Deep Learning-Based Affine and Deformable 3D Medical Image Registration

T.J.T. Roelofs

School of Science

Thesis submitted for examination for the degree of Master of Science in Technology.

Espoo 27.09.2021

Supervisor

Prof. J. Kannala

Advisor

Dr. S.U. Akram

Aalto University
School of Science

Aalto University
School of Science

Aalto University, P.O. BOX 11000, 00076 AALTO
www.aalto.fi
Abstract of the master's thesis

Author T.J.T. Roelofs		
Title Deep Learning-Based Affine and Deformable 3D Medical Image Registration		
Degree programme ICT Innovation		
Major Data Science		**Code of major** SCI3095
Supervisor Prof. J. Kannala		
Advisor Dr. S.U. Akram		
Date 27.09.2021	**Number of pages** 82+8	**Language** English

Abstract

In medical image registration, medical scans are transformed to align their image content. Traditionally, image registration is performed manually by clinicians or using optimization-based algorithms, but in the past few years, deep learning has been successfully applied to the problem. In this work, deep learning image registration (DLIR) methods were compared on the task of aligning inter- and intra-patient male pelvic full field-of-view 3D Computed Tomography (CT) scans. The multistage registration pipeline used consisted of a cascade of an affine (global) registration and a deformable (local) registration.

For the affine registration step, a 3D ResNet model was used. The two deformable methods that were investigated are VoxelMorph, the most commonly used DLIR framework, and LapIRN, a recent multi-resolution DLIR method. The two registration steps were trained separately; For the affine registration step, both supervised and unsupervised learning methods were employed. For the deformable step, unsupervised learning and weakly supervised learning using masks of regions of interest (ROIs) were used. The training was done on synthetically augmented CT scans. The results were compared to results obtained with two top-performing iterative image registration frameworks. The evaluation was based on ROI similarity of the registered scans, as well as diffeomorphic properties and runtime of the registration.

Overall, the DLIR methods were not able to outperform the baseline iterative methods. The affine step followed by deformable registration with LaPIRN managed to perform similarly to or slightly worse than the baseline methods, managing to outperform them on 7 out of 12 ROIs on the intra-patient scans. The inter-patient registration task turned out to be challenging, with none of the methods performing well consistently. For both tasks, the DLIR methods achieve a very significant time speedup compared to the baseline methods.

Keywords Medical image registration, deep learning, convolutional neural network, cascade, pelvic CT

Acknowledgments

This report contains the results of my industrial thesis project that serves as the capstone to my Master's in Data Science at KTH Royal Institute of Technology and Aalto University through EIT Digital Master School. In this thesis project, I was able to dive into an interesting academic machine learning topic and gain industry experience at the same time. My thanks go out to my two almae matres as well as EIT Digital, that all made this experience possible. I specifically want to thank Professor Juho Kannala from Aalto University as well, for supervising me in this final project.

I would also like to thank the team at MVision AI for the opportunity to conduct my research at their company, and their trust in my work. For me, the internship was a great chance for learning and professional development. Special thanks go out to MVision's CTO and my advisor Saad Ullah Akram, for always taking the time out of his busy schedule to share his expertise and suggestions.

Lastly, I also want to thank all the kind and special people that helped me along the way during my degree: To my family and friends back home, *ver weg maar toch dichtbij*; to Veronika, for her friendship and her feedback on an earlier version of this thesis; and finally, to all the other friends that I made and the amazing people that I met during my master's degree. Without you all, this would not have been possible. Thank you for bringing light to these times, in spite of the pandemic and everything else going on around us, and making this an unforgettable experience.

Espoo, 27.09.2021

Tim Roelofs

Contents

Symbols and abbreviations

Symbols

F	Fixed image (reference or target)
M	Moving image (to be aligned with F)
$T(M)$	The transformation (mapping) of M (to F)
p	A voxel in an image, $p \in P$, where P is a patch or image
$\mathcal{P}$	A collection of patches, $P \in \mathcal{P}$
c	A segment in image (or in a set of images F and M), $c \in C$
φ_c	Binary volume of c in F
μ_c	Binary volume of c in $T(M)$
v_A^c	Volume of a segment c in an image A
s_A^c	Surface of a segment c in an image A
$e_A^{c_\tau}$	Border region of tolerance τ of the surface of a segment c in an image A
$\mathfrak{f}$	A bin of intensity values for image F, $\mathfrak{f} \in \mathfrak{F}$
$\mathfrak{m}$	A bin of intensity values for image $T(M)$, $\mathfrak{m} \in \mathfrak{M}$
σ	Smoothing variable

Operators

$\lvert \cdot \rvert$	Cardinality of a set
$\Pr(\cdot)$	Probability
$\delta[\,\cdot\,]$	Kronecker delta
$U(\cdot)$	Uniform distribution

Abbreviations

BE	Bending energy
CBCT	Cone-beam CT
CC	Cross-correlation
CNN	Convolutional neural network
CT	Computed tomography
CV	Computer vision
DL	Deep learning
DLIR	Deep learning-based image registration
DoF	Degrees of freedom
DSC	Dice similarity coefficient
DVF	Deformation vector field
FFoV	Full field of view
GAN	Generative adversarial network
HU	Hounsfield unit
JC	Percentage of voxels with a non-positive Jacobian determinant
MI	Mutual information
ML	Machine learning
MR	Magnetic resonance
MSE	Mean squared error
NCC	Normalized cross correlation
NMI	Normalized mutual information
OAR	Organs at risk
PET	Positron emission tomography
ReLU	Rectified linear unit
ROI	Region of interest
RL	Reinforcement learning
sDSC	Surface Dice similarity coefficient
ST	Spatial transformer
STN	Spatial transformer network
SVF	Stationary velocity field

1 Introduction

In radiotherapy, different types of imaging are used in the diagnosis, prognosis, treatment, and aftercare of cancer. For these purposes, information from images taken from different viewpoints, at different times, using different modalities, or taken in different dimensionalities may need to be combined. To aid this, these images can be transformed to align their image content. This is called *image registration*.

Image registration is increasingly used in clinical radiotherapy practice, for example in diagnostics and tumor staging, treatment planning and guidance, response monitoring, and patient positioning in linear accelerators (LINAC) [1].

In the past, registration was mostly performed manually by clinicians [2]. However, today it can also be done (semi-)automatically, for which optimization-based tools have mainly been adopted. These tools can perform registrations by iteratively optimizing the image similarity. This is useful, as it is less time- and labor-intensive, and leads to more standardized results, whereas in manual registration, the quality of the alignment is highly dependent on the expertise of the clinician [2].

Traditional iterative optimization-based registration methods (hereafter referred to as iterative methods) have two main drawbacks: Firstly, the iterative calculation is very slow, with runtimes of tens of minutes being the norm; secondly, due to the existence of many local optima around the global one, optimization algorithms suffer from premature convergence or stagnation, especially for multi-modal registration (registration of scans of different modalities, i.e., different types) [3]. For medical image registration, this is a significant issue as the accuracy requirements for clinical use are very high (sub-millimeter) for diagnosis and surgery guidance purposes [4].

In recent years, there has been an interest in improving the quality and speed of automatic image registration through the use of deep learning (DL), a subfield of machine learning (ML). The aim of this is to improve the quality and speed of registration, making it more widely usable. For a variety of medical imaging tasks, such as image segmentation, DL has achieved state-of-the-art performance [5]. However, for medical image registration, DL has yet to convincingly outperform state-of-the-art iterative methods in terms of registration quality [5], although interest in DL-based image registration (DLIR) is increasing [2]. Furthermore, in contrast to other fields, the research community has not settled on the best way to utilize DL for image registration, with various approaches being investigated in literature [6]. This is partly due to a lack of benchmarking datasets and accepted evaluation methods, which hinder the comparison of different methods in the literature thus far [7], especially for more difficult registration problems, such as multi-modal registration. Thus, there is a clear need for more evaluation and comparison of DL methods on a variety of image registration tasks.

1.1 Research aim and relevance

1.1.1 Research aim

This research aims to apply DL to the task of multistage registration of full-field-of-view (FFoV) computed tomography (CT) scans of the male pelvis.

As medical image registration research is ongoing into a wide variety of tasks, it is imperative to carefully define the scope of this project. Viergever, Maintz, Klein, *et al.* [1] propose multiple criteria that can be used to classify medical image registration projects. These include the dimensionality of the images, the nature of the transformation, and how subjects are involved. Based on these criteria, the task at hand in this research can be defined using the following labels:

- *Dimensionality:* 3D-3D – Registration of two 3D scans.
- *Nature of registration basis:* Intrinsic – Intrinsic registration, i.e. registration based on image information generated by the patient, without any foreign objects (e.g. markers) introduced into the image space.
- *Nature of the transformation:* Affine and deformable – The final registration will be deformable, though as is usual in the literature, first an affine registration step is performed. Both stages will be covered in this work.
- *Domain of the transformation:* Global and local – The affine registration step is intended for a global registration, with the deformable registration step handling the local, more precise registration.
- *Degree of interaction:* Automatic – No input, except for the scans to be aligned, should be required for the registration task.
- *Modalities involved:* CT-CT – The registration is done between CT scans. This makes it a uni-modal registration task.
- *Subjects involved:* Intra-subject and inter-subject – CT-CT registration is mainly used for intra-subject registration. This is thus the primary focus of this thesis. However, as inter-subject registration also has clinical applications, evaluation is also done on this task.
- *Objects involved:* Male pelvic region – The chosen dataset contains scans of the lower male body, around the prostate area.

1.1.2 Research context

This thesis is conducted as part of a research and development project by MVision AI Oy[1], a medical startup based in Helsinki, Finland that develops AI solutions for radiotherapy, with the aim to improve cancer patients' quality of life both during and after treatment. MVision AI has a growing customer base in Finland and abroad in Sweden, Spain, and Germany amongst other countries.

Next to the segmentation and contouring software it currently offers, MVision aims to automate a range of other tasks in medical imaging through DL, including registration. The task of CT-CT registration of the male pelvic area was chosen by MVision AI, as it was determined to be one of the most important registration tasks.

[1] https://www.mvision.ai/

1.1.3 Clinical relevance

Medical imaging, in general, is increasingly used in clinical practice for diagnosis, planning treatment, guiding treatment, and monitoring disease progression [8]. This naturally leads to an increased demand for registration, which is a critical component of quantitative image analysis workflows [7]. The most prominent applications of medical image registration are: (1) registration of multi-modal scans, in order to combine the information acquired to facilitate diagnosis and treatment planning; (2) registration of scans taken at different times (i.e. longitudinal studies), especially for treatment planning, where anatomical changes need to be monitored; and (3) atlas creation, so the production of a statistical atlas to model (the anatomical variability within) a population [9]. Furthermore, image registration can be used to improve other aspects of the medical imaging workflow, for example, segmentation and dose calculation, where current registration techniques have already led to significant performance increases [7].

Viergever, Maintz, Klein, *et al.* [1] express the hope that through DL, image registration will become an integral part of the entire spectrum of routine medical imaging. However, thus far, the usage of DL in medical imaging remains in the stages of infancy [4]. Further DLIR research is needed to develop methods that lead to clinically relevant improvements, and to bring this research knowledge to the market.

So far, to the best of my knowledge, the only company that offers DLIR software is TheraPanacea[2]. Its SmartFuse module, part of its ART-Plan software, offers AI-based image registration and fusion for a variety of modalities including CT, cone-beam CT (CBCT), positron emission tomography (PET), and magnetic resonance (MR). A number of other companies offer automated image registration algorithms not based on DL. These are MIM Software Inc.[3], Mirada Medical Ltd.[4], Koninklijke Philips N.V.[5], RaySearch Laboratories AB[6], and Varian Medical Systems[7]. This shows that there is currently a market gap for DLIR software, which MVision AI could potentially fill. Note that, as an academic work, developing this project into a product that can be used clinically is far beyond the scope of this thesis. However, being an industrial thesis, MVision AI does aim to reach this developmental stage in the future, and thus this thesis is envisioned to contribute to the development of a clinical DLIR product.

Next to the general importance of DLIR, the specific task of CT-CT registration is also especially clinically relevant, because CT is an important modality in radiotherapy practice. In treatment planning, CT is typically used for the contouring of organs at risk (OAR), and other structures. It is also the only 3D modality used in dose calculation [10].

The registration of multiple CT scans can have various use cases. In image-guided radiotherapy, CT scans are used for treatment planning (*planning CT scans*), as they are used for OAR contouring and the calculation of the radiation dosage [11]. Even though the course of the radiation plan usually spans over more than 7 weeks,

[2] https://www.therapanacea.eu/ [3] https://www.mimsoftware.com/
[4] https://mirada-medical.com/ [5] https://www.philips.com/
[6] https://www.raysearchlabs.com/ [7] https://www.varian.com/

it is usually planned on a single planning CT scan [12]. To check for any anatomical changes that would lead to a change in the treatment plan, images taken at the time of treatment are used to reduce the geometric uncertainty between the planning and delivery of the treatment [12]. In this way, image registration can be used to translate segmentations or annotations between points in time [13]. It should be noted that for the male pelvic region, rescans are not as widely used as they are for, for example, the breast region. However, in recent years, they have become increasingly common, with the increased usage of hypofractionated stereotactic body radiotherapy (SBRT) for the pelvic region, which warrants rescans due to the higher radiation dosages used.

Note that these rescans are not always CT scans. Instead, CBCT scans are often used, as they are quicker and use a lower radiation dosage. However, as their quality is not sufficient to perform dose calculation, in case of a major change, patients are still referred to a rescan CT [11].

1.1.4 Academic relevance

As noted, in the field of medical image registration, research is ongoing into a wide variety of tasks. The chosen focus (see section 1.1.1) is mostly in line with current trends in research: Fu, Lei, Wang, *et al.* [5] report, for example, that 60% of papers in medical image registration are about 3D-3D scans and 72% include deformable registration. However, multistage registration where DL methods are used for both global and local image alignment has not been widely researched: Most publications on deformable registration use iterative methods for the global registration step.

Furthermore, as stated, as of yet, there is no accepted best practice in the field, and a lack of benchmarking makes it difficult to compare methods. Therefore, studies such as this thesis that explicitly compare various methods can shed light on the relative performance of these methods, and contribute towards finding areas for future research.

Considering the modality and region chosen for this thesis, this work investigates a relatively unexplored task of CT-CT registration: The (FFoV) male pelvic region. While this is an increasingly clinically relevant task, it has rarely been attempted in the literature, with almost all CT-CT registration studies focusing on the lungs.

In the following section, previous developments in image registration will be reviewed, and the used methods will be detailed.

2 Background

2.1 Medical image registration

Registration can be more formally defined as the determination of a geometrical transformation that aligns different sets of data [14]. These sets can consist of images, or specifically in the radiology domain, three-dimensional images acquired by tomographic modalities such as CT, MR, and PET. However, on a short note, image registration is not limited to the medical domain (though it is the largest application area), as it is also used in other fields, such as in remote sensing [15]. Furthermore, next to images, a set of data can, for instance, also be a physical space [16]. In the medical domain, image registration need not necessarily be between images of the same patient (intra-patient registration), but can also be between images of different patients or a patient and an atlas [14].

In image registration, a moving image M (also called floating or sensed image) is aligned to a fixed image F (also called target or reference image) through some transformation T such that $T(M)$ is sufficiently similar to F. The level of similarity depends on the quality of the registration and the deformation model that is used. The latter determines the nature of the calculated registration: This can range from a simpler global registration – such as a rigid (translation, rotation) or affine (translation, rotation, scaling, shearing) registration[8] – to a local, deformable registration. Local registrations can be parameterized in various ways as well: In the literature, different deformable registration models are distinguished, for example, those based on curves (higher-order parametric transforms) versus non-parametric models. Local models, especially more complex ones, can lead to more accurate registrations, but these are also more challenging to calculate. To deal with both larger and smaller displacements, generally, global and local models are used as subsequent steps [17], for example, with a deformable method superseding an affine method. In the literature, deformable methods have also been stacked, e.g. in a multi-resolution, coarse-to-fine manner to predict deformations of varying sizes more accurately [18].

Note that according to the aforementioned definition, the output of a registration is not a transformed image, but the transformation T itself, which can then be applied to an image. This transformation T can consist of transformation parameters such as a transformation matrix in the case of global registration, or a deformation vector field (DVF) in the case of local registration. The former consists of 16 values[9] determining the transformation of every point in the image; The latter consists of individual transformations of each point. Once T is determined, the transformed moving image $T(M)$ can be calculated. In the case of multi-modal imaging, this image is then often combined with fixed image F in a process called *image fusion*, which can be as simple as summing the intensity values of the two images [14]. Figure 1 shows an example of the fusion of two registered scans. This fusion process, however, lies beyond the scope of this thesis. In the case of longitudinal studies, where changes

[8] Note that in registration literature, the term *rigid* is sometimes used to refer to any type of global registration, even if it is a linear, affine, or projective transformation.

[9] This number only holds for 3D-3D registration.

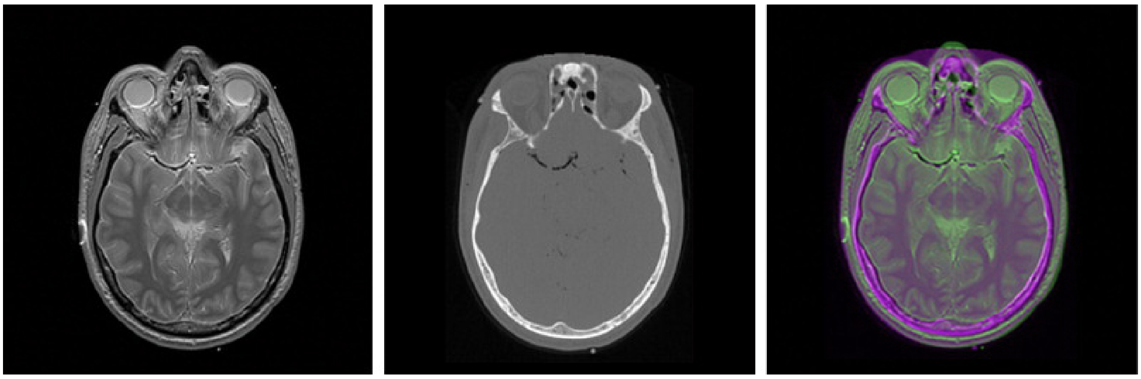


Figure 1: Axial slice of an image fusion (right) between a cranial PD-weighted MRI scan (left) and CT scan (center). Adapted from [19].

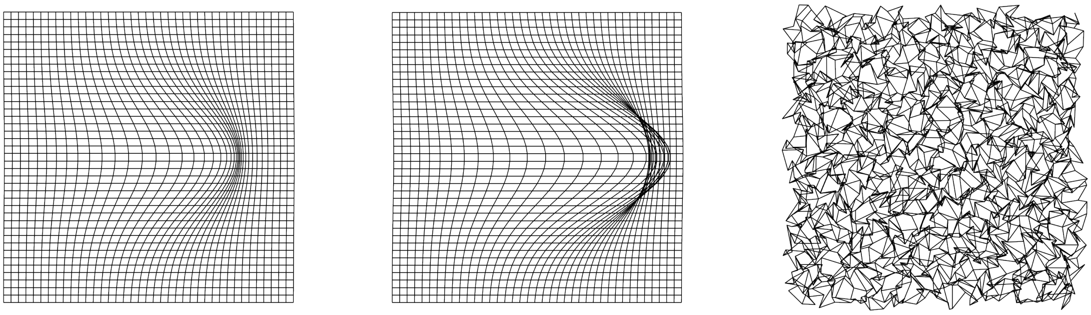
between an earlier image M and a more recent F are investigated, T can also be applied to other images, e.g. a dose distribution image that was calculated based on M.

