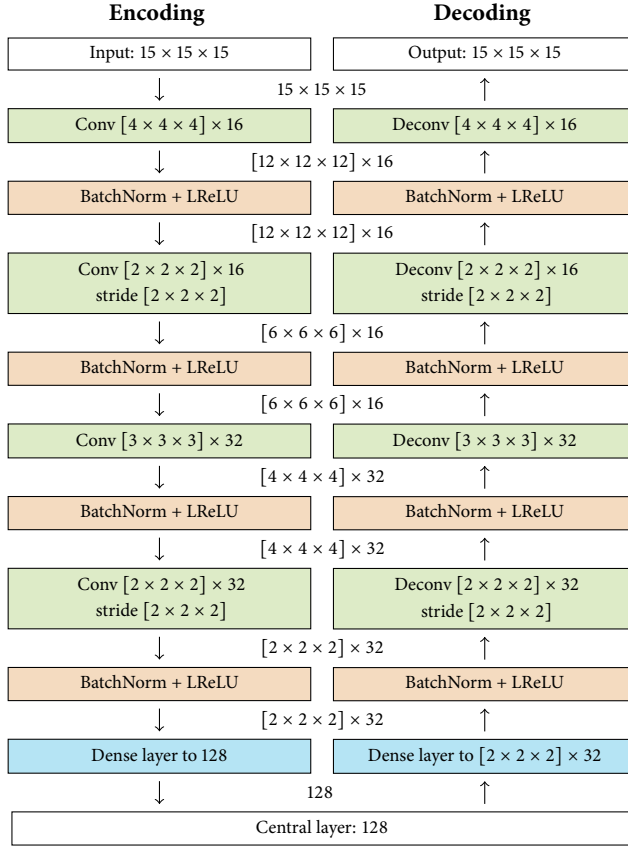


Figure 4.2: The structure used for the encoding and decoding parts of the network, with the size of intermediate representations shown between the blocks.

4.2.2 Modality dropout

The first approach (used, for example, in [62] and [65]) modifies the training procedure. In the default training procedure, the network is never explicitly forced to learn to reconstruct one modality from another, because all modalities are always available for all training samples. If the representation is sufficiently large, the network might learn to use a separate part of the representation for each modality. Modality dropout prevents this by disabling modalities at random during training, computing the mean representation

from a random subset of modalities while still optimizing the reconstructions for all modalities. For a model with M modalities, we select a random subset of 1 to M input modalities in each update step. We generate a random subset each time a sample is included in a minibatch: the modalities can be different each time a sample is used, and each minibatch can contain multiple modality combinations (see Figure 4.3). Using incomplete inputs for training means that the network can no longer rely on the original modality for its reconstruction, but is forced to learn cross-modality reconstructions and representations.

4.2.3 Similarity term in the learning objective

The second approach explicitly adds cross-modality learning to the learning objective, similar to the approach in [63]. We compute the difference between the modality-specific representations $f^m(\mathbf{x}^m)$ and the mean representation $f(\mathbf{x})$. We add this to the original learning objective (Equation (4.1)) with a tunable weight $\alpha \in [0, 1]$:

$$\mathcal{L}_{\text{sim}} = \sum_{m=1}^M |f^m(\mathbf{x}^m) - f(\mathbf{x})|, \quad (4.2)$$

$$\mathcal{L}_{\text{combined}} = (1 - \alpha) \mathcal{L}_{\text{recon}} + \alpha \mathcal{L}_{\text{sim}}. \quad (4.3)$$

Choosing α large enough will cause the network to reduce the differences between the representations for each modality. However, it is equally important not to set α too high: choosing a value very close to 1 will disregard the reconstruction error and can produce representations that may be very similar, but are also very uninformative.

The similarity term as defined in Equation (4.2) can have another undesired effect: it can be trivially minimized by reducing the absolute feature values, so it might lead to very small or completely disabled feature values. This reduces the loss but does nothing to improve the cross-modality similarity. To prevent this trivial optimization, we normalize all feature vectors to zero mean and unit standard deviation.

4.2.4 Per-feature normalization

Global normalization across all features still allows cross-modality differences between individual features: they can be active for one modality and disabled in another. Our third approach therefore normalizes each individual feature to

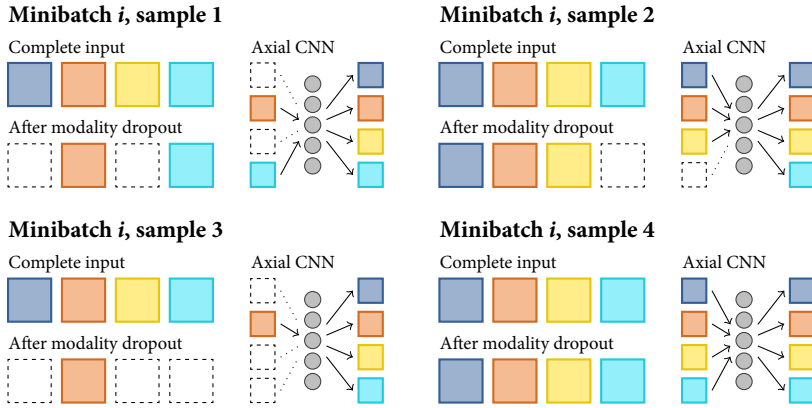


Figure 4.3: Schematic illustration of modality dropout with four modalities. We select a random subset of 1 to 4 modalities for each sample in each minibatch. The network is only given the selected input modalities to compute the central representation, but we ask it to reconstruct all modalities and optimize the full reconstruction error. The subsets are generated independently for each sample, so a minibatch can contain multiple modality combinations. We generate a new random subset each time a sample is used for training.

zero mean and unit standard deviation, before averaging the modality-specific representations to get the mean representation. This per-feature normalization helps to remove a large part of the differences between modalities, and allows the network to focus on more meaningful ways to improve the representation similarity. We implement this normalization using a standard batch normalization procedure [79] to learn estimates of standard deviation and mean for each feature, per-modality, and to normalize the feature to zero-mean and unit standard deviation. The batch normalization formula provides scaling and shift parameters (β and γ in [79]), which allow the model to scale and shift the features away from a zero mean and unit standard deviation. In our case, doing so could reintroduce differences between modalities. We fix the parameters to $\beta = 1$ and $\gamma = 0$ to prevent this. (Note that we only make this change for this specific per-feature normalization step. We use the standard batch normalization formula for the batch normalization layers in the network, as shown in Figure 4.1.)

4.3 Data

We performed experiments for two tasks: knee cartilage segmentation and brain tumor segmentation. In both cases, we evaluate our methods on a patch-based classification task in which we train classifiers to label the center voxel of a $15 \times 15 \times 15$ voxel neighborhood. We take paired patches from all modalities of a subject, such that the patch in each modality represents the same physical location.

For the experiments on knee segmentation, we used knee MRI images from the Osteoarthritis Initiative (OAI) [2], with the manual cartilage and meniscus segmentations from the iMorphics subset. For each subject, the dataset provides normal (N) and fat-suppressed (FS) MRI scans (Figure 4.4a), made shortly after each other, which disagree on the intensity of some tissue types. The normal scans also have a somewhat better resolution. The dataset provides registered and resampled scans for each subject, to a common voxel spacing of $0.36 \times 0.36 \times 0.7$ mm. We extracted paired patches of $15 \times 15 \times 15$ voxels, using the annotation of the center voxel in the normal scan as the patch label to define a three-class classification problem (cartilage, meniscus and background). The background voxels were sampled from a background mask, which we constructed by dilating the cartilage and meniscus segmentations with 10 voxels. We used N-FS pairs from baseline and 12-month follow-up sessions from 88 subjects, excluding two pairs that were not properly aligned. For each of the 172 pairs we extracted a randomly sampled, balanced set of 5000 cartilage, 5000 meniscus and 5000 background patches. Before extracting the patches, we normalized each scan to have a zero mean and unit standard deviation in the background and foreground voxels.

Our second dataset uses data from the BRATS brain tumor segmentation challenge [1], which provides T1, contrast-enhanced T1 (T1+c), T2 and FLAIR scans for each subject (Figure 4.4b). The challenge dataset (BRATS 2015) provides manual segmentations of four tumor components and a brain mask for each subject. The images and segmentations for each subject have been registered to the contrast-enhanced T1 scan and resampled to a $1 \times 1 \times 1$ mm voxel size. For each subject, we extract patches of $15 \times 15 \times 15$ voxels at the same position in each modality, and use the label of the center voxel as the label of this sample. Because some of the tumor components are only visible on some of the modalities and we evaluate single-modality cross-modality classification,

we merged the four tumor components into a single class to formulate a two-class classification problem (tumor vs. non-tumor brain tissue). The dataset contains scans of 220 subjects, for each of which we extracted a balanced set of 5000 foreground and 5000 background patches. Before extracting the patches, we normalized each scan to have a zero mean and unit standard deviation for the voxels inside the brain mask.

4.4 *Experiments*

We present a comparison of all combinations of the three techniques: modality dropout, per-feature normalization and a range of weights α for the similarity term. For each combination, we trained axial neural networks to learn a common feature representation. We then used the resulting networks to compute a feature vector for each modality.

To evaluate the suitability of the common representation for classification, we trained random forest classifiers on the features extracted by each axial neural network. We distinguish two scenarios: same-modality and cross-modality classification. For same-modality classification, we trained the classifier on the features obtained from one modality and evaluate it on a feature vector obtained from the same modality. For cross-modality classification, we trained the classifier using the features derived from a different modality from the one used in testing.

As part of our analysis, we investigate to what extent the models learn modality-specific or shared features. We do this by training classifiers with only a subset of features, ranked by the normalized cross-modality correlation (following the suggestion in [80]). We start with the feature that has the most similar values across modalities and gradually add more, training a new random forest for each subset.

Our networks were implemented using Keras [81] and Theano [82]. We used stochastic gradient descent for 100 epochs on the OAI dataset and 50 epochs on the BRATS dataset, which was sufficient for the networks to converge to a stable state. The minibatch size was 64 patches, the learning rate was 0.3 and the learning rate decay was 0.000002. We used Scikit-learn [40] with the default settings to train random forest classifiers with 30 trees.

We compare the results of our axial CNNs with those of two baseline methods. Both baselines use the same layer architecture as our axial networks

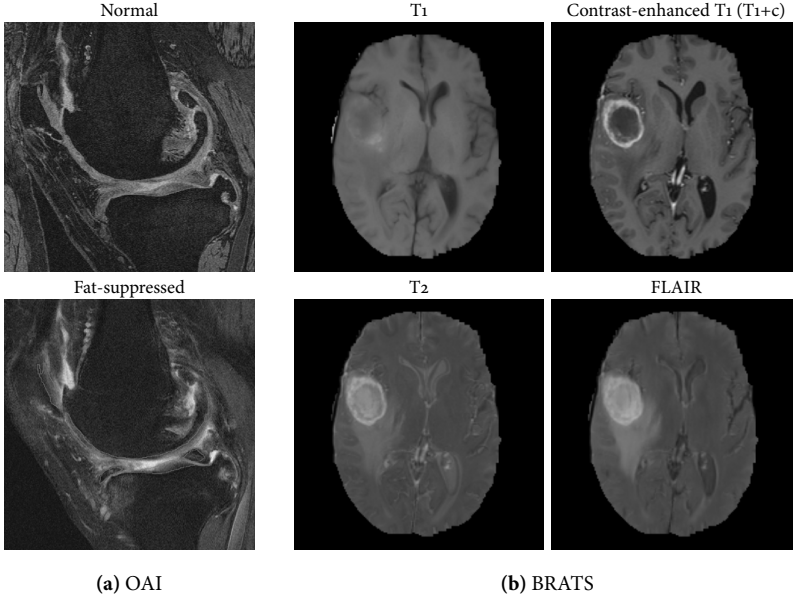


Figure 4.4: Example scans from OAI and BRATS, showing the two knee MRI modalities and four brain MRI modalities.

(Figure 4.2), but instead of learning multiple modality-specific transformations, the baseline methods learn only a single transformation that is shared by all modalities. In this way, they resemble normal autoencoders that encode and decode a single input patch and optimize its reconstruction error.

The two baselines use different training data to learn a common representation. The first baseline method is trained to reconstruct the training modality from itself, which produces a transformation that we also apply to the testing modality. For example, in a cross-modality classification experiment with modalities A and B, the first baseline method learns its representation only from patches of modality A, but the same representation is used to compute the features for the patches from modality B at test time. The second baseline learns the transformation from all modalities combined. In the example with A and B, this baseline would learn its representation from a mixture of patches from A and patches from B, without knowing which modality is represented in each patch.

We report results obtained in five-fold cross-validation. For each dataset, we divided the paired scans in five random subsets of approximately equal size, making sure that all scans of a subject were kept in the same subset. Using each subset in turn for testing, we first trained the axial neural networks on the remaining four subsets and used these networks to compute features for the training and test samples. For each subset, we trained the random forest classifiers on data from the training set and evaluate it on the test set. We report the mean accuracy over all five folds.

We used a slightly modified procedure for the experiments with subsets of most-correlated features, since these cross-modality correlations need to be computed on data that was not used to learn the representation. For these experiments, we introduced a second, two-fold cross-validation step to compute the results: we split each test subset in two halves, ensuring that all data from the same subject is in the same half, and in turn use one half to select the features and the other half to evaluate the classifier. We report the mean results over all 5×2 subsets, covering all samples in the dataset. If all features are selected, this is equivalent to the normal five-fold cross-validation.

4.5 *Results*

Section 4.5.1 presents the same-modality and cross-modality classification accuracy for the various models. This provides an overview of the performance of the proposed methods and that of the baseline methods. We present the results for both datasets, averaged over all five cross-validation folds.

Section 4.5.2 looks at the feature representations learned by each model, showing the standard deviation, cross-modality correlation, and mutual information scores of the individual features. This provides an insight into how modality dropout, per-feature normalization and the similarity term affect the feature learning process. Since each network is initialized randomly, it is not possible to average the measurements for individual features over multiple cross-validation folds. We show the plots for one fold on the OAI dataset, but found similar results for the other OAI folds and on the BRATS dataset.

Finally, Section 4.5.3 tries to identify whether models learn mostly shared or mostly modality-specific features. We show the classification accuracy obtained using subsets of features with the highest cross-modality correlation. This section shows the results for the OAI dataset, averaged over all five folds.

4.5.1 *Same-modality and cross-modality classification accuracy*

Table 4.1 shows the classification accuracy for each combination of methods, measured on both datasets, as well as the performance of the baseline methods on the same data. The table shows the average results over all modality pairs: the exact performance depends on which modalities are combined, because some modalities have more in common than others. However, the general pattern and the ordering of the methods were similar for all modality pairs.

The results show that the axial neural network with the additions discussed in this chapter can provide much better cross-modality results than the baseline methods that do not take cross-modality differences into account. On both datasets, the baseline methods achieve a much lower accuracy in cross-modality classification than in same-modality classification. The axial neural network also shows a drop in performance going from same-modality to cross-modality classification, but this drop is much smaller. On the knee dataset, the best-performing axial neural network obtains a cross-modality accuracy that is very close to its same-modality accuracy. On the brain tumor dataset, the performance drop is larger, but the axial neural network still performs much better on cross-modality classification than the baseline methods.

Table 4.1 shows the results for axial networks with all combinations of the three techniques. The best cross-modality accuracy was obtained with a combination of modality dropout, per-feature normalization and the similarity term. Removing the similarity term from this combination of methods decreased the cross-modality performance only a little, suggesting that modality dropout and per-feature normalization are the most important.

Comparing individual techniques over all different combinations, both modality dropout and per-feature normalization consistently provide an improvement of the classification accuracy. The contribution of the similarity term is less clear: it can give an important improvement if either modality dropout or per-feature normalization is missing, but if both are present the additional improvement of the similarity term is small. However, while the improvement from adding the similarity term might be large or small, it is usually positive: adding the similarity term with an appropriate weight never lead to a large decrease in same-modality or cross-modality performance.

To illustrate the reconstruction part of the network, Figure 4.5 shows some of the reconstructions produced by the best-performing network for the OAI

		Same-modality classification					Cross-modality classification				
OAI knee dataset		Weight of the similarity term α					Weight of the similarity term α				
<i>Axial neural network</i>		0.0	0.1	0.2	0.5		0.0	0.1	0.2	0.5	
No modality dropout, no per-feature normalization		80.0	79.1	79.2	78.9		33.5	52.5	53.7	57.0	
No modality dropout, with per-feature normalization		80.4	79.7	79.3	78.6		43.6	65.2	63.1	62.2	
Modality dropout, no per-feature normalization		81.8	81.1	80.7	80.3		43.4	63.2	67.1	70.7	
Modality dropout and per-feature normalization		81.6	81.5	81.1	80.7		77.0	78.7	78.7	78.2	
<i>Baseline network</i>											
Features from all modalities		79.6					70.0				
Features from the training modality only		79.4					69.0				
BRATS brain tumor dataset		Weight of the similarity term α					Weight of the similarity term α				
<i>Axial neural network</i>		0.0	0.1	0.2	0.5		0.0	0.1	0.2	0.5	
No modality dropout, no per-feature normalization		73.0	71.9	71.5	69.8		50.2	51.8	51.7	51.4	
No modality dropout, with per-feature normalization		72.6	73.4	73.9	72.9		52.2	55.9	57.5	60.0	
Modality dropout, no per-feature normalization		77.5	76.9	76.8	76.5		51.7	55.3	55.9	57.4	
Modality dropout and per-feature normalization		77.5	77.5	77.2	76.8		65.8	66.8	67.0	67.9	
<i>Baseline network</i>											
Features from all modalities		69.4					54.9				
Features from the training modality only		70.0					55.1				

Table 4.1: Classification accuracy of random forests trained with features learned using the axial neural network or the baseline models, comparing same-modality and cross-modality classification. A combination of modality dropout, per-feature normalization and the similarity term gives the best cross-modality classification performance. The table shows the mean classification accuracy over five cross-validation folds and over all modality combinations (normal and fat-suppressed for OAI, T1/T1+c/T2/FLAIR for BRATS).

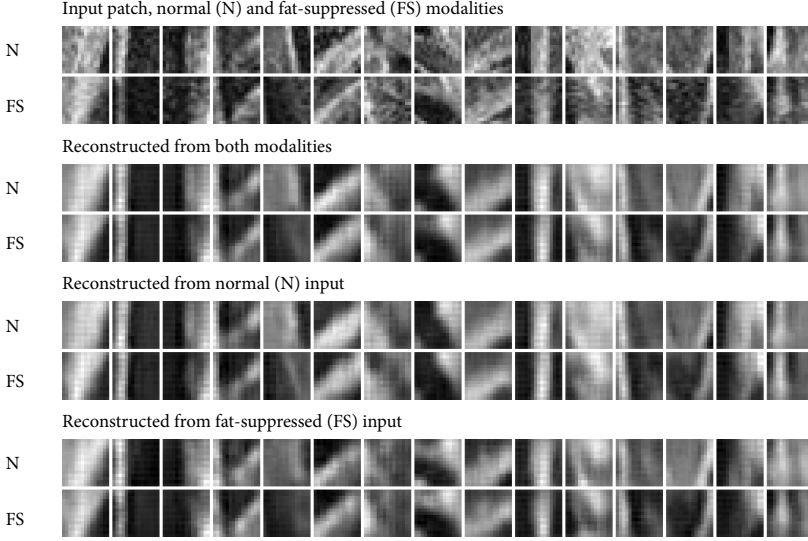


Figure 4.5: Original input and reconstructions for 15 patches from the OAI dataset, showing the center slice from each 3D patch. The reconstructions are generated by an axial neural network trained with modality dropout, per-feature normalization and a weight $\alpha = 0.1$ for the similarity term. The first reconstruction is generated from the central representation computed from both input modalities. The second and third reconstructions are computed using the central representation from one modality only.

dataset. These reconstructions are not used for classification, which is based only on the central feature representation, but it is still useful to see that the network is able to reconstruct the main structures in the image and can also reproduce some of the inter-modality differences.

4.5.2 Feature characteristics

The second part of our investigation considers the information content of individual features. For each feature in each modality, Figure 4.6 shows the mutual information score between the feature value and the class label, the standard deviation, and the normalized cross-modality correlation for each feature. The features are sorted by mutual information in the first modality: from the most informative feature (left) to the least informative (right).

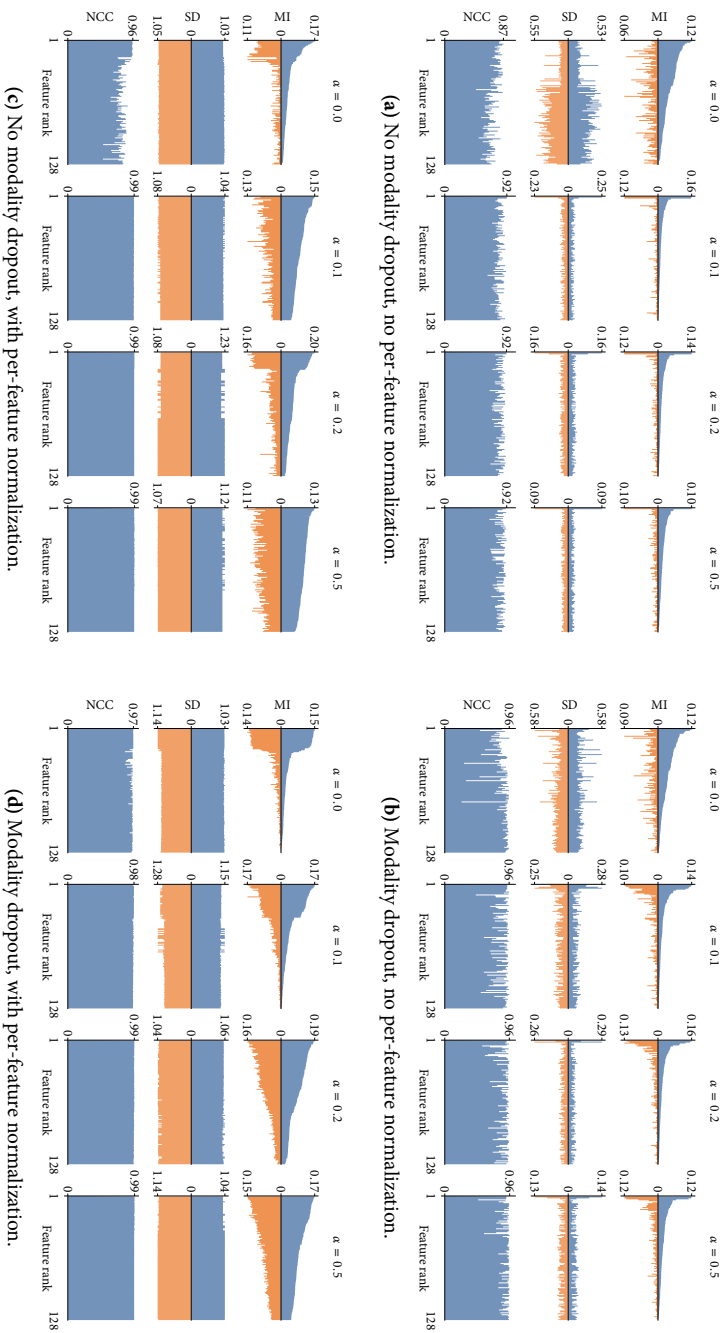


Figure 4.6: Characteristics of the 128 features learned for one fold of the OAI knee data, for different network configurations. The features are sorted by decreasing mutual information scores. First row: mutual information (MI) between the feature values and the class label, when feature values are computed from normal (up/blue) or fat-suppressed (down/orange) scans. Second row: standard deviation (SD) of each feature. Third row: normalized cross-correlation (NCC) between the values computed from both sources. Without modality dropout or per-feature normalization (a) there is a large difference in values between modalities. Combining all three methods (d) produces features that are much more similar, and might be more useful for cross-modality classification.

We interpret these plots by comparing the values of each feature in the two modalities: a cross-modality feature will have a similar meaning in both modalities, and will show similar values in these plots. Conversely, if the plots show a large difference between the values for both modalities, the feature is unlikely to be useful for cross-modality classification.

The most basic model, without modality dropout, per-feature normalization or similarity term, produces features that have very different standard deviations and mutual information scores in each modality (Figure 4.6a). This suggests that this basic model learns some modality-specific features that are informative for one modality, but not for the other.

The complete model with modality dropout, per-feature normalization and the similarity term learns features that are much more similar across domains. The plots for the combination of all methods (Figure 4.6d) show very similar values for the features in both modalities. This suggests that this model learns many cross-modality features, which is consistent with the good performance of this model observed in Section 4.5.1.

The plots of the standard deviations in Figure 4.6 also provide more insight into the interaction between the similarity term and per-feature normalization. Because the similarity term attempts to reduce the difference between feature values for different modalities, it encourages the model to reduce the absolute feature values. This is visible in the plots of the standard deviation, which show that the similarity term reduces the standard deviation of the features. This reduction does not necessarily improve cross-modality correlation, but it does decrease the similarity term of the learning objective. Applying per-feature normalization prevents this problem: the improved normalization brought the standard deviation reasonably close to 1 for all features.

4.5.3 *Classification accuracy for feature subsets*

In the final part of our investigation, we look at the classification accuracy obtained using subsets of features, sorted by decreasing normalized cross-modality correlation. Figure 4.7 shows the cross-modality correlation of individual features, sorted in decreasing order, for the various models. Figure 4.8 shows the classification accuracy obtained using subsets of features with the highest cross-modality correlation. We show the results for the knee dataset only, but the results for the brain tumor dataset show similar patterns.

Figure 4.7 shows how the three techniques affect the cross-correlation of the features. For the basic model without modality dropout, without per-feature normalization and with a zero weight for the similarity term (Figure 4.7a), the feature representation contains a combination of features with a reasonably high cross-modality correlation (0.9), as well as features that are less correlated across modalities (0.6). The optimal model combines modality dropout, per-feature normalization and a non-zero weight for the similarity term (Figure 4.7d), producing a feature representation in which all features have a high cross-modality correlation (values close to 1 for all features). The results for other models in Figure 4.7 show that all three techniques individually can improve the cross-modality correlation of the feature representation.

Figure 4.8 shows the same-modality and cross-modality classification accuracy for subsets of features with the highest cross-modality correlation: from only the most correlated feature on the left, to all features on the right. For same-modality classification (Figure 4.8a–d, left column), the accuracy for most methods increases monotonically with the number of features. Adding more features improves the results, although the improvement becomes fairly small after a sufficient number of features have been added.