Image registration is an ill-posed problem, as multiple deformations could transform image M to resemble image F [20]. For a local registration, the aim of the registration may therefore not only be to find a transformation T that accurately aligns the intensities of M to F, but to find a T that does so in an anatomically realistic manner. This is especially important for longitudinal applications (i.e. those featuring images taken at different times), where T is often not only used to deform M, but also to deform related images in other modalities or associated spatial distribution maps (such as dose maps) [21]. How to achieve these anatomically realistic deformations is still an open question in the image registration community [22]. One of the conditions of an anatomically realistic deformation that is commonly used as a constraint is diffeomorphism: That the transformation is a one-to-one, smooth and continuous mapping, with derivatives that are invertible [23]. These concepts are illustrated in Figure 2. Oftentimes, simpler constraints, such as smoothness of the DVF, are often used in the literature [24].

It has been argued, however, that diffeomorphic DVFs do not guarantee realistic results: For example, in the case of moving organs, sharper deformations in the organ boundaries are required in order to preserve the anatomy [22]. Other methods to ensure realistic results have thus also been proposed.

Lastly, it is worth noting that from an implementation point-of-view, the transformation T is often calculated as a backward mapping, so mapping homologous locations from the fixed space to the moving space. This is done so that, to determine $T(M)$, the value of each of its voxels can be pulled from a(n interpolated) position in M, which leads to a less complex interpolation task [9].

As stated, the goal of registration is to determine T. To do this, a wide variety of methods can be used, including iterative methods and DL-based methods. The former traditionally consist of an optimizer that determines a transformation to maximize a certain similarity function between $T(M)$ and F. These methods can



(a) A smooth, invertible deformation.

(b) A smooth but non-invertible deformation.

(c) A non-continuous deformation.

Figure 2: Examples of 2D deformations represented as grids. Only the first grid has the desired diffeomorphic properties.

be described by three components: (1) a transformation model, (2) an objective function, usually a similarity function, and (3) an optimization method [9], [25]. Just like iterative methods, DL-based methods also use a transformation model, and during their training, they use an objective function. However, rather than using an optimization method to maximize the objective function at test time, in DLIR, a deep neural network learns to determine transformations in a training phase, so that during deployment, the model can output a transformation based on the images in a single pass rather than iteratively. The two components that are used in both registration paradigms, namely the transformation model and the similarity function, will be detailed in the following sections.

2.1.1 Transformation model

There are many ways in which a transformation can be modeled. Models can range from simple, global models with few degrees of freedom (DoF), to complex and computationally expensive local models.

For global registrations, the nature of the transformation depends on the transformation parameters that are estimated. The simplest case is a rigid (euclidean) transformation, consisting only of a translation and rotation, with just 6 DoF. The similarity transformation adds scaling, with one extra DoF. Affine transformations add shearing, leading to 12 DoF. Lastly, there is the projective transformation, which adds projection, has 15 DoF.[10]

For local transformations, the number of DoF is always much larger, as different parts of the image can be deformed in different ways. This causes iterative optimization-based methods for these transformations to require a lot of computation time, which has been a major drawback for many clinical applications [25]. However, how complex the transformation is, depends on the model used.

Sotiras, Davatzikos, and Paragios [9] distinguish various types of deformation

[10] Note that all the numbers of DoF given here only apply to 3D-3D transformations.

models. One type is deformation models derived from interpolation theory, referred to as spline models by Mani and Arivazhagan [25]. These models consider displacements in a restricted set of locations and interpolate them to the rest of the image. Free Form Deformations (FFD) based on B-splines, one of the models from this category, have gained wide acceptance in and are used extensively by the medical image registration community.

Next are models inspired by physical models. This includes linear and nonlinear elastic body models; viscous fluid flow models; diffusion models, which are usually based on the Demons algorithm; and flows of diffeomorphisms, commonly using the Large Deformation Diffeomorphic Metric Mapping (LDDMM) framework [25] or (time-)stationary velocity fields (SVFs). The latter two are especially popular model types.

Last are knowledge-based models, which are models that impose some sorts of known constraints. This includes statistically constrained models, and biophysical models, e.g. biomechanical models of the prostate or tumor growth models.

In DLIR, Most recent deformable methods parameterize the deformation model using a dense displacement field, where a displacement is given for every voxel [26]. Spline models are only used occasionally, and knowledge-based models are virtually never used.

2.1.2 Similarity function

For the objective function, a distinction can be made between extrinsic and intrinsic methods. As this thesis is focused on intrinsic registration, only this category will be discussed.

Intrinsic registration can be based on aligning features, segmentations, or image intensity [25]. Feature-based methods are based on corresponding landmarks detected in the images. These methods used to be the most commonly used [27]. Segmentation-based methods are based on aligning contours identified in the images through image segmentation (either manually or automatic). Lastly, image intensity-based methods match intensity patterns between the two images. To do this, a similarity function is used; Mean square error (MSE), cross-correlation (CC) based metrics, and mutual information (MI) based metrics are most commonly used [25]. Of course, different objective functions can also be combined.

One type of objective function used in DLIR that is not used in iterative registration methods is similarity based on a ground-truth transformation. Since for DLIR models, the objective function is used during training rather than inference, training data with ground-truth transformations can be used as a method of supervising the model. More information on the different types of supervision in DLIR is given in section 2.5.

Now that medical image registration has been discussed in general terms, in the next sections, iterative registration and DL-based registration will be discussed separately in more detail.

2.2 Iterative registration

Iterative registration uses an optimization method to maximize the chosen similarity metric based on the chosen transformation model. A good optimization algorithm can determine the transformation reliably and quickly; For local registrations, this can be challenging due to the large number of parameters required to describe the transformation.

Because of the computational complexity of optimization, the aim of the registration is not to find the globally optimal solution, but rather, finding a sufficiently good registration [25].

Commonly used optimization methods include (stochastic) gradient descent methods, conjugate gradient method, quasi-Newton methods, Powell's method, downhill simplex optimization, and Levenberg-Marquardt algorithm [9], [15], [28], [29]. Most of these are continuous [9], non-linear [15] optimization methods. Various methods were compared by Klein, Staring, and Pluim [28] using MI similarity, a B-splines deformation model, and various input modalities and sizes. In their experiments, a stochastic gradient descent technique came out as the preferred method; though quasi-Newton and nonlinear conjugate gradient methods resulted in slightly higher precision, these used ten to a hundred times more computation time.

2.2.1 Frameworks

To implement iterative registration methods, various frameworks exist. These frameworks are often used in DLIR research to provide a baseline to which the performance of a new method can be compared. Nazib, Fookes, and Perrin [30] made a comparative analysis of the most popular tools.

The Image Registration Toolkit (IRTK) [31] is a popular early image registration tool that sequentially used rigid, affine, and non-rigid registrations using an FFD model based on B-splines. Normalized MI (NMI) is used as a similarity metric. The method was originally conceptualized for breast MR images. An extension of IRTK is offered in NiftyReg [32], suitable for GPU execution and parallel processing. NiftyReg also adds the normalized CC (NCC) similarity metric. NiftyReg was used as a baseline for DLIR comparison by Guo [33].

Elastix [34] is another one of the most popular registration tools, that comes with a set of registration algorithms. The deformation models available are rigid, affine with different numbers of DoF, and B-spline with physics-based control points in uniform and non-uniform grids. Optimization methods include gradient descent, quasi-Newton, and nonlinear conjugate gradient methods, as well as a number of stochastic gradient descent methods. It also includes most common loss functions, such as MI, CC, and MSD. In DLIR literature, Elastix was used for comparison in [35]–[37].

Advanced Normalization Toolkit (ANTs) [38] is an open-source framework that is mainly used for its symmetric diffeomorphic normalization method (SyN). As its name suggests, SyN uses a symmetric flow-of-diffeomorphisms deformation model. It has been shown to perform well in the literature, and was used as a comparison method in various DLIR publications [26], [33], [39], [40].

Table 1 shows a short comparison of the four methods discussed. Elastix is the framework with the most flexibility for the user, giving many options for the similarity metric used. Furthermore, it also offers a variety of optimizers. ANTs also offers flexibility in the type of registration, and a few different optimization techniques or similarity metrics, although less than Elastix. IRTK and NiftyReg are less flexible, however, having implemented just one deformable registration algorithm, and two rigid/affine algorithms. In the table, the results of a test by Nazib, Fookes, and Perrin [30] on two scans resampled at 10% of their original size are also shown. In this experiment, ANTs and Elastix achieved top performance; ANTs took significantly longer to finish its task, however. Similar results were reported by Mogadas, Sothmann, Knopp, *et al.* [41], who noted that ANTs (with a 2.1 mm average target registration error) outperformed Elastix (with 2.2 mm) and NiftyReg (with 2.7 mm) on the DIR-Lab dataset (see section 2.7.2), although the registrations took around 2.5 times as long as Elastix and more than 10 times as long as NiftyReg.

Method	Types	Similarity metrics	Avg. CC after reg.*	Run-time* (s)
IRTK	Rigid, affine, deformable	NMI	0.7486	00:02:42
NiftyReg	Rigid, affine, deformable	NMI, NCC	0.8556	00:16:32
Elastix	Rigid, affine, deformable	MI (variants), CC (variants), MSE, etc.	0.9489	00:05:03
ANTS	Translation, rigid, similarity, affine, deformable (var. models)	MI, NCC, MSE	0.9545	01:23:44

Table 1: Comparison of four iterative registration frameworks. The columns marked with * show values taken from an experiment by Nazib, Fookes, and Perrin [30]; See their paper for details.

2.3 Deep learning for image registration

The previously discussed iterative registration methods have two main drawbacks: Firstly, the optimization process is time-intensive; and secondly, their performance is not always good, with algorithms suffering from premature convergence or stagnation. To mitigate these issues, the image registration community has started focusing on DL. DL registration methods are generally not iterative, so their model inference is very fast. Secondly, DL methods are also a solution to the risk of falling into local minima [15], leading to better results. Furthermore, DL also alleviates the burden of choosing and refining parameters and features that are important to achieve high performance in iterative image registration [3].

DLIR is a relatively new field. Within the medical imaging domain, research has been done into ML methods for automation since the 1960s [42]. DL, however, only became feasible in the early 2010s and has been rising in popularity since. In computer vision (CV), a breakthrough came in 2012 with AlexNet [43], a convolutional neural network (CNN) that, for that time, reached remarkable accuracy in a difficult image classification task, greatly outperforming traditional CV algorithms. After this, DL started being used in many CV tasks, including for medical applications. Within medical imaging, active research areas for the application of DL include organ detection and segmentation, landmark localization, treatment planning, follow-up prediction, and more [3]. In the case of image registration, DL-based methods have already been more popular than iterative methods in the literature since 2017 [15].

In the next sections, DLIR will be discussed in more detail, starting from the building blocks of DLIR methods, the various types of DLIR algorithms that exist, and literature on DLIR that was tested on CT scans specifically.

2.4 Network elements

DL is based on deep neural networks, which have the ability to learn tasks, making them different from iterative methods. In this section, some architectural building blocks of registration networks will be briefly discussed, in order to be able to understand the various methods used in DLIR.

2.4.1 Neural networks

The artificial neural networks used in DL are compositions of multiple layers: an input layer, one or multiple hidden layer(s), and an output layer. In vanilla neural networks, the hidden and output layers are made up of *perceptrons*: Basic computational elements that output a non-linear function of a weighted sum of their inputs. In essence, *training* a neural network means determining these weights. As these non-linear functions are differentiable, they allow the weights to be updated so that the loss is minimized during training through the *backpropagation* algorithm. This algorithm computes the gradient of the loss function with respect to the weights of the network in order to update them: A loss value calculated on the network output gets passed through the network backward, from the output layer to the input layer, using the chain rule to determine how the weights should be changed.

To use images as inputs in a vanilla neural network, one can simply take the intensity value as each pixel (or for 3D images: each voxel) as the value for a node in the input layer. However, this essentially flattens the input image, meaning that spatial information is lost. Furthermore, since the weights are tied to specific neurons (and thus inputs), the network is not invariant to translations.

2.4.2 Convolutional neural network

To solve this problem, in order to use any type of image in DL, CNNs are commonly used. Rather than simple layers of perceptrons, these add convolutional layers, which perform convolutional operations. These consist of moving a filter (a smaller grid of weights) over the input grid (in the first layer: the image). In each moving step, corresponding elements of the two grids are multiplied and summed to compute one scalar value of the output, a grid called a feature map. This feature map is then used as the input for the subsequent layer. The aim of training a network with these layers is thus to learn the weights of these filters.

Similar to how hidden layers in fully connected (vanilla) neural networks can be made more complex by adding more perceptrons, more filters (and thus feature maps) can be added as separate channels in a CNN. Convolutional layers can have hundreds of filters and channels, depending on the architecture.

This architecture solves the issues of the vanilla neural network: Every input feature gets connected to its activation signal in the feature map through the same filter (i.e. the same weights). This means that all of the input features (i.e. all pixels/voxels in the image) share the filter weights, leading to space invariance. Furthermore, the number of weights is greatly reduced, lowering the computational complexity.

Another element of CNNs is the pooling layer, which is used to scale down the feature maps by summing or averaging a small grid of the feature map. There are no learnable weights of these layers, which is why in some resources, they are not seen as separate layers [42]. Scaling down can also be done using strided convolutional layers. In fact, this is less computationally expensive than having a convolution layer followed by a pooling layer; however, if very deep features with high resolution are desired (e.g. in pixel-level prediction tasks), it may not be preferred [44].

Figure 3 illustrates the block of convolution and pooling common in CNNs. These layers are often combined with batch normalization (BN) or activation layers.

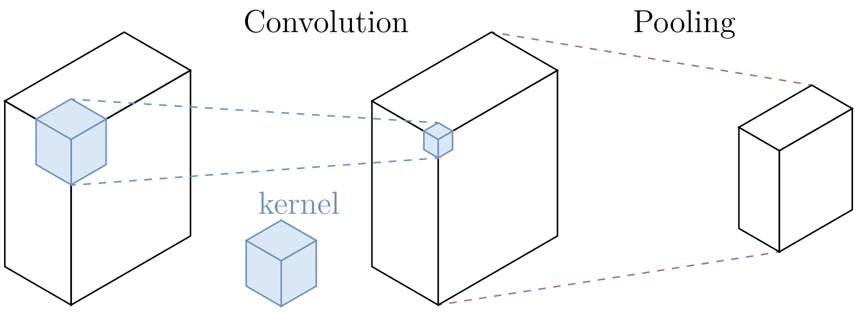


Figure 3: A convolution and pooling operation in a CNN. The convolution uses a kernel with learnable weights. Note that the size of the output of a convolution can be different than the input depending on the stride and padding used.

Many types of CNNs exist. Two architectures commonly used in registration, ResNet and U-Net, will be discussed in the following subsections.

2.4.3 ResNet

Since the introduction of CNNs, their architectures have become increasingly deep. Because of this, gradients can vanish during backpropagation, meaning that when propagating the loss back through the layers of the network, it gradually decreases, making the updates for the weights in the earlier layers negligibly small. Furthermore, accuracy saturation or degradation can occur in deeper networks.

Residual neural networks, or *ResNets*, originally introduced by He, Zhang, Ren, *et al.* [45], are networks that introduce skip connections to mitigate this. As the name suggests, these connections skip one or more layers in the neural network. Figure 4 illustrates how a ResNet building block with a skip connection can look. Because of these connections, during backpropagation, error values are already passed to layers further down the network, rather than only to the previous sequential layer. One way to envision this is that skip connections essentially turn the deep network into an ensemble of relatively shallow networks [46]. Thereby, they combine the benefits of both shallow and deep networks.

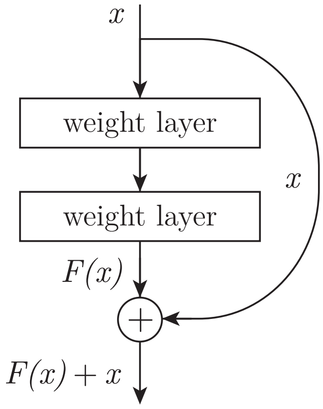


Figure 4: An example of a residual learning building block, where the input skips two layers.

He, Zhang, Ren, *et al.* [45] introduced a number of default ResNet architectures that they tested on the ImageNet classification challenge, namely ResNet-18, -34, -50, -101, and -152, referring to the number of layers in these networks. ResNets, and also these architectures in particular, have grown exceptionally popular as DL approaches [47]. Multiple variants of the ResNet architectures have been introduced with new types of residual blocks, such as the Wide ResNet and ResNeXt architectures.

2.4.4 Encoder-decoder architectures

CNNs are often used for classification or regression problems, where one or multiple images are taken as an input, and one or multiple (numerical) values are predicted. In DLIR, such an architecture can be used for global transformations, as these are parametric; For example, in the case of a global transformation, the model can simply output a transformation matrix, which means it is necessary to predict at most 15 values (in the case of a projective transformation) per image pair or, for example, just 12 (3×4) in the case of an affine model or 6 for a rigid model (these are the number of DoF; see section 2.1.1).

In many visual tasks, especially in biomedical imaging, the output is more complex than that, as the desired output should include localization: For instance, rather than indicating the presence of an object, the network should tell where in the image the object is located. In the case of DLIR, this is the case for local models, where the deformation can change for each voxel.

In order to do so, Long, Shelhamer, and Darrell [48] introduced fully convolutional networks. This architecture was improved by Ronneberger, Fischer, and Brox [49] to create U-Net, a CNN developed for biomedical image segmentation. U-Net features an encoder-decoder architecture, where the first part is a regular CNN with successions of convolution, activation, and pooling layers; Then in the second part of the network, the pooling operations are replaced by upsampling operations, increasing the resolution. Symmetrical skip connections are formed between the same-resolution layers in the two parts, through which fine-grained details can be recovered. The architecture is illustrated in Figure 5.

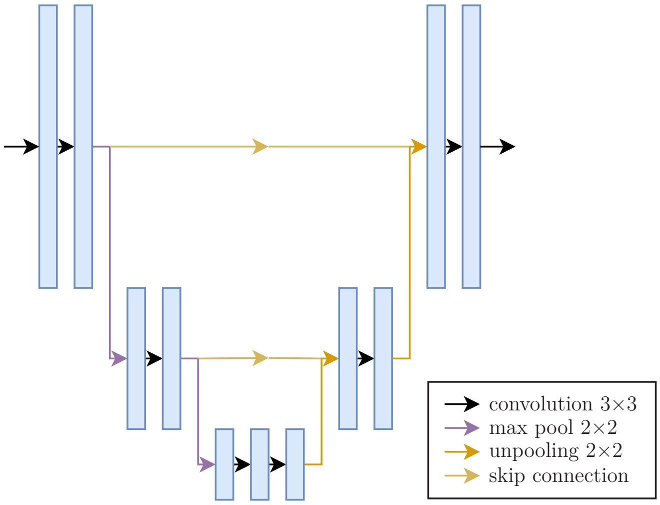


Figure 5: An example of a U-Net architecture.

Encoder-decoder architectures are commonly used in medical imaging in registration and other tasks such as segmentation [3]. Their usage in registration will be explained in the next section.

2.4.5 Spatial transformer network

A key framework for registration models is the spatial transformer network (STN), introduced by Jaderberg, Simonyan, Zisserman, *et al.* [50]. A spatial transformer (ST) is a model block that explicitly allows the spatial manipulation of data within a CNN. This can be used to correct for larger, more complex transformations of the model input, for example, to straighten an image in a text recognition task. Of course, in image registration, finding a suitable transformation is the goal of the network in itself.

The ST consists of three subsequent components: (1) A localization network which, through a number of hidden layers, regresses the transformation parameters that should be applied to the input feature map. This can be a U-Net that outputs a deformation map. (2) These are then applied to a grid over the feature map in the grid generator, to create a warped grid. Lastly, (3) the sampler applies the warped grid to the original feature map to produce the output. As the whole module is differentiable, it can be inserted anywhere into CNN architectures and learn through backpropagation. A CNN with an ST is called an STN.

In registration, STNs are of key importance. They are commonly used in the following manner: A U-Net is used to output transformation parameters or a transformation field directly (T), such as a sampling grid that determines from where in M every voxel in $T(M)$ should be sampled. The shape of DVF (grid) is that of M with 3 channels, showing the coordinates from where a sample is taken. Next, the sampler applies this to the moving image M. Because of this last step, a loss can be calculated based on $T(M)$, rather than on T directly. This makes training a network easier, as a similarity between $T(M)$ and F can be calculated to direct the training, rather than comparing T to the true transformation, which is often unknown, or having to output $T(M)$ directly out of the network. This general structure is illustrated in Figure 6.

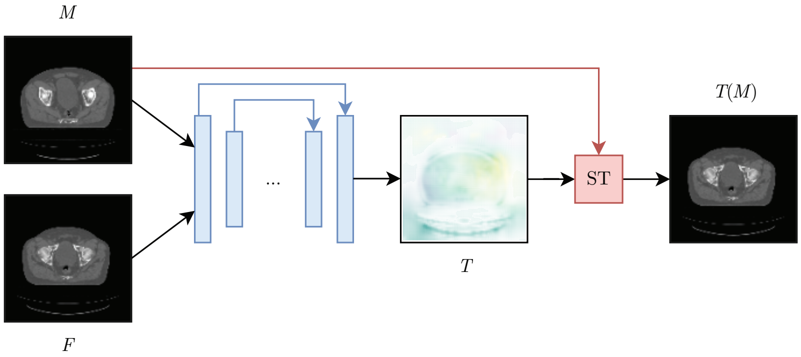


Figure 6: An illustration of an STN with a U-Net, a common DLIR architecture.

2.5 Training methods

Most DLIR models found in the literature are based on the architectural building blocks previously discussed, although other types of models have also been used. To learn the weights of these networks, some kind of training method needs to be used, based on an objective function (see section 2.1.2). There are various streams of training approaches that have been explored within DLIR. In this section, these are briefly outlined.

In the literature, especially in survey papers on DL in medical image registration, of which multiple have been published recently, multiple ways to classify ongoing research have been proposed. In these, firstly, often a distinction is made between learning-based iterative methods and (un)supervised learning methods [2], [15].

The former includes deep similarity metrics, methods that use DL for calculating image similarity but use iterative methods for determining the transformation, and reinforcement learning (RL), a type of ML that iteratively learns through trial and error. While these approaches may lead to better registration accuracy, their speed is still held back by their iterative nature at test time.

In contrast, in the latter category, once a model has been trained, a registration can be done in one pass, making it almost instantaneous. In this category, supervised, unsupervised, and weakly supervised methods can be distinguished [3], [5]. In (weakly) supervised learning, some type of annotated (ground-truth) information is known: Ground-truth deformation maps in supervised learning, or deformations of points or objects in weakly supervised learning. This type of learning thus needs large-scale annotated data sets, and largely depends on the quality of these available annotations [2]. Unsupervised registration on the other hand, which has more recently risen in prominence, has fewer data requirements. Usually, unsupervised methods use the similarity between the warped images $T(M)$ and fixed images F to assess the registration quality. However, currently it is not frequently nor successfully used in multi-modal cases, due to the difficulty associated with defining efficient multi-modal similarity measures [2], [3].

In the literature, supervised and unsupervised registration methods are most frequently used, reportedly for 26 and 28% [5] or 37.5 and 25% [3] of DLIR publications respectively.[11] Since in this work, supervised, unsupervised and weakly supervised techniques are used, these types will be discussed in more detail in the following sections. Two upcoming areas in DLIR research, RL and generative adversarial networks (GANs), are furthermore discussed in Appendix C.

2.5.1 Supervised transformation estimation

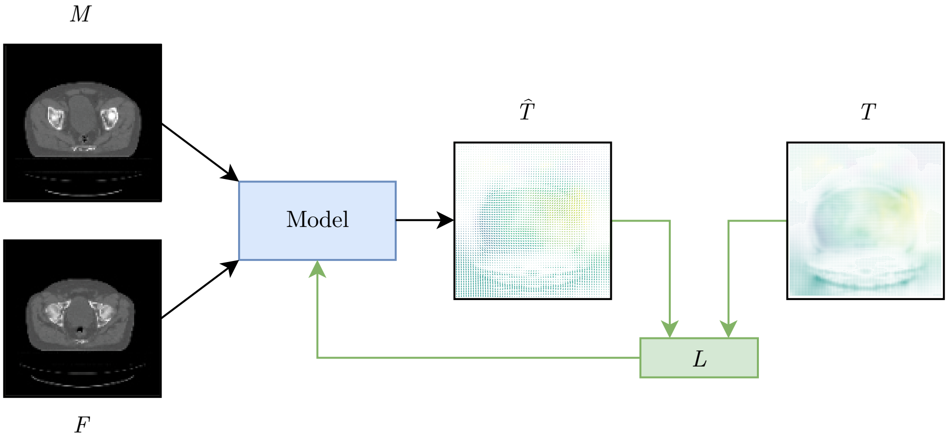


Figure 7: The supervised transformation estimation framework.

[11] The difference in numbers between these two studies may be explained by the fact that different selection criteria were used to select papers.

In supervised learning, the training data contains both a pair of input data and the correct model output. The model must thus learn to, based on the inputs, predict an output as close as possible to the correct output. In supervised methods for image registration, this means that the training data contains not only scans but also the ground-truth deformation between them. In the case of a global transformation, this ground-truth deformation (in its simplest form) entails the transformation parameters. In the case of local (deformable) transformations, the ground-truth deformation takes the form of a DVF. The supervised learning method is illustrated in Figure 7.

As Boveiri, Khayami, Javidan, *et al.* [3] note, for supervised learning, CNNs and U-Nets are the predominant techniques for supervised global and supervised deformable transformation prediction.

For supervised methods, acquiring the required training data can be challenging. Moreover, the quality of the registration model is heavily dependent on the quality of the ground-truth data. For example, Sentker, Madesta, and Werner [51] uses an existing iterative registration method to acquire transformations. While they were able to achieve a good performance, a drawback of this method is that an algorithm trained using such data may only reach a performance up to the quality of the traditional registration approach used for the transformation generation. As improving registration performance over traditional methods is one of the main reasons for the use of DLIR, this is not desirable. However, it must be said that currently, DLIR generally does not manage to outperform traditional methods, or not significantly, so depending on the application, this may not be an issue.

Another method of acquiring ground-truth transformations is to use artificial transformations. Of course, the challenge here is to generate realistic transformations: If the method used for generating transformed scans does not manage to output transformations that are sufficiently similar to real (anatomical) deformations, an algorithm trained on this data will poorly generalize to real scans. As Haskins, Kruger, and Yan [2] argue, this is especially true for the simulation of realistic deformable transformations, which is much more challenging than the simulation of realistic rigid transformations. Fu, Lei, Wang, *et al.* [5] report two options to simulate transformations. Firstly, random transformations can be applied to scans. Sun, Moelker, Niessen, *et al.* [52], for example, used this method in combination with image translation to create a synthetic transformed image in another modality in order to train networks for multi-modal CT-US registration. While results on the synthetic data were good, results on real scans remained poor. To generate more realistic deformations, Fu, Lei, Wang, *et al.* [5] note that models can be used, e.g. a statistical appearance model based on a few registered samples, or a respiratory motion model.

On a side note, in contrast to this, in a recent publication, Hoffmann, Billot, Iglesias, *et al.* [53] show that it is possible to train a well-performing model on completely unrealistic training data, challenging the idea that synthetic data for registration tasks needs to be realistic. The authors developed a tool for nonlinear contrast-agnostic registration (e.g. registration between different types of MR scans), using no actual scans for training: Rather, they use segmentation maps that are warped using large elastic deformations, from which synthetic images are generated with arbitrary contrast. The segmentation maps they used were either actual segmentation maps from brain MR scans or, more interestingly, shapes generated purely from noise. While these shapes hold no relation to the real world, the model trained on the shapes using weak supervision managed to perform almost as well as the model trained on the actual segmentation maps, and both outperformed various iterative and DL registration tools, showing that even geometrically completely unrealistic synthetic scans can still be a successful basis for training.

2.5.2 Unsupervised transformation estimation

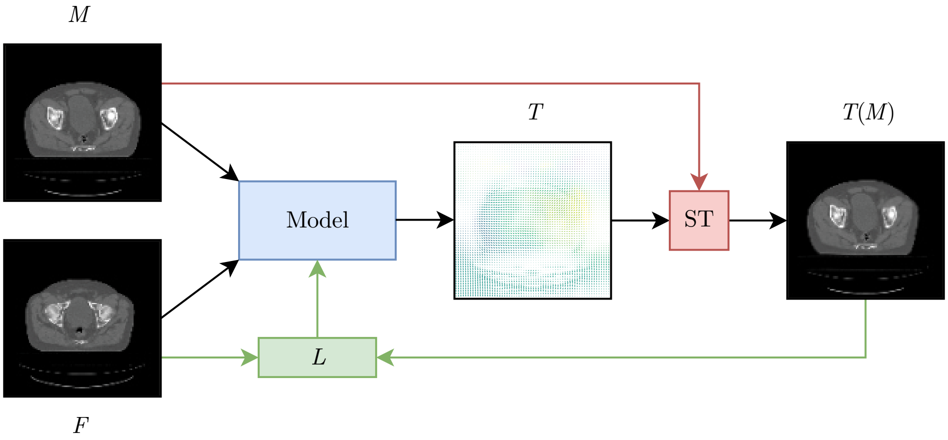


Figure 8: The unsupervised transformation estimation framework.

In unsupervised learning, the training data contains only the input data. In the case of image registration, these are the scans. This solves the issue of a lack of datasets with known transformation parameters or maps, and issues with generating realistic transformations. Instead, the training is based on maximizing the similarity between the transformed moving image and the fixed image, just as is done in iterative registration. This framework is illustrated in Figure 8.

For unsupervised learning of deformable transformations, generally, U-Net-based networks that include STs are used (see section 2.4.5). This setup lends itself well to deformable transformation prediction, which is used in almost all unsupervised methods. For global registration, CNN architectures can again be used (for example, Kori and Krishnamurthi [54] used a 3-layer CNN for affine parameter estimation). As was the case for deformable transformations, for estimation of affine transformations the predicted parameters need to be applied to the image in order to determine the similarity of the deformed moving image and the fixed image.

A loss function is used to guide the learning process. Generally, this loss function consists of a similarity component, which indicates the similarity between the warped moving image and the fixed image, and a regularization component, which can impose constraints such as smoothing constraints. Again, this is equivalent to the similarity metric used in iterative methods (see section 2.1.2). Just as was the case there, many common similarity functions for unsupervised DLIR are intensity-based. Common metrics include: CC or its variants, MSE (or similar metrics like mean absolute error (MAE) or sum of squared distances (SSD)), and MI or its variants. For deformable registrations, the regularization terms that are often added are frequently modeled as a linear operator on spatial gradients of the DVF [55]: For example, the bending energy (BE; the sum of the second derivatives of the DVF), the L_2 norm of the spatial gradient, and the Kullback–Leibler (KL) divergence.

Next to intensity-based measures, feature-based similarity metrics have also been used. These can use the same metrics, but rather than calculating them for all of the content of the image, they are used to calculate the distance between certain features extracted from the images to be registered by some kind of feature extractor component of the network.

Noise and artifacts in images such as US and CBCT often cause similarity metrics to perform poorly [5]. Furthermore, most metrics do not work for multi-modal registration. For multi-modal registration, MI-based metrics are currently considered to be the gold-standard similarity metric [15]. However, it is not sufficiently robust, leading to poor performance [2]. This is an inherent problem for many multi-modal image registration tasks, as often, multi-modal images are acquired because they provide complementary information, and are thus inherently dissimilar [56]. This is an issue that has also been observed for iterative registration methods. There, this led to some early DLIR research where DNNs were trained to learn image similarity [2], as discussed previously. In the case of unsupervised transformation prediction, more recently, GANs have similarly been used for similarity calculation; this is further discussed in Appendix C.

2.5.3 Weakly supervised transformation estimation

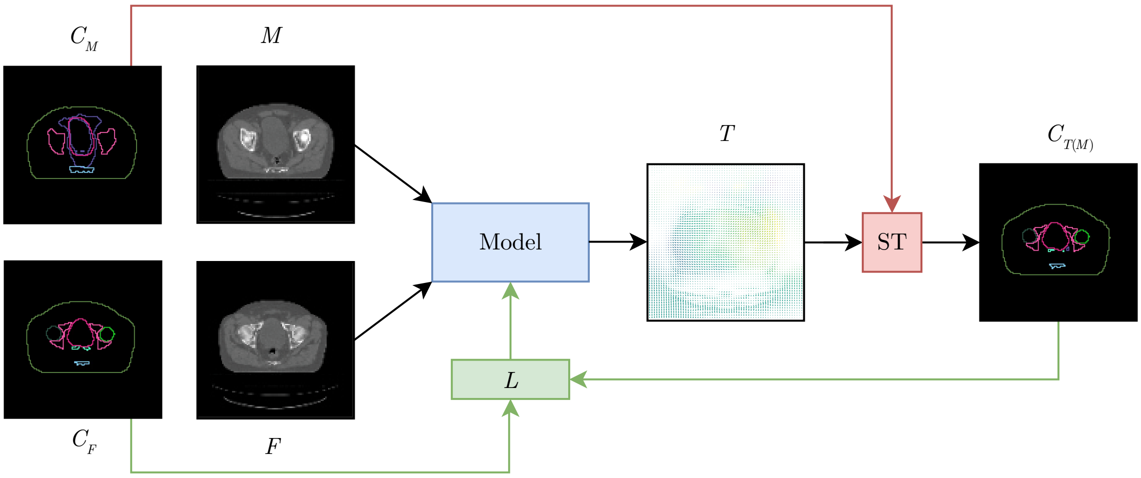


Figure 9: The weakly supervised transformation estimation framework.

Weak supervision refers to ML where limited or imprecise sources of supervision are used. In the case of medical image registration, this mainly refers to using the overlap of segmentations of corresponding anatomical structures as a source of supervision. Other than segmentations, other sources of weak supervision can be used as well, such as corresponding key points, though this is less common; In this thesis, weak supervision will thus be used to refer to supervision using segmentations. This concept is illustrated in Figure 9. While having some data requirements, i.e. that segmented regions of interest (ROIs) need to be available for both the fixed and moving images, this is arguably much easier to obtain than ground-truth transformations between the images.

Of course, only optimizing the overlap of certain labels is not likely to lead to a proper registration of the image as a whole; Therefore, generally, this sort of supervision is used in a *dual supervision* setup.

A related concept is dual supervision. This strictly just means that a combination of loss functions is used; In the medical image registration context, it generally refers to the combination of unsupervised and weakly supervised loss functions. During training, a network would then optimize both the similarity between the transformed moving image and the fixed image (possibly constrained by some sort of regularization), while simultaneously maximizing the overlap between the labels of these. Note that this setup does not require labels to be available for every scan; the unsupervised loss can be used regardless.

2.5.4 Model improvements

In order to improve the results of the aforementioned methods, some authors have used more complex training schemas or models.

Cascading models One common model improvement is the use of multi-stage models, where multiple models are applied after each other. This resembles traditional registration methods, where models are often also applied in sequence, as multi-stage strategies make conventional iterative image registration less sensitive to local optima and image folding [57]. The most common combination is applying a rigid or affine transformation model first, followed by a deformable model. This was notably done by Hu, Modat, Gibson, *et al.* [58], whose "global-net" and "local-net" scheme used to register MR and US scans was considered to be "highly appreciated and valuable" [3, p. 26] by the DLIR research community. In another influential [3] work, De Vos, Berendsen, Viergever, *et al.* [57] proposed a scheme where multiple deformable models were concatenated behind an affine transformation network, in order to get to a more precise final transformation.

Ouyang, Liang, and Xie [17] effectively showed how cascading multiple deformable models can lead to improved performance, and that including an affine transformation network can lead to large improvements, with a Dice Similarity Coefficient (DSC) improvement of over 10%. Zhao, Dong, Chang, *et al.* [59] also showed this feat, repeating models up to 30 times. They found that repeating models with shared weights can lead to a small increase, but that a much larger increase comes from cascades models without shared weights, with more cascades generally leading to better results. They note that repeating a model many times does make the model's runtime much slower: In one of their experiments, a 10-cascade of a model had a 8.7 times longer runtime than the non-cascaded version when run on CPU. Zhao, Lau, Luo, *et al.* [60] published an architecture specifically for cascaded models, the Volume Tweening Network, that features an integrated affine registration network.

Note that for efficient training of these types of networks, the similarity loss is not only calculated and backpropagated after a full forward pass (so on the output of the cascaded model), but also on the outputs of the individual models [17], [60].

Multi-resolution methods One-pass deformable registration models often struggle with predicting larger deformations. Several researchers have attempted to combat this problem by using a multi-resolution, coarse-to-fine setup. By predicting the deformation of a scan resampled at a lower resolution, larger deformations become easier to predict, as these deformations will not cover as big of a voxel distance, and smaller deformations may not be visible anymore.

Multi-resolution approaches have been implemented in multiple ways. The aforementioned work by De Vos, Berendsen, Viergever, *et al.* [57] used a cascading scheme with two deformable models, where the first model predicted deformations on a lower-resolution version of the scan, and the second one predicted the residual deformations on the higher-resolution scan. Mok and Chung [26] use a similar scheme with three resolution levels, but instead of just passing the output of one model to the next model, they use a more complex method: Next to passing the deformed scan, they also upsample the output DVF from the previous level and concatenate it with the input scans on the next level, and add the feature embeddings from the lower level the next level via a skip connection to increase the receptive field. Eppenhof, Lafarge, Veta, *et al.* [18] used a "progressive training" scheme where at the start of the training, only the middle layers of the encoder-decoder architecture with the smallest feature maps are used, which are trained on images and deformation fields that have been downsized to the same resolution. Then, gradually, the layers with larger feature maps get added, so that the sizes of the images get progressively bigger until finally the whole network with the full-resolution images as inputs is complete.