For cross-modality classification (Figure 4.8a–d, right column), the accuracy does not increase monotonically, but first increases and then decreases again as features with a lower correlation are added. The low cross-modality correlation indicates that these features have a different meaning in each modality, which will confuse a cross-modality classifier. However, the proposed techniques can alleviate this problem. For the combination of modality dropout and per-feature normalization (Figure 4.8d), the cross-modality classification accuracy increases monotonically with the number of features. Including the similarity term leads to an earlier peak in the classification accuracy. This is consistent with the high cross-modality correlations of all features (Figure 4.7d), which indicates that this combination of methods learns mostly cross-modality features. For the other models, there is a larger range of cross-modality correlations (e.g., Figure 4.7a), which together with the decrease in accuracy suggests that these models learn a mixture of modality-specific and shared features.

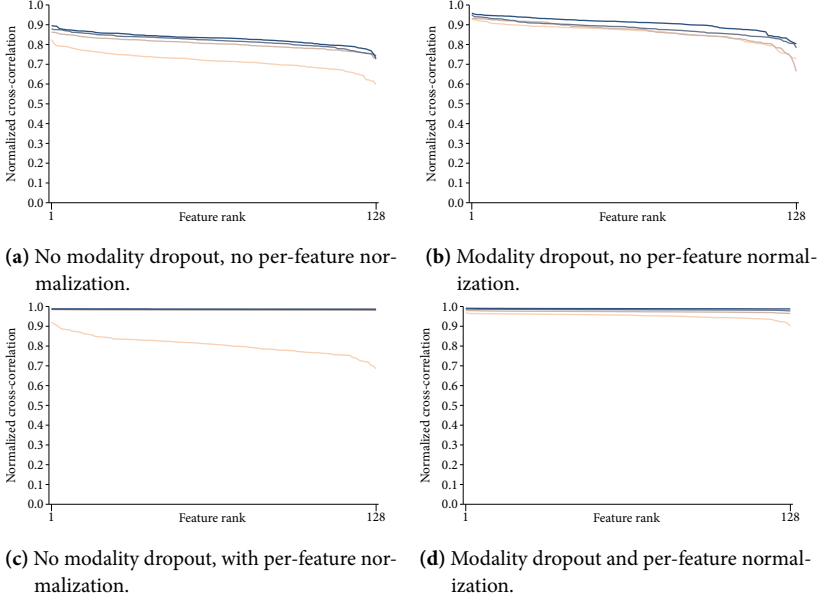


Figure 4.7: Normalized cross-correlation (NCC) of features on the OAI knee dataset, sorted from high to low correlation and averaged over five folds. The correlation is computed between corresponding features from both modalities. Combining modality dropout and per-feature normalization produced the most similar features.

4.6 Discussion

We evaluated three strategies to improve cross-modality feature learning in an axial neural network: modality dropout, per-feature normalization, and a similarity term. The best results were obtained using a combination of all three methods (Table 4.1). For both of our datasets, the features learned using this combination of techniques resulted in the best cross-modality classification accuracy, without affecting the same-modality classification accuracy too much. The cross-modality classification accuracy obtained using this combination of methods was higher than that obtained with the baseline method, a similar feature-learning model that used the same transformation for all modalities.

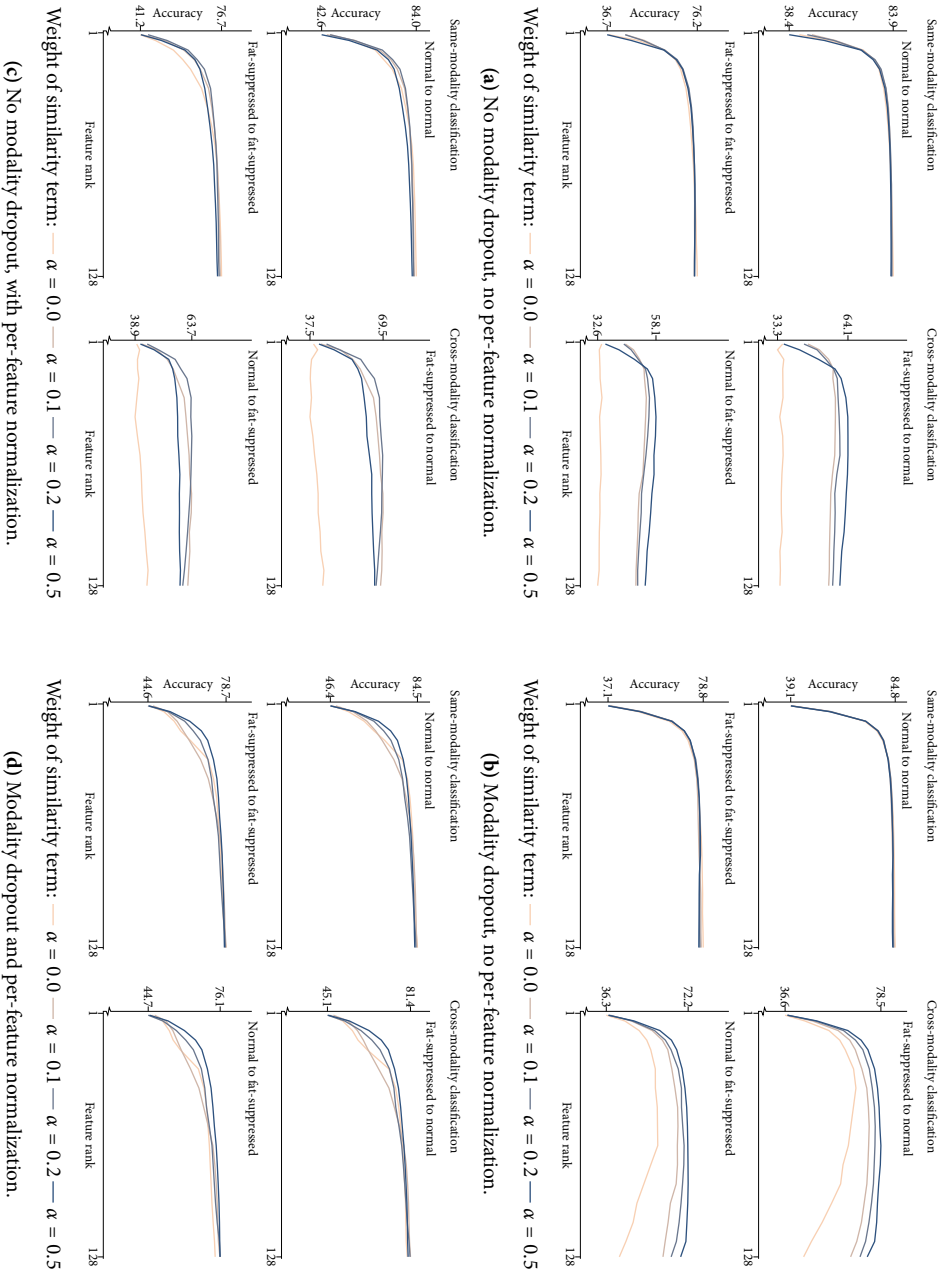


Figure 4.8: Classification accuracy on the knee dataset, for models trained with and without modality dropout and per-feature normalization. The horizontal axes indicate the number of selected features, starting with the features with the highest cross-correlation. All plots use the same scale; vertical tick marks indicate the minimum and maximum accuracy in each plot.

4.6.1 *Comparing the three techniques*

Modality dropout improved the accuracy of the axial neural network in both same-modality and cross-modality classification experiments (Table 4.1), perhaps because it explicitly trains the model to work well in both scenarios. Modality dropout forces the model to reconstruct the target modality from itself, which is useful for same-modality classification. It also forces the model to reconstruct the target modality from another modality, which helps the cross-modality case because it forces the network to learn representations that encode sufficient cross-modality information and prevents it from depending too much on one modality.

The second important factor was per-feature normalization (Table 4.1). Forcing the features to have a similar mean and standard deviation in all modalities turns out to be an effective way of minimizing cross-modality representation differences (Figure 4.6). It prevents the network from learning features that are used for one modality but not for another, and it simplifies the optimization by removing part of the cross-modality differences. The features learned with per-feature normalization also had higher mutual information scores, suggesting that they contained more discriminative information.

Adding a similarity term in the objective function had a positive influence on the cross-modality classification accuracy, but the strength of this influence depended on whether it was combined with the other techniques (Table 4.1). For the combination of modality dropout and per-feature normalization, the additional effect of the similarity term was fairly small: the results of this combination were only slightly better if the similarity term was included. This was different for all other combinations: there, increasing the weight of the similarity term produced more similar features and a better cross-modality classification accuracy. This suggests that the combination of modality dropout and per-feature normalization is powerful enough to remove most of the need for the extra similarity term, but that the term can still have a positive effect in other cases. It is important, however, to limit the weight of the similarity term: setting it too close to 1 can cause the model to learn trivial, non-informative and non-discriminative features [63].

Because the similarity term tries to minimize the absolute difference between feature representations, it also tends to reduce the absolute feature values. This is visible in Figure 4.6: the features learned with the similarity term have

lower standard deviations than the features learned without the term. Per-feature normalization counters this side-effect and stabilizes the feature values, improving the accuracy in the process.

While the three methods greatly improved the cross-modality performance, they also maintained most of the original same-modality performance (Table 4.1). This is a useful property if the same representation is used for both same-modality and cross-modality classification.

4.6.2 *Modality-specific vs. shared features*

One of the hypotheses behind our experiments was that the models might learn a combination of modality-specific and shared features. Shared features are good for cross-modality classification, but the models might still learn modality-specific features to preserve crucial modality-specific information. To investigate this further, we tried to separate shared and modality-specific features by sorting the features based on the cross-modality correlation (Figure 4.7) and training on subsets of highly correlated features. Our approach produced different results for each of the methods (Figure 4.8). For the best combination of modality dropout, per-feature normalization and the similarity term, we found that almost all features had a very strong cross-modality correlation, and that the classification performance improved monotonically with the number of features. This suggests that this combination of methods learned mostly cross-modality features. The other methods produced both highly correlated features and features that had a lower cross-modality correlation. In these cases, although the same-modality accuracy increased as we added more features, we obtained the best cross-modality accuracy by training on a smaller subset of highly correlated features. This suggests that these representations contained not only shared but also modality-specific features, which help same-modality classification but can harm the cross-modality case.

Although shared representations may be best for cross-modality classification, preserving modality-specific information is important for the same-modality performance. This is somewhat reflected by the results of our baseline methods (Table 4.1), which show that features learned for one specific modality give a slightly better same-modality accuracy. In applications where the same-modality performance is as important as the cross-modality performance, it may be useful to give the model a way to preserve modality-specific features

without including them in the shared representation. One way to do this could be to reserve a separate, modality-specific part of the representation that is only used for a single modality.

4.6.3 *Data requirements*

The approach discussed in this chapter makes some assumptions about the data and the problem to which the methods are applied. Firstly, the approach assumes that data is available for all modalities and that at least some of this (unlabeled) data is registered with a voxelwise correspondence. The axial neural network learns its feature representation from corresponding patches, which represent the same physical area in each modality. For models with more than two modalities, it is not strictly necessary to have all modalities available for all subjects: as the modality dropout method shows, it is possible to train with patches for which only a subset of modalities is available.

Secondly, learning a shared representation for multiple modalities assumes that the modalities have something in common. Because our model learns a separate transformation for each modality, it can handle large differences between modalities. However, the shared representation can only preserve and transfer information that is available in all modalities: if a modality provides information that is not visible in the other modalities, this information can not be used in cross-modality classification. The performance of the proposed methods depends therefore on the type of problem and on the differences between the modalities. If the modalities are very different and the modality-specific information has important discriminative value, removing it from the shared representation may reduce the same-modality classification performance.

In our experiments, the modalities in the knee dataset have more in common than the modalities in the brain tumor dataset. The knee images have a different resolution and have different intensities for some of the structures, but the image structures that are important for classification are recognizable in both images. As a result, the cross-modality classification performance on this dataset comes fairly close to that in the same-modality case. This suggests that transfer learning could be successful in this scenario. In the brain tumor dataset, the four modalities have larger differences, and some tumor structures are clearly visible in some images but not in others (Figure 4.4b). In our cross-modality classification experiments, this meant that the cross-

modality classification accuracy was noticeably lower than the same-modality accuracy. The performance differed per modality: in our classification task, T2 and FLAIR gave much better results than T1 and contrast-enhanced T1. This preference for T2 and FLAIR is most likely an artifact of how we grouped the tumor components into a single class, and would be different when classifying other components (for example, contrast-enhanced T1 would be important for identifying the necrotic core, which we grouped with the other tumor components in our experiments). While our experiments clearly show the potential of our method as a transfer learning method, accurate tumor classification in this dataset will require the use of multiple modalities. However, it might be possible to use transfer learning between pairs of modalities that together contain sufficient information (e.g., T1/T2 and T1+c/FLAIR).

4.6.4 *Remaining thoughts*

In this chapter, we used autoencoder-like models to learn features without discriminative training. The advantage of this approach is that the representation learning does not require labels, only paired scans. Labels are required for training the classifier, but they can also come from unpaired scans from only one of the modalities. A disadvantage of this unlabeled feature learning is that the representations may contain some features that have no discriminative value, but are needed to compute the reconstruction. An alternative network that combines feature learning and classification might be able to obtain a better performance by focusing only on discriminative features. Although this is outside the scope of this chapter, the approaches discussed here could also be applied to such classification networks.

The axial neural network discussed in this chapter learns a separate transformation for each modality, as opposed to models that use a single transformation for all modalities (such as our baseline methods). Single-transformation models essentially learn to extract modality-invariant features with transformations that are insensitive to the source modality, which limits them to features that can be extracted in the same way from all modalities. In contrast, multi-transformation models such as ours learn a shared feature representation by learning modality-specific transformations. This is a more flexible approach that can, in theory at least, extract any information that is common to all modalities, even if it is represented differently in each modality.

Since this chapter is focused on analyzing cross-modality classification, and not on finding the best knee cartilage or brain tumor segmentation segmentation method per se, it is difficult to compare our results with those of state-of-the-art approaches. Many knee cartilage segmentation methods use shape-based post-processing methods [83]. Brain tumor segmentation methods, such as those for the BRATS challenge [1], generally use multi-modal information to get good classification results. The results of these more specialized methods are better than those presented in this chapter.

4.7 Conclusion

Differences in appearance make it difficult to apply a classifier trained on data from one source to data from another. The proposed representation learning method attacks this problem by transforming data from different sources to a shared feature representation. We found that this yields both modality-specific and cross-modality features. The basic axial neural network architecture can be extended with three methods that further improve cross-modality performance. Modality dropout trains the network by randomly removing some modalities during training, which forces the model to learn cross-modality reconstructions. Per-feature normalization improves cross-modality similarity by normalizing all features to zero mean and unit standard deviation. A similarity term explicitly adds cross-modality similarity to the learning objective of the network. Based on our experiments on two different datasets, we found that modality dropout and per-feature normalization are crucial to maximize the number of cross-modality features and obtain the best cross-modality classification results. The similarity term has a strong influence in models without either modality dropout or per-feature normalization, but has only a minor positive contribution if both other techniques are used.

Acknowledgments

This research is financed by the Netherlands Organization for Scientific Research (NWO). We used supercomputer facilities sponsored by NWO Exact and Natural Sciences and used public data from the Osteoarthritis Initiative (OAI) and the Brain Tumor Segmentation Challenge (BRATS).

5

Unpaired, unsupervised domain adaptation assumes your domains are already similar

Unsupervised domain adaptation is a popular method in medical image analysis, but it can be tricky to make it work: without labels to link the domains, domains must be matched using feature distributions. If there is no additional information, this often leaves a choice between multiple possibilities to map the data, which may be equally likely but not equally correct. In this chapter, we explore the fundamental problems that may arise in unsupervised domain adaptation, and discuss conditions that might still make it work. We demonstrate these conditions in experiments with synthetic data, MNIST digits, and medical images. We observe that practical success of unsupervised domain adaptation relies on existing similarities in the data, and is anything but guaranteed in the general case.

CHAPTER 5. UNPAIRED, UNSUPERVISED DOMAIN ADAPTATION ASSUMES YOUR DOMAINS ARE ALREADY SIMILAR

Chapter based on

G. van Tulder and M. de Bruijne, “Unpaired, unsupervised domain adaptation assumes your domains are already similar,” Submitted.

5.1 Introduction

Modern deep learning methods for medical image analysis achieve impressive results, but the models they produce often generalize poorly to data from different scanners or different medical centers. This is especially inconvenient in medical imaging because it can be time-consuming and expensive to obtain the ground-truth annotations for a new training set. Domain adaptation methods address this problem by adapting models trained on data from one domain, the *source*, to data from another, the *target*. If the domain adaptation step works well, models trained for existing datasets can be applied to data from new domains with only a limited performance loss. Similarly, domain adaptation can be used to combine data from multiple sources in a single model, either by modelling the differences between domains or by reducing them.

5.1.1 Unsupervised domain adaptation

Domain adaptation comes in many shapes and forms (see Guan et al. [84] for a recent overview of applications in medical imaging). In this chapter we study *unsupervised domain adaptation*, which assumes that labelled data is only available for the source domain. Some methods for unsupervised domain adaptation learn the translation between domains from paired data, such as scans of the same patient in different scanners. Here, we investigate a more challenging setting: unsupervised domain adaptation without paired samples.

Without information on individual sample pairs, the mapping between domains must be learned on a distribution level. To do this, a common assumption is that although the data from the source and target domains *looks* different, the *underlying structure and tissue types* are quite similar. For example, a brain scan might look different in different scanners, but the anatomical information is the same. This correspondence can be exploited to learn a mapping between domains: if the domains have similar underlying structure and tissue types, we should expect the features and outputs to have a similar distribution as well.

5.1.2 Image-to-image translation

Many unsupervised domain adaptation methods are based on *image-to-image translation*: by translating images from the target domain to the source do-

main, they can be analyzed using the existing classifiers trained on source data. For example, the popular CycleGAN model [85] is optimized using a cycle-consistency loss, which minimizes the reconstruction loss of a source–target–source translation, and an adversarial loss that discriminates between real and translated target images.

Image-to-image translation is complex and relatively inefficient. The translation model must translate all information in the images, but only some of that is useful to the subsequent classification or segmentation model. Moreover, the focus on reconstruction loss may remove useful information that is difficult to translate between images. In many cases, finding a perfect translation might be impossible. For example, a translation between MRI and CT may only preserve information that is captured by both modalities.

5.1.3 *Learning domain-invariant representations*

An alternative to image-to-image translation is *domain adaptation in feature space* by learning domain-invariant representations. After mapping domain-specific inputs to a common, domain-invariant feature representation, the same classifier or segmentation model can be used for all domains. If the datasets contains paired samples, the domain-specific mappings can be learned with a loss that compares the representation of the same sample across domains. Without paired samples, the mappings can be learned by aligning the feature distributions for both domains. There are many ways to do this, such as by optimizing a distribution similarity metric such as the Maximum Mean Discrepancy loss (MMD, [86]), or by training a variational autoencoder [87].

In this chapter, we use the popular approach of *domain adversarial learning* [88]. This method relies on a domain discriminator that is trained to predict the domain of a sample given its feature representation. By using this discriminator in an adversarial learning objective for the feature encoding model, the encoder is encouraged to learn domain-invariant representations. Tzeng et al. [89] describe a general framework for adversarial discriminative domain adaptation (ADDA) that covers many variants of this approach. Kamnitsas et al. [75] present an early application of domain adversarial learning in a paper on brain lesion segmentation.

We investigate the application of representation learning to unsupervised domain adaptation with unpaired samples, where we assume that labels are

only available for the source domain and there is no direct link between samples in the source and target domains. We use domain adversarial learning to implement our domain adaptation objective, but we believe that many of our conclusions also hold for other methods.

5.1.4 *Why does this even work?*

Domain adversarial learning is a popular method in medical image analysis [84], often with good results, but there has been relatively little research into *why* it works. At first glance, domain adversarial learning makes very few assumptions about the data, and should be able to align any pair of domains just by matching their feature distributions. In practice, we argue in this chapter, aligning distributions is not sufficient: there is usually more than one way to match the domains, which means that additional assumptions about the data are needed to find the correct solution.

In early work on this topic, Ben-David et al. [90, 91] explored the theoretical bounds of the error of a domain adaptation model [90] and discussed the assumptions for a successful domain adaptation result [91]. Most importantly, they suggest that the unlabeled source and target distributions should be similar. More recently, Zhao et al. [92] provided a theoretical analysis of domain adaptation by learning invariant representations, i.e., intermediate features which have a similar distribution in the source and target domains. Zhao et al. show that in general, learning an invariant representation and achieving a small error on the source domain is not sufficient to guarantee a small error on the target domain, because the labelling function may be different for both domains.

In this chapter, we explore these themes from a medical imaging perspective. We hypothesize that a successful domain adaptation using adversarial learning requires explicit or implicit assumptions about the data, or more specifically: assumptions about the similarities between domains. We explore what these assumptions can be, and show why they help to obtain useful domain adaptation results. We investigate a number of data and model characteristics that often appear in medical imaging and that might explain why medical domain adversarial learning is successful. We explore these properties in several practical experiments, comparing results for datasets with different properties and different network architectures.

5.1.5 *Outline*

Section 5.2 presents an overview of related work in adversarial domain adaptation for medical images. Section 5.3 describes the unsupervised domain adaptation approach. Section 5.4 discusses the problems with this approach, and why it should not work in theory. Section 5.5 explains why it sometimes does work in practice. Section 5.6 introduces the metrics used to evaluate the results. Section 5.7.1 describes the technical implementation of the experiments. Section 5.7.2 shows the experiments on a synthetic dataset, followed by Section 5.7.3 on MNIST digits and Section 5.7.4 on brain tumor classification. Section 5.8 and Section 5.9 provide a discussion and conclusion.

5.2 *Related work*

We summarize the main trends on adversarial domain adaptation in a medical context. We discuss two approaches: image-level domain adaptation, which translates images between domains, and feature-level domain adaptation, the approach used in this chapter, which learns domain-invariant feature representations. Guan et al. [84] provide a recent survey of domain adaptation in medical imaging, covering adversarial learning and other methods.

5.2.1 *Image-level domain adaptation with a cycle-consistency loss*

Many adversarial domain adaptation works use image-to-image translation with a cycle-consistency loss, based on the CycleGAN model [85]. Cohen et al. [93] point out that this type of image-to-image translation may not be ideal. They argue that distribution matching is sensitive to differences in the sample distribution between the source and target domains, which can lead to unrealistic and incorrect translations. They illustrate this with a CycleGAN model that adds spurious tumor patterns when translating between brain MRI protocols. The CyCADA model [94] adds a semantic consistency loss that aligns the translated image on a feature level or on a task-specific level, such as the output of a classification model.

In medical imaging, the CycleGAN approach has been used for MRI-to-CT image synthesis [95–97], multi-contrast MRI [98], fundus imaging [99], chest X-ray [100], histopathology [101], and ultrasound [97] images. The basic cycle-consistency loss is sometimes extended with additional, application-specific

constraints, e.g., by encouraging structural or anatomical consistency between domains [102–104]. Other works using CycleGAN align domains based on the output of auxiliary tasks such as segmentation [105–107], or by directly matching feature values [108, 109]. Some other works use atlas registration [110] or a student-teacher model with inter- and intra-domain teachers [111] to improve the results.

In general, cycle-consistency alone is not sufficient to learn reliable translations [94]. Additional constraints, and corresponding assumptions about the domains, are required to get usable results. Even without additional constraints, these image-to-image translation models use a convolutional approach that assumes that images have similar spatial arrangements across domains.

5.2.2 *Feature-level domain adaptation*

Adversarial domain adaptation by learning domain-invariant feature representations [89], without explicitly reconstructing images from the target domain, is also commonly used for medical image classification and segmentation. Kamnitsas et al. [75] presented an early version of this approach for brain lesion segmentation. The method was later applied for many other tasks, such as anatomical structure segmentation [112], multi-modal brain MRI [113], colonoscopy images [114], or fundus imaging [115]. Instead of learning a fully domain-invariant model, some approaches try to disentangle domain-invariant and domain-specific features [116, 117], which allows them to exploit domain-specific information where necessary.

Feature-level domain adaptation can be extended with additional constraints, e.g., by adding structural constraints on the output of a segmentation model. Bateson et al. [118] argue that adversarial training may not be suitable for adapting segmentation networks, and suggest using domain-invariant prior knowledge about common anatomical structures to direct the adaptation. Similarly, Cui et al. [119] used several structural constraints to capture common cardiac structure across MRI and CT. More indirectly, Wang et al. [120] applied an adversarial domain discriminator to a segmentation output. Li et al. [111] provided additional semantic feature maps to the discriminator, to exploit domain-invariant spatial patterns.

Domain adaptation can also be guided by adding auxiliary tasks to the learning objective. For example, Koohbanani et al. [121] used domain-specific

pretext tasks in a self-supervision setup. Luo et al. [122] used task-specific discriminators to improve domain invariance. Chen et al. [123] proposed a combination of feature-level and image-level methods.

5.3 *Methods*

5.3.1 *Domain adaptation with a neural network*

In this chapter, we consider domain adaptation in a deep neural network with the following architecture: an encoder that maps the domain-specific input to a latent, domain-invariant feature representation, and a shared prediction model that uses the intermediate representation to make a prediction. We use classification as the prediction task in this chapter, but this could also be a segmentation or regression task. The domain adaptation in the encoder can take two forms: using a single encoder that is used for both domains, or using a separate, domain-specific encoder for each domain.

The first approach requires a single, common model that works well for data from both domains. Since it uses the same feature extraction path for both domains, it will automatically map both domains to the same representation if the domains are fairly similar. However, the approach provides limited flexibility to adapt to larger differences between domains, and is likely to focus on domain-invariant features that have similar appearance in both domains.

The second approach uses a separate encoding path for each domain. We use this architecture in this chapter. In contrast to a shared encoder, domain-specific encoders can accommodate large differences between domains: if the encoders are complex enough, they can map the inputs to a shared encoding that is common to both domains. However, the increased power and flexibility also increase the risk that the encoders learn inconsistent mappings, since there are no shared features that link the two encoding branches. We will revisit this limitation in Section 5.4.