Different transformation models As stated, most recent deformable DLIR methods parameterize the deformation model using a dense displacement field, where a displacement is given for every voxel [26]. Although this method is common and intuitive, it does not guarantee properties like smoothness or diffeomorphism. However, as previously discussed in section 2.1.1, there are many other ways in which a displacement can be parametrized: For example, using B-Splines, as was done by De Vos, Berendsen, Viergever, *et al.* [57] and Qiu, Qin, Schuh, *et al.* [61].

SVFs have been used in the literature as well, for example by Mok and Chung [26], Qiu, Qin, Schuh, *et al.* [61], Dalca, Balakrishnan, Guttag, *et al.* [62], and Krebs, Mansi, Mailhé, *et al.* [63]. The diffeomorphic transformation T is the group exponential of the SVF v, $T = \exp(v)$. T can be found efficiently through the *scaling and squaring* algorithm. This algorithm is formulated to compute matrix exponentials. It is based on the idea that the matrix exponential is much simpler to compute for matrices close to zero [64]. In particular, rather than calculating $\exp(v)$ directly, the scaled version $\exp\left(\frac{v}{2^N}\right)$ is computed (approximated), which then needs to be squared N times, as $\exp\left(\frac{v}{2^N}\right)^{2^N} = \exp(v)$. The number of integration steps N determines the accuracy of the approximation through how closely $\frac{v}{2^N}$ approaches zero, but also raises the computational complexity.

Advanced loss and regularization Some authors have also used more complex loss terms, aiming to produce more realistic deformations. Regularization using GANs, which is discussed in Appendix C, is an example of this. In other papers (such as [59], [65]) the similarity is calculated bidirectionally to regularize their model, ensuring the predicted deformation is invertible by calculating not only the similarity between $T(M)$ and F but also the similarity between $\hat{T}(A)$ and M, where $\hat{T}$ is the inverse of T.

With a similar purpose, Kim, Kim, Lee, *et al.* [66] trained two models simultaneously to register F to M with T_{FM} and M to F with T_{MF}. These models were applied subsequently, creating the cycle constraints that $T_{MF}(T_{FM}(F))$ should have high similarity to F, and that $T_{FM}(T_{MF}(M))$ should have high similarity to M. They also added identity constraints, where using the same image for both F and M should not lead to a deformation (as an image is already perfectly aligned to itself). These additions, again, led to smoother and more accurate deformations.

Lastly, Mansilla, Milone, and Ferrante [22] used weak supervision with a loss term meant to encourage anatomically plausible deformations. This loss term was based on passing a mask and a deformed mask through the encoder part of a pretrained denoising autoencoder, and then calculating the similarity of the outputs.

2.6 CT registration

Now that the general registration task has been discussed extensively, this section will cover DLIR literature specifically on the registration of CT scans, as this is the focus of this work.

2.6.1 Uni-modal CT registration

Most image registration research (about 60% [5]) focuses on uni-modal registration. The most common modalities used are MR and CT, with 53 and 21% of publications respectively [3].

CT-CT registration using DLIR has been attempted in over 20 publications. Most of these publications use images of the lungs, e.g. [67]–[71]. This is largely due to the existence of some open lung CT datasets, such as the commonly used DIR-Lab dataset, which features annotated 4D-CT images (see [72] and [73]). It should be noted that while many papers only test their method on one particular region, in DLIR, methods can often generalize to other problems as well. Next to lungs, some papers have focused on the heart [74], [75], chest [36], [76], or abdomen [77]. However, to the best of my knowledge, Cabrera Gil [78] was the only one to apply DLIR to CT-CT registration of the male pelvic region.

It should be noted that few works focus on rigid or affine CT registration. This is likely because, since it is possible to do a deformable registration for this problem, this is the desired approach, as it is more precise. Global registration approaches are only commonly used for multi-modal registration of 2D images to 3D scans in the case of CT.

The first works on CT-CT registration using DLIR used deep similarity methods [71] and RL for affine registration [79].

Soon after that, supervised and unsupervised methods gained prominence. However, for CT-CT registration, works generally do not focus on global transformations. While various studies focused on supervised or unsupervised rigid registration have been published [2], none of these concern CT-CT registration. Instead, in uni-modal global registration studies, MR is usually the chosen modality.

Instead, for CT-CT registration, deformable transformation estimation methods started gaining prominence early on. Sokooti, De Vos, Berendsen, *et al.* [76], for example, used a supervised approach with synthetic deformations, which their multi-scale U-Net based network, was able to predict on par with a conventional B-spline method. Onieva, Marti-Fuster, Puente, *et al.* [70] used the same network, but trained using a reinforced-sequential training strategy, though this did not seem to lead to huge performance improvements.

Other works using similar U-Net-based networks and supervised training approaches followed. Eppenhof and Pluim [80] presented a supervised method for registration of pulmonary CT images, using synthetic augmentations as ground truth data. Later, Eppenhof, Lafarge, Veta, *et al.* [18] extended this work using a progressive learning architecture, where the smaller feature maps lower in the U-Net were learned first, before moving on to the larger feature maps. This new learning schema led to significant improvements, being able to capture larger displacements much better.

Cabrera Gil [78] used the model presented by Eppenhof and Pluim [80] with some changes (e.g. added residual connections in each convolutional block) to register CT-CT and CT-MR scans of the male pelvis. Again, supervised learning with synthetic augmentations was used.

Sentker, Madesta, and Werner [51] used a more complex network architecture for estimating lung motion based on 4D-CT lung scans, named GDL-FIRE^{4D}. This architecture consisted of a (pretrained) autoencoder with an embedded Inception-Resnet-v2 model. Rather than using synthetic augmentations on singular scans, they acquired ground-truth data using various existing DIR frameworks. As the network that was used is probabilistic, they were able to generate uncertainty maps; However, the authors showed that these widely differ based on the algorithm chosen to generate ground-truth data, raising doubts about their accuracy.

As an unsupervised method, Fu, Lei, Wang, *et al.* [20] used a GAN model (see Appendix C) to register frames of 4D-CT lung scans. They used a two-stage coarse-to-fine setup, trained using NCC loss. They compared their method extensively to 6 DLIR methods, namely the aforementioned methods from [18], [51], [76] as well as methods from [37], [57], [81], and 9 conventional registration methods, showing that they were able to achieve state-of-the-art performance.

De Vos, Berendsen, Viergever, *et al.* [57] also used an unsupervised approach.

An unsupervised method for both the affine pre-registration and the deformable registration was described, though only the deformable registration method was implemented. The method used a three-stage cascading coarse-to-fine network. They tested their architecture on chest CT scans and cardiac cine MR scans.

Deshpande and Bhatt [82] used an unsupervised approach but trained with synthetically augmented scans, rather than inter-subject scans (similar to some of the supervised methods described earlier). Their Bayesian approach managed to outperform various other DLIR methods in lung CT registration and cardiac MR registration.

Various other approaches have also been used. Hansen and Heinrich [83] use an unsupervised approach, but rather than using a U-Net architecture, they extract features at sparse keypoints and map these to a displacement space, from which the registration is constructed. They tested their approach on thoracic CT images. Different types of networks have also been applied successfully, such as binary tree architectures by Blendowski and Heinrich [67] and the probabilistic dense displacement network by Heinrich [77].

Elmahdy, Jagt, Zinkstok, *et al.* [84] worked on CT-CT registration of planning and daily CT scans of the pelvic region. However, they did not use a fully DL-based method: DL was only used for generating contours from the scans, which were used to aid the deformable registration. Their work serves as an example of the interplay of segmentation and registration methods, something that is to be explored further for DLIR methods.

2.6.2 Multi-modal CT registration

Fu, Lei, Wang, *et al.* [5] assert that over 40% of image registration methods in the literature are used in the multi-modal image domain. While this work does not include multi-modal registration, insights from this harder problem can still be applied to the easier (uni-modal) problem, and hence some registration literature that registers CT scans with scans of some other modality will be discussed in this section.

Unlike in the uni-modal case, for multi-modal CT registration there are a few works that focus on rigid or affine registration. For 2D-2D CT-MR registration, Cheng, Zhang, and Zheng [85] used a deep similarity approach, and Sun, Hu, Yao, *et al.* [86] used RL. For CT (3D) and X-ray (2D) registration, Miao, Piat, Fischer, *et al.* [87] and Zheng, Miao, Wang, *et al.* [88] used RL methods as well. Lastly, Sun, Moelker, Niessen, *et al.* [52] focused on CT-US (2D-2D) registration using a supervised approach. As all of these works include some kind of 2D modality, and thus are not particularly relevant for this thesis, they will not be discussed in more detail. The two exceptions to this are De Vos, Berendsen, Viergever, *et al.* [57], whose model included both affine and deformable registration, and Liao, Miao, Tournemire, *et al.* [79], who used RL for CT-CBCT registration. Both will be discussed in more detail later.

There are many papers on deformable multi-modal CT registration, however. In this category, unlike in the previously discussed uni-modal works, unsupervised methods are underrepresented, likely due to the aforementioned difficulty of inter-modal similarity calculations [5].

To combat this, various papers use deep similarity metrics for multi-modal deformable registration, usually in combination with iterative registration methods, e.g. [56], [89], [90].

Other papers use transformation prediction networks. Sun, Moelker, Niessen, *et al.* [52] used a supervised method with ground-truth DVFs. Although the method worked well on synthetic multi-modal images, these did not manage to perform well for true multi-modal images. Hu, Modat, Gibson, *et al.* [91] also used a weakly supervised method, which was tested on MR-US registration, with good results. They used a multi-stage, multi-resolution method where first, a network called Global-Net would predict affine transformation parameters, after which a network called Local-Net was used for the further deformable registration. The combination of the two was labeled as Composite-Net. Multiple loss functions were tested, with multi-scale Dice loss giving the best result as a label-similarity loss function (over multi-scale cross-entropy), and bending energy as the regularization term, where it surpassed L_2 norm loss.

CT-CBCT Some past work has also specifically dealt with the multi-modal registration task of CT and CBCT scans. Since CT and CBCT are similar, the same (e.g. unsupervised) methods used for CT-CT registration have also been used here. For example, Van Kranen, Kanehira, Rozendaal, *et al.* [92] used the VoxelMorph architecture (see section 3.1.2). The accuracy was found to be acceptable, albeit less than clinically applied B-spline DIR and failed at occasional large deformations. Jiang, Yin, Ge, *et al.* [37] showed that even networks trained for CT-CT registration can be successfully applied to CT-CBCT registration, managing to outperform commercial registration software Velocity by Varian Medical Systems[7] in a lung registration task, with an average TRE of 1.66 mm versus Velocity's average of 2.73 mm on the popular DIR-Lab dataset. The model used was a multi-scale unsupervised network, trained with NCC similarity loss and L_2-norm regularization. Being multi-scale, it was trained at three resolutions, first separately and then jointly.

Of course, some papers also take other approaches; Liao, Miao, Tournemire, *et al.* [79], for example, used RL for rigid registration of CT and CBCT.

MR-CT Another combination of modalities for which registration is vital is CT and MR scans, as MR scans are used for tumor and OAR segmenting and CT scans for dose mapping. However, as Tanner, Ozdemir, Profanter, *et al.* [56] argue, "[d]eformable Image Registration (DIR) of MR and CT images is one of the most challenging registration task[s], due to the inherent structural differences of the modalities and the missing dense ground truth" (p. 1).

One approach taken in the literature is using learning from paired image data. For example, Cao, Yang, Wang, *et al.* [39] took an unsupervised training approach for CT-MR registration, with a standard U-Net-based network with an ST layer. They trained this network using an intra-modal similarity: Rather than calculating the similarity between the warped MR image and the CT image, it was calculated between the warped MR image and the original MR image aligned to the CT input, and the other way around. The dual-modality similarity, i.e. this last part where the similarity is calculated in both directions, leads to a more robust network.

Intra-modality loss functions can also be used without paired data if image translation is used. For example, image translation using cycle-GANs was used by Cabrera Gil [78] to create paired images, which were then synthetically augmented to train a DL network with supervised learning.

Tanner, Ozdemir, Profanter, *et al.* [56] also used a GAN-assisted approach to CT-MR registration, but with an unsupervised training method. Images were synthesized using cycle-GANs so that intra-modal images could be used for loss calculation using local NCC. However, they did not use a DL method for the transformation prediction. These were compared to models based on multi-modal NMI or MIND loss, or a combination of the two. Experiments were done on scans of the abdomen and thorax. For the abdomen, some of the results of the model using the synthetic images were almost as good as the NMI- and NMI+MIND-based models, though for the thorax they were worse. While this paper did not use a learning-based architecture, like is done in this thesis, it shows that a "simple" similarity function like NMI can still outperform more complex methods for multi-modal registration.

Similarly, Blendowski, Bouteldja, and Heinrich [93] also used a DL method to assist similarity calculation, in this case using segmentations obtained from a segmentation network; However, again, the registration is done using an iterative method, rather than DLIR.

Like Blendowski, Bouteldja, and Heinrich [93], Hering, Kuckertz, Heldmann, *et al.* [94] also used segmentations, but in this case, as weak supervision with labels for CT-MR registration. They proposed a 2.5D convolutional transformer architecture. 2.5D networks use three (one axial, one coronal, and one sagittal) 2D slices of a scan as model input to approximate the whole scan; The final 3D deformation is the average of the nonzero components of the three deformation fields. The network is U-Net based, using the normalized gradient field distance measure for the similarity loss.

2.7 Evaluation datasets

As previously stated, one of the current issues in the field of DLIR is the lack of common public datasets and benchmarking datasets [3]. According to [5], most DLIR publications use private datasets. Nonetheless, there have been various challenges aimed at comparing registration datasets, as well as some datasets that are quite popularly used and thus also allow for model comparison. In this section, these will briefly be discussed.

2.7.1 Registration challenges

The Learn2Reg challenge[12] is a registration challenge that is a part of MICCAI (a large international conference on medical image computing and computer-assisted

[12] https://learn2reg.grand-challenge.org/

intervention). The challenge was organized in 2020, with 11 participating groups, and again in 2021. In 2020, the challenge consisted of 4 tasks, concerning the registration of: (1) multi-modal T2 MR-ultrasound scans of the brain, (2) CT thorax scans, (3) CT abdomen scans, and (4) T1 MR scans. More information on the challenge winner, LapIRN, is given in section 3.1.2. The 2021 edition, which is still ongoing at the time of writing, consists of 3 tasks concerning the registration of: (1) CT-MR thorax-abdomen scans, (2) CT lung scans, and (3) MR brain scans.

In older editions of MICCAI, challenges that involved registration were also organized. For example, Multi-Atlas Labeling Beyond the Cranial Vault[13] was a challenge that was part of MICCAI 2015 on the atlas registration abdomen and cervix CT scans. The data from this challenge has been used for training by Heinrich and Hansen [95].

The Continuous Registration Challenge[14] is another registration challenge, focused on the registration of lung and brain scans. The challenge was part of ISBI 2019 (the 2019 IEEE 16th International Symposium on Biomedical Imaging), and will remain open for submissions indefinitely. The challenge uses 3 CT lung datasets (POPI, DIR-Lab, and EMPIRE10) and 4 MR brain datasets (LPBA40, ISBR18, CUMC12, and MGH10) as training data, and two datasets (SPREAD for the CT lung task and BRAINS for the MR brain task) for evaluation.

In the same conference, the ANHIR challenge[15] on the registration of 2D microscopy images was also organized. This task is quite far removed from the focus of this thesis, however.

2.7.2 Datasets

Next to the aforementioned benchmarking challenges, there are also some datasets that are commonly used for evaluating DLIR publications, facilitating the comparison between publications.

According to Fu, Lei, Wang, *et al.* [5], the most commonly used CT dataset is DIR-Lab[16], which contains 2 sets of 10 4D CT scans, on which registration between frames can be performed. This dataset has been used in DLIR research by [20], [37], [51], [57], [76], [81] amongst others.

Similarly, the POPI-model[17] and EMPIRE10[18] datasets, both of which contain 4D CT scans, have been used in various publications, albeit not as frequently. Both were also included in the aforementioned Continuous Registration Challenge.

For multi-modal CT-MR and PET-MR registration, RIRE[19] is an evaluation dataset.

[13] DOI: 10.7303/syn3193805 [14] https://continuousregistration.grand-challenge.org/
[15] https://anhir.grand-challenge.org/ [16] https://www.dir-lab.com/
[17] https://www.creatis.insa-lyon.fr/rio/popi-model/
[18] https://empire10.grand-challenge.org/ [19] https://www.insight-journal.org/rire/

3 Methods

In this project, a multi-stage DLIR approach consisting of an affine registration followed by a deformable registration is taken. An overview of this pipeline is shown in Figure 10. As is shown in the figure, first an affine transformation matrix T_A is estimated by the affine model, which is then applied to the scan; Next, the deformable model estimates a DVF to correct local misalignment between the affinely registered moving scan and the fixed scan. Note that while in this approach, the affine and deformable registrations are applied separately, this is not a requirement for multi-stage approaches. The affine transformation matrix could instead be converted to a grid (which is essentially what is already done in the first ST), which can be added to the DVF that is outputted by the deformable model, so that the transformation can be applied to the moving image at once.

In this thesis, the two stages of the model are trained separately. For the deformable registration model, two types of models are tested: A single registration network, and a multi-resolution cascaded network. The different architectures used are described in this chapter. Next to that, the various methods used for training these models are also detailed.

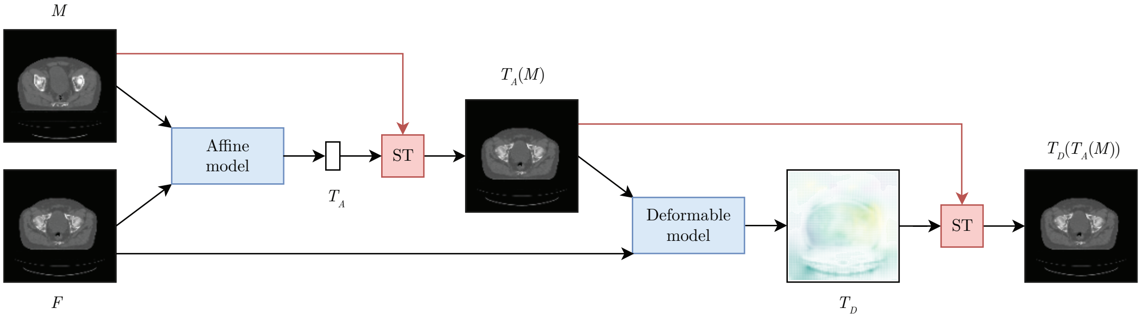


Figure 10: The registration pipeline, consisting of the affine model and the deformable model.