5.3.2 *Adversarial domain adaptation*

The source encoder and the shared prediction model can be trained with a supervised learning objective, computed on labelled data from the source domain. To train the target encoder and learn a domain-invariant feature

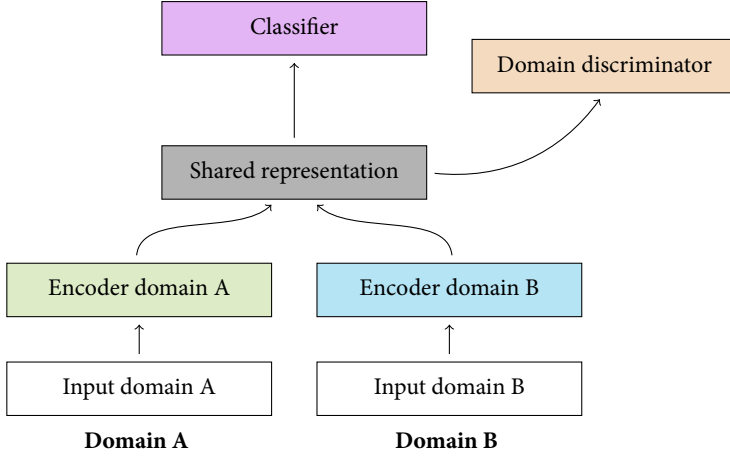


Figure 5.1: The domain adaptation model uses a separate encoding branch for each domain. The output of these encoders is forwarded to a shared classification network and to a domain discriminator. A domain adversarial learning objective is applied to encourage the encoders to learn a shared, domain-invariant representation space.

representation, we need an unsupervised objective that can use unlabelled target data. In this chapter, we use an adversarial domain adaptation objective.

Adversarial learning [124] is commonly used to train generative models. A discriminator is trained to discriminate between samples from a real distribution and samples generated by a generator model. By optimizing the generator to maximize the loss of the discriminator, the samples generated by the model will start to resemble those from the real distribution.

In domain adversarial learning [89, 125], the discriminator is presented with feature representations of samples from the source and target domain, and is trained to predict the domain of each sample. The discriminator loss is included as an adversarial term in the learning objective for the encoders, which encourages them to learn domain-invariant representations that have similar distributions in both domains.

5.3.3 Architecture and learning objectives

Figure 5.1 shows the model with domain-specific encoders as it is used in this chapter. We denote the domain-specific encoders as F_{src} for the source and

F_{tgt} for the target domain. Given an input \mathbf{x} , we use the appropriate encoder $F \in \{F_{\text{src}}, F_{\text{tgt}}\}$ to compute the representation $F(\mathbf{x})$. This representation is then used as input for a shared classification model G to compute the prediction

$$\tilde{y} = G(F(\mathbf{x})). \quad (5.1)$$

The learning objective consists of a classification component and a domain-adversarial component. The classification component is computed only for the source samples, using the ground-truth label y to compute the binary cross-entropy loss:

$$\mathcal{L}_{\text{class}} = -y \log \tilde{y} - (1 - y) \log (1 - \tilde{y}). \quad (5.2)$$

A separate domain discriminator D is used to encourage the two encoders to produce domain-invariant representations. The discriminator is trained with a binary cross-entropy loss to predict the domain of a sample given its intermediate feature representation:

$$\mathcal{L}_{\text{disc}} = \begin{cases} -\log D(F_{\text{src}}(\mathbf{x})), & \text{if } \mathbf{x} \text{ is from the source domain;} \\ -\log (1 - D(F_{\text{tgt}}(\mathbf{x}))), & \text{if } \mathbf{x} \text{ is from the target domain.} \end{cases} \quad (5.3)$$

We reuse this learning objective $\mathcal{L}_{\text{disc}}$ as an adversarial term in the learning objective for the encoder. We optimize the encoder and classifier to minimize the classification loss and maximize the discriminator loss:

$$\mathcal{L}_{\text{combined}} = \lambda_{\text{class}} \mathcal{L}_{\text{class}} - \lambda_{\text{disc}} \mathcal{L}_{\text{disc}}. \quad (5.4)$$

5.4 Problem analysis

In the absence of paired samples, the domain adaptation model can only compare domains at a distribution level. This has consequences for the quality and correctness of the results.

5.4.1 Two phases of domain adaptation

For the following analysis, we will divide the unsupervised domain adaptation task in two phases. First, the method must determine the structure of the input space for each domain, e.g., by identifying clusters of samples. Second, the

method must match the structures in both domains in order to map the feature representations of samples from one domain to the other. If both phases are successful, the domain adaptation will result in the correct classification on the target domain.

Our analysis is further based on the smoothness assumption in machine learning, which states that samples that are close in the input space are likely to belong to the same class. Similarly, domain adaptation learns a smooth mapping between domains: samples that are close together in the target domain will most likely be mapped close together in the source domain.

For simplicity, for this problem analysis, we will assume that the data distribution is so smooth that the samples in each domain can be grouped in a number of distinct clusters. In practice, we may not be able to find perfectly distinct clusters in the data – for example, because samples from different classes may have very similar appearance and classes may overlap – but this will not affect our general conclusion.

Ideally, each class would correspond to a single cluster in each domain, and the task of domain adaptation would be to link each cluster to the correct cluster in the other domains. In practice, it is likely that the classes are more heterogeneous and consist of multiple subclusters. This complicates the task of the domain adaptation algorithm, which must now identify all subclusters and link them to the correct classes in the other domain.

Both domain adaptation phases must be successful to obtain a good classification result. Observing the target classification accuracy at the end is not sufficient to identify which of the two parts failed: a low target accuracy combined with a high source accuracy could mean that both clustering and mapping failed, but it could also mean that the model found the right clusters but mapped them incorrectly between domains.

5.4.2 *Unsupervised domain adaptation requires additional assumptions*

Consider a thought experiment with a balanced binary classification problem, in which each class contains fairly homogeneous samples. Given the in-class homogeneity, it is easy to find the correct clusters. Linking those clusters across domains is more difficult: without additional information, it is impossible to say which cluster in the target domain belongs to which cluster in the source domain. As a result, domain adaptation has only a 50% chance of success.

In unsuccessful cases the clustering may still work, while the classification accuracy may be close to zero because the clusters are linked incorrectly.

Observe that the problem in this simple example would not occur if the classes were not balanced. If the class imbalance was similar for both domains, the model could use the size of each cluster to learn a correct mapping.

However, the result also depends on the assumption that the samples within each class are sufficiently homogeneous. In practice, this will almost never be the case. For example, in some applications different types of tissue might map to the same class. In segmentation tasks, voxels near the edge of a structure may have a different appearance from voxels located in the center, even if the whole structure belongs to a single class, and the representation of near-edge voxels may even vary with orientation. For this analysis, we therefore assume that each class consists of multiple subclusters that are internally homogeneous. This makes it more difficult to find the correct solution, since the required class balance can be achieved with different combinations of subclusters.

Consider an experiment in which the data is subdivided in 10 homogeneous subclusters of equal size. If the class balance in the source domain is 80–20, that is, 8 and 2 subclusters per class, this can be replicated in the target domain by mapping any combination of two subclusters to the minority class. Since there is no way for the algorithm to identify which combination is correct, the domain adaption is likely to fail even if it discovers the clusters correctly.

In this chapter, we argue that the conclusions for these thought experiments can be extended to domain adaptation on real datasets. We provide experimental verification of these specific results on synthetic data in Section 5.7.2.

5.5 *Exploiting domain-invariant properties*

In the previous section, we argued that unsupervised domain adaptation is unlikely to learn correct mappings if there is no information to link subclusters across domains. In practice, of course, this is too pessimistic. Unlike the dataset in our example, most real-world datasets will have some domain-invariant properties that can be exploited to align domains.

The outcome of adversarial domain adaptation depends on the initial representations, which usually depend on randomly initialized weights. Since the training makes small, incremental changes to the encoders to match distributions, it can increase similarity of clusters that are already similar, but

it is unlikely to swap entire clusters. If the initial guess was correct, the final mapping is likely to be correct as well.

Fortunately, the initial mapping and subsequent optimization are not completely random, but depend on biases in the data and the model. If these biases are helpful, domain adaptation is more likely to succeed. In this section, we introduce four domain-invariant properties that often appear in medical images and may provide a useful domain adaptation bias. We will then discuss how these properties can affect the domain adaptation results implicitly.

5.5.1 *Similar class imbalance*

In Section 5.4.2, we argued that class imbalance might be used to link domains with homogeneous classes. Many real-life datasets show some class imbalance, but most are also heterogeneous. Our thought experiment showed that this makes the imbalance less useful, because the subclusters in the data can be combined in arbitrary ways to obtain the required class balance. The experiments later in this chapter confirm this.

5.5.2 *Similar intensities*

If the average image intensities are consistent between domains, e.g., if a class that is brighter in one domain is also brighter in the other domain, this similarity can be used to learn the mapping between domains. This assumption often holds for images from the same imaging modality. For example, CT images from different scanners will show roughly similar intensity patterns.

This similarity can be exploited explicitly (models with shared encoders are based on this assumption), but it can also affect the domain adaptation implicitly. Here, we argue that the architecture and initialization of the model can interact with intensity similarities in the data to bias the model towards particular mappings. Given a random initialization of the weights and a standard activation function, the magnitude of the input intensities is reflected in the representation: on average, a class with inputs around zero will produce smaller absolute feature values in the encoder output than a class with larger input values. This initial bias is consistent for all domain-specific encoders, and can be used to map classes with similar intensity to similar feature values.

5.5.3 *Similar spatial structures*

In applications with spatial inputs, source and target domains may have similar spatial arrangements. For example, in MRI and CT images of the same anatomy, the modalities produce images that show the same anatomical structures, even if the appearance is different. We argue that domain adaptation could exploit spatial similarities like these if the models use convolution.

With convolutional encoders, the latent representation preserves the spatial structure of the input. Even with a random initialization of the weights, the output of convolutional encoders in different domains will generate representations that are spatially similar. As long as the classes have the same spatial arrangement in both domains, these similarities could be exploited by the model to link the domains, even if the structures themselves have a different appearance.

5.5.4 *Similar local texture and intensity distributions*

A fourth source of similarities is local texture. Especially in segmentation tasks, texture information could be used to identify components if the textures are similar across domains. Using convolution makes the encoders sensitive to type and amount of texture: heavily textured areas may produce a different convolution output than areas with a lighter texture, even with random initialization of the weights. This could bias the encoders to learn similar representations for similarly textured areas, which would lead to a correct mapping if the texture has similar meaning across domains. In medical imaging, this kind of texture similarity can appear in multi-view images from the same imaging modality, such as multi-modal MRI or smaller variations in scanning protocol. On the other hand, cross-modality applications such as MRI-to-CT could have different textures in each domain, which could lead to a bias towards incorrect mappings.

5.6 *How to measure domain adaptation success?*

We use several metrics to measure the performance of the models, based on the two phases in the domain adaptation process that we identified in Section 5.4.1: finding clusters in each domain, and linking those clusters across domains.

5.6.1 Measuring the correctness of the mapping

Ultimately, the performance of domain adaptation is defined by the *classification accuracy* on the target domain. In the experiments in this chapter, we compute the classification accuracy on the source domain and on the target domain. Since the classifier is trained only on the source domain, we expect the performance on the target domain to be lower, but ideally the two should be as close as possible.

5.6.2 Measuring mapping quality

However, as discussed in Section 5.4.1, classification accuracy alone does not provide the full picture, since it measures the combined success of both domain adaptation phases. We use three metrics to evaluate the clustering phase independent of the linking phase.

Compensated accuracy. A simple case of cross-domain confusion in a binary classification task is a scenario where the domain adaptation method correctly finds two clusters that correspond to the two classes in the target domain, but maps these clusters to the incorrect class in the source domain. To measure this effect, we define the *compensated accuracy* as

$$\text{compensated accuracy} = \max(\text{accuracy}, 100\% - \text{accuracy}).$$

Mapping confidence. In more complicated problems with heterogeneous classes, we can assume that each class is made up of several subclusters. We define a domain adaptation *confidence* score that measures whether the domain adaptation model correctly identifies the subclusters in the data, independent of whether they are assigned to the correct class.

The metric is defined using subcluster labels. We first compute the subcluster confusion matrix CM and the class balance CB:

$$\text{CM}(Y, C) = \sum_i I(\hat{y}_i = Y, c_i = C), \quad (5.5)$$

$$\text{CB}(Y) = \frac{1}{N} \sum_i I(y_i = Y), \quad (5.6)$$

where $I(\cdot)$ is the indicator function, $Y \in 0, 1$ is a binary class, C is a subcluster, N is the number of samples, and \hat{y}_i, y_i, c_i are the predicted class, the ground-

truth class, and subcluster of sample i , respectively. We then compute the class-balanced weighted confusion matrix WCM and the class difference CD:

$$\text{WCM}(Y, C) = \text{CM}(Y, C) / (2 \cdot \text{CB}(Y)) \quad (5.7)$$

$$\text{CD} = \sum_C \text{WCM}(0, C) - \text{WCM}(1, C). \quad (5.8)$$

Finally, we compute the confidence as

$$\text{Confidence} = \sum_C \max_Y (\text{WCM}(Y, C)) - |\text{CD}|. \quad (5.9)$$

The confidence score ranges between 0% and 100%. If the model identifies all subclusters correctly (i.e., samples from one subcluster are all assigned to the same class), the confidence score will be 100% independent of the correctness of the classification. If the model achieves no clustering (e.g., samples from one subcluster are equally distributed over the two classes), the score is 0%. For a dataset with two homogeneous classes, the confidence is equal to the compensated accuracy.

Linear CKA. At the level of the encoder outputs, we compute the *representation similarity* using linear CKA (centered kernel alignment [126]). Linear CKA measures the content-based feature similarity while allowing for differences in representation, giving an indication of how much information is shared by both domains. The method is often used to compare the feature representations of different networks trained on the same data, but we use it to compare representations of paired samples across domains. We refer to Kornblith et al. [126] for the full definition. In our experiments, the linear CKA ranges from 0 (no alignment) to 100 (complete alignment).

5.7 Experiments and results

5.7.1 Implementation

We use neural networks to implement the domain-specific encoders F_{src} and F_{tgt} , the classifier G , and the domain discriminator D . The architectures of these networks are described in the following sections. In some experiments, we vary the level of the intermediate representation: we use the same set of layers for $F + G$ combined, but change how they are divided between the

encoders F and the classifier G . The discriminator and classifier are optimized with a binary cross-entropy objective, using a gradient reversal layer between the discriminator and the encoders to implement the adversarial objective. All models are implemented in PyTorch and trained using the Adam optimizer until convergence. Detailed architectures and hyperparameters are shown at the end of this chapter.

5.7.2 Experiments with synthetic data

Data and architecture. We construct a synthetic, binary classification problem with 10 input features, $\mathbf{x} \in \mathbb{R}^{10}$, and generate samples for two domains with identical or different input representations (Table 5.1), according to the following settings:

- For “Two $-1/+1$ ”, we construct a problem with two clusters: samples $[-1, -1, -1, -1, -1, -1, -1, -1, -1, -1]$ for class 0 and $[1, 1, 1, 1, 1, 1, 1, 1, 1, 1]$ for class 1 in both domains (i.e., both domains receive the same input).
- The variant “Two $-1/+1$, inverted” uses the same type of samples, but we invert the labels in the target domain: $[-1, \dots, -1]$ corresponds to class 1 and $[1, \dots, 1]$ to class 0, simulating a very strong difference between domains.
- Similarly, “Two 0/1” and “Two 0/1, inverted” use samples with values $[0, \dots, 0]$ and $[1, \dots, 1]$ with equal or swapped classes, respectively.
- Finally, “Ten” includes samples with one-hot encoding, representing 10 different clusters: from $[1, 0, 0, 0, 0, 0, 0, 0, 0, 0]$ to $[0, 0, 0, 0, 0, 0, 0, 0, 0, 1]$. In our experiments, we assign each cluster to one of the output classes, depending on the required class balance: for example, we assign 2 clusters to class 0 and 8 clusters to class 1 to simulate a 20 – 80 class balance.

For all settings, we add random noise to all features, sampled from a uniform $[-0.5, 0.5]$ distribution. This creates many unique samples, without introducing class overlap. All experiments use the same, very simple architecture with linear encoders and decoders (Figure 5.2).

Results. We run the experiment described in Section 5.4.2 with the synthetic datasets. With homogeneous and balanced classes (experiment “Synthetic two $-1/+1$, Balanced 50–50” in Table 5.2), the model obtains a perfect classification

Synthetic dataset	Clusters	Source	Target
Two $-1/+1$	2	$-1/+1$	$-1/+1$
Two $-1/+1$, inverted	2	$-1/+1$	$+1/-1$
Two $0/1$	2	$0/1$	$0/1$
Two $0/1$, inverted	2	$0/1$	$1/0$
Ten	10	One-hot	One-hot

Table 5.1: Synthetic datasets. Datasets with two clusters use a feature vector filled with the same value. Datasets with ten clusters use a one-hot encoding with the feature corresponding to the cluster set to 1. Uniformly distributed noise is added to all features.

accuracy on the source domain. On the target domain, however, the average classification accuracy is much lower. Looking closer, we observe that the target accuracy in individual runs is either 0% or 100%, while the compensated accuracy is always 100% for both domains. This confirms our earlier prediction that the model can easily find the clusters in the data, but is unable to reliably find the correct link between domains.

Next, we try an experiment with unbalanced classes (experiments “Synthetic two $-1/+1$, Unbalanced 20–80” and “— 80–20”). The class imbalance helps the model to find the correct mapping, resulting in a perfect target accuracy in all runs. As hypothesized in Section 5.4.2, class imbalance is not sufficient in datasets with heterogeneous classes. When we perform the same experiment with heterogeneous classes (experiments “Synthetic ten”), we see that the model fails to learn a good target classification. The high confidence scores indicate that the model is able to find the subclusters, but it is unable to link them correctly between domains. We confirmed this by inspecting the confusion matrices (Table 5.5).

Finally, we find that the domain adaptation is sensitive to the representation of the input features. We repeat the experiments with homogeneous classes, but switch the input features from $\{-1, +1\}$ to $\{0, 1\}$ (experiments “Synthetic two $0/1$ ”). With these input values, even with balanced classes, the model now learns a perfect accuracy on the target domain in almost all runs. We explain this surprising result with the bias predicted in Section 5.5.2: the representation of the data interacts with the model, introducing a bias that causes the model to

	Accuracy (%)		Compensated accuracy (%)		Confidence	
	Source	Target	Source	Target	Source	Target
Synthetic two +1/ -1						
Unbalanced 20-80	100.0	100.0	100.0	100.0	98.9	99.1
Balanced 50-50	100.0	64.0	100.0	100.0	99.1	99.3
Unbalanced 80-20	100.0	100.0	100.0	100.0	98.8	98.9
Synthetic two +1/ -1, inverted target						
Unbalanced 20-80	100.0	100.0	100.0	100.0	99.1	99.1
Balanced 50-50	100.0	40.0	100.0	100.0	99.0	99.0
Unbalanced 80-20	100.0	100.0	100.0	100.0	98.8	99.0
Synthetic two 0/1						
Unbalanced 20-80	100.0	99.9	100.0	99.9	99.3	98.1
Balanced 50-50	100.0	92.0	100.0	100.0	99.4	99.2
Unbalanced 80-20	100.0	99.2	100.0	99.2	99.2	94.8
Synthetic two 0/1, inverted target						
Unbalanced 20-80	99.6	58.4	99.6	66.2	98.2	43.4
Balanced 50-50	100.0	4.0	100.0	100.0	99.2	99.0
Unbalanced 80-20	100.0	96.0	100.0	96.0	98.9	79.2
Synthetic ten						
Unbalanced 20-80	100.0	72.2	100.0	72.2	98.9	98.9
Balanced 50-50	100.0	47.6	100.0	65.3	99.3	98.1
Unbalanced 80-20	100.0	68.8	100.0	68.8	99.0	98.7

Table 5.2: Results for experiments with synthetic data, showing mean validation performance averaged over 25 models. Models were trained to convergence. Sparkline plots show the distribution of results for individual runs.

learn the same representation for both domains. We find confirmation in the results for experiments with inverted target features (experiments “Synthetic 0/1, inverted”), in which the models reliably learn the incorrect mapping.

5.7.3 Experiments with MNIST digits

Data. We use the 28×28 -pixel MNIST¹ digit images with intensities scaled to $[0, 1]$, using the original training and test splits. We convert the original 10-class problem into a binary classification task by grouping the digits $\{0, 1, 2, 3, 4\}$ and $\{5, 6, 7, 8, 9\}$. To simulate a 20–80 or 80–20 class imbalance, we respectively classify $\{0, 1\}$ vs $\{2, 3, \dots, 9\}$ and $\{0, 1, \dots, 7\}$ vs $\{8, 9\}$. In all cases, we use the original digit labels to compute our subcluster-based confidence metric.

We perform experiments with three variations for the target domain: 1. standard, with original images, similar to the source domain; 2. inverted, with inverted intensities ($1 - \text{the original intensity}$) to remove intensity-based similarities; 3. flipped, with images horizontally and vertically mirrored to remove most of the spatial similarities between domains.

Architectures. We use a convolutional network with domain-specific encoders (Figures 5.3 and 5.4). For the *spatial encoder* models, we join the domain-specific encoders at the final spatial layer, just before global pooling. This gives the domain adaptation access to the final spatial feature maps. For the *dense encoder* models, we join the encoders just after the global pooling layer, which means that the domain adaptation method does not receive any spatial information.

Results. The results in Table 5.3 show that the domain adaptation model relies on spatial and intensity similarities to link the domains. In all experiments, the models with spatial encoders achieve a higher target accuracy than the models with dense encoders. The spatial encoders fail if they are applied to a data with a flipped target domain, because there are no spatial similarities to rely on. At the same time, the models with spatial encoders are able to learn with large intensity shifts: the target accuracy on a target domain with inverted images is similar to that on standard images. However, the linear CKA scores are lower, which suggests that the representation still depends on intensity information.

1. <http://yann.lecun.com/exdb/mnist/>

	Accuracy (%)		Confidence		Linear CKA	
	Source	Target	Source	Target	Source	Target
MNIST, spatial encoder						
Unbalanced 20-80	99.6	98.2	97.9	92.2	95.7	95.7
Balanced 50-50	98.7	97.3	95.4	92.0	99.6	99.6
Unbalanced 80-20	99.3	89.3	97.4	46.8	66.5	66.5
MNIST, dense encoder						
Unbalanced 20-80	99.3	79.2	97.4	78.1	40.2	40.2
Balanced 50-50	98.8	53.4	95.4	71.7	32.1	32.1
Unbalanced 80-20	99.2	66.5	97.1	76.8	33.7	33.7
MNIST inverted target, spatial encoder						
Unbalanced 20-80	99.6	96.4	97.9	93.9	74.7	74.7
Balanced 50-50	96.7	96.6	91.7	91.7	74.7	74.7
Unbalanced 80-20	99.2	99.2	97.0	97.2	78.6	78.6
MNIST inverted target, dense encoder						
Unbalanced 20-80	99.4	66.9	97.5	33.6	34.9	34.9
Balanced 50-50	97.5	48.8	94.5	31.1	20.4	20.4
Unbalanced 80-20	97.7	61.9	91.4	54.2	14.5	14.5
MNIST flipped target, spatial encoder						
Unbalanced 20-80	99.6	80.2	98.1	29.9	52.5	52.5
Balanced 50-50	96.4	59.6	92.7	56.0	39.4	39.4
Unbalanced 80-20	98.0	75.3	94.3	66.4	36.8	36.8
MNIST flipped target, dense encoder						
Unbalanced 20-80	99.4	77.9	97.4	77.3	53.0	53.0
Balanced 50-50	91.2	51.0	80.2	53.4	18.3	18.3
Unbalanced 80-20	99.3	65.5	97.2	77.9	24.0	24.0

Table 5.3: Results for experiments with MNIST data, showing mean validation performance averaged over 25 models. Models were trained to convergence. Sparkline plots show the distribution of results for individual runs.

The dense encoders have a low target accuracy on the standard domain with balanced classes, but show a reasonably high confidence. This indicates that they can still find some clusters in the data. The models fail completely when the target images are inverted, which shows that they rely on intensity similarities to link the domains.

Surprisingly, the linear CKA scores of the dense encoder model drop when the target images are flipped. This is counter-intuitive, because these models do not receive any spatial feature maps. We suspect this could be evidence for our fourth bias (Section 5.5.4): the early convolution layers encode local texture information that influences the later, global feature representations.

5.7.4 Experiments with brain MRI scans

Data. We present a brief demonstration on brain MRI scans from the BRATS 2015 dataset [1]. We think it is likely that subclusters as discussed in this chapter appear in any realistic dataset. However, to properly evaluate and observe the behavior, we require a dataset with known subset labels.

The brain tumor segmentation dataset includes four MRI sequences (T1, T1 with contrast, T2, FLAIR) and manual segmentations of four tumor components (necrosis, edema, non-enhancing tumor, and non-enhancing tumor). We extract 2D patches of 15×15 pixels, labelled with the class of the center pixel and balanced to have an equal number of samples per class. We define a binary classification problem by combining the BRATS labels into two classes: necrosis/edema and non-enhancing/enhancing tumor, which roughly correspond to the outer and inner part of the segmentation, respectively. We use the original class labels as the subclusters in our analysis.

Architectures. We compare four models, all based on the same architecture but joining the source and target branches at different levels (Figures 5.5 to 5.8). The *spatial encoder, early join* model joins the representations at an early spatial level (after the first pooling layer). This makes it relatively easy to join the domains if the domains are fairly similar, but also limits the complexity of the transformations that can be modelled. The *spatial encoder, late join* model joins the representations before the global pooling layer. This allows the model to learn more complex transformations, supporting larger differences between domains, but the increased complexity will also make it more difficult to learn

the correct transformation. Because the representations are joined before global pooling, this architecture can still exploit spatial similarities. The *dense encoder* model joins the representations after global pooling, removing spatial information. The *posterior join* model joins the domain-specific branches only at the level of the final output. This model has the least information, and must link the domains based on the posterior distributions.

Results. Table 5.4 shows the results of these four models on the BRATS dataset. This task is more complicated than those in our previous experiments. The early-join spatial encoder achieves a reasonable target accuracy in a number of runs, but not all. The confidence and linear CKA scores suggest that this model also has modest success at identifying the clusters. The scores for the late-join spatial encoder and the non-spatial models are much worse. Neither the confidence, nor the accuracy on the target domain are very good, indicating that the domain adaptation fails to find clusters or link them between domains. The linear CKA scores are very low, indicating that the models learn very dissimilar representations. On this dataset, spatial information is crucial for the model to learn a correct mapping between domains.