3.1 Models

3.1.1 Affine image registration

For the affine registration, one type of architecture is used: a 3D ResNet model. While, as previously discussed, affine CT-CT registration has not received much attention in DLIR publications, various publications on 3D-3D affine registration tasks with other modalities have used types of CNNs and ResNets specifically, e.g. Salehi, Khan, Erdogmus, *et al.* [96], to predict the transformation parameters.

3D ResNet As the model for the affine registration, the 3D ResNet implementation of Kataoka, Wakamiya, Hara, *et al.* [97] is used. This implementation consists of updated models of those earlier published by Hara, Kataoka, and Satoh [98] and include 3D versions of the standard ResNet architectures, as well as 3D versions of ResNet architectures with different residual blocks, such as the ResNeXt, DenseNet, and R(2+1)D ResNet architectures. These architectures were initially created to handle video data for classification tasks. A number of frames from a video would be concatenated into a 3D array, and the output consists of a 1D array where the length is the number of classes. The architectures, as well as various pretrained models, were published on Github[20] under the MIT License.

In this case, the input of the model does not consist of a 3D array of 2D video frames, but rather of a 3D scan; and rather than classification, a regression is done, where the output consists of a 1D array of 12 values, representing the first 12 values of the 4×4 transformation matrix (as the last row is always scaled to $[0, 0, 0, 1]$). In some publications, e.g. [99], [100] values for the translations and rotations across all

Layer/block	Output size	Architecture (filters, kernel size, stride)		
		ResNet-10	ResNet-18	ResNet-34
input	$128 \times 128 \times 48$	-		
conv1	$64 \times 64 \times 24$	$64, 7^3, 2$		
max. pool	$32 \times 32 \times 12$	$3^3, 2$		
conv2_x	$32 \times 32 \times 12$	$\begin{bmatrix} 64, 3^3, 1 \\ 64, 3^3, 1 \end{bmatrix}$	$\begin{bmatrix} 64, 3^3, 1 \\ 64, 3^3, 1 \end{bmatrix} \times 2$	$\begin{bmatrix} 64, 1^3, 1 \\ 64, 3^3, 1 \\ 256, 1^3, 1 \end{bmatrix} \times 3$
conv3_x	$16 \times 16 \times 6$	$\begin{bmatrix} 128, 3^3, 1 \\ 128, 3^3, 1 \end{bmatrix}$	$\begin{bmatrix} 128, 3^3, 1 \\ 128, 3^3, 1 \end{bmatrix} \times 2$	$\begin{bmatrix} 128, 1^3, 1 \\ 128, 3^3, 1 \\ 512, 1^3, 1 \end{bmatrix} \times 4$
conv4_x	$8 \times 8 \times 3$	$\begin{bmatrix} 256, 3^3, 1 \\ 256, 3^3, 1 \end{bmatrix}$	$\begin{bmatrix} 256, 3^3, 1 \\ 256, 3^3, 1 \end{bmatrix} \times 2$	$\begin{bmatrix} 256, 1^3, 1 \\ 256, 3^3, 1 \\ 1024, 1^3, 1 \end{bmatrix} \times 6$
conv5_x	$4 \times 4 \times 1$	$\begin{bmatrix} 512, 3^3, 1 \\ 512, 3^3, 1 \end{bmatrix}$	$\begin{bmatrix} 512, 3^3, 1 \\ 512, 3^3, 1 \end{bmatrix} \times 2$	$\begin{bmatrix} 512, 1^3, 1 \\ 512, 3^3, 1 \\ 2048, 1^3, 1 \end{bmatrix} \times 3$
avg. pool	$1 \times 1 \times 1$	-		
fc	12	512		

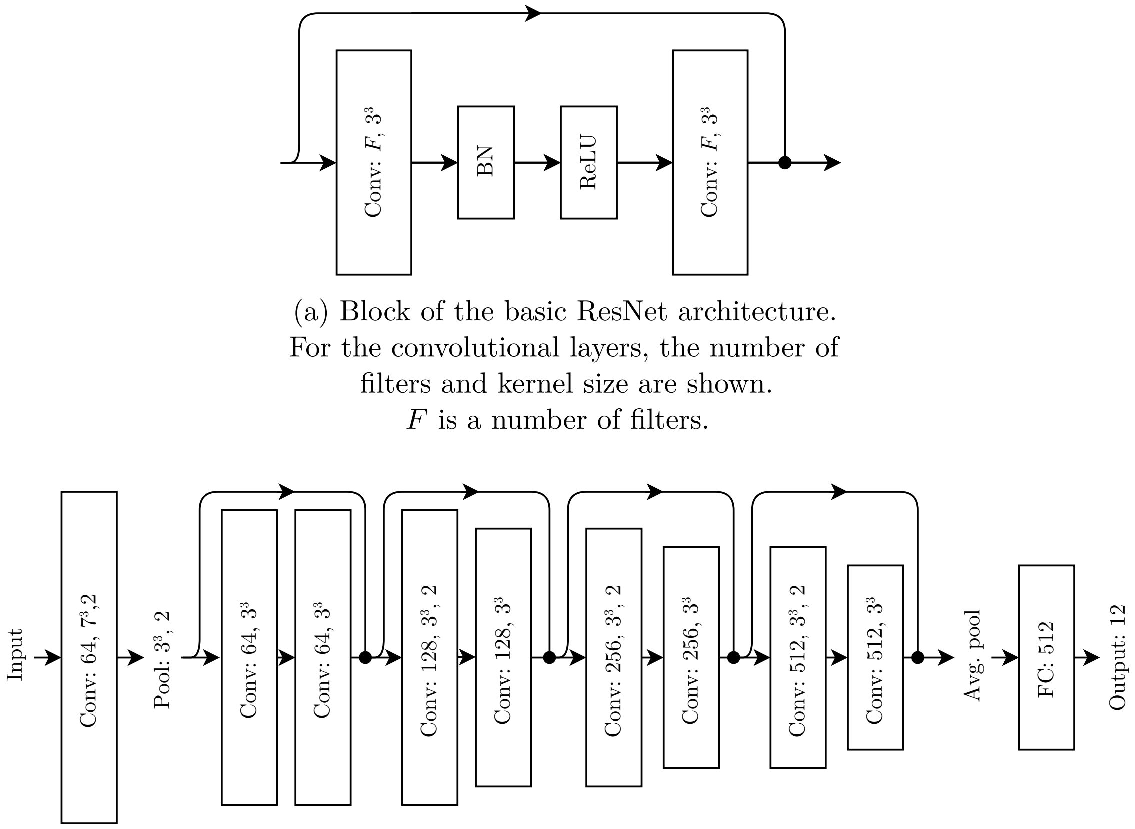
Table 2: Architectures of the 10-, 18- and 34-layer ResNets. For each convolutional layer and residual block, the number of filters, their size, and the strides of the convolution are shown respectively. BN and ReLU layers are not shown, though BN follows after every convolutional layer, and ReLU after every second convolutional layer (see Figure 11a). Downsampling is performed by conv3_1, conv4_1, and conv5_1 with a stride of 2.

[20] https://github.com/kenshohara/3D-ResNets-PyTorch/

three axes were predicted separately rather than in a transformation matrix. In other papers, e.g. [54], [96], [101], a matrix approach is used, however. This approach was deemed more widely applicable, as it can be used regardless of the type of global registration (e.g. rigid, affine, projective), and was therefore also used in this thesis.

The 10-layer, 18-layer and 34-layer 3D ResNets with basic ResNet blocks were used (see Figure 11a). Although Hara, Kataoka, and Satoh [98] showed that their more complex models can achieve higher accuracies, it is common practice in DL to start with smaller and simpler models, and only use more complex ones if the desired performance is not achieved, as bigger models can take significantly longer to train and often require more training data.

The full model architectures for the three ResNet models are shown in Table 2. For easier interpretation, a visual representation of the ResNet-10 architecture is shown in Figure 11b.



(a) Block of the basic ResNet architecture. For the convolutional layers, the number of filters and kernel size are shown. F is a number of filters.

(b) 10-layer ResNet architecture. For the convolutional layers, the number of filters, kernel size, and the stride (if not 1) are shown. BN and ReLU layers are not shown. In some blocks, the first layer is followed by pooling. The FC block at the end can have different sizes, see Table 2.

Figure 11: 3D ResNet-10 architecture and basic block.

3.1.2 Deformable image registration

Two deformable models are used in this work. Firstly, VoxelMorph, a single resolution U-Net-based architecture that is arguably the most popular DLIR framework; Secondly, LapIRN, a more complex multi-resolution architecture that was the winner of two of the Learn2Reg 2020 MICCAI Registration Challenge tasks, and can be considered one of the top-performing current DLIR methods.

Single resolution U-Net using VoxelMorph VoxelMorph is an architecture originally proposed by Balakrishnan, Zhao, Sabuncu, *et al.* [55]. Available as permissive free software (licensed under Apache License 2.0) through Github[21], it is a popular method and has been named as the state-of-the-art DLIR method in various publications, e.g. [63], [102], [103], often serving as a baseline for comparison of new methods. Furthermore, it has been tested quite extensively against traditional methods such as ANTs, Elastix, and NiftyReg [104]. In these past tests, VoxelMorph mainly performed (slightly) worse than these three (ANTs: [30], [40], [59], [60], [105]; Elastix: [30], [59], [66]; and NiftyReg: [62], [95], [105]), though in some studies, it has also performed slightly better (ANTs: [33], [62]; NiftyReg: [30], [33]).

The VoxelMorph network takes both fixed and moving images to create a 2-channel input image, which is then passed through a U-Net-like architecture that outputs a DVF. This DVF is then applied to the moving image in the ST layer, returning the final registered moving image. Note that VoxelMorph was only designed to handle relatively small deformations, and was meant to be used for images that have previously been aligned globally. Originally, it was used with an unsupervised learning method for 3D images, though it was extended to weakly supervised versions soon after publication: By Balakrishnan, Zhao, Sabuncu, *et al.* [105] for masks, by adding these as extra channels, and further by Dalca, Yu, Golland, *et al.* [103]) for surfaces from point clouds. Furthermore, it was extended to deliver an approximately diffeomorphic registration with uncertainty estimates in the probabilistic version presented by Dalca, Balakrishnan, Guttag, *et al.* [65] and Dalca, Balakrishnan, Guttag, *et al.* [62]. In this version, the model output is passed through a vector integration layer that uses scaling and squaring with $N = 7$ integration steps to calculate the final approximately diffeomorphic DVF.

Figure 12 illustrates the VoxelMorph architecture. The architecture can be implemented with various model depths. In this research, a model was implemented with an encoder consisting of 4 blocks containing a convolutional layer, a max-pooling layer, and Leaky ReLU activation layer, and a decoder consisting of 3 blocks containing a convolutional layer, an upsampling layer, and a skip connection to the corresponding encoder block. There is one less layer in the decoder to predict an SVF at every two voxels, following Dalca, Balakrishnan, Guttag, *et al.* [65]. The decoder is followed by two more convolutional layers. Furthermore, the vector integration using scaling and squaring layers presented by Dalca, Balakrishnan, Guttag, *et al.* [65] were used to deliver a DVF with less folding. Table 3 summarizes the architecture.

[21] http://voxelmorph.mit.edu/

Layer/block		Output size	Architecture
input		$128 \times 128 \times 48$	-
Encoder	conv1	$64 \times 64 \times 24$	$16, 3^3, 1$
	conv2	$32 \times 32 \times 12$	$32, 3^3, 1$
	conv3	$16 \times 16 \times 6$	$32, 3^3, 1$
	conv4	$8 \times 8 \times 3$	$32, 3^3, 1$
Decoder	conv3-	$16 \times 16 \times 6$	$32, 3^3, 1$
	conv2-	$32 \times 32 \times 12$	$32, 3^3, 1$
	conv1-	$64 \times 64 \times 24$	$32, 3^3, 1$
conv		$64 \times 64 \times 24$	$16, 3^3, 1$
conv		$64 \times 64 \times 24$	$3, 3^3, 1$

Table 3: Architecture of the VoxelMorph model that was used. Every convolutional layer is followed by Leaky ReLU activation, and either downsampling (in the encoder) or upsampling (in the decoder). For each convolutional layer and residual block, the number of filters, their size, and the strides of the convolution are shown respectively. Skip connections go from the input layer to conv0, from conv1 to conv1-, etc. The last convolutional layer is followed by diffeomorphic integration through scaling and squaring.

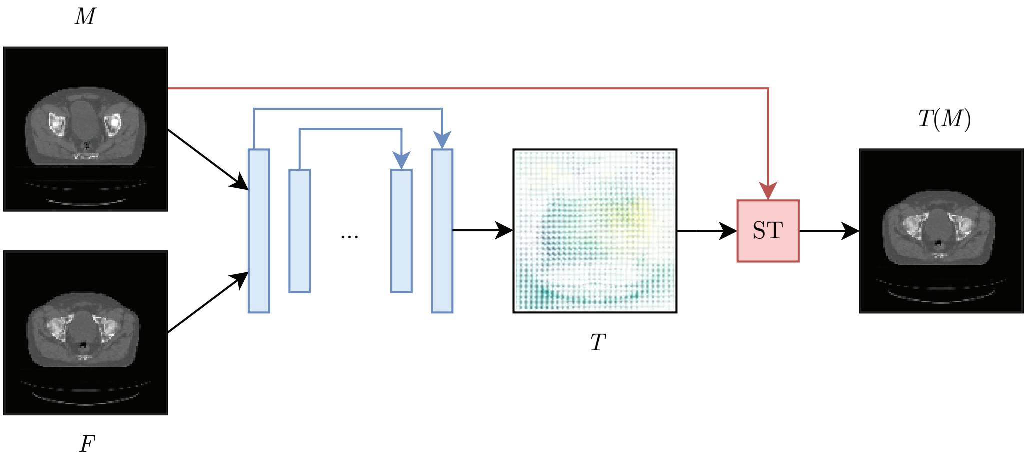


Figure 12: VoxelMorph model architecture. The encoder-decoder and ST modules are shown in blue and red respectively. Note that the ST module is not an obligatory part of the architecture, and could be left out in case of supervised learning. The model output is T, processed through scaling and squaring.

Multi-resolution U-Net using LapIRN LapIRN is a network proposed by Mok and Chung [26]. It deals with DLIR tasks in a coarse-to-fine manner and aims for diffeomorphic registration. Using its multi-resolution architecture, it is meant to be able to handle larger deformations than single-resolution architectures like VoxelMorph. As the lower-resolution levels take care of larger deformations, the higher levels can focus on smaller remaining deformations. The deformations are all added together for the final DVF, similar to a Laplacian pyramid (hence the name of the network).

LapIRN was the winner of two of the four tasks of the Learn2Reg 2020 MICCAI Registration Challenge, namely task 3 on registration of abdominal CT scans (see [106]) and task 4 on the registration of MR scans of the hippocampus (see [107]). The method also outperformed VoxelMorph (in both the original and the diffeomorphic formulation) and various traditional registration algorithms on two brain MR registration tasks [26].

The LapIRN architecture consists of a three-level (in the original formulation, though the authors note this can be extended to any number) network, utilizing three identical CNN-based registration networks to mimic multi-resolution registration. The input pyramid consists of the original scans and their downsampled versions at half their size and a quarter of their size. The CNNs output deformation fields, that are added together (which, in the case of the smaller levels, requires upsampling to the original resolution).

The CNN-based registration networks have an identical architecture, similar

Layer/block		Output size			Architecture
		lvl. 1	lvl. 2	lvl. 3	
input		$32 \times 32 \times 12$	$64 \times 64 \times 24$	$128 \times 128 \times 48$	-
Encoder	conv1	$32 \times 32 \times 12$	$64 \times 64 \times 24$	$128 \times 128 \times 48$	$\left[32, 3^3, 1\right] \times 2$
	conv2	$16 \times 16 \times 6$	$32 \times 32 \times 12$	$64 \times 64 \times 24$	$32, 3^3, 2$
Residual block		$16 \times 16 \times 6$	$32 \times 32 \times 12$	$64 \times 64 \times 24$	$\begin{bmatrix} 32, 1^3, 1 \\ 28, 3^3, 1 \\ 28, 3^3, 1 \end{bmatrix} \times 5$
Decoder	conv3$_T$	$32 \times 32 \times 12$	$64 \times 64 \times 24$	$128 \times 128 \times 48$	$32, 2^2, 2$
	conv4	$32 \times 32 \times 12$	$64 \times 64 \times 24$	$128 \times 128 \times 48$	$16, 3^3, 1$
	conv5	$32 \times 32 \times 12$	$64 \times 64 \times 24$	$128 \times 128 \times 48$	$3, 3^3, 1$

Table 4: Architecture of the LapIRN model that was used. For each convolutional layer and residual block, the number of filters, their size, and the strides of the convolution are shown respectively. conv3$_T$ is a transposed convolutional layer. In the residual blocks, the convolutional layers are preceded by Leaky ReLU layers, and there is a skip connection from the first convolutional layer to the last. conv1_x, the residual blocks, and conv4 are followed by Leaky ReLU layers. conv5 is followed by a SoftSign layer and diffeomorphic integration through scaling and squaring; The upsampled output from the previous level is then added in the case of levels 2 and 3. The output of conv3$_T$ at the first and second levels is connected to the output of conv2 on levels 2 and 3 respectively through skip connections.

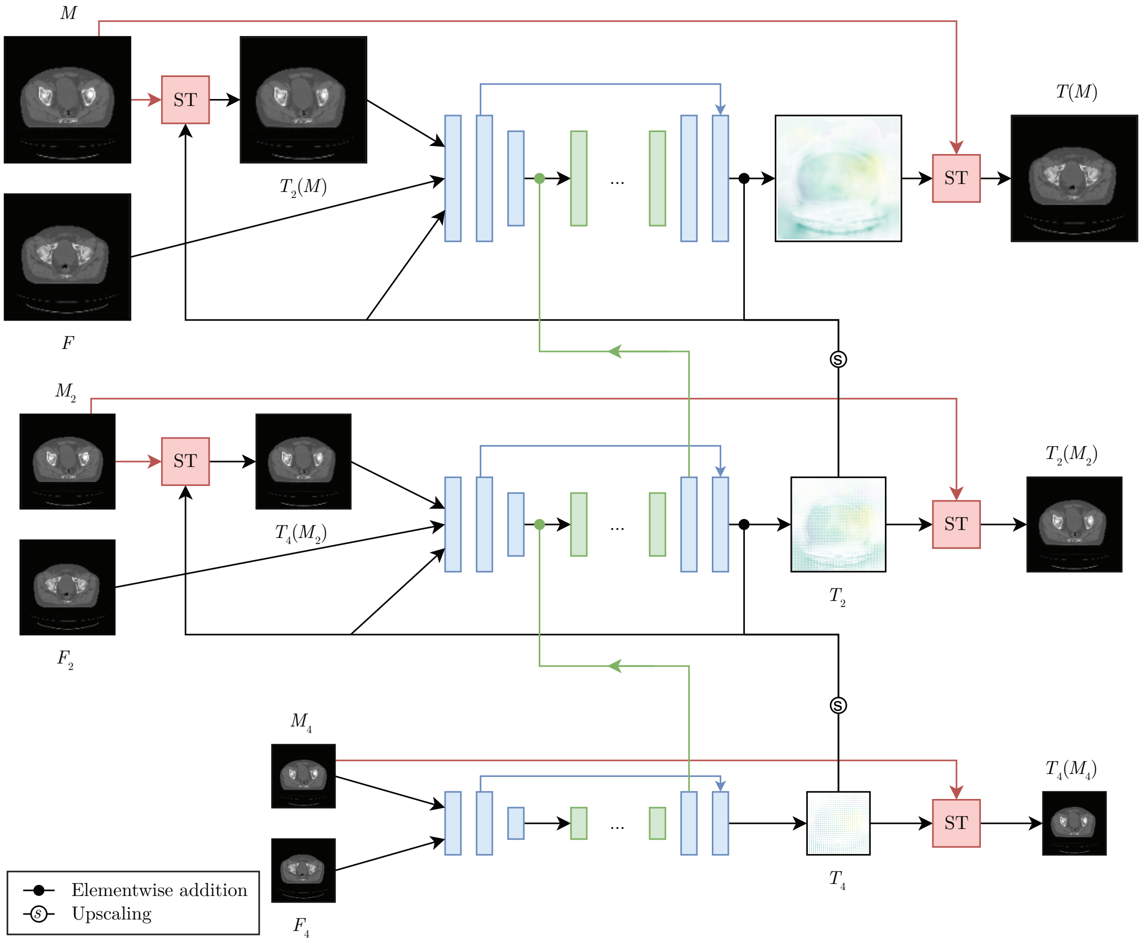


Figure 13: LapIRN architecture. The encoder and decoder, residual blocks, and ST modules are shown in blue, green, and red respectively. M_2 and M_4 indicate resampled instances of M at one half and one quarter of the size respectively; The same holds for F_2 and F_4. The model is trained from the first level (at the bottom) upwards. For the ST modules, the DVFs are first processed through scaling and squaring.

in function to the encoder-decoder architecture used in VoxelMorph. In this case, it contains an encoder, a number of residual blocks, and a decoder; The middle layers do not contain down- or up-sampling. The feature encoder is comprised of two convolutional layers with stride 1 and one convolutional layer with stride 2. After this, five residual blocks follow, consisting of two convolutional layers with a pre-activation structure (meaning that the activation is applied before the convolution, rather than after) and a skip connection. Finally, the decoder consists of one transpose convolutional layer and two consecutive convolutional layers with stride 1, followed by SoftSign activation. The output is then put through scaling and squaring to create a smoother DVF, in the same fashion used in the diffeomorphic VoxelMorph implementation.

For the first level, the input of the model consists of the downsampled scans;

For the next levels, the moving scan is first warped with the output of the previous level. Furthermore, the DVF from the previous level itself is also added as input (so instead of 2 channels, there are 5). This DVF is then also added to the output at the end. Similarly, the output of the upsampling layer is added to the input of the residual blocks in the next layer through a skip connection.

This model architecture is illustrated in Figure 13, with details given in Table 4.

Note that compared to VoxelMorph, the scans do not get downsampled as much with pooling layers in the three levels. Instead, the downsampling is done between the levels; the smallest feature maps in the network are the same size as the smallest feature maps in VoxelMorph, namely 1/16th of the size of the original scan in all directions.

The network is trained in a coarse-to-fine manner, meaning that first, training is done on the coarsest level alone, before the next level gets trained. The weights of the previous level are first frozen; However, eventually, they are unfrozen so they can still be updated during the training of the subsequent levels.

3.2 Training methods and loss functions

The aforementioned models can be trained in different ways, as was previously described in section 2.5. In this section, the training methods and loss functions used in this thesis are detailed.

In DLIR, training methods generally aim to minimize a certain loss function (where $\overline{T}$ is the optimal transformation):

$$\overline{T} = \arg\min_{T} L_{total}(F, M, T) \tag{1}$$

for any F and M from the training data. The loss function L_{total} consists of multiple elements: One or multiple similarity metric(s), and a regularizing metric, as shown in Equation 2 (with optional terms in square brackets).

$$L_{total}(F, M, T) = -L_{sim}(F, T(M))\,[-\lambda_1 L_{sim_2}(F, T(M))]\,[+\lambda_2 L_{reg}] \tag{2}$$

The λ weights regularize the influence of the additional parameters.

3.2.1 Supervised learning

To train a supervised model, synthetic augmentations are used, so that transformation parameters T (the first three rows of the transformation matrix in case of an affine model, and the DVF in case of a deformable model) are known. The similarity function used for supervised training of the affine and deformable models in this paper is MSE, as defined in Equation 3, where $\hat{T}$ are the true and T are the predicted transformation parameters.

$$\text{MSE}\Big(\hat{T}, T\Big) = \frac{1}{|P|} \sum_{p \in P} \Big(\hat{T}(p) - T(p)\Big)^2 \tag{3}$$

3.2.2 Unsupervised learning

Rather than through supervised learning, most models in this paper are trained in an unsupervised manner. The similarity loss function L_{sim} used for this is either NCC loss or MI loss.

NCC loss CC is a metric that assumes that corresponding voxel intensities in the images indicate a linear relationship [108]. Its normalized version, NCC, is less sensitive to linear changes in the intensities in the images [109]. In the general formulation, the NCC is calculated for each voxel $p \in P$. Rather than comparing $F(p)$ and $T(M)(p)$ directly, the metric is based on a volume of width n around p, consisting of a number of voxels p_i. The NCC is defined in Equation 4, where $\overline{F}(p) = \frac{1}{n^3}\sum_{p_i} F(p_i)$ and $\overline{T(M)}(p)$ is defined similarly.

$$\mathrm{L_{NCC}}\Big(F, T(M)\Big) = \sum_{p \in P} \frac{\Big(\sum_{p_i}\Big(F(p_i) - \overline{F}(p)\Big)\Big(T(M)(p_i) - \overline{T(M)}(p)\Big)\Big)^2}{\Big(\sum_{p_i}\Big(F(p_i) - \overline{F}(p)\Big)^2\Big)\Big(\sum_{p_i}\Big(T(M)(p_i) - \overline{T(M)}(p)\Big)^2\Big)} \tag{4}$$

MI loss For the MI loss, the differentiable MI implementation by Guo [33] is used. In its general formulation, MI measures the mutual dependence between the two variables by qualifying the distance between their joint distribution and the product of their marginal distributions. For images, the MI is based on histograms of the voxel intensities, which are divided into a number of equally spaced histogram bins.

Guo [33] introduces two different formulations of the MI: the *global* and *local* versions, where the latter is a patch-based version of the former. First, the general (global) version will be detailed, after which this local formulation will be explained.

The (global) MI metric is defined in Equation 5, where $\Pr(\mathfrak{f})$ is the probability of the voxels in image F to have an intensity in bin $\mathfrak{f} \in \mathfrak{F}$, and likewise $\Pr(\mathfrak{m})$ for bin $\mathfrak{m} \in \mathfrak{M}$ image $T(M)$.

$$\mathrm{MI}\Big(F, T(M)\Big) = \sum_{(\mathfrak{f},\mathfrak{m}) \in (\mathfrak{F},\mathfrak{M})} \Pr(\mathfrak{f},\mathfrak{m}) \log \frac{\Pr(\mathfrak{f},\mathfrak{m})}{\Pr(\mathfrak{f})\Pr(\mathfrak{m})} \tag{5}$$

For MI to be used as a loss function, it needs to be formulated in a differentiable way. Therefore, in the formulation by Guo [33], instead of a voxel only contributing to the bin it belongs to, each voxel contributes to a range of histogram bins. This is achieved using Parzen windowing, which calculates $\Pr(\mathfrak{f})$ based on the values of the n voxels $f \in F$, where each sample contributes to $\Pr(\mathfrak{f})$ with a function of its distance to the bin average value (also denoted as $\mathfrak{f}$). The probability is then calculated as shown in Equation 6:

$$\mathrm{Pr}_f(\mathfrak{f}) = \frac{1}{n}\sum_{f \in F} W(\mathfrak{f} - f) \tag{6}$$

with Gaussian weighting function W that is weighted by parameter λ:

$$W(\mathfrak{f} - f) = \frac{1}{\lambda\sqrt{2\pi}} \exp\Big(-\frac{(\mathfrak{f} - f)^2}{2\lambda^2}\Big)$$

$\mathrm{Pr}_m(\mathfrak{m})$ is calculated in the same manner. To compute the joint probability, similarly, the voxels from corresponding locations in the images are taken, as shown in Equation 7.

$$\mathrm{Pr}_{f,m}(\mathfrak{f},\mathfrak{m}) = \frac{1}{n} \sum_{(f,m)\in(F,T(M))} W(\mathfrak{f}-f)W(\mathfrak{m}-m) \tag{7}$$

Based on Equations 6 and 7, the MI loss is defined in Equation 8, where a smoothing parameter σ is added.

$$\mathrm{L}_{\mathrm{MI}}\Big(F,T(M)\Big) = \sum_{(\mathfrak{f},\mathfrak{m})\in(\mathfrak{F},\mathfrak{M})} \mathrm{Pr}_{f,m}(\mathfrak{f},\mathfrak{m}) \log \frac{\mathrm{Pr}_{f,m}(\mathfrak{f},\mathfrak{m})}{\mathrm{Pr}_f(\mathfrak{f})\mathrm{Pr}_m(\mathfrak{m})+\sigma} \tag{8}$$

Calculating the MI using intensity distributions over the whole image is not desired, however. Given two voxels that have the same intensity but are spatially far away from each other, global MI will treat them the same manner. Instead, to only consider voxels with the same intensity if they are spatially close together, the local MI is used. This means that instead of doing global calculations, the MI is calculated over patches. The calculation is repeated for every patch $P \in \mathcal{P}$ to get to the loss function in Equation 9. Guo [33] achieved better performance with this local MI implementation but notes that there is a risk for overfitting. Both implementations are therefore tested.

$$\mathrm{L}_{\mathrm{MI}_{local}}\Big(F,T(M)\Big) = \frac{1}{|\mathcal{P}|} \sum_{P\in\mathcal{P}} \mathrm{L}_{\mathrm{MI}}\Big(F(P),T(M)(P)\Big) \tag{9}$$

The lowest the mutual information can be is 0, indicating equal probability distributions. There is no maximum value.

Multi-resolution loss As LapIRN uses a pyramid structure where it starts predicting deformations at a lower resolution first, it is trained with a pyramid similarity metric to capture both large and small misalignments adequately and avoid local optima [26]. Therefore, a multi-resolution version of the loss function is used, as defined in Equation 10, where K is the total number of layers in the pyramid, with A_k being the downsampled F at that level of the pyramid. In this case, $K = 3$.

$$\mathrm{L}_K\Big(F,T(M)\Big) = \sum_{k=1}^{K} \frac{1}{2^{(K-k)}} L\Big(F_k,T(M)_k\Big) \tag{10}$$

Note on padding Often in literature, instead of looking at all voxels in an image, a masked part of the image containing only the relevant section is taken. For example, in a lung alignment task, masks of the lungs have been used. In this project, body masks could have been used to exclude the areas outside of the body. However, it is still undesirable to use full-sized scans to calculate the similarity loss. One scan may be larger than another, which is solved by padding both scans. Therefore, one scan can contain parts of the body not present in the other scan, where instead only zeros from the padding are present. Thus, using the intensities from all voxels to calculate the similarity may lead to different similarity scores than is warranted: For example, if the patient's legs are shown in the fixed scan but not in the moving scan, the NCC of a scan stretched to cover the legs may be higher than one that is perfectly aligned to the part of the body actually present in both scans. To combat this, the voxels over which the loss is calculated are only those from non-padded parts present in both scans.

3.2.3 Weakly supervised learning

The unsupervised learning schema can be extended to a weakly supervised training schema by adding a loss function based on masks segmented in F and M. For this, the Dice loss is used. This loss is a measure of the overlap between the predicted and true masks, and is virtually identical to the Dice similarity coefficient (DSC; see section 4.3). The Dice loss is defined in Equation 11, where $\varphi_c(p)$ is the value of the binary volume of segment c in F for voxel p, $\mu_c(p)$ is the binary volume of the same voxel and the same segment in $T(M)$, and σ is a smoothing parameter.

$$\mathrm{L}_{\mathrm{DSC}}\Big(F, T(M)\Big) = \sum_{c\in C} \frac{2 \sum\limits_{p\in P} \varphi_c(p)\mu_c(p) + \sigma}{\sum\limits_{p\in P} \varphi_c(p) \sum\limits_{p\in P} \mu_c(p) + \sigma} \tag{11}$$

3.2.4 Regularization

For the deformable models, L_2-regularization or BE-regularization are used to promote smoother DVFs.

L_2-regularization L_2-regularization, which squares the spatial gradients of the DVF, is defined in 12.

$$\mathrm{L}_{L_2}(T) = \sum_{p\in P} ||\nabla T(p)||^2 \tag{12}$$

The spatial gradients in this formula are approximated by the differences between neighboring voxels. Specifically, for $\nabla T(p) = \left(\frac{\partial T(p)}{\partial x}, \frac{\partial T(p)}{\partial y}, \frac{\partial T(p)}{\partial z}\right)$, the partial derivative $\frac{\partial T(p)}{\partial x}$ is approximated by $T((p_{x+1}, p_y, p_z)) - T((p_x, p_y, p_z))$, and similar for $\frac{\partial T(p)}{\partial y}$ and $\frac{\partial T(p)}{\partial z}$, mirroring [105].

BE-regularization The BE loss [31] is denoted in Equation 13. The metric comes from plate theory, as it was originally formulated to calculate the bending energy of a thin plate of metal.

$$\mathrm{L}_{\mathrm{BE}}(T) = \sum_{p\in P} \left|\left|\nabla^2 T(p)\right|\right|^2 \tag{13}$$

In this equation, ∇^2 denotes the Hessian matrix, i.e.

$$\nabla^2 T(p) = \begin{bmatrix} \frac{\partial^2 T(p)}{\partial x^2} & \frac{\partial^2 T(p)}{\partial xy} & \frac{\partial^2 T(p)}{\partial xz} \\ \frac{\partial^2 T(p)}{\partial xy} & \frac{\partial^2 T(p)}{\partial y^2} & \frac{\partial^2 T(p)}{\partial yz} \\ \frac{\partial^2 T}{\partial xz} & \frac{\partial^2 T(p)}{\partial yz} & \frac{\partial^2 T(p)}{\partial z^2} \end{bmatrix}$$

where the same approximations of the partial derivatives are used as previously.

4 Experiments

The methods explained in the previous section are tested to assess their registration performance on FFoV CT scans of the male pelvic region, i.e. the area around the prostate. In this section, the data used and the pre-processing done for training and testing the models are described, as well as the criteria used for evaluation of the experiments.

4.1 Dataset

4.1.1 Dataset 1: Training, validation and inter-patient test set

Firstly, a proprietary dataset of MVision AI was used composed of 390 scans of patients from 15 collaborating clinics or hospitals. All scans in the dataset belong to different patients.

The pixel spacing of the scans varies but is centered around 1 mm in the x- and y-directions. In the z-direction, the slice thickness is larger, centered around 2.5 mm. This resolution is typical for medical scans [13]. The scans have size $512 \times 512 \times s$ voxels, with the number of slices s between 44 and 677, except for one smaller scan, which was excluded.

For most scans, some manual segmentations are available. The segmented ROIs differ per scan; For most scans, only 1 or 2 segmentations are available (usually only the prostate, or the prostate and the seminal vesicles). For 4 scans, no annotations were available; These scans were discarded from the dataset.

Out of all the ROIs that were segmented for the various scans, not all are relevant to assess the registration quality. Brock, Mutic, McNutt, *et al.* [13] defined lists of ROIs recommended for assessment of image registration quality per site. For the pelvic region, they recommended the symphysis pubis, sacroiliac joint, sacrum, iliac crest, femoral head, prostate, and penile bulb. The last two are relevant for prostate cancer. MVision AI compiled a similar expert list of the most clinically relevant ROIs, based on the ROIs distinguished by its segmentation models. They picked the following 12 ROIs: (1) The whole body, (2) the bowel bag, (3) the left and (4) right femurs, (5) the L4 and (5) L5 vertebrae (VB), (7) the pelvic bone, (8) the penile bulb, (9) the prostate, (10) the sacrum, (11) the seminal vesicles, and (12) the bladder. These ROIs were used for training and evaluation. The ROI locations are shown in Figure 14.

As for many ROIs, no manual segmentations were available, automatic segmentations were used instead for the training for the ROIs. These segmentations were not checked by an expert, so their quality may be lower than the manual segmentations. Out of the 4103 masks available for the training scans (note that not every ROI is visible in every scan), 1166 (28%) were manual segmentations. The most commonly available manual segmentations were for the bladder (299 of 345 masks) and prostate (272 of 341 masks). The least commonly available were the L4 and L5 vertebrae, sacrum, pelvic bone, and bowel bag, all being available for just 41 training scans.

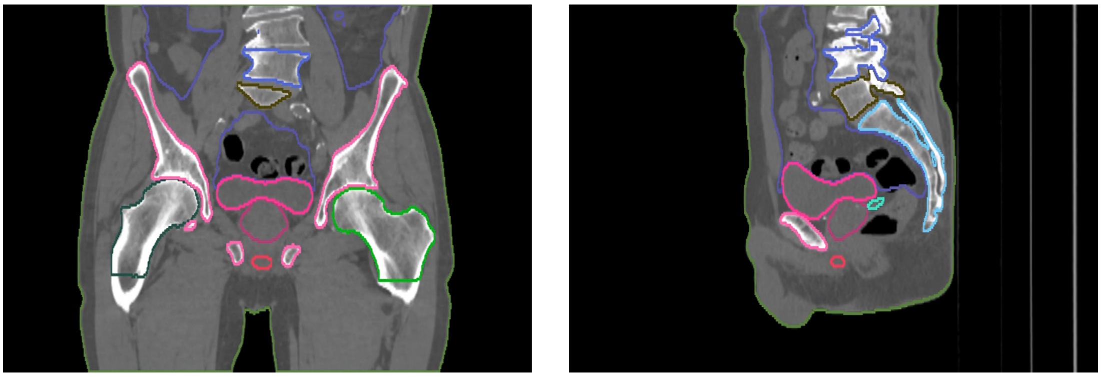


Figure 14: A coronal and sagittal slice of a patient showing the 12 ROIs.
ROIs: ● Bladder, ● Body, ● Bowel bag, ● Femur L, ● Femur R, ● L4 VB, ● L5 VB, ● Pelvic bone, ● Penile bulb, ● Prostate, ● Sacrum, ● Seminal ves.

Out of the remaining scans, the 20 scans with the most manual segmentations out of the 12 ROIs used were selected for model selection and testing, as these segmentations are guaranteed to be accurate. 20 other random scans were used for tracking performance during training. The other 345 scans were used for training.

4.1.2 Dataset 2: Intra-patient test set

An additional dataset was utilized consisting of 54 scans belonging to 25 patients, meaning that for every patient, 2 or 3 scans were available, taken at different times. These scans were used as the test set, to evaluate the intra-patient registration capacity of the models.