5.8 Discussion

In this chapter we explored the limitations of unsupervised domain adaptation, using adversarial learning to learn domain-invariant representations. We addressed a common domain adaptation scenario where labelled training data is available for the source domain but not for the target domain, and where there are no paired samples that can be used to learn correspondences between domains. In this setting, adversarial domain adaptation attempts to learn a domain-invariant representation by aligning the source and target distributions in the latent feature space. We showed that this unsupervised distribution matching may lead to incorrect results, because there is no guarantee that similar samples in different domains will be mapped to similar latent representations. However, we also observed that domain-invariant properties of the data can introduce a bias that helps the model find the correct mapping.





















	Accuracy (%)		Confidence		Linear CKA
	Source	Target	Source	Target	Target
BRATS, spatial encoder, early join					
Balanced 50–50	73.9 	62.7 	65.6 	57.0 	61.4 
BRATS, spatial encoder					
Balanced 50–50	73.9 	49.6 	69.1 	45.6 	4.3 
BRATS, dense encoder					
Balanced 50–50	77.4 	51.3 	66.9 	45.3 	2.4 
BRATS, posterior join					
Balanced 50–50	77.5 	50.5 	69.3 	46.2 	0.5 

Table 5.4: Results for experiments with BRATS data, showing mean validation performance averaged over 25 models. Models were trained to convergence. Sparkline plots show the distribution of results for individual runs.

5.8.1 *Unsupervised domain adaptation without paired samples is flexible but unpredictable*

In our experiments, we used models with domain-specific encoders. Using domain-specific encoders instead of a single, shared encoder allows the model to accommodate large differences between domains. This is convenient if the domains are very different, because the encoders can learn a domain-specific mapping for each domain. In comparison, a model with a single encoder is restricted to extracting domain-invariant features that have a similar appearance in both domains.

The flexibility afforded by the domain-specific encoders comes at a cost: without labelled target data or paired samples, it is difficult to link the domains correctly. In Section 5.4, we discussed that there are many possible ways to map samples between domains, and there is no guarantee that the model will automatically find the correct solution. The synthetic experiments in Section 5.7.2 showed a clear example of this problem: the models learned a random mapping that was either completely correct or completely wrong.

5.8.2 *Similarities between domains may help or hinder the domain adaptation*

Despite the lack of guarantees, unsupervised domain adaptation can still succeed if the domains are sufficiently similar. In Section 5.5, we discussed four domain-invariant properties that are commonly seen in medical imaging data, and which may provide a useful source of domain adaptation bias:

- The model can use the class imbalance to identify classes, if this is similar between the source and target domains. This is more likely to work in datasets with fairly homogeneous classes, such as our synthetic example.
- The model can match classes based on average intensity, if this is similar in both domains. We saw evidence for an intensity-based bias in the experiments with synthetic and MNIST data.
- The model can use the large-scale spatial similarities to match classes. This is sensitive to rotations and inversions, but can be very powerful if the images in both domains have a similar spatial structure. The convolutional feature extraction layers preserve the spatial arrangement of the input, if

the encoding branches are joined at a spatial level. We observed that spatial information was important in our MNIST and BRATS experiments.

- The model might use local textures to match classes based on the strength of the textures in the image. This requires that the texture is comparable between domains, which might be difficult in more complex tasks, such as between CT and MRI. This effect is more difficult to measure, but we saw signs of this in the confidence scores of the dense encoders in some of our experiments.

Since many medical datasets exhibit some of these similarities, the domain adaptation process may be biased towards learning the correct mapping. Many domain adaptation approaches from the medical imaging literature rely on these similarities between domains explicitly, either by using an architecture with shared encoders or by introducing additional constraints in the domain adaptation process. However, we found that these assumptions are also used implicitly in a model with domain-specific encoders.

5.8.3 *Limitations and practical consequences*

The internal behavior of domain adaptation methods is difficult to observe in practice. Our experiments on synthetic and MNIST data provided useful insights in the process, but the models and data are simpler than those in most real-world applications. The relatively homogeneous data allowed us to compute the subcluster-based metrics required for our analysis, but real data will be more heterogeneous and usually comes without subcluster labels. Our experiments on BRATS used more realistic data, but were less transparent.

The observations in this paper were made on models using domain-specific encoders. While this allows a very flexible mapping between domains, it also makes it harder to learn a correct mapping. In contrast, models with shared encoders may be more likely to find a correct mapping if the domains are somewhat similar, but may have problems with larger differences between domains.

Despite these limitations, we believe that most of our conclusions also apply to more advanced models. Since there are no guarantees that unsupervised domain adaptation works in the general case, its success for specific applications must mean that the models exploit some underlying similarities in

the data. The four assumptions discussed in Section 5.5 suggest what those similarities could be. We believe that many medical imaging tasks satisfy some or all of these assumptions, and suspect that this is why domain adaptation often succeeds.

It is important to be aware of these properties when applying domain adaptation to a new dataset. Even if the assumptions are not explicitly encoded in an auxiliary learning objective or constraint, they may still affect the outcome through implicit biases in the models. We would also like to note that this is not unique to domain adaptation at a feature level. Image-to-image translation methods such as CycleGAN, which constrain the translation to maintain the global spatial structure of the translated images, will face similar problems when translating local textures and intensities.

5.9 Conclusion

Learning unsupervised domain adaptation from unpaired samples is an ambitious goal, and to some extent it is surprising that it works at all. In this chapter, we argued that successful unsupervised domain adaptation relies on similarities between domains. We explored several types of similarity that are common in medical images, and find that they can indeed help to push the domain adaptation in the right direction. However, even if those assumptions are satisfied, a correct domain adaptation is not guaranteed. In our experiments on the BRATS dataset, unsupervised domain adaptation failed for anything but the simplest case. In practice, we suspect that unsupervised domain adaptation can work well if domains are already similar, but needs additional constraints if they are not.

Implementation details

All experiments were implemented with PyTorch. In all experiments, domains A and B were trained with independent training samples from the same distribution. For the synthetic experiments, we used an infinite stream of random samples. For the MNIST experiments, we used the official training and test split. For the BRATS experiment, we used the high-grade glioma subset and split the data in separate training, validation and testing sets, keeping samples from the same subject in a single subset. We used 80 subjects for training domain A, 80 subjects for training domain B, 30 subjects for validation, and 30 subjects for testing. For each subject, we selected patches centered on pixels from the ground-truth segmentation, while maintaining the class balance.

Our aim was to identify scenarios where domain adaptation could potentially work, but was unable to link the two domains. Consequently, we selected hyperparameters based on the results on domain A, while checking the confidence on domain B to ensure that the adaptation did not map all samples to a single class. Using the selected hyperparameters, we ran 25 experiments with different random initializations to obtain the results shown in the tables.

We fixed the weight of the classification term in the learning objective to $\lambda_{\text{class}} = 0.1$ for all experiments. For the discriminative, we chose one of $\lambda_{\text{disc}} \in \{0.3, 0.2, 0.1, 0.01, 0.001, 0.0001\}$ based on the performance on the source domain.

The learning rate was chosen from $\{0.001, 0.0005, 0.0001, 0.00001\}$. For the synthetic experiments, we used 0.001 for all experiments. For MNIST and BRATS, we used 0.001, 0.0005, 0.0001 depending on the setting, but all three values gave similar results.

We optimized the models using Adam with a minibatch size of 128, for 200 epochs (MNIST) or 100 epochs (other experiments). This was sufficient for all networks to converge. We report the results at the end of the final epoch.

Run	Prediction	Clusters from source domain										Clusters from target domain									
		1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10
1	Class 1	9	10	9	10	9						9				9		10		10	10
	Class 2					10	9	10	10	9			9	10	9		9		10		
2	Class 1	10	9	10	9	10						10			9	10			9	9	
	Class 2					9	10	9	9	9			9	9			9	10			10
3	Class 1	10	10	9	9	9						9			10	10		9		9	
	Class 2					10	10	9	9	9			10	10			10		9		10
4	Class 1	10	9										10	10							
	Class 2		9	9	10	10	10	10	9	9	10	9			9	9	10	10	9	10	10
5	Class 1	9	9										10				10				
	Class 2			10	10	10	10	9	9	9	10	10		9	9	10		10	9	9	9
6	Class 1	9	9												9						
	Class 2			9	9	10	9	10	9	10	9	9	10	9		10	10		10	9	10

Table 5.5: Confusion matrices (%) for example runs on the synthetic ten dataset, with balanced (runs 1–3) or unbalanced (runs 4–6) data. Domain adversarial learning finds the correct class balance, but creates random combinations of clusters to do so.

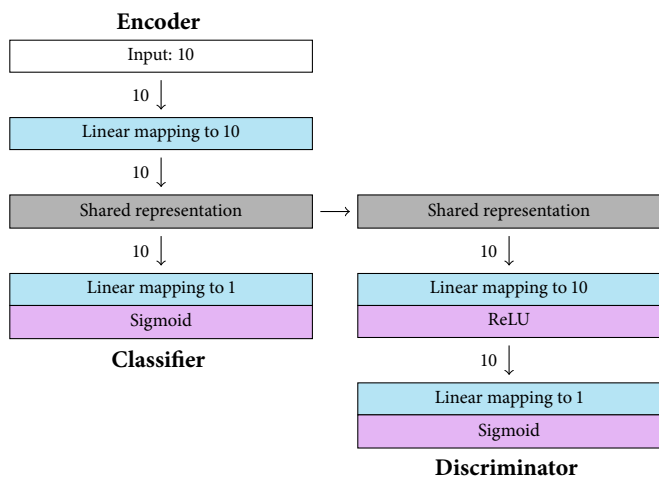


Figure 5.2: Network architecture for the synthetic experiments.

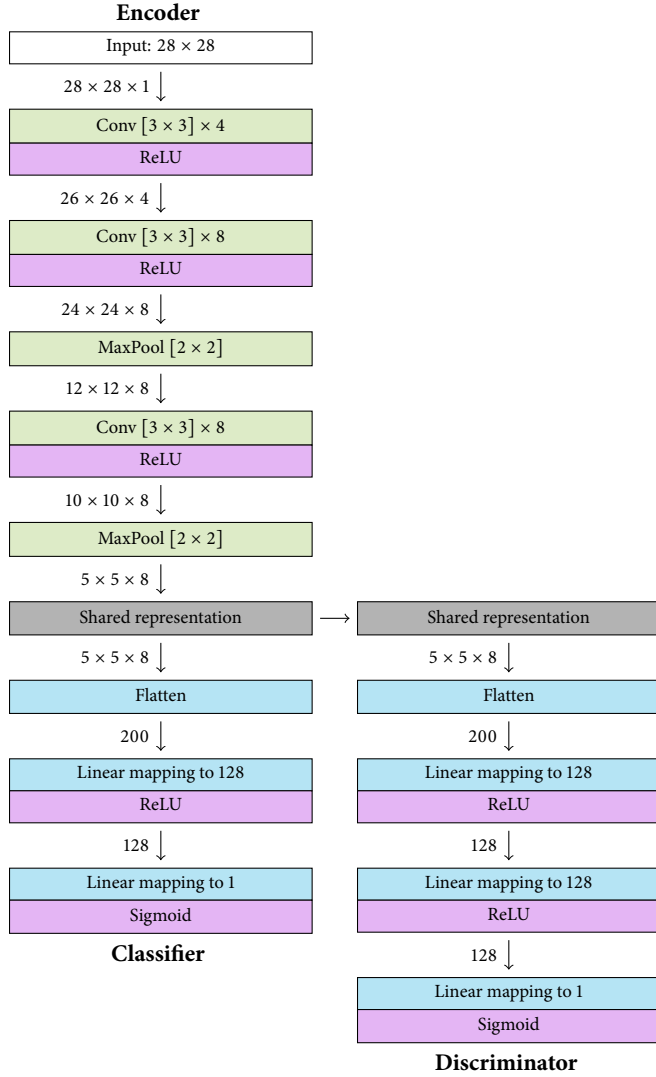


Figure 5.3: MNIST model: Spatial encoder.
 Network architectures for the MNIST experiments. The division between domain-specific encoders and the shared classifier depends on the experiment.

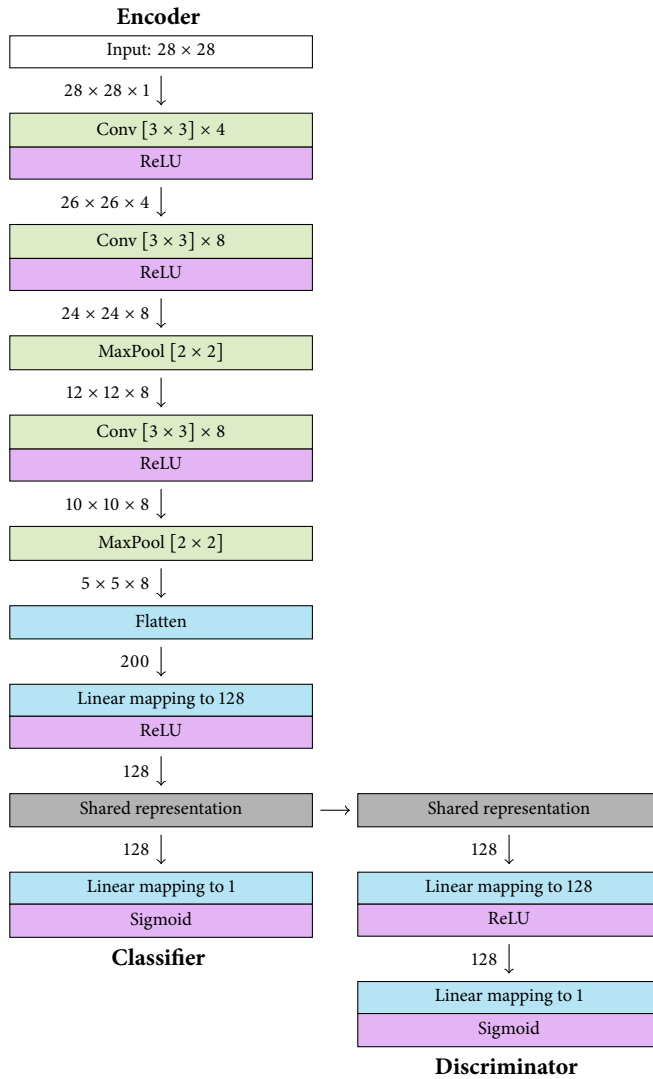


Figure 5.4: MNIST model: Dense encoder.
 Network architectures for the MNIST experiments. The division between domain-specific encoders and the shared classifier depends on the experiment.

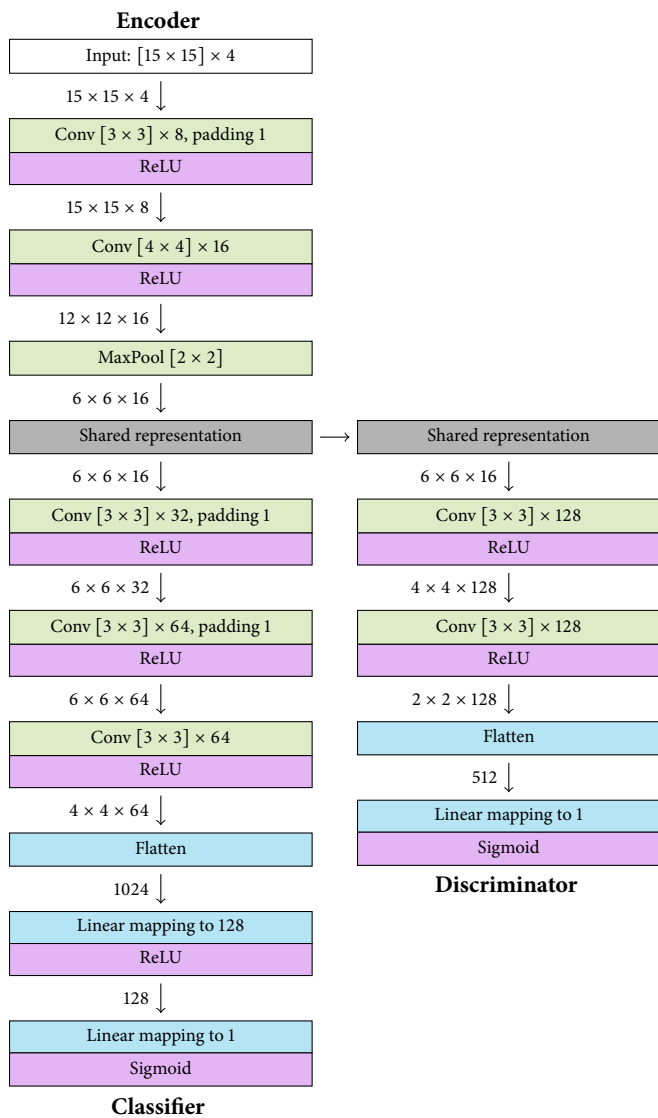


Figure 5.5: BRATS model: spatial encoder, early join.

Network architecture for the BRATS experiments with spatial encoders. The division between domain-specific encoders and the shared classifier depends on the experiment.

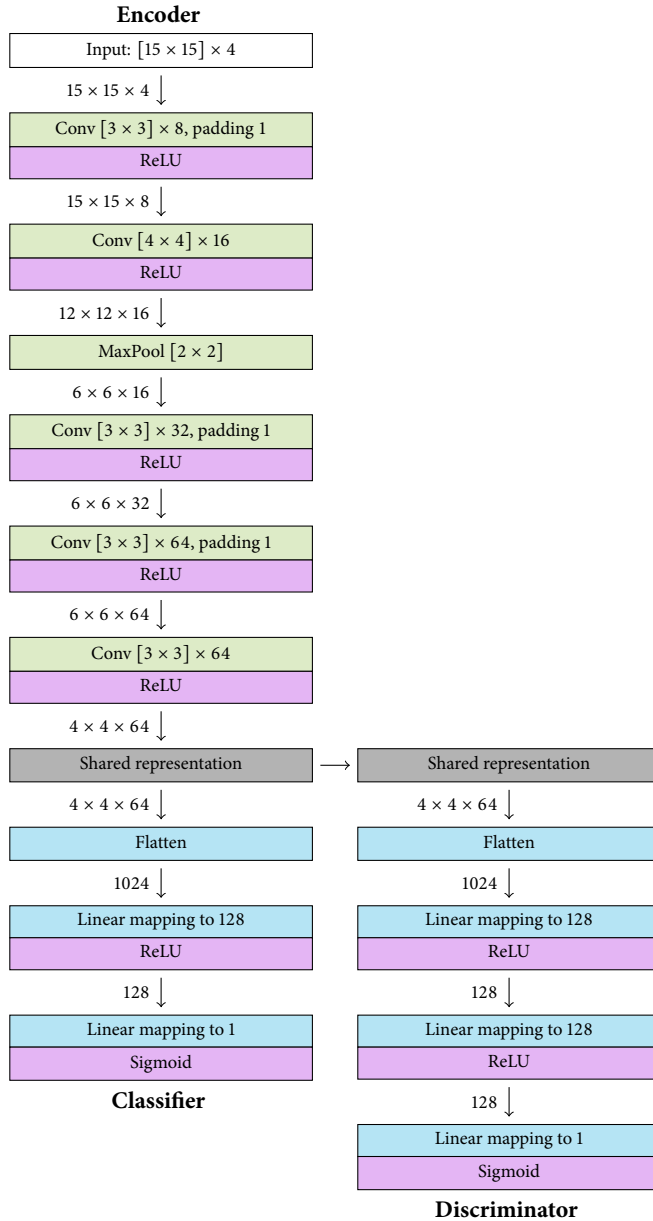


Figure 5.6: BRATS model: spatial encoder, late join.
 Network architecture for the BRATS experiments with spatial encoders.
 The division between domain-specific encoders and the shared classifier
 depends on the experiment.

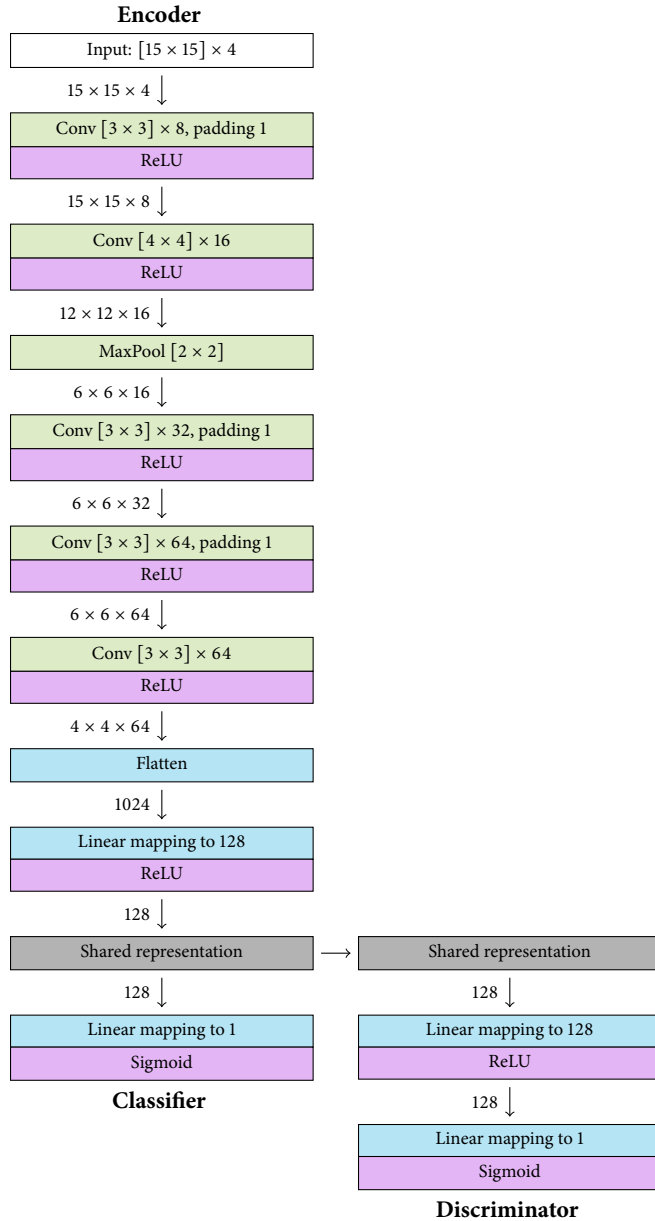


Figure 5.7: BRATS model: dense encoder.

Network architecture for the BRATS experiments with dense encoders. The division between domain-specific encoders and the shared classifier depends on the experiment.

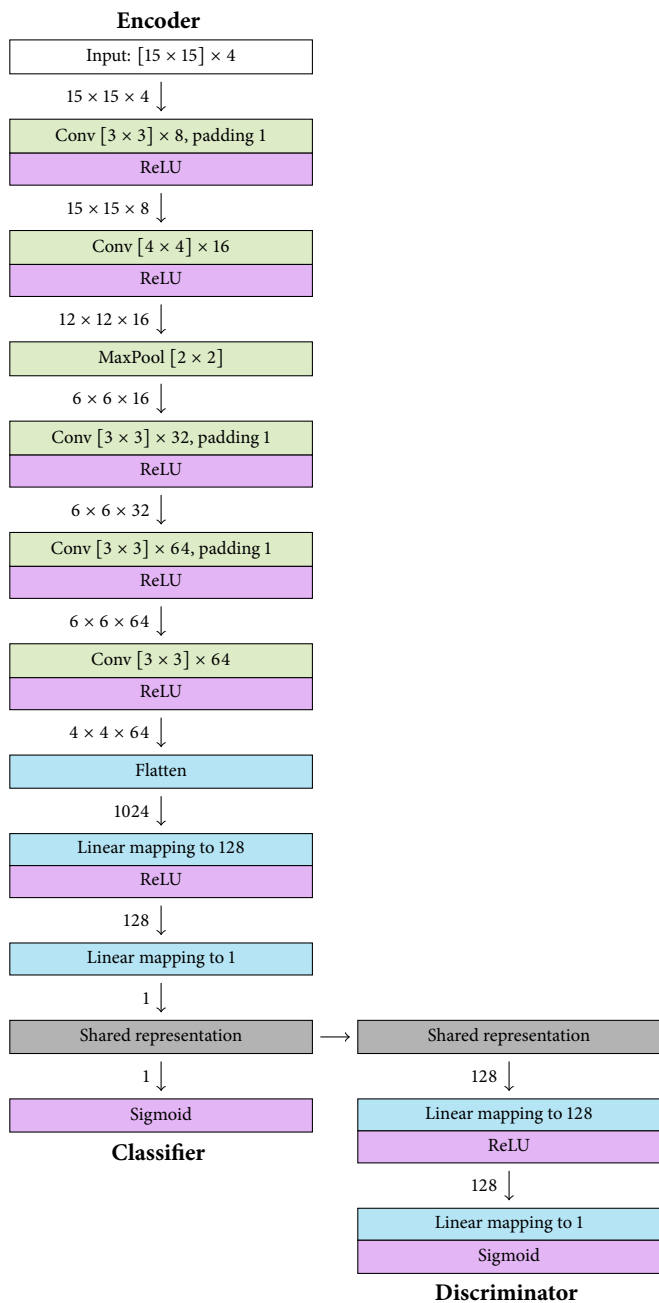


Figure 5.8: BRATS model: posterior join.

Network architecture for the BRATS experiments with dense encoders. The division between domain-specific encoders and the shared classifier depends on the experiment.

6

Discussion

This thesis presented views on representation learning as a method for domain adaptation in medical image analysis. Domain adaptation can improve the performance of models that are trained on data from different domains, such as images from different scanners, hospitals, or patient groups. We evaluated this approach in experiments on multi-modal data, with images made with different MRI settings. In this section, we combine the different perspectives and make some final observations.

This discussion is organized as follows. Section 6.1 summarizes the main findings of each chapter. We then connect the work in this thesis to the general field of domain adaptation for medical imaging: Section 6.2 addresses the main assumptions behind the domain adaptation approach, and Section 6.3 discusses how these assumptions translate to the different domain adaptation methods. Section 6.4 discusses some limitations. For a more general outlook, Section 6.5 discusses the challenges of adapting computer vision methods to the medical domain. Section 6.6 considers possible future work, including the application of domain adaptation in clinical practice. Section 6.7 concludes the thesis.

6.1 Main findings

In Chapter 2, we compared discriminative and generative representation learning on two lung CT analysis tasks. We evaluated features learned by convolutional restricted Boltzmann machines (RBMs) with and without a discriminative learning objective. The RBM-learned features often outperformed

predefined Gaussian filter banks. By observing the performance of a classification model, we found that combining discriminative and generative learning always produced better features than generative learning alone, and often also outperformed pure discriminative learning.