In total, 34 pairs of scans were available (one pair if two scans were available for a patient, and three pairs if three scans were available). The scans come from one of the hospitals that also provided scans for the aforementioned dataset. Although they are scans from the same hospital, the different scans of each patient were not necessarily taken with the same scanner.

For these scans, no manual segmentations were available; Therefore, automatic segmentations of all the 12 ROIs were created, as was done for the training set. Again, these were not checked manually by an expert, potentially leading to errors in the segmentations.

As was the case for the other dataset, these scans all have a size of $512 \times 512 \times s$ voxels, with a spacing that varies per scan.

4.1.3 Training and test sets

As said, the training scans are from different patients. These could be combined as a registration task (inter-patient registration) during training. However, as the focus of this thesis is mainly intra-patient registration, this is not ideal, as inter-patient variation is much larger than intra-patient variation, thus leading to a much more difficult task. Therefore, a training schema with synthetic augmentations was also used, where a scan would be registered to a synthetically augmented version of itself or the other way around.

For validation and model selection, one set was used, consisting of a set of 20 scans from dataset 1 with synthetic augmentations. For model testing, two sets were used. Firstly, from the scans from the validation set, 20 inter-patient combinations (out of the 380 possible combinations) were used as the inter-patient test set. Secondly, the 34 intra-patient combinations of scans from dataset 2 were used as the intra-patient test set, as stated previously. Three samples from each of the validation and test sets are shown in Figure 15.

4.2 Data processing

4.2.1 Preprocessing

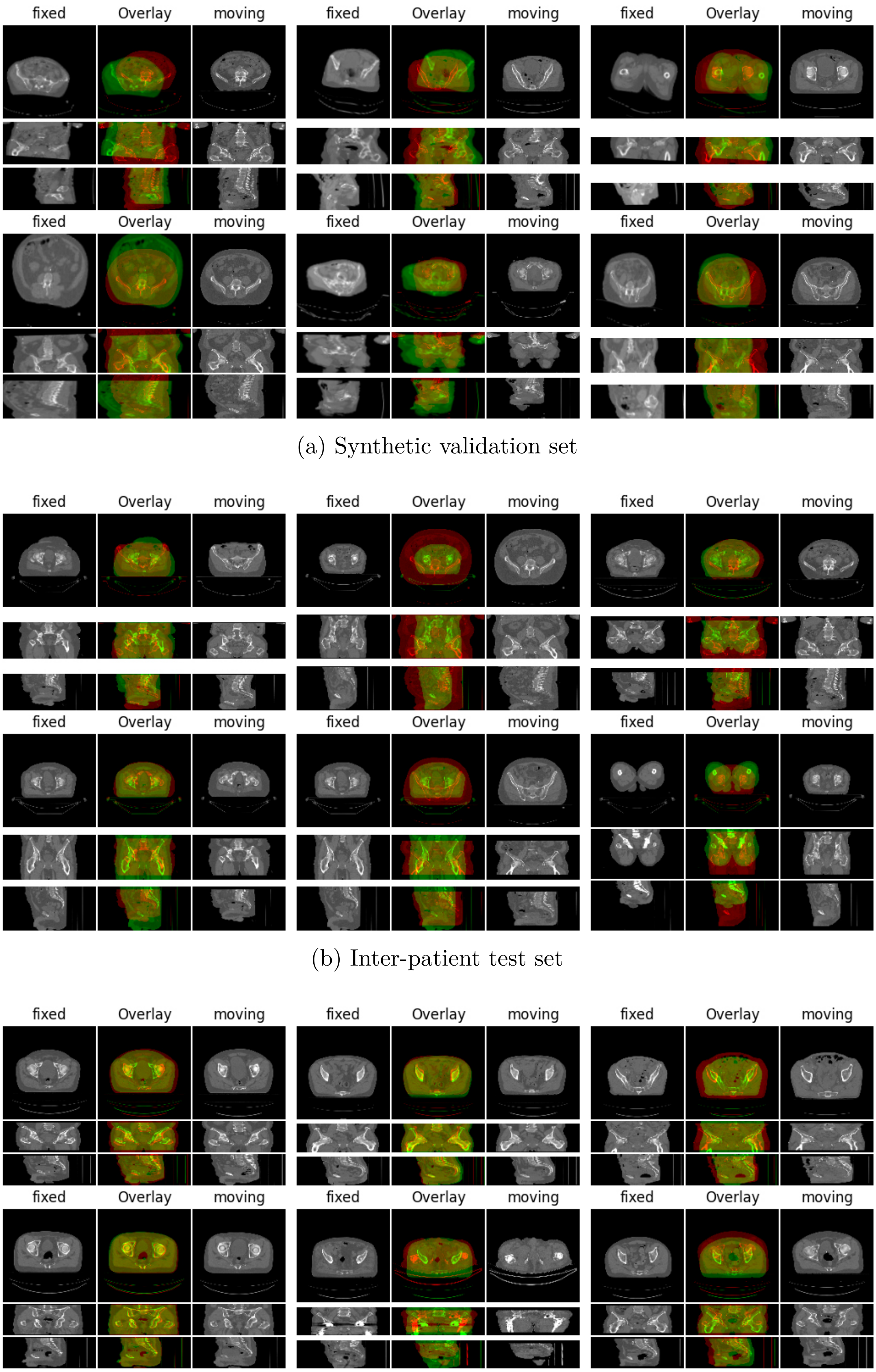
The scans were downsampled by a factor of 4, making the scan size $128 \times 128 \times s$ voxels, where s ranges between 11 and 169, in order to reduce the computation time of the experiments. As the scans were not resampled isotropically, the image spacing remains larger in the z-direction. Although isotropic resampling is commonly done in image registration research, this step leads to more interpolation steps, which was thought to affect the results.

Furthermore, the intensities of the scans were clamped with the lower and upper bound set to -500 and 800 HU respectively, following Mok and Chung [26], in order to have the model focus on the anatomy and ignore any noise outside of the body or artifacts due to metal implants. After this, the intensities were normalized to the unit interval $[0, 1]$.

4.2.2 Sizing the scans

For the affine model, a pair of moving and fixed scans is processed at once as a whole. To keep the model input size consistent, the scans thus need to be cropped or padded. In this thesis, the scans were cropped or padded to a size of $128 \times 128 \times 48$ voxels; The depth of 48 voxels is slightly above the mean depth of 46.36 the downsampled scans, and the median of 41 voxels. Thus, around 68% of scans needed padding to fit this size, while the rest of the scans were cropped.

For the deformable model, the scans were not inputted as a whole, but in patches. Due to GPU memory limitations, patches are quite frequently used in registration research, reportedly in roughly half of the publications on 3D-3D registration [5]. Patch-based training can negatively impact performance if the patches are too small, as this leads to observed deformations that go out of the boundaries of the patch. In this work, affine pre-registration and a relatively large patch size were used to mitigate this. One advantage of using a patch-based method is that, if not used, the scans would need to be padded quite heavily to allow all scans to fit into the input size. Since the scan thickness differs quite a lot, the scans would be padded to at least 169 voxels thickness (the thickness of the largest scan). And as the smallest scan is only 11 voxels thick, a padded version of this scan would have over 90% padding. In contrast, using the patch-based approach, only scans smaller than the patch size need to be padded.



(a) Synthetic validation set

(b) Inter-patient test set

(c) Intra-patient test set

Figure 15: Samples from the two validation sets and the test set. Axial, coronal, and sagittal center slices are shown for fixed and moving scans from the three sets, as well as an overlay of the fixed and moving image.

In this thesis, slices of thickness $a = 48$ were used as patches. The input size is thus $128 \times 128 \times 48$, which is the same as in the affine model; However, if the scan is bigger than the patch, in the deformable model, multiple patches are passed through the model, while in the affine model, the scan only needs to be passed once.

One challenge of using patches is the patch fusion process during inference, as discussed by Fu, Lei, Wang, *et al.* [5]. Patch fusion can lead to grid-like artifacts, especially along the edges of the patches. One way to mitigate this problem is to use a large patch overlap. This method was used in this research, with a patch-overlap of $a - 1 = 47$. One downside of this method is that it increases the processing time if the patch overlap is large, as is the case here. However, as the aim is to optimize performance, rather than to create the fastest application, this is not a major issue.

Zero padding was used for all padding; Due to the normalization used, the empty parts of the scan also have 0 intensity, so this value is appropriate.

4.2.3 Data augmentation

Data augmentation encompasses a range of techniques that enhance the size and quality of a training dataset. It has successfully been applied to prevent overfitting of the model and improve its generalizability, especially on smaller datasets [110]. Common augmentations include color augmentations (e.g. changes in brightness or contrast, adding noise, blurring or sharpening), rotation, shearing, scaling, flipping, random cropping, random erasing, elastic deformations, and combinations of these techniques.

As stated, synthetic augmentations were applied to the scans from the first dataset to create artificial patient scans. The augmentations consisted of applying both affine and warped augmentations to the scans to create a second scan for the patients. This second scan would either be used as the fixed or moving scan for the registration task.

Next to the synthetic augmentations, additional data augmentations were also used. The data augmentations applied consist of gamma adjustments, blurring, adding noise, small deformable augmentations, small affine augmentations, and flipping. The former four were added to the fixed and moving scan separately; the latter two were added to both scans. Furthermore, the moving and fixed scans were also randomly switched.

Data augmentations can either be applied during training or in advance. If one does not have storage space limitations, the latter approach can be preferred, as no computational time and power have to be spent applying augmentations during training. However, the number of different applied augmentations needs to be large enough for the model to be able to generalize well. In contrast, the former approach leads to every sample having a different augmentation. In this thesis, the former method was primarily used. For the synthetic validation set, naturally, the latter approach is used so that the samples are always the same.

Implementation details of the data augmentations can be found in Appendix A.

4.3 Evaluation

To assess the performance of the models, they are compared to registration using state-of-the-art iterative registration tools Elastix and ANTs (see section 2.2.1).

Evaluation of the methods is done based on three metrics. Firstly, the DSC and Surface DSC (sDSC) are used, which measure the quality of the registration through the overlap of segmentations in the fixed and registered moving scan. Next to that, the percentage of voxels with a non-positive Jacobian Determinant (JC) is used, which is a measure of the smoothness of the transformation field of the deformable registration. As the methods use vector integration through scaling and squaring to approximate diffeomorphism, which should thus lead to smooth deformations, the JC is expected to be low.

The **DSC** is defined in Equation 14, where v_F^c represents the voxels, so the volume, of segment c in F.

$$\mathrm{DSC}\Big(F, T(M)\Big) = \frac{1}{|C|} \sum_{c \in C} \frac{2\left|v_F^c \cap v_{T(M)}^c\right|}{\left|v_F^c\right| + \left|v_{T(M)}^c\right|} \tag{14}$$

Note that the calculation of the Dice loss presented in section 3.2.3 is virtually the same as that of the DSC, as the sum of a binary volume $\sum_{p \in P} \varphi_c(p)$ is the same as the cardinality of the volume $\left|v_F^c\right|$ and the intersection of two volumes is the same as the voxel-wise multiplication of their binary volumes. The differences between the DSC and the Dice loss are that the Dice loss is differentiable, has a smoothing parameter, and takes the sum instead of the mean over the masks.

The DSC ranges from 0 to 1, with 1 indicating perfect alignment (overlap) of all masks, and 0 indicating complete misalignment (no overlap).

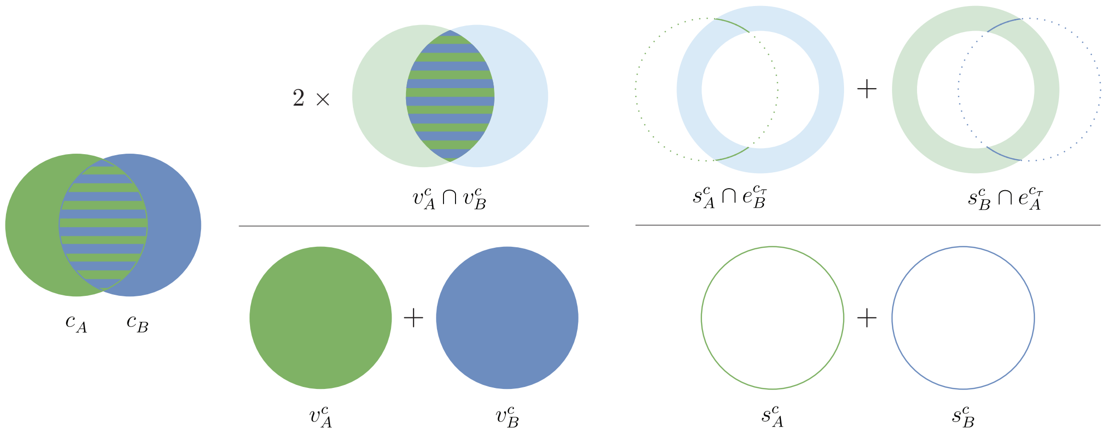
Secondly, the **sDSC** is used. Instead of looking at the overlap of the volumes of two masks, the sDSC measures the overlap of two surfaces at a specified tolerance. In segmentation, this is an important metric. It addresses the bias of the volumetric DSC to larger ROIs, where the internal volume accounts for the bulk of the score.

As v_F^m was defined as the volume of segment c in F, its surface will be denoted as s_F^c, and a border region $e_F^{c_\tau}$ at tolerance level τ. The sDSC is now defined as follows:

$$\mathrm{sDSC}_\tau\Big(F, T(M)\Big) = \frac{1}{|C|} \sum_{c \in C} \frac{\left|s_F^c \cap e_{T(M)}^{c_\tau}\right| + \left|s_{T(M)}^c \cap e_F^{c_\tau}\right|}{\left|s_F^c\right| + \left|s_{T(M)}^c\right|} \tag{15}$$

Just like for the volumetric DSC, if the surfaces all perfectly align, the sDSC would be 1; in case of a total misalignment, the sDSC would be 0. In this case, perfect alignment depends on the chosen tolerance τ: The higher τ is, the higher the score will be. It is imperative that the tolerance is set at a clinically acceptable level. In this research, τ was set to 2 mm, as this is typically clinically desired [13].

Note that the sDSC is not differentiable, so it cannot be used as a loss function, like the DSC. For more information on the sDSC, see Nikolov, Blackwell, Zverovitch, *et al.* [111]. The DSC and sDSC metrics are illustrated in Figure 16.



(a) Segment c in scans A and B

(b) DSC

(c) sDSC at tolerance τ

Figure 16: Illustrations of the DSC and sDSC metrics calculated on a segment c in two scans, A and B. The segments are visualized as circles, and their surfaces are visualized as rings, with thick rings showing the tolerance region around the surface.

Lastly, the **JC** is used to assess the smoothness of the predicted deformation. The Jacobian matrix $J_T(p) = \nabla T(p)$ captures the local properties of T around voxel p. At locations where $\det J_T(p) > 0$, the local deformation is diffeomorphic, and as such, no folding occurs. This metric, as defined in Equation 16, measures the fraction of voxels where this is not the case, i.e. in what percentage of the T folding occurs. The most optimal value for this metric is 0 and, as it is a percentage, the maximum (and worst) value is 100.

$$\text{JC}(T) = \frac{100}{|P|} \sum_{p \in P} \delta\Big[\det J_T(p) \leq 0\Big] \tag{16}$$

The JC is only calculated for deformable registrations, as affine transformations can never lead to folding.

4.4 Overview of experiments

Experiments were first done with the affine model, testing two training methods (supervised and unsupervised learning) and training datasets (synthetic and inter-patient). The best affine model was evaluated in detail on the test sets. This model was then used as the basis for the experiments with the two deformable architectures, i.e. both during training and testing, scans were first passed through the affine model before the deformable model was used.

For the deformable model, three training methods (supervised learning, weakly supervised learning, and unsupervised learning) were tested. For the unsupervised learning, three loss functions (NCC and global and local MI) were tested; Furthermore, regularization with L_2-loss and BE-loss was also compared. For VoxelMorph, bidirectional loss calculation was used. Again, the two training datasets (synthetic and inter-patient) were tested. The best deformable models (used on top of the affine model) achieved with either architecture were extensively evaluated on the test sets.

In the evaluation of both the affine and the deformable models, registrations with ANTs and Elastix were taken as a baseline for comparison. Although many papers use these methods with just the default settings, since these are rarely optimal, some testing was done to achieve better registrations by changing various hyperparameters (number of resolutions, number of iterations, similarity metric, regularization, etc.). More information about these settings, can be found in Appendix A, with results in Appendix B.3.

4.5 Implementation details

Training of the DLIR models was done with the Adam optimizer. The maximum batch size that could be fit on the GPU was used for the experiments. For training the affine model, a learning rate of 1e-5 was used, with a batch size of 32. For training the VoxelMorph models, a larger learning rate of 3e-4 was used, with a batch size of 8. For training the LapIRN models, a learning rate of 1e-4 was used, with a batch size of 2. The models were trained until convergence was reached, or to a standard 86,000 steps for VoxelMorph and 180,000 steps for LapIRN. For the affine model, a much larger amount of steps was needed to bring the training and validation error close to 0.

The DLIR models were implemented in Python using PyTorch [112], one of the most popular DL libraries [5]. Data augmentations were done using MONAI [113] and TorchIO [114]. Implementation details can be found in Appendix A.

Elastix was implemented using SimpleElastix [115], an open-source extension of SimpleITK that provides bindings of Elastix for a variety of programming languages; In this case, Python was used.

ANTs was implemented in Python using the ANTsPy library [116] that wraps the ANTs C++ library.

The models were trained on multiple computers, all with an NVIDIA GeForce RTX 3090 GPU, and either an Intel Core i9-10900K CPU @ 3.70GHz or an Intel Core i7-11700K @ 3.60GHz.

5 Results

In this section, the results of the conducted experiments are presented. Firstly, the best performing affine model is evaluated, and compared to the baseline affine results achieved with ANTs and Elastix. Next, the results of the best affine model followed by the best LapIRN and VoxelMorph models are evaluated and again compared to the baseline results from ANTs and Elastix. Lastly, the influence of some key hyperparameters on the performance of the deformable models is explored.

5.1 Affine registration

The best affine model found was trained with the 3D ResNet-18 architecture, using both supervised learning using synthetic data augmentations, followed by unsupervised learning using both combinations of scans and synthetically augmented scans. Training a model in a completely unsupervised method from scratch is difficult, quickly falling into the trap of local minima. Therefore, starting the training using supervised learning is beneficial. However, completely relying on supervised learning is not preferred, as this relies on synthetically augmented scans with rather small elastic deformations, not to make the ground-truth transformation inaccurate. Thus, in order for the model to generalize better to real intra-patient scans, the unsupervised learning step is useful.

The best model was trained for over 2 million steps of supervised learning and 700,000 unsupervised steps. Tests were done with various synthetic augmentations in order to determine what augmentations should be used in the synthetic training data for the model to generalize well to the intra-patient scans. The final augmentation settings, as well as other settings can be found in Appendix A.

Performance metrics of the best affine model are found in Table 5. For comparison, the performance metrics of ANTs and Elastix, run with their default affine models, are also shown. For a more detailed comparison of the test sets, in Figure 17, boxplots of the DSC scores per ROI for the model are compared to the baseline affine models. Similar tables and figures are available for the validation set in Appendix B.1.