In Chapter 3, we used representation learning for image synthesis in a multi-modal brain segmentation task, with MRI scans acquired with different settings. We also simulated scenarios with missing modalities and used autoencoders and RBMs to synthesize the missing images. Surprisingly, we found that classifiers trained on these synthesized images sometimes performed *better* than classifiers trained on original data. The synthesized data especially helped our linear SVMs, but hardly improved our random forests. We suspect that the synthesis models applied complex, non-linear transformations to the data, which were more useful to linear classifiers than to classifiers that could learn these transformations themselves. The improved performance may be due not to the synthesis of missing modalities, but to the underlying transformation.

In Chapter 4, we considered the task of cross-modality representation learning for two multi-modal MRI datasets. We trained models to translate images between modalities, encouraging them to learn modality-invariant representations that could be used as input for a single, shared classifier. We compared several techniques – modality dropout, feature normalization, and an explicit similarity term – and found that a combination of all three produced the most modality-invariant features. Most models tended to learn a combination of modality-specific and modality-invariant features. While modality-invariant features are important for cross-modality classifiers, we observed that same-modality classifiers may perform better if the representation also contains some modality-specific information.

In Chapter 5, we explored the limits of representation learning in unpaired, unsupervised domain adaptation. We used domain-adversarial learning to train models with domain-specific encoding branches, learning shared representations for cross-domain classification. We found that this relies on similarities between the domains, and suggested four types of similarity that often apply in medical imaging. Although these similarities are sometimes included explicitly as a constraint or learning objective, we found that they can also affect the models implicitly: the architecture and initialization may bias the networks towards learning similar representations for similar inputs.

Several themes appear throughout this thesis. Representation learning for one modality (Chapter 2) is extended to multiple modalities (Chapter 3) and used for cross-modality learning, using domain-specific encoders to learn cross-modality features (Chapters 4 and 5). Each chapter covers a different representation learning objective: generative learning (Chapter 2), image synthesis (Chapter 3), representation similarity on paired features (Chapter 4), and domain-adversarial learning (Chapter 5). All chapters use a hybrid learning objective, either to combine generative and discriminative learning (Chapter 2), or to combine discriminative learning and domain adaptation (Chapters 3 to 5). Finally, the various ingredients are combined in cross-domain classification and domain adaptation experiments (Chapters 4 and 5).

6.2 Assumptions

6.2.1 *Do we need domain adaptation?*

When training a machine learning model with data from multiple sources, we should first determine whether domain adaptation is necessary, and if so, to what extent and in what form.

At one extreme, we could ignore the problem and pretend that all samples come from the same domain. This can be attractive if the domains are similar enough and sufficient training data is available from all domains. In that case, the model would learn features that work for all domains, a generalization problem not unlike handling differences between patients. If required, the cross-domain performance could be further improved using simple tricks such as normalization or data augmentation, for example, by training the model to be invariant to intensity differences.

The other extreme was considered in Chapter 5, where we assumed that the domains were very different and that there was no labelled training data from the target domain. It turns out that this scenario is difficult to solve without introducing additional assumptions about the data.

Most practical applications with data from multiple domains will fall somewhere in between: domains that are different enough to need domain adaptation, but sufficiently similar to make it a success.

6.2.2 *Translating images vs learning shared representations*

If we use domain adaptation, we must decide if we do this at the input level, by translating images between domains, or at the level of an intermediate feature representation, by learning domain-invariant features.

Translating images is conceptually simple and its results can be evaluated visually, but it is computationally inefficient and more difficult than necessary, since most information in an image will not be required for a classification or segmentation task. The approach is also restricted in what it can learn: it usually assumes that translated images are spatially similar, and can only translate information that is visible in both domains. We tried image synthesis with RBMs in Chapter 3, but found that the hidden feature representation was more useful for classification. In recent years image-to-image translation was popularised by the CycleGAN model [85], which uses generative adversarial networks and a cycle-consistency loss to learn the translations.

It can be more effective and efficient to perform domain adaptation at a feature level, by learning a domain-invariant representation. This approach is often implemented with domain-adversarial learning [89], using an adversarial domain discriminator to improve the domain-invariance of the features (Chapter 5), but can also be implemented by directly minimizing the difference between feature representations for paired samples from each domain (Chapter 4). Compared with image translation, translating features can be more efficient, because the model can build on the feature representation that is already used for the main classification or segmentation model, and can ignore irrelevant details. It is more flexible, at least theoretically, and can handle large differences between domains, especially when using a separate encoding path for each domain. It is even possible to combine domain-specific and domain-invariant features if the prediction task requires this. However, the result of feature-level domain adaptation can be hard to interpret and verify.

6.2.3 *How much information is available?*

Domain adaptation requires data from the target domain – or it would be domain generalization – but there is some flexibility in what it needs. Paired samples provide the most information, with known links between samples from the source domain to corresponding samples in the target domain. In

medical imaging, these could be images from the same patient in different scanners, or images from a multi-sequence dataset, for example. Given these paired samples, the domain adaptation method can directly minimize the difference between representations of the same sample in each domain. Paired images are especially informative if they are spatially registered, but this is not always required. We used this approach in Chapters 3 and 4.

If paired samples are not available, domains can be linked in other ways. A popular method is based on distribution matching, e.g., by adversarial domain adaptation [89], which assumes that the distribution of samples is similar in all domains. We examined this method in Chapter 5. This can yield undesired results, because unlabelled distributions can be matched in many ways, and it is difficult to find the correct solution without labels. This problem can be addressed by including additional information, such as labels for a small number of target samples or labels for auxiliary tasks, or by including constraints based on prior knowledge about the domains. This may require additional assumptions about similarities between domains.

6.2.4 *Which assumptions do we make about the domains?*

Like all machine learning methods, domain adaptation methods make assumptions about the data to achieve their results. Domain adaptation methods assume that there are similarities between domains.

For example, many domain adaptation methods in medical imaging implicitly assume that the differences between domains are fairly small. For models based on representation learning, we know this because many of them use a shared encoding path for all domains. This greatly simplifies the domain adaptation task, but restricts the models to features that can be extracted in a similar way for all domains (Chapter 5).

But even in models with independent encoding paths, domain adaptation will exploit existing similarities between the domains. Chapter 5 discussed several types of similarity that often occur in medical images, such as intensity-based or spatial similarities. These similarities are sometimes encoded explicitly in the form of constraints or additional tasks for the domain adaptation model, but we observed that they can also implicitly affect the results, by introducing a bias that leads models to map inputs to similar features. It is useful to be aware of these biases, because they can have positive or negative impact.

6.3 *Methods*

6.3.1 *How should we learn domain-invariant representations?*

In this thesis, we discussed several methods to learn representations of image data. The restricted Boltzmann machines (RBMs) from Chapters 2 and 3 are no longer part of the state-of-the-art in representation learning. These probabilistic models are computationally expensive and can be difficult to train, especially when extended to more than one hidden layer. This is disappointing, because they are fun little models with useful properties, such as the flexibility to accept and sample from incomplete inputs (Chapter 3), and are perhaps less quick to overtrain (Chapter 2). Meanwhile, feed-forward networks, such as CNNs and U-Nets and their many variations, have proven to be more successful. We have used them exclusively in Chapters 4 and 5. They need a supervised learning objective, but are reasonably efficient and very effective.

For cross-domain classification, the methods need to be adapted to make the representations domain-invariant. This can be achieved by changing the architecture, the training procedure, the loss, or a combination. This thesis covered several of these approaches.

In Chapter 4, we used autoencoder-like networks trained with a cross-modality reconstruction loss. Using modality dropout and feature normalization as additional components, the models learned cross-domain representations. This approach required paired samples, but was relatively easy to train. We compared a cross-modality reconstruction loss with optimizing a similarity term directly on the feature representation, and found that a combination of both methods worked best.

In Chapter 5, we used domain adversarial learning [89]. This is a common approach in medical image analysis. The usual architecture consists of an encoder that maps the input to a latent representation, a classifier or other model that makes a prediction based on the representation, and a domain discriminator that tries to predict the domain of the samples based on their feature representation. By using this domain discriminator in an adversarial learning objective, the encoders are encouraged to learn domain-invariant representations, without requiring paired samples.

Both approaches offer a choice between a shared encoder and domain-specific encoders. Most commonly used with adversarial learning is the shared

encoder, which learns a single, domain-invariant feature extractor and is relatively easy to train, but is limited in flexibility. We used modality-specific encoders in Chapters 4 and 5. Each encoder learns a domain-specific mapping to the common feature space, allowing for much larger differences between domains. However, the model is so flexible that it requires paired samples or other assumptions to learn a correct mapping.

A third approach is image-to-image translation, using models such as CycleGAN [85], which learn domain-specific translations and have a flexibility that is most comparable with models with domain-specific encoders. However, they require other assumptions such as spatial similarity between the domains, and like the representation-based methods, can learn incorrect mappings [93].

6.3.2 *Domain-specific or domain-invariant features?*

Most domain adaptation methods assume that both domains provide the same information, in different forms, that can be mapped to a common representation without excessive information loss. Since this assumption never holds completely, it is fairly common to see a decrease in same-domain performance on the original domain. Although domain adaptation is primarily concerned with learning *domain-invariant* features for cross-domain predictions, it may be useful to keep *domain-specific* features as well, in scenarios where the original same-domain performance is also important. For example, users may prefer to have a single model that works well for both domains.

We briefly explored this question in Chapter 4, observing that our models usually learned a combination of modality-specific and modality-invariant features. We found that forcing representations to be too similar removed too many modality-specific features, and would hurt the same-modality classification accuracy. One way to allow models to learn domain-specific features is by adding additional, domain-specific encoding branches. This approach is used in recent works on domain disentanglement (e.g., [127]), for example.

Besides same-domain and cross-domain performance, users may also value cross-domain *consistency*. An important advantage of automated predictions is their objectivity and reproducibility: unlike human annotators, an automated model does not suffer from inter-observer and intra-observer variability. Domain shift can introduce new variability in the form of domain-specific biases, but domain adaptation may help to reduce those.

6.3.3 *How do we know if domain adaptation worked?*

Domain adaptation can be unpredictable, especially in its unsupervised forms. The prediction results on the target domain, such as classification accuracy, give a global indication of its performance, but do not provide any deep insights: models with a similar low accuracy may fail for different reasons.

For practical applications, it is important to know why and how models fail. A lower target accuracy could indicate a relatively harmless, random error that applies equally to all samples, but it could also indicate a more serious, systematic bias in the domain adaptation process. For example, it might be acceptable if a tumor classifier makes small random prediction errors if they are equally distributed over all samples, but not acceptable if the classifier consistently misidentifies two tumor subtypes.

Observing the inner workings of a deep learning-based domain adaptation model is not easy. In Chapter 5, we discussed domain adaptation as a two-step process: first, the model must identify clusters in the data, then, it must link these clusters between domains. The classification accuracy on the target domain only measures the performance of both steps together. We applied several metrics to measure the correctness of the clustering and linking steps, but these metrics required detailed label information for the target samples that is unlikely to be available in real applications.

It might seem that image-to-image translation, with its visual output and the cycle-consistency, would be easier to check for undesired results. Indeed, the convolutional components may ensure that the global spatial structures are preserved. However, the underlying domain adaptation problems are similar to those in feature-level adaptation: without paired samples, domains must be matched at the distribution level, introducing ambiguities that are difficult to resolve. This can lead to systematic errors and incorrect translations (see, for example, the CycleGAN models that hallucinate brain tumors when translating MRI scans of healthy subjects, as shown by Cohen, Luck, and Honari [93]).

How best to evaluate domain adaptation remains an open research question. For some applications, observing the final performance on the target set may be sufficient, while for others the requirements may be stricter. In general, supervised domain adaptation may be considered easier to verify and therefore ‘safer’ than unsupervised domain adaptation, but unexpected problems can also arise in the supervised case.

6.4 Limitations

While the limitations of individual experiments are discussed in their respective chapters, there are two points that need to be addressed here.

6.4.1 Are imaging modalities domains?

In much of this thesis we looked at *modalities* – or more precisely: MRI sequences¹ – rather than *domains*. Multi-modal image analysis is interesting in itself (e.g., when imputing missing data, as in Chapter 3), but it also provides a useful playground for domain adaptation. Multi-modal data is relatively easy to obtain: there are many public, multi-modal datasets, and the datasets usually include images for each subject in all modalities. This makes it easy to train and evaluate cross-modality translation models, as in Chapters 3 and 4.

In many ways, multi-modal image analysis may be more difficult than ‘real’ domain adaptation. When using data from different scanners or slightly different imaging protocols, images from each domain may have a different appearance, but they often contain similar information. In multi-modal datasets, the differences between domains can be much larger: images from different modalities may have a different appearance, but often also contain different information. Most multi-modal datasets are not created for domain adaptation enthusiasts, but because the modalities provide complementary information that is needed for the classification or segmentation task. This modality-specific information can make domain adaptation much more difficult, because a modality-invariant representation may lack some important details that are required to achieve a good performance.

Nevertheless, it would be useful to compare our methods in a ‘real’ cross-domain setting as well. The closest approximation in this thesis can be found in Chapter 4, which used knee cartilage images acquired with fat-suppressed and non-fat-suppressed MRI protocols. Although the images in this dataset were paired and registered, the protocols provide fairly similar information, and in most practical applications only one of them would be acquired.

1. Early on in my PhD, a friendly radiologist warned a colleague that “when Gijs says ‘modality’, he actually means ‘sequence’”. This is a long-standing source of confusion between the technical and clinical sides of medical imaging. In keeping with my more technical background, I use “modality” throughout this thesis. This conforms with datasets such as BRATS, a “multi-modal” brain tumor segmentation dataset that only includes multi-sequence MRI.

For technical studies of domain adaptation algorithms, like this thesis, cross-modality image analysis is a useful proxy for other domain adaptation tasks. While slightly less realistic, the multi-modal tasks can be more challenging and the available data enables an in-depth analysis of the results.

6.4.2 *Generalizing to other, larger models*

Over the last few years, improvements in GPU hardware and an increased availability of data made it possible to use larger and larger machine learning models. In comparison with contemporary models, the ones in this thesis are smaller than usual: they have fewer layers, fewer parameters, and sometimes work on small patches instead of full images. Using smaller models makes the experiments more efficient, and allows for more extensive comparisons of hyperparameters and architectures. We expect that the behaviour of the representation learning and domain adaptations will be comparable. Discriminative learning may improve performance (Chapter 2), modality synthesis may have unexpected effects (Chapter 3), modality-invariant features improve cross-domain performance (Chapter 4), and unpaired, unsupervised domain adaptation is most likely still not guaranteed (Chapter 5).

6.5 *Is medical image analysis just computer vision?*

Computer vision is a great inspiration for medical image analysis, and many techniques, including those in this thesis, were developed for natural images before being applied to X-ray, CT, or MRI. Much of this success can be linked to representation learning: a method that learns its own features from the data is much easier to transfer than a method that requires hand-made feature descriptors for each new task.

However, besides the similarities between computer vision and medical imaging – most importantly, both have images that can be analyzed with convolutional neural networks – there are also substantial differences. Understanding these differences is important to obtain good and reliable results. A thesis on domain adaptation in medical imaging would be incomplete without a brief discussion of this even greater transfer learning problem.

6.5.1 *Different data with different characteristics*

Many differences between medical and natural images derive from the type of variation in the data. While natural images can have variable lighting and perspective, medical images are usually acquired in more controlled conditions. Compared with the tigers in the forests of ImageNet, the primates in medical images are pictured in a much more consistent and standardized way.

This standardization is needed because medical imaging tasks are usually more subtle than the computer vision equivalents. Finding a tiger in a forest is difficult, but mainly because of the variety and less because the difference between tiger and not-tiger is very small. Medical imaging tasks, such as classifying brain tumors, require more finely tuned thresholds.

Paradoxically, it is the same standardization that makes medical imaging applications very sensitive to changes in scanners, protocols, modalities, centers, populations, or one of the many other parameters that affect the appearance of the images. The robustness that comes natural to computer vision tasks is much harder to achieve in medical imaging.

The subtlety and sensitivity make medical imaging an excellent target for domain adaptation, because models that rely on standardized images from a specific dataset will not directly work on images with different properties.

6.5.2 *Adapting methods from computer vision*

Methods from computer vision need to be adapted to work on medical images. While standard approaches may work reasonably well, performance may be improved by using application-specific preprocessing and architectures.

For preprocessing, it is important to preserve the image characteristics such as scale and intensity. While natural images may contain objects at many different orientations and scales, medical images often have a known pixel or voxel size. Instead of using the common computer vision approach to squash all images to a fixed size – such as 256×256 pixels for ImageNet – preserving the resolution can make it easier to analyze medical images: rather than learning to detect anatomical structures at many different scales, the model can focus on one specific scale. This is also useful if absolute measurements provide discriminative information, such as brain atrophy in Alzheimer classification. In some cases, it may also help to register images to a fixed reference space.

Similarly, intensity information in medical images may be more standardized than in natural images, which may have to deal with lighting, shadows, white balance et cetera. For medical applications, using very aggressive normalization methods can remove discriminative intensity information. In some cases, using preset windowing settings directly from the DICOM headers might already provide sufficient standardization.

Medical imaging and computer vision might also benefit from slightly different network architectures. Much of the capacity in large computer vision networks such as ResNet or DenseNet is used to handle the variation in the data, which can only be modelled with a large number of parameters, and large training sets. In comparison, medical image analysis receives smaller sets of standardized images and relies on finely tuned thresholds, which may be easier to learn with a smaller network with fewer parameters.

6.5.3 *Pretraining on ImageNet*

One of the more intriguing aspects of medical deep learning is the use of pretrained features from ImageNet and other computer vision datasets. This has proven to be a very effective way to initialize the weights of CNNs, often outperforming models trained from scratch.

The reusability of pretrained weights is usually explained with the observation that early layers in a CNN learn generic, Gaussian-filter-like feature descriptors, which can be reused for other tasks and images (see also our feature learning experiments in Chapter 2). Since ImageNet is much larger than most medical datasets, it is easier to learn a good set of features.

However, pretraining a medical imaging model on natural images is counterintuitive. It involves a clumsy conversion of grayscale medical images to pseudo-RGB, even though the pretrained color features will be useless. Moreover, medical images do not *look* like natural images. It would seem better to pretrain models on medical datasets instead.

Although many works use pretrained weights (in fact, a search for “transfer learning” in medical imaging lists many papers that do just that), there has been less research into the idea itself. There are some interesting findings. For example, Raghu et al. [128] observed that for some problems, simple, lightweight models trained from scratch can have similar performance to large networks with ImageNet pretraining.

Pretraining on medical images may yield different features than pretraining on natural images. For example, natural images may have stronger edges than most medical images. Wen et al. [129] suggest that pretraining on medical images may be more effective for classification than for segmentation, since medical images are visually homogeneous and lack morphological information. Along similar lines, but with opposing conclusions, Hosseinzadeh Taher et al. [130] report that segmentation and classification tasks may benefit from pretraining with different natural imaging datasets, depending on their need for global or more fine-grained features. Earlier, Schlegl, Ofner, and Langs [131] found that pretraining on a computer vision dataset performed better than pretraining on data from a different medical imaging domain.

For obvious reasons, reusing features from ImageNet is not common for 3D networks. Chen, Ma, and Zheng [132] provide some models for 3D medical image analysis, pretrained on public challenge data. Alternatively, models can be pretrained with self-supervised learning (e.g., Taleb et al. [133]).

6.5.4 *Technical challenges*

From a technical perspective, there are a few important differences between medical imaging and general computer vision. While many computer vision datasets include a large number of relatively small images, most medical imaging datasets consist of a smaller number of large images. This has practical consequences, especially for images with a high resolution, such as histopathology slides, or images with a high number of dimensions, such as 3D or 3D+t MRI. Large images can be difficult to fit in the GPU memory, requiring smaller minibatch sizes, smaller models, or patch-based training.

6.5.5 *Methodological challenges*

From a methodological perspective, medical imaging also provides several unique challenges. First, medical imaging solves different tasks, with a stronger focus on segmentation, detection, and quantification than in general computer vision. This requires different network architectures and training procedures, such as U-Net and multi-task learning. Second, the availability of smaller datasets requires techniques such as data augmentation, domain adaptation, and semi-supervised learning. Third, some applications may pose specific problems, such as detection tasks that look for small structures in large images,

which may require attention-based methods. Fourth, the medical context may pose questions about reliability, uncertainty prediction, explainability, fairness, privacy, and data sharing.

6.6 *Looking forward*

At the end of this thesis, there are a few questions left.

6.6.1 *Is domain adaptation still useful if you have a lot of data?*

Medical imaging datasets have traditionally been small, but the availability of data is rapidly increasing. Perhaps encouraged by the success of deep learning, more and more data is collected within hospitals to facilitate medical imaging research. Some of this data is also shared in standardized ways, in projects such as ADNI² and the Osteoarthritis Initiative (OAI [2], Chapter 4). In the Netherlands, initiatives such as Health-RI³ and BBMRI⁴ aim to improve data sharing among hospitals, and similar projects have started elsewhere.

This influx of data is useful for domain adaptation research, but it might also reduce the need for it. While there is still variation between scans from different scanners or hospitals, networks that are trained on large and heterogeneous datasets can model this variation without additional changes.

On the other hand, since large datasets are usually created by combining smaller datasets from multiple centers, the data might include unwanted biases. For example, if populations in different hospitals have different characteristics, the prediction outcomes might become correlated with properties such as scanner models and image quality. Domain adaptation methods may be used to remove these biases from the data.

Finally, semi-supervised domain adaptation methods may be useful if the new data is partially unlabelled. Since it is generally easier to collect images than to produce annotations, not all new datasets may have the correct application-specific labels. Semi-supervised methods could learn from all available data by combining discriminative learning for the labelled samples with unsupervised domain adaptation for the unlabelled target samples.

2. <http://adni.loni.usc.edu/>

3. <https://www.health-ri.nl/>

4. <https://www.bbmri.nl/>

6.6.2 *How do we know if domain adaptation was successful?*

Domain adaptation can sometimes produce unexpected and unwanted results (Chapter 5 and Section 6.3.3), but these problems may be difficult to detect. Comparing the performance on a validation set with ground-truth labels is not sufficient: a low classification accuracy on the target domain could be due to general mistakes in the classifier, but it could also point to more harmful, biased errors. Segmentations are easier to check, but may also include hidden biases that are difficult to identify with a visual inspection alone.

More advanced methods are needed to evaluate and improve the reliability of domain adaptation. We suggested some metrics in Chapter 5, but these require detailed knowledge about the target data. Practical applications would require methods that require less detailed knowledge, but still give insights in the domain adaptation process. These methods could also be useful to monitor the performance of algorithms after development, as a form of “quality assurance” to maintain the performance of models in clinical use.

6.6.3 *Should domain adaptation use more prior knowledge?*

Domain adversarial learning and the other representation-based methods studied in this thesis make few assumptions about the differences between the domains. While this can be very effective, it may also be useful to consider a more principled approach. In many medical imaging applications, the relation between domains is not completely random: for example, MRI scanners all work in similar ways, and the differences between them are limited and have a physical explanation. This prior knowledge could be used in a more model-based domain adaptation approach, by limiting the optimization to transformations that are physically likely.

Prior knowledge can also be included in the learning process as regularization, through explicit constraints or in additional learning tasks. In medical imaging, these constraints can be based on knowledge about anatomical structures, intensity information, or other known domain invariances. Instead of relying on the model to find these similarities automatically, they may be included in the learning objective to make the process more predictable. Finding the right invariances and ways to include them in the domain adaptation procedure is an interesting direction for future research.

Domain adversarial learning is just one method to learn domain-invariant representations. It is a general, somewhat undirected way that can be difficult to train (see, e.g., [134]). There is a large number of alternative approaches that could be used instead or in combination, with different advantages and disadvantages (e.g., [135, 136]).

6.6.4 *How does this work in clinical practice?*

Despite the broad range of research on domain adaptation for medical imaging [84] and the promise to reduce labelling cost while improving performance, it is hard to find reports on implementations in clinical practice. Like this thesis, most works present experiments on public and proprietary research datasets.

This may have several reasons. Developers and users of clinical software may prefer models that work everywhere, i.e., models that are generalizable, over models that need to be adapted to each new use case. As is clear from this thesis and other work on this topic, domain adaptation is far from trivial. It requires training data from the target domain, it requires retraining of complex deep learning models, and it may require technical expertise to inspect and validate the results. Even then, there is a real risk that it will not end well.

Perhaps the most likely use case for domain adaptation are large epidemiological studies. Although studies like ADNI or the Rotterdam Study use carefully designed protocols to make images as similar as possible, there will still be differences between images acquired in different centers, using different scanners, or using upgraded protocols. For these large datasets in a research context, domain adaptation is both useful and feasible (see, e.g., [137]).

For application in software for clinical practice, the models will need to adapt to target data based on feedback from clinical users, preferably without interference from a machine learning expert. It may be interesting to investigate methods such as online learning and reinforcement learning to do this without retraining the original model completely. Models that adapt gradually to new inputs are computationally attractive, but may also produce more predictable and reliable results.

From a regulatory perspective, the adaptability of domain adaptation models may be a disadvantage. Regulatory approval is a relatively static process: a model is validated, approved, and then applied in clinical practice. This leaves little room for a model that continuously updates based on new data.

6.7 *Wrapping up*

Domain adaptation is an attractive idea: it allows you to combine data from multiple sources and train a single prediction model that works for all of them. This is especially useful for medical imaging, where datasets are small, annotations are time-consuming, and methods are sensitive to the small differences between inputs from different domains. Domain adaptation can improve performance and reduce annotation cost.

In this thesis, we used representation learning to map inputs from different domains to a common feature space, which means that a single classifier or segmentation model can be used for all domains. Given some labelled or paired samples from the target domain, this method can give reliable and useful cross-domain results. With only unpaired and unlabelled samples, the outcome depends on existing similarities and biases in the data.

Domain adaptation is very popular in medical imaging research, but far less common in clinical practice. This may be a matter of time – with the growing reliance on machine learning-based image analysis and the increasing availability of large but heterogeneous datasets, the big break-through of domain adaptation might very well be imminent. The findings from this thesis can help to improve its results.

On the other hand, domain adaptation looks simpler than it is, and there are several problems that must be solved for each new application: how domain adaptation should be included in the model, how flexible the adaptation should be, which assumptions it should make, and how the results should be evaluated. This thesis presented answers, but also found new problems. While domain adaptation improves the generalizability of the models, its own generalizability leaves something to be desired.

Summary

Machine learning is an essential tool for medical image analysis, but the models it creates do not always generalize well to new domains. For example, a model that works well on data from one type of scanner might work less well on data from another, because the models are sensitive to subtle differences in the images. Scanner-specific models may perform better, but training them requires new labelled training data for each domain, which may be time-consuming and expensive to obtain. In these cases, domain adaptation methods may help to transfer knowledge between domains.

This thesis explores domain adaptation using deep learning, by learning shared representations that are similar across domains. Using these shared representations, data from different domains can be combined and analysed with a single model. This may reduce labelling cost when applying existing models to data from new sources, but may also improve performance on datasets that combine data from multiple sources.

Chapter 2 investigates representation learning for lung tissue classification, using restricted Boltzmann machines (RBMs) to learn high-level representations that can be used as input for a classification model. The standard unsupervised learning objective can be extended with a supervised, discriminative learning objective, which helps the model to extract features that are useful for classification. We evaluate this model on two tasks in lung CT images: airway detection and tissue classification for interstitial lung disease. We compare the RBM-learned features with predefined feature banks, finding that RBM-learned features can outperform the predefined features, especially if the RBM is trained with a hybrid generative-discriminative learning objective.

Chapter 3 investigates RBMs and feed-forward neural networks for cross-modality image synthesis, as a method to impute missing modalities in a multi-modality classification setting. After training networks to reconstruct different modalities from a shared representation, we impute missing MRI

sequences in a brain tumor segmentation task. The design of the RBMs makes a single model sufficient for all cross-modality combinations, while the feed-forward networks require a separate synthesis model for each input-output combination. We compare the reconstructed images in classification experiments with linear support vector machines (SVMs) and random forests. Image synthesis improves the results if some images are missing at test time. We observe the largest improvement for the linear SVMs, with a smaller improvement for the random forests, which suggests that the nonlinear nature of the synthesis models might also contribute to the improved performance.

Chapter 4 learns cross-modality representations from multi-modal images, using autoencoder-like architectures to learn a shared representation for all modalities, which is then used as input for a cross-modal classification model. We compare four strategies to improve the quality of these cross-modal representations: cross-modal reconstruction, an explicit similarity loss, per-feature normalization, and modality dropout. We investigate whether these methods learn modality-specific or shared features. In experiments with knee cartilage and brain tumor segmentation, both on multi-modal MRI data, we find that a combination of all methods produces the best cross-modal features.

Chapter 5 investigates the possibilities and limitations of unsupervised, unpaired domain adaptation. While this approach is very flexible and requires no labelled data from the target domain, it is also likely to produce incorrect results. We explore the problems that may occur, and discuss whether existing similarities between domains may bias the model towards the correct solution. We present four assumptions that often hold for medical images, such as the assumption that images from two domains have a similar spatial structure or a similar intensity distribution. We conclude that while unsupervised, unpaired domain adaptation makes few explicit assumptions, these implicit biases may have a strong influence on the results.

Chapter 6 discusses the different perspectives on cross-modality medical image analysis and domain adaptation that are presented in this thesis. We compare the assumptions and choices that can be made, and discuss how they are reflected in the methods. We discuss the main limitations of the work, and look at the similarities and differences between computer vision and medical imaging. We end with a number of open questions about the methodological and clinical aspects of domain adaptation in medical image analysis.

Samenvatting

Machine learning is in de afgelopen jaren een essentieel hulpmiddel geworden voor medische beeldanalyse. Door verbeterde methodes, snellere hardware en een toegenomen beschikbaarheid van medische beelden is het mogelijk geworden om steeds complexere modellen te gebruiken voor een breed scala aan classificatie- en segmentatietaken. Automatische beeldanalyse bespaart tijd en is soms ook beter dan handmatige annotatie door menselijke experts, bijvoorbeeld op punten als precisie, objectiviteit en reproduceerbaarheid.

Machine learning-methodes leren van voorbeelden. In medische beeldverwerking zijn dit vaak scans die zijn geannoteerd door radiologen, bijvoorbeeld met een label voor de aanwezigheid van een bepaalde ziekte, of met een segmentatiemasker dat de locatie en contouren van een anatomische structuur aangeeft. Soms worden ook andere bronnen gebruikt, zoals resultaten van histopathologisch onderzoek of klinische gegevens over het ziekteverloop.

De gelabelde voorbeelden worden gebruikt om een model te trainen dat de labels kan voorspellen voor nieuwe beelden, bijvoorbeeld van nieuwe patiënten voor wie nog geen handmatig label beschikbaar is. Het model doet deze voorspellingen door te kijken naar kenmerken en patronen in de beelden, zoals de pixel-intensiteit of textuur. Op basis van de gelabelde voorbeelden leert het model welke kenmerken en patronen belangrijk zijn en wat ze betekenen.

De modellen die op deze manier worden getraind geven meestal goede resultaten voor scans die lijken op de voorbeelden waarmee ze zijn getraind, maar ze werken soms minder goed voor nieuwe data, zoals scans van een ander merk of type scanner. Doordat de modellen zijn getraind op een specifieke set voorbeelden, leren ze welke kenmerken voor dat type data belangrijk zijn. Dat werkt goed als de nieuwe scans lijken op de oorspronkelijke voorbeelden, maar zorgt voor problemen als de nieuwe scans er anders uitzien – bijvoorbeeld omdat ze zijn gemaakt met een ander model scanner, met iets andere

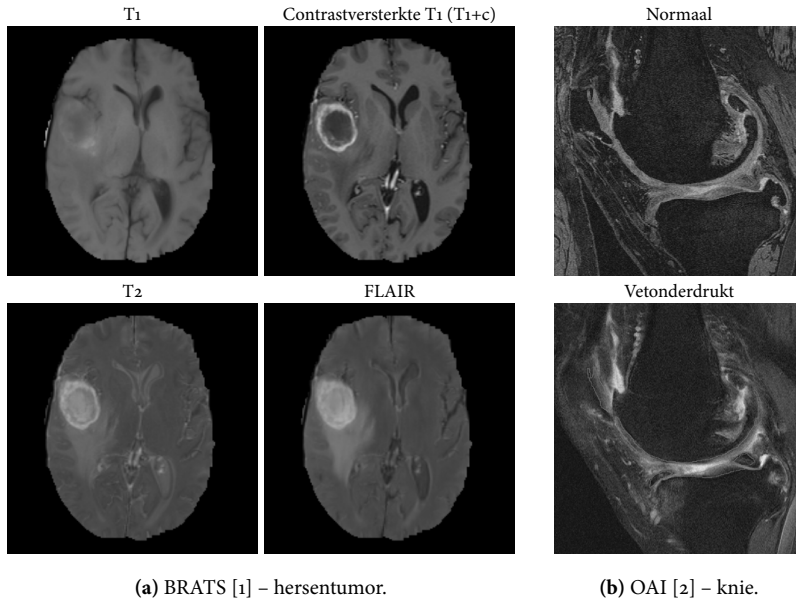
instellingen, of in een ander ziekenhuis. Als het model kijkt naar kenmerken die in de nieuwe scans ontbreken, of die daar een andere betekenis hebben, zal het model voor die nieuwe data minder goed werken.

Dit probleem heet *domain shift*: een model dat is getraind op data van één domein (het brondomein, bijvoorbeeld scanner A) moet worden toegepast op data van een ander domein (het doeldomein, bijvoorbeeld scanner B) waarin de scans er anders uitzien. Figuur 1 toont hiervan twee voorbeelden: MRI-scans van dezelfde patiënt die zijn gemaakt met verschillende scanner-instellingen zien er duidelijk verschillend uit. Door deze verschillen kan het zijn dat een model dat is getraind voor één soort scans niet automatisch even goed werkt voor scans van de andere soorten.

Domain shift is een veelvoorkomend probleem in medische beeldverwerking, omdat het tijdrovend, duur of onpraktisch kan zijn om voor ieder nieuw domein nieuwe data te verzamelen en annoteren. Bij medisch onderzoek is het vaak noodzakelijk om datasets opnieuw te gebruiken of om data van verschillende bronnen te combineren om zo een voldoende grote dataset te krijgen. Ook voor klinische toepassingen, bijvoorbeeld voor bedrijven die medische software ontwikkelen, is het handig als hetzelfde model kan worden gebruikt voor beelden van verschillende scanners, gemaakt met verschillende instellingen of in verschillende ziekenhuizen.

Domain shift kan worden bestreden met *domain adaptation*, een groep machine learning-methodes die een model dat is getraind op data van één domein kunnen aanpassen zodat het goed werkt voor data van een ander domein. Dit kan op grofweg twee manieren: door één model te maken dat ongevoelig is voor de verschillen tussen de domeinen, en daardoor goed werkt voor beide domeinen; of door een nieuw, domein-specifiek model te maken dat is afgeleid van het oorspronkelijke model, maar speciaal is afgestemd op de eigenschappen van het doeldomein.

In dit proefschrift combineren we domain adaptation met deep learning, een populaire techniek die wordt gebruikt in de meeste huidige medische beeldverwerkingsmodellen. Deep learning is gebaseerd op *representation learning*, een methode waarbij een neurale netwerk met meerdere lagen wordt gebruikt om nieuwe, abstracte representaties van de data te leren. Voor afbeeldingen, zoals medische scans, zijn dit meestal convolutionele neurale netwerken (CNNs).



Figuur 1: Hoe een MRI-scan eruitziet hangt onder andere af van de instellingen van de scanner. Links: vier scans van dezelfde hersentumor, gemaakt met verschillende MRI-sequenties. Rechts: twee verschillende scans van dezelfde knie, gemaakt voor kraakbeensegmentatie (groene contouren). Een model dat is getraind op één type beeld (zoals T1+c) werkt waarschijnlijk minder goed voor een ander type (zoals FLAIR), omdat sommige kenmerken daar ontbreken of een verschillende betekenis hebben. (Illustratie uit Hoofdstuk 4.)

Zoals ieder deep learning-model bestaat een CNN uit een aantal lagen waarin de invoer stap voor stap wordt ontleed. De eerste laag herkent basale patronen, zoals lijnen en hoeken. Dit levert een tussenliggende representatie op die beschrijft waar in het beeld welk patroon voorkomt. De tweede laag combineert de eenvoudige patronen tot iets ingewikkelder patronen, zoals simpele vormen en structuren, en produceert een nieuwe representatie die de scan op een iets abstracter niveau beschrijft. De volgende lagen in het netwerk leren steeds complexere, abstractere representaties, tot aan de laatste laag die zo abstract is dat daarmee het label of de segmentatie kan worden voorspeld. Welke patronen belangrijk zijn en hoe ze moeten worden gecombineerd tot nuttige nieuwe representaties, leert het model van de gelabelde voorbeelden.

Representation learning biedt de ideale mogelijkheid voor domain adaptation: als je toch een nieuwe representatie van de data leert, dan kun je net zo goed een representatie leren die beelden uit verschillende domeinen een vergelijkbare representatie geeft. Als de beelden van verschillende domeinen op dezelfde manier kunnen worden gerepresenteerd, kunnen ze vervolgens door één gemeenschappelijk model worden beoordeeld.

Er zijn twee manieren om domain adaptation toe te voegen aan een neurale netwerk. De eerste aanpak gebruikt een gemeenschappelijke encoder: een aantal gemeenschappelijke lagen aan het begin van het netwerk, die beelden van alle domeinen op dezelfde manier omzetten naar een gemeenschappelijke representatie. Dit werkt het best als de domeinen al redelijk op elkaar lijken. De tweede aanpak gebruikt voor ieder domein een aparte encoder: voor ieder domein zijn de eerste lagen van het netwerk verschillend, zodat de beelden op een domein-specifieke manier kunnen worden behandeld. Deze aanpak is heel flexibel, omdat voor ieder domein de beste transformatie kan worden gebruikt, maar is meestal ook lastiger te leren.

Domain adaptation-methodes hebben informatie nodig over de verschillen en overeenkomsten tussen de domeinen. Sommige methodes hebben genoeg aan voorbeelden uit het brondomein, en maken wat aannames over de verschillen met het doeldomein. De meeste methodes gebruiken echter ook voorbeelden van het doeldomein. Dit kunnen gepaarde voorbeelden zijn, zoals scans van dezelfde patiënt in verschillende scanners. Gepaarde scans geven heel veel informatie over de verbanden tussen de domeinen, maar zijn vaak moeilijk te krijgen. Sommige methodes gebruiken daarom ongepaarde maar gelabelde voorbeelden uit het doeldomein. Deze zijn eenvoudiger te vinden, maar bieden minder aanknopingspunten om de domeinen te combineren. Ten slotte zijn er methodes die alleen ongepaarde, ongelabelde voorbeelden uit het doeldomein gebruiken: die voorbeelden zijn het makkelijkst te krijgen, maar bieden nog minder informatie.

Een model voor domain adaptation heeft twee taken: de hoofdtaak, zoals het classificeren of segmenteren van nieuwe scans, en het leren van een gemeenschappelijke representatie. De hoofdtaak kan worden geleerd met een standaard supervised learning objective, op basis van gelabelde data van het brondomein. De gemeenschappelijke representatie kan worden geleerd met een extra domain adaptation objective, dat samen met de hoofdtaak

wordt geoptimaliseerd. Veelvoorkomende keuzes voor dit extra objective zijn representation similarity, waarbij het representatieverschil tussen gepaarde voorbeelden in beide domeinen zo klein mogelijk wordt gemaakt, en feature distribution similarity, waarvoor geen gepaarde data nodig is, maar waarbij een methode zoals domain adversarial learning wordt gebruikt om de verdeling van de representaties voor beide domeinen gelijk te maken.

In dit proefschrift onderzoeken we de combinatie van domain adaptation en deep learning, waarbij een gemeenschappelijke representatie wordt geleerd voor verschillende domeinen. Op basis van deze gemeenschappelijke representatie kan data van verschillende domeinen worden gecombineerd en door één model worden geanalyseerd. Dit is efficiënter omdat er minder labels nodig zijn om bestaande modellen aan te passen voor nieuwe data, en kan daarnaast worden gebruikt om data van verschillende bronnen te combineren.

Hoofdstuk 2 onderzoekt representation learning voor longweefselclassificatie, waarbij restricted Boltzmann machines (RBMs) worden gebruikt om representaties te leren als input voor een classificatiemodel. Het standaard unsupervised learning objective van de RBMs kan worden uitgebreid met een supervised, discriminative objective, dat het model helpt om representaties te leren die nuttig zijn voor een classificatiemodel. We evalueren dit model voor twee taken met CT-longscans: detectie van luchtwegen en weefselclassificatie voor interstitiële longziekten. We vergelijken de features geleerd door de RBM met vooraf gedefinieerde features uit standaard feature banks. RBM-geleerde features blijken beter te presteren dan vooraf gedefinieerde features, in het bijzonder als de RBM is getraind met een hybride learning objective dat generatief en discriminatief leren combineert.

Hoofdstuk 3 onderzoekt RBMs en feed-forward neurale netwerken voor beeldsynthese bij het herkennen van hersentumoren. Bij de diagnose van hersentumoren worden verschillende MRI-beelden vergeleken (Figuur 1), maar in de praktijk kunnen sommige beelden ontbreken. Om de ontbrekende beelden aan te vullen maken we modellen die beelden kunnen vertalen: we trainen modellen die de verschillende types vertalen naar één gemeenschappelijke representatie, en modellen die die representatie weer terugvertalen naar de verschillende types. We evalueren de synthetische beelden in experimenten met lineaire support vector machines (SVMs) en random forests. Beeldsynthese

verbetert de resultaten als het model moet worden toegepast op incomplete data waarin sommige modaliteiten ontbreken. We zien daarbij de grootste verbetering voor de lineaire SVMs en een kleinere verbetering voor de random forests, wat suggereert dat ook de niet-lineaire aard van het synthesemodel zou kunnen bijdragen aan de verbeterde classificatie.

Hoofdstuk 4 leert gemeenschappelijke representaties voor verschillende verschillende soorten MRI-scans. We gebruiken autoencoder-achtige architecturen om een gemeenschappelijke representatie te leren en gebruiken die vervolgens voor een classificatiemodel. We vergelijken vier strategieën om de kwaliteit van deze representaties te verbeteren: door de representatie terug te vertalen naar de verschillende beelden, door een expliciete similarity loss, door normalisatie van individuele features, en door modality dropout. We onderzoeken of deze methodes domein-specifieke of gemeenschappelijke features leren. In segmentatie-experimenten met kniekraakbeen en herstentumoren, beide op multi-modale MRI-scans, zien we dat een combinatie van alle methodes de beste gemeenschappelijke features oplevert.

Hoofdstuk 5 onderzoekt de mogelijkheden en beperkingen van unsupervised, unpaired domain adaptation. Hoewel deze aanpak zeer flexibel is en geen gelabelde data van het doeldomein vereist, is er ook een grote kans op onjuiste resultaten. We onderzoeken welke problemen zich kunnen voordoen en of overeenkomsten tussen de domeinen het model een voorkeur voor een bepaalde oplossing zouden kunnen geven. We bespreken vier aannames die vaak gelden voor medische afbeeldingen, zoals de aanname dat beelden uit verschillende domeinen een vergelijkbare spatiële structuur of een vergelijkbare intensiteitsverdeling hebben. We concluderen dat unsupervised, unpaired domain adaptation weliswaar weinig expliciete aannames maakt, maar dat de overeenkomsten tussen domeinen toch een sterke invloed kunnen hebben.

Hoofdstuk 6 bespreekt de verschillende perspectieven op domain adaptation in medische beeldverwerking, zoals die in dit proefschrift worden gepresenteerd. We vergelijken de aannames en keuzes die kunnen worden gemaakt, bespreken hoe deze tot uiting komen in de methodes, en bespreken de belangrijkste beperkingen. We kijken ook naar de verschillen en overeenkomsten tussen algemene computer vision en medische beeldverwerking. We besluiten met een aantal open vragen over de methodologische en klinische aspecten van domain adaptation in medische beeldverwerking.

Acknowledgements

All good things must come to an end, and while it is good that this particular thing ends here, it does bring me a touch of nostalgia. I've had great fun over the years, in no small part due to the people at BGR and in its surroundings.

Before I get to those long-term contributors, I would like to thank the members of the doctoral committee for agreeing to read my thesis and for attending the defense. It is great to have your expert opinions. I am particularly grateful to dr. Kostas Kamnitsas for his extensive list of suggestions for the manuscript. They certainly helped to improve the final version of this thesis.

Marleen, it was a pleasure working with you. I very much enjoyed our weekly meetings, which even at times when I didn't have much to report, usually gave me new ideas and motivation. (I remember that we discussed this last point at one of our yearly review meetings. You thought providing motivation was not part of your role, but I still think it was.) I liked the trust and freedom you gave me: my project was supposed to be about transfer learning for lung images, but I think I managed to go off on tangents for most of the time (perhaps that was also part of your plan). It might not have been the fastest way to finish a PhD, but in any case, I'm happy that it was possible. Thanks.

It could have been different. When we first met just before Christmas 2012, I was unsure whether I would like medical imaging, a topic that I had tried very hard to avoid until then, and I think you had to make quite an effort to convince me that the project and the group were something for me. You were successful, because after thinking for a few days, I emailed you to accept your offer. This introduced me to your long (or odd) working hours: by emailing late at night, I had hoped to postpone my decision just a little bit longer, but to my horror, you replied within an hour – so much for my precious state of indecision. However, after almost ten years, I can say it was a good choice: you were right about the project and the group, and if you press me today, I might even concede that medical imaging is actually quite interesting.

Speaking of how I got here, I must also thank Marco – my other scientific parent, you might say – who while supervising my Master's thesis managed to convince me that doing research is fun, and who played a pivotal role in connecting me with Marleen and BIGR.

Wiro, thank you for being my first and then second promotor. We did not spend too much time working together, but your input in the first year and while finishing my thesis was certainly very useful. Thank you for your support in the background, as well as for your general presence in BIGR. While the composition of the group changed over the years, I always found it a pleasant environment. No doubt, that must have had something to do with you.

Long-term office mates then, who without exception made it fun to come to work every day. In order of appearance:

Annegreet, fellow transfer learner, first roommate in the new Na building, and tour guide on my first day at BIGR. I remember being a bit surprised that so few people were in at 10.00 in the morning, but I later found out how that worked. Thank you for the interesting and entertaining discussions. From the brief chuckles about exercise regimens or the nutritional qualities of new types of food, to the uncontrollable hilarity of the octopus incident (sorry again for that), it was always a pleasure to have you around.

Florian, by far the noisiest of my BIGR roommates. I was lucky to share an office with you from the day you arrived to the day I left, interrupted only by your brief exile in the open plan office and the US. Thank you for the many funny and philosophical conversations. From my side of the office, it was nice to see you develop from a misunderstood genius to an equally ambitious but properly certified scientist, and whether it was a misplaced bicycle or your perennial hydranet, both your frustrated and happy sides were a joy to watch. Given the enviable speed at which you work, it was no surprise that you finished your PhD first. I should be the last person to comment on pacing, but I hope you don't forget to slow down to enjoy the view from time to time.

Veronika, we first met when you were part of my MSc thesis committee, and it was nice to see you again at BIGR a few years later. It was great to hear your machine learning insights at the model-based meetings, interesting to learn about your views on science, and impressive (if a bit humbling) to observe the efficient and structured way in which you worked.

Sebastian, I'm not sure if we can really call it long-term, but you were at least there until the end. It was nice to have you in the office for a while, if only so I could observe the preparations for the BIGR board game nights from up close. I have good memories of your impromptu music quiz on my last afternoon in the office. (I won from Florian by being slightly less bad, I think.)

This thesis would also have been impossible without the other members of the model-based meeting group, who provided useful feedback and inspiration at our meetings. Those were certainly one of the highlights of the week.

Adrià, thanks for the many coffee breaks, our joint work with Annegreet on the Dirty Catalan Phrasebook, your pragmatic advice on everything from lung imaging to colour schemes, and, of course, the first-and-only interdisciplinary seminar on astronomy and medical image analysis.

Antonio, thanks for the nice discussions and staircase walks.

Arna, thanks for the many shared lunch breaks, where we were usually the first to sit down as a result of bringing lunch from home.

Arno, erstwhile MBM member, thanks for hosting my GPU computer.

Gerda, thanks for being one of my paranympths, and for your many insightful contributions during our meetings. Although you arrived as an MSc student who I was supposed to co-supervise, it quickly became clear that you were very good at self-supervision: I didn't have to do much. You have an impressive amount of knowledge, be it on deep learning algorithms or board games with compulsory 40-minute instruction videos.

Kim, thanks for your insights on distance transforms and other deep learning-related topics. I am also grateful for your demonstration of what happens if you fill a small office with too many GPUs running at full power.

Robin, it was nice to have in the group the first time, and good to see you come back. I look forward to seeing more of your work.

Shuai, thanks for your deep learning ideas, and for your general up-beat attitude. It's an honour to have been involved with some of your work.

Zahra, thanks for being one of my paranympths, for the conversations at lunch and elsewhere, and for the brief time that we shared an office. You were a very nice presence in the group.

And, of course, Andrés, Deep, Dirk, Esther, Fedde and Hakim, the Danish cousins, and the many visiting students. Thank you for the many interesting discussions over the years.

ACKNOWLEDGEMENTS

From the wider BGR family, there is a long list of people I should thank for the nice coffee breaks, course and conference visits, outings, social events, seminars, and literature meetings.

Jean-Marie, starting around the same time, it was very nice to see you at courses, lunch, dinner, and the regular coffee breaks together with Adrià.

Marcel and Hakim, thanks for having me along in the cooking team at our two BGR outings. That was a great experience – it was the best part of the outings, if you ask me – and, as an added bonus, it gave my less-than-social side a welcome excuse to hide in the kitchen from time to time.

Regular visitors to lunch (you know who you are), it was always nice to join you in Sophia or the new building.

Those at board game nights, pre-Christmas-party dinners, and Sinterklaas.

Petra, Desiree, Tineke, Marise, Annemarijn, and the others at the radiology department, thanks for your administrative help.

And finally, Adriaan, Bo, Carolyn, Chaoping, Danilo, Diego, Dirk, Emanoel, Emilie, Erik, Erwin, Esben, Eugene, Gennady, Gerardo, Ghassan, Gokhan, Guillaume, Harm, Henk, Henri, Hortense, Hua, Hui, Ihor, Jifke, Jiahang, Joer, Jose, Jyotirmoy, Kasper, Karin, Luisa, Luu, Marius, Mart, Martijn, Mattias, Miroslav, Nóra, Oleh, Pierre, Rahil, Riway, Roman, Stefan, Theo, Thomas, Valerio, Vikram, Wei, Willem, Wietske, Wyke, Yao, Yuanyuan, and everyone who I forgot: thanks, it was a pleasure to share this time with you.

Harm, Pierluigi, Piotr, Pieter, and Laurike, thanks for teaching me some lung analysis tricks, and for showing me how our techniques can be applied in medical practice. It was nice to get this experience from the other side.

Elena, thank you for inviting me to Nijmegen. I found it a very welcoming environment. Thanks also to everyone else in the data science group: I didn't see you nearly enough in the past two years, but what I did see did not disappoint.

And finally, of course, my parents and sister. Thanks for your support, for asking curious questions, and, most importantly, for knowing what *not* to ask.

Gijs van Tulder
Rotterdam, April 2022

About the author

Gijs van Tulder was born in November 1983 in Leiderdorp, the Netherlands, but he never actually lived there. His early years were pleasant and happily uneventful. Having finished a BSc in Economics and Business at the Erasmus University Rotterdam, he said goodbye to the world of money and decided to do computer science instead. At the Delft University of Technology, he studied Computer Science and Media and Knowledge Engineering, which introduced him to the exciting world of machine learning. In 2012, he graduated cum laude with an MSc thesis on active learning.

In 2013, Gijs started his PhD project at the Biomedical Imaging Group Rotterdam (BIGR) at Erasmus MC, supervised by Marleen de Bruijne and Wiro Niessen. His project was about transfer learning and domain adaptation for medical imaging, which he decided to try with the then-new approach of representation learning and deep learning.

From 2017, Gijs worked as a researcher in collaborations between BIGR and the LungAnalysis group at Erasmus MC. One project involved MRI-based perfusion and ventilation analysis for cystic fibrosis patients, using a method based on Fourier decomposition. For a project with the Erasmus MC Pompe Center, Gijs developed a lung segmentation and measurement algorithm for the analysis of diaphragm function in Pompe patients.

Since January 2020, Gijs works as a researcher in the Data Science group at the Faculty of Science, Radboud University, Nijmegen. He investigates transfer learning for breast cancer screening and teaches deep learning to computer science MSc students.

Publications

Journal papers

G. van Tulder and M. de Bruijne, “Unpaired, unsupervised domain adaptation assumes your domains are already similar”, Submitted.

S. Chen, Z. Sedghi Gamechi, F. Dubost, **G. van Tulder**, and M. de Bruijne, “An end-to-end approach to segmentation in medical images with CNN and posterior-CRF”, *Medical Image Analysis*, vol. 76, p. 102 311, Feb. 2022.
DOI: 10.1016/j.MEDIA.2021.102311.

L. Harlaar, P. Ciet, **G. van Tulder**, E. Brusse, R. G. M. Timmermans, W. G. M. Janssen, M. de Bruijne, A. T. van der Ploeg, H. A. W. M. Tiddens, P. A. van Doorn, and N. A. M. E. van der Beek, “Diaphragmatic dysfunction in neuromuscular disease, an MRI study”, *Neuromuscular Disorders*, vol. 32, no. 1, pp. 15–24, Jan. 2022.
DOI: 10.1016/j.NMD.2021.11.001.

G. Bortsova, D. Bos, F. Dubost, M. W. Vernooij, M. K. Ikram, **G. van Tulder**, and M. de Bruijne, “Automated Segmentation and Volume Measurement of Intracranial Internal Carotid Artery Calcification at Noncontrast CT”, *Radiology: Artificial Intelligence*, vol. 3, no. 5, p. e200226, Sep. 2021. DOI: 10.1148/RYAI.2021200226.

L. Harlaar, P. Ciet, **G. van Tulder**, A. Pittaro, H. A. van Kooten, N. A. M. E. van der Beek, E. Brusse, P. A. Wielopolski, M. de Bruijne, A. T. van der Ploeg, H. A. W. M. Tiddens, and P. A. van Doorn, “Chest MRI to diagnose early diaphragmatic weakness in Pompe disease”, *Orphanet Journal of Rare Diseases*, vol. 16, no. 1, p. 21, Jan. 2021. DOI: 10.1186/s13023-020-01627-x.

F. Dubost, H. Adams, P. Yilmaz, G. Bortsova, **G. van Tulder**, M. A. Ikram, W. Niessen, M. W. Vernooij, and M. de Bruijne, “Weakly supervised object detection with 2D and 3D regression neural networks”, *Medical Image Analysis*, vol. 65, p. 101 767, Oct. 2020.
DOI: 10.1016/j.MEDIA.2020.101767.

G. van Tulder and M. de Bruijne, “Learning Cross-Modality Representations From Multi-Modal Images”, *IEEE Transactions on Medical Imaging*, vol. 38, no. 2, pp. 638–648, Feb. 2019. DOI: 10.1109/TMI.2018.2868977.

G. van Tulder and M. de Bruijne, “Combining Generative and Discriminative Representation Learning for Lung CT Analysis With Convolutional Restricted Boltzmann Machines”, *IEEE Transactions on Medical Imaging*, vol. 35, no. 5, pp. 1262–1272, 2016. DOI: 10.1109/TMI.2016.2526687.

Conference papers

G. van Tulder, Y. Tong, and E. Marchiori, “Multi-view Analysis of Unregistered Medical Images Using Cross-View Transformers”, in *Medical Image Computing and Computer Assisted Intervention – MICCAI 2021*, M. de Bruijne, P. C. Cattin, S. Cotin, N. Padoy, S. Speidel, Y. Zheng, and C. Essert, Eds., Lecture Notes in Computer Science, Cham: Springer International Publishing, 2021, pp. 104–113.
DOI: 10.1007/978-3-030-87199-4_10.

S. Chen, G. Bortsova, A. García-Uceda Juárez, **G. van Tulder**, and M. de Bruijne, “Multi-task Attention-Based Semi-supervised Learning for Medical Image Segmentation”, in *Medical Image Computing and Computer Assisted Intervention – MICCAI 2019*, D. Shen, T. Liu, T. M. Peters, L. H. Staib, C. Essert, S. Zhou, P.-T. Yap, and A. Khan, Eds., Lecture Notes in Computer Science, Cham: Springer International Publishing, 2019, pp. 457–465. DOI: 10.1007/978-3-030-32248-9_51.

G. Bortsova, **G. van Tulder**, F. Dubost, T. Peng, N. Navab, A. van der Lugt, D. Bos, and M. de Bruijne, “Segmentation of Intracranial Arterial Calcification with Deeply Supervised Residual Dropout Networks”, in *Medical Image Computing and Computer Assisted Intervention – MICCAI 2017*, M. Descoteaux, L. Maier-Hein, A. Franz, P. Jannin, D. L. Collins, and S. Duchesne, Eds., Lecture Notes in Computer Science, Cham: Springer International Publishing, 2017, pp. 356–364.
DOI: 10.1007/978-3-319-66179-7_41.

G. van Tulder and M. de Bruijne, “Representation Learning for Cross-Modality Classification”, in *Medical Computer Vision and Bayesian and Graphical Models for Biomedical Imaging*, H. Müller, B. M. Kelm, T. Arbel, W. Cai, M. J. Cardoso, G. Langs, B. Menze, D. Metaxas, A. Montillo, W. M. Wells III, S. Zhang, A. C. Chung, M. Jenkinson, and A. Ribbens, Eds., Lecture Notes in Computer Science, Cham: Springer International Publishing, 2017, pp. 126–136.
DOI: 10.1007/978-3-319-61188-4_12.

G. van Tulder and M. de Bruijne, “Why Does Synthesized Data Improve Multi-sequence Classification?”, in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, N. Navab, J. Hornegger, W. M. Wells, and A. Frangi, Eds., Lecture Notes in Computer Science, Cham: Springer International Publishing, 2015, pp. 531–538. DOI: 10.1007/978-3-319-24553-9_65.

G. van Tulder and M. de Bruijne, “Learning Features for Tissue Classification with the Classification Restricted Boltzmann Machine”, in *Medical Computer Vision: Algorithms for Big Data*, B. Menze, G. Langs, A. Montillo, M. Kelm, H. Müller, S. Zhang, W. Cai, and D. Metaxas, Eds., Cham: Springer International Publishing, 2014, pp. 47–58.
DOI: 10.1007/978-3-319-13972-2_5.

B. Loni, **G. van Tulder**, P. Wiggers, D. M. J. Tax, and M. Loog, “Question Classification by Weighted Combination of Lexical, Syntactic and Semantic Features”, in *Text, Speech and Dialogue*, I. Habernal and V. Matoušek, Eds., Lecture Notes in Computer Science, Berlin, Heidelberg: Springer, 2011, pp. 243–250. DOI: 10.1007/978-3-642-23538-2_31.

Other work

G. van Tulder, *Elasticdeform: Elastic deformations for N-dimensional images*, 2019.
DOI: 10.5281/ZENODO.4569691. <https://github.com/gvtulder/elasticdeform/>.

The Theano Development Team, “Theano: A Python framework for fast computation of mathematical expressions”, Tech. Rep., 2016. arXiv: 1605.02688.

PhD portfolio

Courses	Year	ECTS
Presentation course <i>Medical Informatics / Erasmus MC</i>	2013	1
Visual recognition and machine learning summer school <i>ENS/INRIA, Paris, France</i>	2013	4
Front-end vision and multiscale image analysis <i>ASCI / TU Eindhoven</i>	2013	4
Computer vision by learning <i>ASCI / University of Amsterdam</i>	2014	4
Deep learning summer school <i>KU/DTU, Copenhagen, Denmark</i>	2014	4
Scientific integrity course <i>Erasmus MC</i>	2014	0.3
Knowledge-driven image segmentation <i>ASCI / Leiden University Medical Center</i>	2014	4
Biomedical English writing and communication <i>Erasmus MC</i>	2015	3
Total		24.3

Conferences	Year	ECTS
Attending MICCAI 2014, oral workshop presentation <i>Boston, MA, USA</i>	2014	2
Attending MICCAI 2015, oral presentation <i>Munich, Germany</i>	2015	2
Attending MICCAI 2016, short talk at MCV workshop <i>Athens, Greece</i>	2016	2
NVPHBV	2016	1
Total		7

Supervision	Year	ECTS
Gerda Bortsova (co-supervision of MSc thesis)	2016	1

Internal meetings	Year	ECTS
Medical Informatics research lunch <i>BIGR/MI / Erasmus MC</i>	2013–2019	1
BIGR Seminars <i>BIGR / Erasmus MC</i>	2013–2019	1
Model-based meeting <i>BIGR / Erasmus MC</i>	2013–2019	1
Literature meetings <i>BIGR / Erasmus MC</i>	2013–2019	1
Total		4

Bibliography

- [1] B. H. Menze, A. Jakab, S. Bauer, et al., “The Multimodal Brain Tumor Image Segmentation Benchmark (BRATS)”, *IEEE Transactions on Medical Imaging*, vol. 34, no. 10, pp. 1993–2024, Oct. 2015. DOI: 10.1109/TMI.2014.2377694.
- [2] C. G. Peterfy, E. Schneider, and M. Nevitt, “The osteoarthritis initiative: Report on the design rationale for the magnetic resonance imaging protocol for the knee”, *Osteoarthritis and Cartilage*, vol. 16, no. 12, pp. 1433–1441, 2008. DOI: 10.1016/j.joca.2008.06.016.
- [3] Y. Bengio, A. Courville, and P. Vincent, “Representation Learning: A Review and New Perspectives”, Université de Montréal, Tech. Rep., 2012. arXiv: 1206.5538v2.
- [4] H. Larochelle, M. Mandel, R. Pascanu, and Y. Bengio, “Learning algorithms for the classification restricted Boltzmann machine”, *The Journal of Machine Learning Research*, vol. 13, no. 1, pp. 643–669, Mar. 2012.
- [5] G. Desjardins and Y. Bengio, “Empirical Evaluation of Convolutional RBMs for Vision”, Dept. IRO, Université de Montréal, Tech. Rep. 1327, 2008.
- [6] M. Norouzi, M. Ranjbar, and G. Mori, “Stacks of convolutional Restricted Boltzmann Machines for shift-invariant feature learning”, in *2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2009, pp. 2735–2742. DOI: 10.1109/CVPR.2009.5206577.
- [7] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng, “Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations”, in *Proceedings of the 26th International Conference on Machine Learning (ICML)*, ICML ’09, New York, NY, USA: Association for Computing Machinery, Jun. 2009, pp. 609–616. DOI: 10.1145/1553374.1553453.
- [8] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng, “Unsupervised learning of hierarchical representations with convolutional deep belief networks”, *Communications of the ACM*, vol. 54, no. 10, pp. 95–103, Oct. 2011. DOI: 10.1145/2001269.2001295.
- [9] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition”, *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998. DOI: 10.1109/5.726791.

- [10] A. Depeursinge, A. Vargas, A. Platon, A. Geissbuhler, P.-A. Poletti, and H. Müller, "Building a reference multimedia database for interstitial lung diseases", *Computerized Medical Imaging and Graphics*, vol. 36, no. 3, pp. 227–238, Apr. 2012. DOI: 10.1016/j.COMPIMEDIMAG.2011.07.003.
- [11] A. Depeursinge, A. Foncubierta-Rodriguez, D. Van de Ville, and H. Müller, "Lung Texture Classification Using Locally-Oriented Riesz Components", in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2011*, G. Fichtinger, A. Martel, and T. Peters, Eds., Lecture Notes in Computer Science, Berlin, Heidelberg: Springer, 2011, pp. 231–238. DOI: 10.1007/978-3-642-23626-6_29.
- [12] A. Depeursinge, A. Foncubierta-Rodriguez, D. Van de Ville, and H. Müller, "Multiscale Lung Texture Signature Learning Using the Riesz Transform", in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2012*, N. Ayache, H. Delingette, P. Golland, and K. Mori, Eds., Lecture Notes in Computer Science, Berlin, Heidelberg: Springer, 2012, pp. 517–524. DOI: 10.1007/978-3-642-33454-2_64.
- [13] A. Depeursinge, D. Van de Ville, A. Platon, A. Geissbuhler, P.-A. Poletti, and H. Müller, "Near-Affine-Invariant Texture Learning for Lung Tissue Analysis Using Isotropic Wavelet Frames", *IEEE Transactions on Information Technology in Biomedicine*, vol. 16, no. 4, pp. 665–675, Jul. 2012. DOI: 10.1109/TITB.2012.2198829.
- [14] A. Depeursinge, T. Zrimec, S. Busayarat, and H. Müller, "3D lung image retrieval using localized features", in *Medical Imaging 2011: Computer-Aided Diagnosis*, vol. 7963, SPIE, Mar. 2011, pp. 701–714. DOI: 10.1117/12.877943.
- [15] Y. Song, W. Cai, Y. Zhou, and D. D. Feng, "Feature-Based Image Patch Approximation for Lung Tissue Classification", *IEEE Transactions on Medical Imaging*, vol. 32, no. 4, pp. 797–808, Apr. 2013. DOI: 10.1109/TMI.2013.2241448.
- [16] Y. Song, W. Cai, H. Huang, Y. Zhou, D. D. Feng, Y. Wang, M. J. Fulham, and M. Chen, "Large Margin Local Estimate With Applications to Medical Image Classification", *IEEE Transactions on Medical Imaging*, vol. 34, no. 6, pp. 1362–1377, Jun. 2015. DOI: 10.1109/TMI.2015.2393954.
- [17] A. Foncubierta-Rodriguez, A. Depeursinge, and H. Müller, "Using Multiscale Visual Words for Lung Texture Classification and Retrieval", in *Medical Content-Based Retrieval for Clinical Decision Support*, H. Müller, H. Greenspan, and T. Syeda-Mahmood, Eds., Lecture Notes in Computer Science, Berlin, Heidelberg: Springer, 2012, pp. 69–79. DOI: 10.1007/978-3-642-28460-1_7.
- [18] M. Asherov, I. Diamant, and H. Greenspan, "Lung texture classification using bag of visual words", in *Medical Imaging 2014: Computer-Aided Diagnosis*, vol. 9035, SPIE, Mar. 2014, pp. 678–685. DOI: 10.1117/12.2044162.

- [19] M. Anthimopoulos, S. Christodoulidis, A. Christe, and S. Mougiakakou, "Classification of interstitial lung disease patterns using local DCT features and random forest", in *2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, Aug. 2014, pp. 6040–6043. DOI: 10.1109/EMBC.2014.6945006.
- [20] Q. Li, W. Cai, and D. D. Feng, "Lung image patch classification with automatic feature learning", in *2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, Jul. 2013, pp. 6079–6082. DOI: 10.1109/EMBC.2013.6610939.
- [21] Q. Li, W. Cai, X. Wang, Y. Zhou, D. D. Feng, and M. Chen, "Medical image classification with convolutional neural network", in *2014 13th International Conference on Control Automation Robotics Vision (ICARCV)*, Dec. 2014, pp. 844–848. DOI: 10.1109/ICARCV.2014.7064414.
- [22] M. Gao, U. Bagci, L. Lu, et al., "Holistic Classification of CT Attenuation Patterns for Interstitial Lung Diseases via Deep Convolutional Neural Networks", in *Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization*, vol. 6, 2015, pp. 1–6. DOI: 10.1080/21681163.2015.1124249.
- [23] J. H. Pedersen, H. Ashraf, A. Dirksen, et al., "The Danish Randomized Lung Cancer CT Screening Trial—Overall Design and Results of the Prevalence Round", *Journal of Thoracic Oncology*, vol. 4, no. 5, pp. 608–614, May 2009. DOI: 10.1097/JTO.0B013E3181A0D98F.
- [24] T. Schlegl, S. M. Waldstein, W.-D. Vogl, U. Schmidt-Erfurth, and G. Langs, "Predicting Semantic Descriptions from Medical Images with Convolutional Neural Networks", in *Information Processing in Medical Imaging (IPMI)*, S. Ourselin, D. C. Alexander, C.-F. Westin, and M. J. Cardoso, Eds., Lecture Notes in Computer Science, Cham: Springer International Publishing, 2015, pp. 437–448. DOI: 10.1007/978-3-319-19992-4_34.
- [25] S.-C. B. Lo, H.-P. Chan, J.-S. Lin, H. Li, M. T. Freedman, and S. K. Mun, "Artificial convolution neural network for medical image pattern recognition", *Neural Networks*, vol. 8, no. 7-8, pp. 1201–1214, 1995. DOI: 10.1016/0893-6080(95)00061-5.
- [26] S.-C. B. Lo, S.-L. a Lou, J.-S. Lin, M. T. Freedman, M. V. Chien, and S. K. Mun, "Artificial convolution neural network techniques and applications for lung nodule detection", *IEEE Transactions on Medical Imaging*, vol. 14, no. 4, pp. 711–718, Dec. 1995. DOI: 10.1109/42.476112.

- [27] W. Shen, M. Zhou, F. Yang, C. Yang, and J. Tian, “Multi-scale Convolutional Neural Networks for Lung Nodule Classification”, in *Information Processing in Medical Imaging (IPMI)*, S. Ourselin, D. C. Alexander, C.-F. Westin, and M. J. Cardoso, Eds., Lecture Notes in Computer Science, Cham: Springer International Publishing, 2015, pp. 588–599.
DOI: 10.1007/978-3-319-19992-4_46.
- [28] D. Kumar, A. Wong, and D. A. Clausi, “Lung Nodule Classification Using Deep Features in CT Images”, in *2015 12th Conference on Computer and Robot Vision*, Jun. 2015, pp. 133–138. DOI: 10.1109/CRV.2015.25.
- [29] H. R. Roth, L. Lu, A. Seff, K. M. Cherry, J. Hoffman, S. Wang, J. Liu, E. Turkbey, and R. M. Summers, “A New 2.5D Representation for Lymph Node Detection Using Random Sets of Deep Convolutional Neural Network Observations”, in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2014*, P. Golland, N. Hata, C. Barillot, J. Hornegger, and R. Howe, Eds., Lecture Notes in Computer Science, Cham: Springer International Publishing, 2014, pp. 520–527. DOI: 10.1007/978-3-319-10404-1_65.
- [30] S. Shin, S. Lee, and I. D. Yun, “Classification based micro-calcification detection using discriminative restricted Boltzmann machine in digitized mammograms”, in *Medical Imaging 2014: Computer-Aided Diagnosis*, vol. 9035, SPIE, Mar. 2014, pp. 415–420. DOI: 10.1117/12.2043316.
- [31] J. Berry and I. Fasel, “Dynamics of tongue gestures extracted automatically from ultrasound”, in *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2011, pp. 557–560.
DOI: 10.1109/ICASSP.2011.5946464.
- [32] T. Schmah, R. S. Zemel, G. E. Hinton, S. L. Small, and S. L. Strother, “Generative versus discriminative training of RBMs for classification of fMRI images”, in *Advances in Neural Information Processing*, 2008, pp. 1409–1416.
- [33] G. E. Hinton, “A Practical Guide to Training Restricted Boltzmann Machines”, in *Neural Networks: Tricks of the Trade: Second Edition*, ser. Lecture Notes in Computer Science, G. Montavon, G. B. Orr, and K.-R. Müller, Eds., Berlin, Heidelberg: Springer, 2012, pp. 599–619. DOI: 10.1007/978-3-642-35289-8_32.
- [34] T. Ojala, M. Pietikäinen, and D. Harwood, “A comparative study of texture measures with classification based on feature distributions”, *Pattern Recognition*, vol. 29, no. 1, pp. 51–59, Jan. 1996.
DOI: 10.1016/0031-3203(95)00067-4.
- [35] A. M. Saxe, P. W. Koh, Z. Chen, M. Bhand, B. Suresh, and A. Y. Ng, “On random weights and unsupervised feature learning”, in *Proceedings of the 28th International Conference on Machine Learning (ICML)*, ICML’11, Madison, WI, USA: Omnipress, Jun. 2011, pp. 1089–1096, ISBN: 978-1-4503-0619-5.

- [36] M. Varma and A. Zisserman, "A Statistical Approach to Material Classification Using Image Patch Exemplars", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 11, pp. 2032–2047, Nov. 2009.
DOI: 10.1109/TPAMI.2008.182.
- [37] T. Leung and J. Malik, "Representing and Recognizing the Visual Appearance of Materials using Three-dimensional Textons", *International Journal of Computer Vision*, vol. 43, no. 1, pp. 29–44, Jun. 2001.
DOI: 10.1023/A:1011126920638.
- [38] C. Schmid, "Constructing models for content-based image retrieval", in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, vol. 2, Dec. 2001, pp. II-39–II-45.
DOI: 10.1109/CVPR.2001.990922.
- [39] J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley, and Y. Bengio, "Theano: A CPU and GPU Math Compiler in Python", in *Python for Scientific Computing Conference (SciPy)*, S. van der Walt and J. Millman, Eds., 2010.
DOI: 10.25080/MAJORA-92BF1922-003.
- [40] F. Pedregosa, G. Varoquaux, A. Gramfort, et al., "Scikit-learn: Machine Learning in Python", *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, Oct. 2011.
- [41] J. Petersen, M. Nielsen, P. Lo, L. H. Nordenmark, J. H. Pedersen, M. M. W. Wille, A. Dirksen, and M. de Bruijne, "Optimal surface segmentation using flow lines to quantify airway abnormalities in chronic obstructive pulmonary disease", *Medical Image Analysis*, vol. 18, no. 3, pp. 531–541, Apr. 2014. DOI: 10.1016/j.MEDIA.2014.02.004.
- [42] Y. Song, W. Cai, H. Huang, Y. Zhou, Y. Wang, and D. D. Feng, "Locality-constrained Subcluster Representation Ensemble for lung image classification", *Medical Image Analysis*, vol. 22, no. 1, pp. 102–113, May 2015.
DOI: 10.1016/j.MEDIA.2015.03.003.
- [43] Y. Song, W. Cai, S. Huh, M. Chen, T. Kanade, Y. Zhou, and D. Feng, "Discriminative Data Transform for Image Feature Extraction and Classification", in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2013*, K. Mori, I. Sakuma, Y. Sato, C. Barillot, and N. Navab, Eds., Lecture Notes in Computer Science, Berlin, Heidelberg: Springer, 2013, pp. 452–459. DOI: 10.1007/978-3-642-40763-5_56.
- [44] Y. Song, W. Cai, H. Huang, Y. Zhou, Y. Wang, and D. Dagan Feng, "Boosted multifold sparse representation with application to ILD classification", in *2014 IEEE 11th International Symposium on Biomedical Imaging (ISBI)*, Apr. 2014, pp. 1023–1026. DOI: 10.1109/ISBI.2014.6868047.

- [45] J. K. Dash, V. Madhavi, S. Mukhopadhyay, N. Khandelwal, and P. Kumar, "Segmentation of interstitial lung disease patterns in HRCT images", in *Medical Imaging 2015: Computer-Aided Diagnosis*, vol. 9414, SPIE, Mar. 2015, pp. 687–692. DOI: 10.1117/12.2079072.
- [46] R. Salakhutdinov and G. Hinton, "An efficient learning procedure for deep Boltzmann machines.", *Neural Computation*, vol. 24, no. 8, pp. 1967–2006, Aug. 2012. DOI: 10.1162/NECO_A_00311.
- [47] G. E. Hinton and R. R. Salakhutdinov, "Reducing the Dimensionality of Data with Neural Networks", *Science*, vol. 313, no. 5786, pp. 504–507, Jul. 2006. DOI: 10.1126/SCIENCE.1127647.
- [48] D. Erhan, Y. Bengio, and A. Courville, "Why does unsupervised pre-training help deep learning?", *Journal of Machine Learning Research*, vol. 11, pp. 625–660, Feb. 2010.
- [49] R. J. A. Little and D. B. Rubin, *Statistical Analysis with Missing Data*, Second Edition. New York: Wiley, 2002, ISBN: 0-471-18386-5.
- [50] B. Fischl, D. H. Salat, A. J. W. van der Kouwe, N. Makris, F. Ségonne, B. T. Quinn, and A. M. Dale, "Sequence-independent segmentation of magnetic resonance images.", *NeuroImage*, vol. 23, pp. S69–S84, Jan. 2004. DOI: 10.1016/j.NEUROIMAGE.2004.07.016.
- [51] A. Johansson, M. Karlsson, and T. Nyholm, "CT substitute derived from MRI sequences with ultrashort echo time", *Medical Physics*, vol. 38, no. 5, pp. 2708–2714, 2011. DOI: 10.1118/1.3578928.
- [52] A. Johansson, A. Garpebring, T. Asklund, and T. Nyholm, "CT substitutes derived from MR images reconstructed with parallel imaging", *Medical Physics*, vol. 41, no. 8 part 1, p. 082302, 2014. DOI: 10.1118/1.4886766.
- [53] K. Eilertsen, L. Nilsen Tor Arne Vestad, O. Geier, and A. Skretting, "A simulation of MRI based dose calculations on the basis of radiotherapy planning CT images", *Acta Oncologica*, vol. 47, no. 7, pp. 1294–1302, Jan. 2008. DOI: 10.1080/02841860802256426.
- [54] M. Kapanen and M. Tenhunen, "T₁/T₂*-weighted MRI provides clinically relevant pseudo-CT density data for the pelvic bones in MRI-only based radiotherapy treatment planning.", *Acta Oncologica*, vol. 52, no. 3, pp. 612–618, Apr. 2013. DOI: 10.3109/0284186X.2012.692883.
- [55] A. Larsson, A. Johansson, J. Axelsson, T. Nyholm, T. Asklund, K. Riklund, and M. Karlsson, "Evaluation of an attenuation correction method for PET/MR imaging of the head based on substitute CT images.", *Magnetic Resonance Materials in Physics, Biology and Medicine*, vol. 26, no. 1, pp. 127–136, Feb. 2013. DOI: 10.1007/s10334-012-0339-2.

- [56] M. Hofmann, F. Steinke, V. Scheel, G. Charpiat, J. Farquhar, P. Aschoff, M. Brady, B. Schölkopf, and B. J. Pichler, “MRI-Based Attenuation Correction for PET/MRI: A Novel Approach Combining Pattern Recognition and Atlas Registration”, *Journal of Nuclear Medicine*, vol. 49, no. 11, pp. 1875–1883, Nov. 2008. DOI: 10.2967/JNUMED.107.049353.
- [57] M. Hofmann, B. Pichler, B. Schölkopf, and T. Beyer, “Towards quantitative PET/MRI: A review of MR-based attenuation correction techniques”, *European Journal of Nuclear Medicine and Molecular Imaging*, vol. 36, no. 1, pp. 93–104, Mar. 2009. DOI: 10.1007/s00259-008-1007-7.
- [58] J. E. Iglesias, E. Konukoglu, D. Zikic, B. Glocker, K. Van Leemput, and B. Fischl, “Is Synthesizing MRI Contrast Useful for Inter-modality Analysis?”, in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2013*, K. Mori, I. Sakuma, Y. Sato, C. Barillot, and N. Navab, Eds., Lecture Notes in Computer Science, Berlin, Heidelberg: Springer, 2013, pp. 631–638. DOI: 10.1007/978-3-642-40811-3_79.
- [59] S. Roy, A. Carass, and J. Prince, “A Compressed Sensing Approach for MR Tissue Contrast Synthesis”, in *Information Processing in Medical Imaging (IPMI)*, G. Székely and H. K. Hahn, Eds., Lecture Notes in Computer Science, Berlin, Heidelberg: Springer, 2011, pp. 371–383. DOI: 10.1007/978-3-642-22092-0_31.
- [60] R. Li, W. Zhang, H.-I. Suk, L. Wang, J. Li, D. Shen, and S. Ji, “Deep Learning Based Imaging Data Completion for Improved Brain Disease Diagnosis”, in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2014*, P. Golland, N. Hata, C. Barillot, J. Hornegger, and R. Howe, Eds., Lecture Notes in Computer Science, Cham: Springer International Publishing, 2014, pp. 305–312. DOI: 10.1007/978-3-319-10443-0_39.
- [61] T. Tieleman, “Training restricted Boltzmann machines using approximations to the likelihood gradient”, in *Proceedings of the 25th International Conference on Machine Learning (ICML)*, ICML ’08, New York, NY, USA: Association for Computing Machinery, Jul. 2008, pp. 1064–1071. DOI: 10.1145/1390156.1390290.
- [62] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng, “Multimodal deep learning”, in *Proceedings of the 28th International Conference on Machine Learning (ICML)*, ICML’11, Madison, WI, USA: Omnipress, Jun. 2011, pp. 689–696, ISBN: 978-1-4503-0619-5.
- [63] G. van Tulder and M. de Bruijne, “Representation Learning for Cross-Modality Classification”, in *Medical Computer Vision and Bayesian and Graphical Models for Biomedical Imaging*, H. Müller, B. M. Kelm, T. Arbel, et al., Eds., Lecture Notes in Computer Science, Cham: Springer International Publishing, 2017, pp. 126–136. DOI: 10.1007/978-3-319-61188-4_12.

- [64] G. van Tulder and M. de Bruijne, "Why Does Synthesized Data Improve Multi-sequence Classification?," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, N. Navab, J. Hornegger, W. M. Wells, and A. Frangi, Eds., Lecture Notes in Computer Science, Cham: Springer International Publishing, 2015, pp. 531–538.
DOI: 10.1007/978-3-319-24553-9_65.
- [65] M. Havaei, N. Guizard, N. Chapados, and Y. Bengio, "HeMIS: Hetero-Modal Image Segmentation", in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2016*, S. Ourselin, L. Joskowicz, M. R. Sabuncu, G. Unal, and W. Wells, Eds., Lecture Notes in Computer Science, Cham: Springer International Publishing, 2016, pp. 469–477.
DOI: 10.1007/978-3-319-46723-8_54.
- [66] P. Moeskops, J. M. Wolterink, B. H. M. van der Velden, K. G. A. Gilhuijs, T. Leiner, M. A. Viergever, and I. Išgum, "Deep Learning for Multi-task Medical Image Segmentation in Multiple Modalities", in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2016*, S. Ourselin, L. Joskowicz, M. R. Sabuncu, G. Unal, and W. Wells, Eds., Lecture Notes in Computer Science, Cham: Springer International Publishing, 2016, pp. 478–486. DOI: 10.1007/978-3-319-46723-8_55.
- [67] Y. He, S. Xiang, C. Kang, J. Wang, and C. Pan, "Cross-Modal Retrieval via Deep and Bidirectional Representation Learning", *IEEE Transactions on Multimedia*, vol. 18, no. 7, pp. 1363–1377, Jul. 2016. DOI: 10.1109/TMM.2016.2558463.
- [68] S. Rastegar, M. S. Baghshah, H. R. Rabiee, and S. M. Shojaei, "MDL-CW: A Multimodal Deep Learning Framework with CrossWeights", in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2016, pp. 2601–2609. DOI: 10.1109/CVPR.2016.285.
- [69] L. Zhao, Q. Hu, and W. Wang, "Heterogeneous Feature Selection With Multi-Modal Deep Neural Networks and Sparse Group LASSO", *IEEE Transactions on Multimedia*, vol. 17, no. 11, pp. 1936–1948, Nov. 2015.
DOI: 10.1109/TMM.2015.2477058.
- [70] F. Feng, X. Wang, R. Li, and I. Ahmad, "Correspondence Autoencoders for Cross-Modal Retrieval", *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 12, no. 18, pp. 1–22, Oct. 2015.
DOI: 10.1145/2808205.
- [71] N. Srivastava and R. Salakhutdinov, "Multimodal learning with deep Boltzmann machines", *Advances in Neural Information Processing Systems (NIPS)*, 2012.

- [72] V. Vukotić, C. Raymond, and G. Gravier, “Bidirectional Joint Representation Learning with Symmetrical Deep Neural Networks for Multimodal and Crossmodal Applications”, in *Proceedings of the 2016 ACM on International Conference on Multimedia Retrieval - ICMR '16*, New York, New York, USA: ACM Press, 2016, pp. 343–346. DOI: 10.1145/2911996.2912064.
- [73] M. Moradi, Y. Guo, Y. Gur, M. Negahdar, and T. Syeda-Mahmood, “A Cross-Modality Neural Network Transform for Semi-automatic Medical Image Annotation”, in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2016*, S. Ourselin, L. Joskowicz, M. R. Sabuncu, G. Unal, and W. Wells, Eds., Lecture Notes in Computer Science, Cham: Springer International Publishing, 2016, pp. 300–307. DOI: 10.1007/978-3-319-46723-8_35.
- [74] Z. Zhang, L. Yang, and Y. Zheng, “Translating and Segmenting Multimodal Medical Volumes with Cycle- and Shape-Consistency Generative Adversarial Network”, in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2018, pp. 9242–9251. DOI: 10.1109/CVPR.2018.00963.
- [75] K. Kamnitsas, C. Baumgartner, C. Ledig, et al., “Unsupervised Domain Adaptation in Brain Lesion Segmentation with Adversarial Networks”, in *Information Processing in Medical Imaging (IPMI)*, M. Niethammer, M. Styner, S. Aylward, H. Zhu, I. Oguz, P.-T. Yap, and D. Shen, Eds., Lecture Notes in Computer Science, Cham: Springer International Publishing, 2017, pp. 597–609. DOI: 10.1007/978-3-319-59050-9_47.
- [76] J. Folkesson, E. B. Dam, O. F. Olsen, P. C. Pettersen, and C. Christiansen, “Segmenting articular cartilage automatically using a voxel classification approach”, *IEEE Transactions on Medical Imaging*, vol. 26, no. 1, pp. 106–115, 2007. DOI: 10.1109/TMI.2006.886808.
- [77] A. Prasoon, K. Petersen, C. Igel, F. Lauze, E. Dam, and M. Nielsen, “Deep Feature Learning for Knee Cartilage Segmentation Using a Triplanar Convolutional Neural Network”, in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2013*, K. Mori, I. Sakuma, Y. Sato, C. Barillot, and N. Navab, Eds., Lecture Notes in Computer Science, Berlin, Heidelberg: Springer, 2013, pp. 246–253. DOI: 10.1007/978-3-642-40763-5_31.
- [78] M. Havaei, A. Davy, D. Warde-Farley, A. Biard, A. Courville, Y. Bengio, C. Pal, P.-M. Jodoin, and H. Larochelle, “Brain tumor segmentation with Deep Neural Networks”, *Medical Image Analysis*, vol. 35, pp. 18–31, Jan. 2017. DOI: 10.1016/J.MEDIA.2016.05.004.

- [79] S. Ioffe and C. Szegedy, “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift”, in *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, PMLR, Jun. 2015, pp. 448–456.
- [80] Y. Li, J. Yosinski, J. Clune, H. Lipson, and J. Hopcroft, “Convergent Learning: Do different neural networks learn the same representations?”, in *International Conference on Learning Representations (ICLR)*, 2016. arXiv: 1511.07543.
- [81] F. Chollet, *Keras*, <http://keras.io/>, 2017.
- [82] The Theano Development Team, “Theano: A Python framework for fast computation of mathematical expressions”, Tech. Rep., 2016. arXiv: 1605.02688.
- [83] A. Arovitola and L. Gallo, “Knee bone segmentation from MRI: A classification and literature review”, *Biocybernetics and Biomedical Engineering*, vol. 36, no. 2, pp. 437–449, 2016. DOI: 10.1016/J.BBE.2015.12.007.
- [84] H. Guan and M. Liu, “Domain Adaptation for Medical Image Analysis: A Survey”, *arXiv:2102.09508 [cs, eess]*, Feb. 2021. arXiv: 2102.09508.
- [85] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, “Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks”, in *2017 IEEE International Conference on Computer Vision (ICCV)*, Oct. 2017, pp. 2242–2251. DOI: 10.1109/ICCV.2017.244.
- [86] A. Gretton, A. Smola, J. Huang, M. Schmittfull, K. Borgwardt, and B. Schölkopf, “Covariate Shift by Kernel Mean Matching”, in *Dataset Shift in Machine Learning*, The MIT Press, 2008. DOI: 10.7551/MITPRESS/9780262170055.003.0008.
- [87] F. Wu and X. Zhuang, “Unsupervised Domain Adaptation With Variational Approximation for Cardiac Segmentation”, *IEEE Transactions on Medical Imaging*, vol. 40, no. 12, pp. 3555–3567, Dec. 2021. DOI: 10.1109/TMI.2021.3090412.
- [88] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, “Domain-Adversarial Training of Neural Networks”, in *Domain Adaptation in Computer Vision Applications*, G. Csurka, Ed., Cham: Springer International Publishing, 2017, pp. 189–209. DOI: 10.1007/978-3-319-58347-1_10.
- [89] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, “Adversarial Discriminative Domain Adaptation”, in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul. 2017, pp. 2962–2971. DOI: 10.1109/CVPR.2017.316.

- [90] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan, "A theory of learning from different domains", *Machine Learning*, vol. 79, no. 1, pp. 151–175, May 2010. DOI: 10.1007/s10994-009-5152-4.
- [91] S. Ben-David, T. Luu, T. Lu, and D. Pál, "Impossibility Theorems for Domain Adaptation", in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, JMLR Workshop and Conference Proceedings, Mar. 2010, pp. 129–136.
- [92] H. Zhao, R. T. D. Combes, K. Zhang, and G. Gordon, "On Learning Invariant Representations for Domain Adaptation", in *Proceedings of the 36th International Conference on Machine Learning (ICML)*, PMLR, May 2019, pp. 7523–7532.
- [93] J. P. Cohen, M. Luck, and S. Honari, "Distribution Matching Losses Can Hallucinate Features in Medical Image Translation", in *Medical Image Computing and Computer Assisted Intervention – MICCAI 2018*, A. F. Frangi, J. A. Schnabel, C. Davatzikos, C. Alberola-López, and G. Fichtinger, Eds., Lecture Notes in Computer Science, Cham: Springer International Publishing, 2018, pp. 529–536. DOI: 10.1007/978-3-030-00928-1_60.
- [94] J. Hoffman, E. Tzeng, T. Park, J.-Y. Zhu, P. Isola, K. Saenko, A. Efros, and T. Darrell, "CyCADA: Cycle-Consistent Adversarial Domain Adaptation", in *Proceedings of the 35th International Conference on Machine Learning (ICML)*, PMLR, Jul. 2018, pp. 1989–1998.
- [95] J. M. Wolterink, P. R. Seevinck, and A. M. Dinkla, "MR-to-CT Synthesis using Cycle-Consistent Generative Adversarial Networks", in *Med-NIPS*, 2017.
- [96] H. Yang, J. Sun, A. Carass, C. Zhao, J. Lee, Z. Xu, and J. Prince, "Unpaired Brain MR-to-CT Synthesis Using a Structure-Constrained CycleGAN", in *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, D. Stoyanov, Z. Taylor, G. Carneiro, et al., Eds., Lecture Notes in Computer Science, Cham: Springer International Publishing, 2018, pp. 174–182. DOI: 10.1007/978-3-030-00889-5_20.
- [97] B. Zhou, Z. Augenfeld, J. Chapiro, S. K. Zhou, C. Liu, and J. S. Duncan, "Anatomy-guided multimodal registration by learning segmentation without ground truth: Application to intraprocedural CBCT/MR liver segmentation and registration", *Medical Image Analysis*, vol. 71, p. 102 041, Jul. 2021. DOI: 10.1016/j.MEDIA.2021.102041.
- [98] S. U. Dar, M. Yurt, L. Karacan, A. Erdem, E. Erdem, and T. Çukur, "Image Synthesis in Multi-Contrast MRI With Conditional Generative Adversarial Networks", *IEEE Transactions on Medical Imaging*, vol. 38, no. 10, pp. 2375–2388, Oct. 2019. DOI: 10.1109/TMI.2019.2901750.

- [99] L. Ju, X. Wang, X. Zhao, P. Bonnington, T. Drummond, and Z. Ge, “Leveraging Regular Fundus Images for Training UWF Fundus Diagnosis Models via Adversarial Learning and Pseudo-Labeling”, *IEEE Transactions on Medical Imaging*, vol. 40, no. 10, pp. 2911–2925, Oct. 2021. DOI: 10.1109/TMI.2021.3056395.
- [100] H. Li, H. Han, Z. Li, L. Wang, Z. Wu, J. Lu, and S. K. Zhou, “High-Resolution Chest X-Ray Bone Suppression Using Unpaired CT Structural Priors”, *IEEE Transactions on Medical Imaging*, vol. 39, no. 10, pp. 3053–3063, Oct. 2020. DOI: 10.1109/TMI.2020.2986242.
- [101] T. de Bel, J.-M. Bokhorst, J. van der Laak, and G. Litjens, “Residual cyclegan for robust domain transformation of histopathological tissue slides”, *Medical Image Analysis*, vol. 70, p. 102 004, May 2021. DOI: 10.1016/j.MEDIA.2021.102004.
- [102] J. Cai, Z. Zhang, L. Cui, Y. Zheng, and L. Yang, “Towards cross-modal organ translation and segmentation: A cycle- and shape-consistent generative adversarial network”, *Medical Image Analysis*, vol. 52, pp. 174–184, Feb. 2019. DOI: 10.1016/j.MEDIA.2018.12.002.
- [103] J. Jiao, A. I. L. Namburete, A. T. Papageorgiou, and J. A. Noble, “Self-Supervised Ultrasound to MRI Fetal Brain Image Synthesis”, *IEEE Transactions on Medical Imaging*, vol. 39, no. 12, pp. 4413–4424, Dec. 2020. DOI: 10.1109/TMI.2020.3018560.
- [104] X. Chen, C. Lian, L. Wang, H. Deng, T. Kuang, S. Fung, J. Gateno, P.-T. Yap, J. J. Xia, and D. Shen, “Anatomy-Regularized Representation Learning for Cross-Modality Medical Image Segmentation”, *IEEE Transactions on Medical Imaging*, vol. 40, no. 1, pp. 274–285, Jan. 2021. DOI: 10.1109/TMI.2020.3025133.
- [105] D. Tomar, M. Lortkipanidze, G. Vray, B. Bozorgtabar, and J.-P. Thiran, “Self-Attentive Spatial Adaptive Normalization for Cross-Modality Domain Adaptation”, *IEEE Transactions on Medical Imaging*, vol. 40, no. 10, pp. 2926–2938, Oct. 2021. DOI: 10.1109/TMI.2021.3059265.
- [106] M. Ren, N. Dey, J. Fishbaugh, and G. Gerig, “Segmentation-Renormalized Deep Feature Modulation for Unpaired Image Harmonization”, *IEEE Transactions on Medical Imaging*, vol. 40, no. 6, pp. 1519–1530, Jun. 2021. DOI: 10.1109/TMI.2021.3059726.
- [107] A. Tomczak, S. Ilic, G. Marquardt, T. Engel, F. Forster, N. Navab, and S. Albarqouni, “Multi-Task Multi-Domain Learning for Digital Staining and Classification of Leukocytes”, *IEEE Transactions on Medical Imaging*, vol. 40, no. 10, pp. 2897–2910, Oct. 2021. DOI: 10.1109/TMI.2020.3046334.

- [108] B. Yu, L. Zhou, L. Wang, Y. Shi, J. Fripp, and P. Bourgeat, "Sample-Adaptive GANs: Linking Global and Local Mappings for Cross-Modality MR Image Synthesis", *IEEE Transactions on Medical Imaging*, vol. 39, no. 7, pp. 2339–2350, Jul. 2020. DOI: 10.1109/TMI.2020.2969630.
- [109] C. Chen, Q. Dou, H. Chen, J. Qin, and P. A. Heng, "Unsupervised Bidirectional Cross-Modality Adaptation via Deeply Synergistic Image and Feature Alignment for Medical Image Segmentation", *IEEE Transactions on Medical Imaging*, vol. 39, no. 7, pp. 2494–2505, Jul. 2020. DOI: 10.1109/TMI.2020.2972701.
- [110] Y. Gao, Y. Liu, Y. Wang, Z. Shi, and J. Yu, "A Universal Intensity Standardization Method Based on a Many-to-One Weak-Paired Cycle Generative Adversarial Network for Magnetic Resonance Images", *IEEE Transactions on Medical Imaging*, vol. 38, no. 9, pp. 2059–2069, Sep. 2019. DOI: 10.1109/TMI.2019.2894692.
- [111] H. Li, T. Loehr, B. Wiestler, J. Zhang, and B. H. Menze, "E-UDA: Efficient Unsupervised Domain Adaptation for Cross-Site Medical Image Segmentation.", in *CORR*, Jan. 2020.
- [112] C. Bian, C. Yuan, J. Wang, M. Li, X. Yang, S. Yu, K. Ma, J. Yuan, and Y. Zheng, "Uncertainty-aware domain alignment for anatomical structure segmentation", *Medical Image Analysis*, vol. 64, p. 101732, Aug. 2020. DOI: 10.1016/J.MEDIA.2020.101732.
- [113] H. Guan, Y. Liu, E. Yang, P.-T. Yap, D. Shen, and M. Liu, "Multi-site MRI harmonization via attention-guided deep domain adaptation for brain disorder identification", *Medical Image Analysis*, vol. 71, p. 102076, Jul. 2021. DOI: 10.1016/J.MEDIA.2021.102076.
- [114] X. Liu, X. Guo, Y. Liu, and Y. Yuan, "Consolidated domain adaptive detection and localization framework for cross-device colonoscopic images", *Medical Image Analysis*, vol. 71, p. 102052, Jul. 2021. DOI: 10.1016/J.MEDIA.2021.102052.
- [115] Y. Shen, B. Sheng, R. Fang, H. Li, L. Dai, S. Stolte, J. Qin, W. Jia, and D. Shen, "Domain-invariant interpretable fundus image quality assessment", *Medical Image Analysis*, vol. 61, p. 101654, Apr. 2020. DOI: 10.1016/J.MEDIA.2020.101654.
- [116] D. Hu, H. Zhang, Z. Wu, F. Wang, L. Wang, J. K. Smith, W. Lin, G. Li, and D. Shen, "Disentangled-Multimodal Adversarial Autoencoder: Application to Infant Age Prediction With Incomplete Multimodal Neuroimages", *IEEE Transactions on Medical Imaging*, vol. 39, no. 12, pp. 4137–4149, Dec. 2020. DOI: 10.1109/TMI.2020.3013825.
- [117] C. Pei, F. Wu, L. Huang, and X. Zhuang, "Disentangle domain features for cross-modality cardiac image segmentation", *Medical Image Analysis*, vol. 71, p. 102078, Jul. 2021. DOI: 10.1016/J.MEDIA.2021.102078.

- [118] M. Bateson, J. Dolz, H. Kervadec, H. Lombaert, and I. B. Ayed, “Constrained Domain Adaptation for Image Segmentation”, *IEEE Transactions on Medical Imaging*, vol. 40, no. 7, pp. 1875–1887, Jul. 2021. DOI: 10.1109/TMI.2021.3067688.
- [119] Z. Cui, C. Li, Z. Du, N. Chen, G. Wei, R. Chen, L. Yang, D. Shen, and W. Wang, “Structure-Driven Unsupervised Domain Adaptation for Cross-Modality Cardiac Segmentation”, *IEEE Transactions on Medical Imaging*, vol. 40, no. 12, pp. 3604–3616, Dec. 2021. DOI: 10.1109/TMI.2021.3090432.
- [120] S. Wang, L. Yu, X. Yang, C.-W. Fu, and P.-A. Heng, “Patch-Based Output Space Adversarial Learning for Joint Optic Disc and Cup Segmentation”, *IEEE Transactions on Medical Imaging*, vol. 38, no. 11, pp. 2485–2495, Nov. 2019. DOI: 10.1109/TMI.2019.2899910.
- [121] N. A. Koohbanani, B. Unnikrishnan, S. A. Khurram, P. Krishnaswamy, and N. Rajpoot, “Self-Path: Self-Supervision for Classification of Pathology Images With Limited Annotations”, *IEEE Transactions on Medical Imaging*, vol. 40, no. 10, pp. 2845–2856, Oct. 2021. DOI: 10.1109/TMI.2021.3056023.
- [122] L. Luo, L. Yu, H. Chen, Q. Liu, X. Wang, J. Xu, and P.-A. Heng, “Deep Mining External Imperfect Data for Chest X-Ray Disease Screening”, *IEEE Transactions on Medical Imaging*, vol. 39, no. 11, pp. 3583–3594, Nov. 2020. DOI: 10.1109/TMI.2020.3000949.
- [123] C. Chen, Q. Dou, H. Chen, J. Qin, and P.-A. Heng, “Synergistic Image and Feature Adaptation: Towards Cross-Modality Domain Adaptation for Medical Image Segmentation”, *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, pp. 865–872, Jul. 2019. DOI: 10.1609/AAAI.V33IO1.3301865.
- [124] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative Adversarial Nets”, in *Advances in Neural Information Processing Systems (NIPS)*, vol. 27, Curran Associates, Inc., 2014.
- [125] Y. Ganin and V. Lempitsky, “Unsupervised Domain Adaptation by Backpropagation”, in *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, PMLR, Jun. 2015, pp. 1180–1189.
- [126] S. Kornblith, M. Norouzi, H. Lee, and G. Hinton, “Similarity of Neural Network Representations Revisited”, in *Proceedings of the 36th International Conference on Machine Learning (ICML)*, PMLR, May 2019, pp. 3519–3529. arXiv: 1905.00414.

- [127] S. Y. Shin, S. Lee, and R. M. Summers, “Unsupervised Domain Adaptation for Small Bowel Segmentation Using Disentangled Representation”, in *Medical Image Computing and Computer Assisted Intervention – MICCAI 2021*, M. de Bruijne, P. C. Cattin, S. Cotin, N. Padoy, S. Speidel, Y. Zheng, and C. Essert, Eds., Lecture Notes in Computer Science, Cham: Springer International Publishing, 2021, pp. 282–292. DOI: 10.1007/978-3-030-87199-4_27.
- [128] M. Raghu, C. Zhang, J. Kleinberg, and S. Bengio, “Transfusion: Understanding Transfer Learning for Medical Imaging”, in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 32, Curran Associates, Inc., 2019.
- [129] Y. Wen, L. Chen, Y. Deng, and C. Zhou, “Rethinking pre-training on medical imaging”, *Journal of Visual Communication and Image Representation*, vol. 78, p. 103 145, Jul. 2021. DOI: 10.1016/J.JVCIR.2021.103145.
- [130] M. R. Hosseinzadeh Taher, F. Haghighi, R. Feng, M. B. Gotway, and J. Liang, “A Systematic Benchmarking Analysis of Transfer Learning for Medical Image Analysis”, in *Domain Adaptation and Representation Transfer, and Affordable Healthcare and AI for Resource Diverse Global Health*, S. Albarqouni, M. J. Cardoso, Q. Dou, et al., Eds., Lecture Notes in Computer Science, Cham: Springer International Publishing, 2021, pp. 3–13. DOI: 10.1007/978-3-030-87722-4_1.
- [131] T. Schlegl, J. Ofner, and G. Langs, “Unsupervised Pre-training Across Image Domains Improves Lung Tissue Classification”, in *Medical Computer Vision: Algorithms for Big Data*, B. Menze, G. Langs, A. Montillo, M. Kelm, H. Müller, S. Zhang, W. (Cai, and D. Metaxas, Eds., Lecture Notes in Computer Science, Cham: Springer International Publishing, 2014, pp. 82–93. DOI: 10.1007/978-3-319-13972-2_8.
- [132] S. Chen, K. Ma, and Y. Zheng, “Med3D: Transfer Learning for 3D Medical Image Analysis”, *arXiv:1904.00625 [cs]*, Jul. 2019. arXiv: 1904.00625.
- [133] A. Taleb, W. Loetzsch, N. Danz, J. Severin, T. Gaertner, B. Bergner, and C. Lippert, “3D self-supervised methods for medical imaging”, in *Advances in Neural Information Processing Systems (NeurIPS)*, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, Eds., vol. 33, Curran Associates, Inc., 2020, pp. 18 158–18 172. arXiv: 2006.03829.
- [134] M. Arjovsky and L. Bottou, “Towards Principled Methods for Training Generative Adversarial Networks”, in *International Conference on Learning Representations*, Jan. 2017. arXiv: 1701.04862.
- [135] W. M. Kouw and M. Loog, “A Review of Domain Adaptation without Target Labels”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 3, pp. 766–785, Mar. 2021. DOI: 10.1109/TPAMI.2019.2945942.

BIBLIOGRAPHY

- [136] G. Wilson and D. J. Cook, “A Survey of Unsupervised Deep Domain Adaptation”, *ACM Transactions on Intelligent Systems and Technology*, vol. 11, no. 5, 51:1–51:46, Jul. 2020. DOI: 10.1145/3400066.
- [137] A. van Opbroek, H. C. Achterberg, M. W. Vernooij, and M. de Bruijne, “Transfer Learning for Image Segmentation by Combining Image Weighting and Kernel Learning”, *IEEE Transactions on Medical Imaging*, vol. 38, no. 1, pp. 213–224, Jan. 2019. DOI: 10.1109/TMI.2018.2859478.