As can be seen in the table, the model is able to achieve a moderately good registration performance on the intra-patient test set, with an average DSC of 0.62, an improvement of 0.19 over the unregistered scans. However, compared to the baseline affine models, the affine model is worse. Elastix performs best overall, with ANTs following closely. For some scans, the difference between the model performance is not as big: In about 35% of intra-patient scans, the difference between the mean DSC achieved by the 3D ResNet and the mean DSC achieved by the baselines is less than 0.05. However, on average, the model scores almost 0.10 lower than the best-performing baseline model for each scan.

On the inter-patient test set, both the 3D ResNet model and the baseline methods perform much worse. The model improved the average DSC by 0.17 over the unregistered scans to an average of just 0.32, which is rather low. As ANTs and Elastix also scored poorly, the difference in performance between the DLIR model and the baseline methods is not as big.

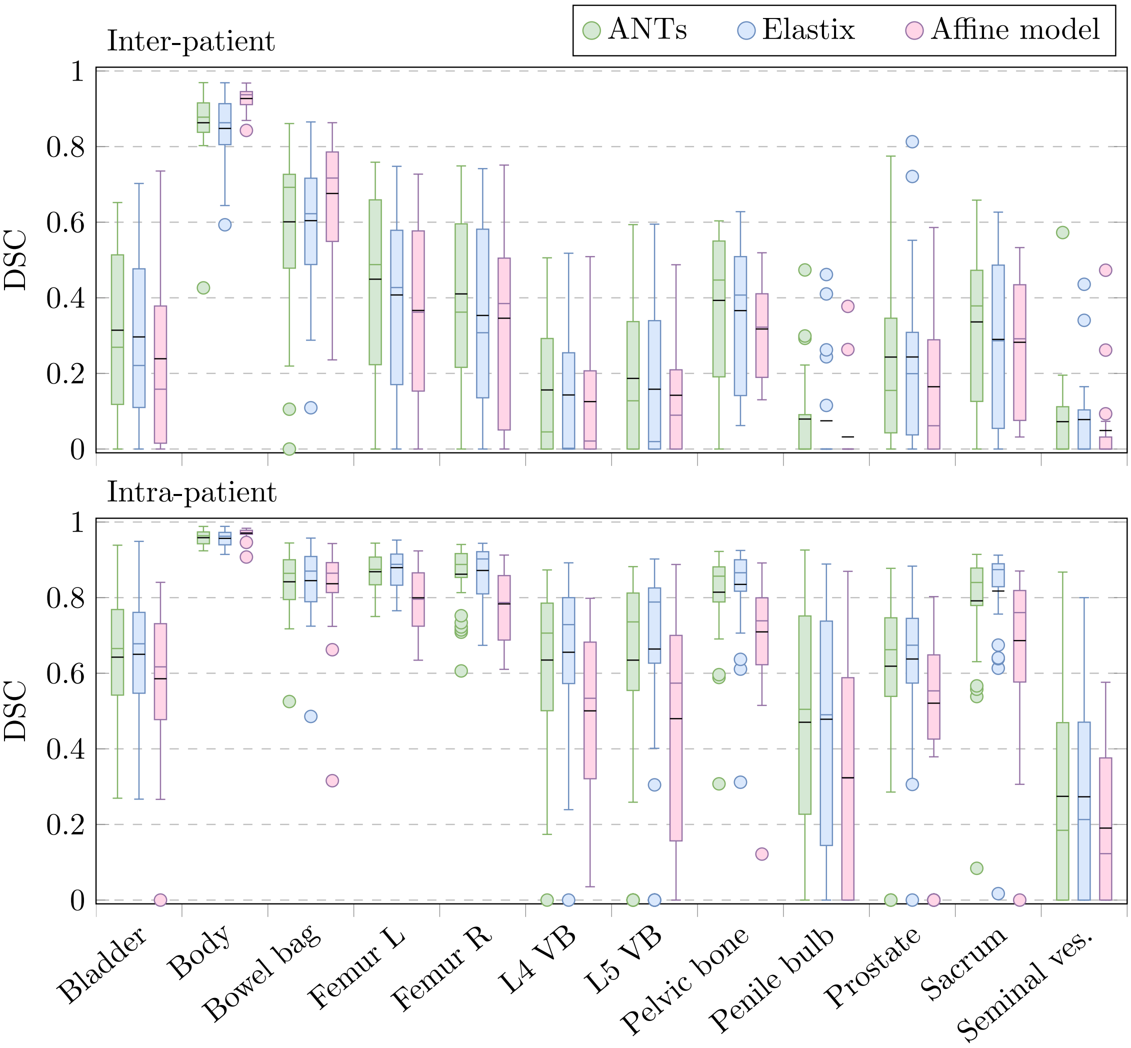


Figure 17: Boxplots of the DSC scores achieved by the best affine model compared to ANTs and Elastix. The mean scores are shown in black.

Test set	Model	DSC	sDSC	Runtime (s)
Inter-patient	3D ResNet	0.31 (0.30)	0.20 (0.15)	**2.42e-3 (1.48e-4)**
	ANTs	**0.34 (0.29)**	**0.22 (0.18)**	0.62 (0.19)
	Elastix	0.32 (0.29)	0.20 (0.17)	1.27 (0.21)
	No reg.	0.14 (0.22)	0.06 (0.08)	-
Intra-patient	3D ResNet	0.62 (0.28)	0.49 (0.21)	**2.38e-3 (1.89e-5)**
	ANTs	0.70 (0.26)	0.60 (0.24)	0.45 (0.05)
	Elastix	**0.71 (0.26)**	**0.62 (0.24)**	1.17 (0.23)
	No reg.	0.43 (0.28)	0.26 (0.17)	-

Table 5: Means and standard deviations of DSC and sDSC scores and the runtimes of the registrations of the best affine model compared to the affine registration models of ANTs and Elastix.

As expected, the model's runtimes are much faster than the baseline, with the DLIR model taking less than 1% of the time required by the baseline algorithms. It is notable that while the baseline models take slightly longer for the inter- than for the intra-patient scans, which is to be expected for iterative algorithms, as these were much more poorly aligned initially. For the DLIR model, there of course is no such difference.

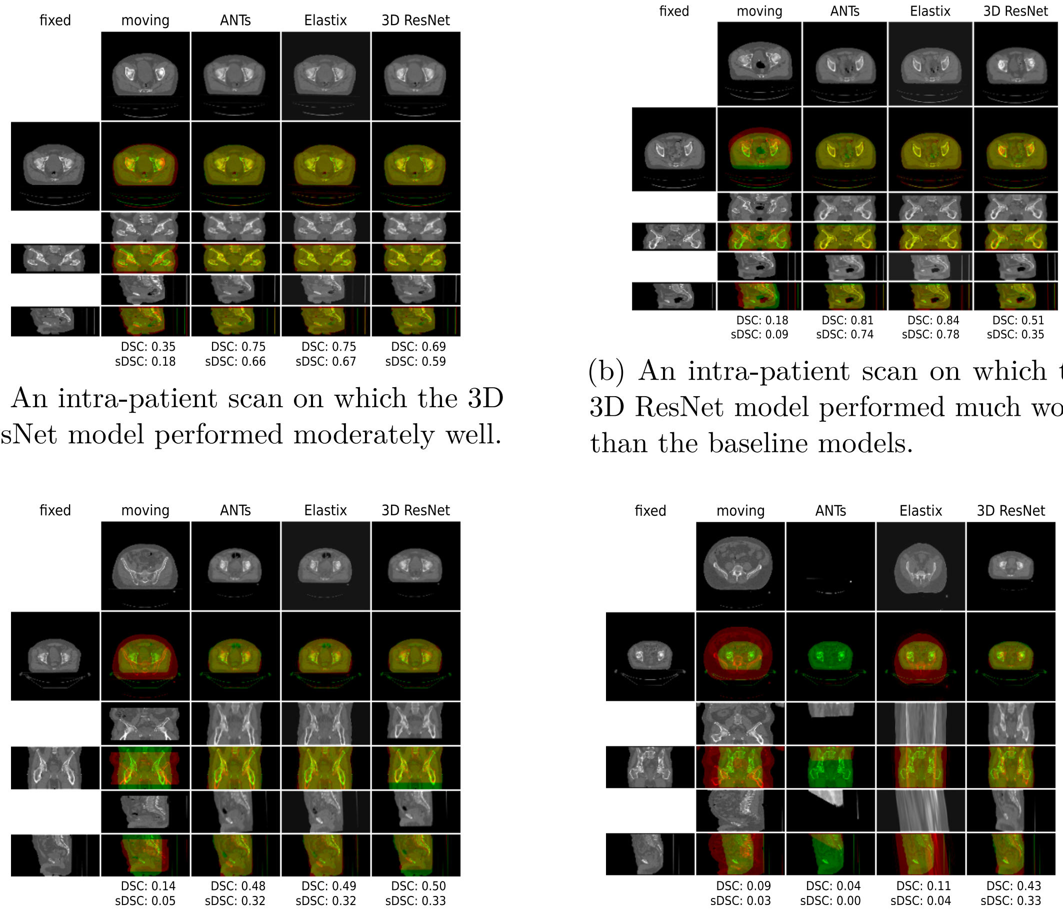
As can be seen in Figure 17, across almost all ROIs, the performance of the model is worse than the baseline methods on both test sets. The only ROI where the affine model outperforms the baseline models is the whole body. This is the ROI where the highest similarity scores are achieved by all models. However, for most of the other ROIs, good similarity scores were not achieved. On the inter-patient test set, the model is clearly worse than the baseline methods on most ROIs, while the performance of the baselines is already rather poor.

To determine the causes of the mixed performance on the intra-patient test set and the generally poor performance on the inter-patient set, individual scans were analyzed.

For the intra-patient test set, Figure 18a shows a typical case where there is a clear difference between the baseline methods and 3D ResNet's scores. In this case, the 3D ResNet scored worse on all ROIs, except for the body, which it seemed to prioritize in the alignment. One clear condition where misalignment occurs is when a part visible in the fixed scan is not visible in the moving scan. In these cases, the 3D ResNet model appears to stretch the moving image to cover the whole fixed image, leading to poor alignment. An example is shown in Figure 18b, where the model stretched the moving image to cover the top part of the fixed scan, even though this part of the body is not visible in the moving scan, leading to worse alignment than was achieved with ANTs and Elastix, and much lower DSC and sDSC scores.

For the inter-patient scans, Figure 18c shows an example where all models score approximately the same on the similarity metrics, despite that their transformations seem quite different. The baseline methods appear to rotate the scan and stretch it more than the 3D ResNet. Here, a similar issue appears as was discussed before, but now for the baseline methods, which stretch the scan to cover the entirety of the body visible in the fixed image. This example illustrates that average DSC and sDSC scores do not always give a good picture of the registration quality.

In some inter-patient cases, the transformations predicted by the baseline methods actually deteriorate the registrations: For 2 scans (10% of scans) for Elastix and 6 scans (30% of scans) for ANTs, the registered result has lower DSC and sDSC scores than the moving scan, meaning that the registration worsened the alignment. An example of this is shown in Figure 18d, where ANTs moved the moving image almost out of the frame completely, and Elastix stretched the moving image excessively. These types of failures do not occur for the 3D ResNet model: Its registrations are always better than the unregistered alignment in terms of DSC and sDSC score. In the example, the 3D ResNet seems to register the scan accurately (although visually, it does not seem like the affine optimal registration either). Regardless, this shows that the model appears to be more robust to outright failures.



(a) An intra-patient scan on which the 3D ResNet model performed moderately well.

(b) An intra-patient scan on which the 3D ResNet model performed much worse than the baseline models.

(c) An inter-patient scan on which all models did moderately well.

(d) An inter-patient scan on which the baseline methods performed poorly.

Figure 18: Examples of affine registration results. Axial, coronal, and sagittal center slices and overlays are shown for the fixed scan and the moving scan, as well as the warped moving scans by the three methods. Average DSC and sDSC scores are shown as well.

5.2 Deformable registration

The best VoxelMorph model was trained on the synthetic training data using a combination of unsupervised learning using NCC loss and weakly supervised learning with BE regularization. It was trained for 86,000 steps, after which it was deemed to have converged.

Similarly, the best LapIRN model was trained on synthetic data using a combination of unsupervised learning and weakly supervised learning. It was trained for 60,000 steps at each level in an unsupervised manner, after which 20,000 steps of combined weakly and unsupervised learning were used to finetune the model.

In this section, these models are evaluated in more detail, and compared against the baseline deformable registrations using ANTs and Elastix. The best registration model found with ANTs uses affine registration followed by a deformable registration using the SyN algorithm with Demons loss run for 100 iterations maximum and otherwise default settings. Results of ANTs with other settings can be found in B.3.

The best model found in Elastix consists of the default affine model, followed by the deformable B-Spline-based model run for 1000 iterations (and default settings other than that). Summaries of results achieved with Elastix with different settings that were tested can also be found in Appendix B.3.

As can be seen in Table 6, both models were able to achieve a relatively good registration performance on the intra-patient set, with average DSC scores of 0.72 for VoxelMorph and 0.79 for LapIRN. This is higher than the scores on the inter-patient test set, where average DSC scores of 0.41 and 0.48 were achieved by VoxelMorph and LapIRN, which is rather low. On the synthetic validation set, the scores were also lower, with average DSC scores of 0.63 and 0.72 respectively; see Appendix B.2. Lower scores for the inter-patient registration were already found in the affine registration, but it is clear that deformable registration is not able to improve the registration to a large extent. Of course, it can be argued that, as the inter-patient scans have much larger differences between them, high similarity scores are not attainable, even in case of a perfect registration. However, when looking at the scan results, it is clear that the registration results are far from ideal. One clear indication for this is the difference compared to the baseline models. While LapIRN is only slightly worse than these on average for the intra-patient test set, for the inter-patient test set, the difference is much larger.

The more complex, multi-resolution LapIRN architecture outperforms the simpler VoxelMorph model across all metrics. Compared to the performance achieved with the registration quality before the deformable registration is applied (i.e. after the affine registration), LapIRN was able to cause an average improvement of just below 0.20 on both the inter-patient and intra-patient test sets; The best VoxelMorph model, however, was only able to achieve an improvement of around 0.10 on both. Compared to the unregistered scans, the full DLIR pipeline of affine and deformable registration with LapIRN was able to improve the registration performance by almost 0.30.

For a more detailed comparison, in Figure 19, boxplots of the DSC and sDSC scores per ROI for the model are compared to the baseline methods. Similar figures showing plots for the validation set can be found in Appendix B.2. As can be seen, for the intra-patient test set, for over half the ROIs, namely the body, left and right femurs, pelvic bone, bowel bag, and L4 and L5 vertebrae, LapIRN scores better in terms of DSC than the baseline methods, though it scores worse for the other ROIs and metrics. VoxelMorph scores significantly worse on almost all metrics. On the inter-patient test set, both models score visibly worse than the baseline methods on all ROIs.

In the table, it is visible that the JC is low for all models, averaging less than 1%. For VoxelMorph, LapIRN, and Elastix, smoother results are achieved on the intra-patient test set, of which the scans were more similar initially. On both sets, the two DLIR models score better than Elastix, with VoxelMorph having a smoother result than LapIRN. ANTs was able to create completely smooth deformations, however, and thus outperforms all other models. This shows that, while the patch-based diffeomorphic approximation using scaling and squaring does not work perfectly, a rather smooth result is still achieved.

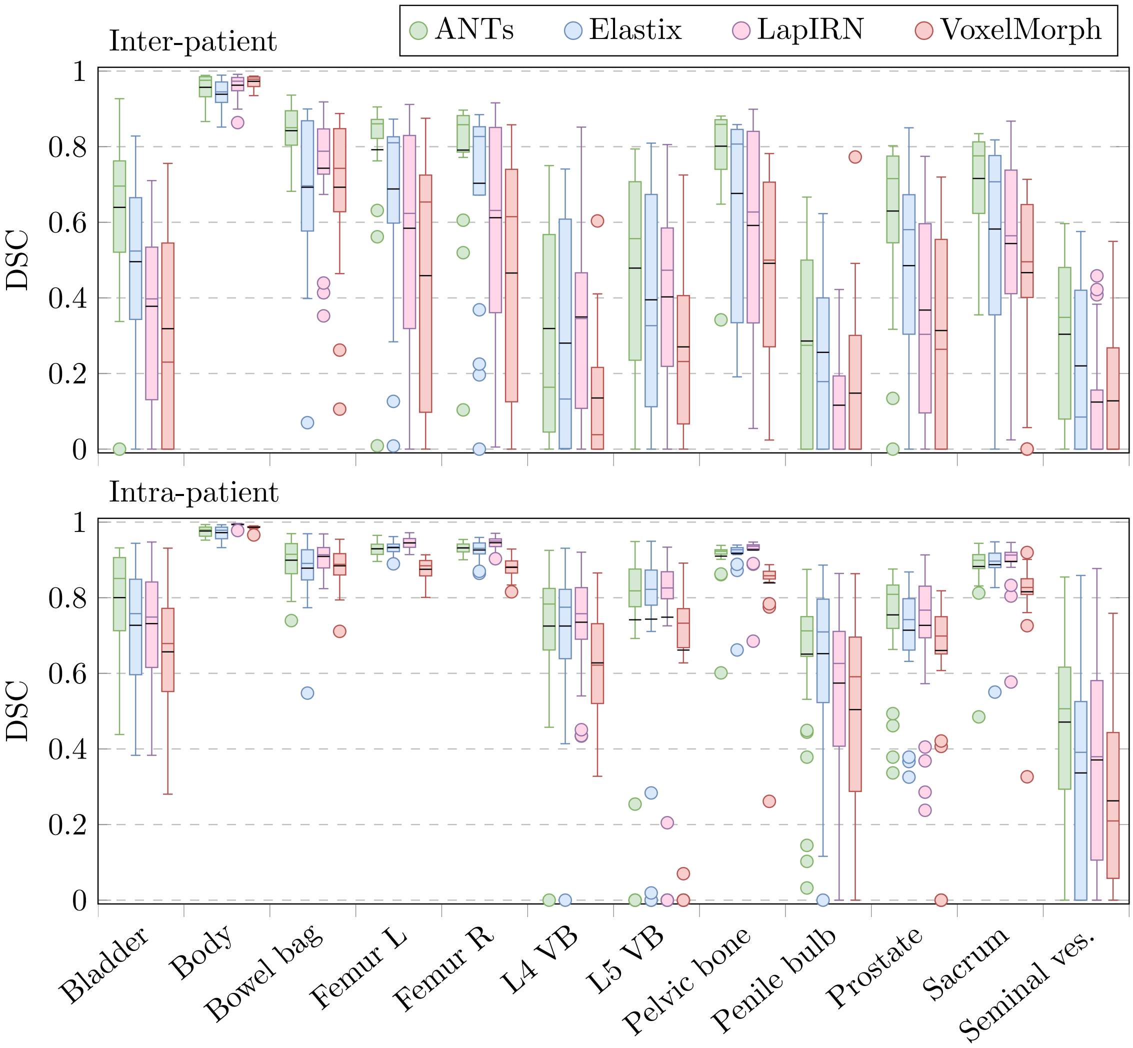
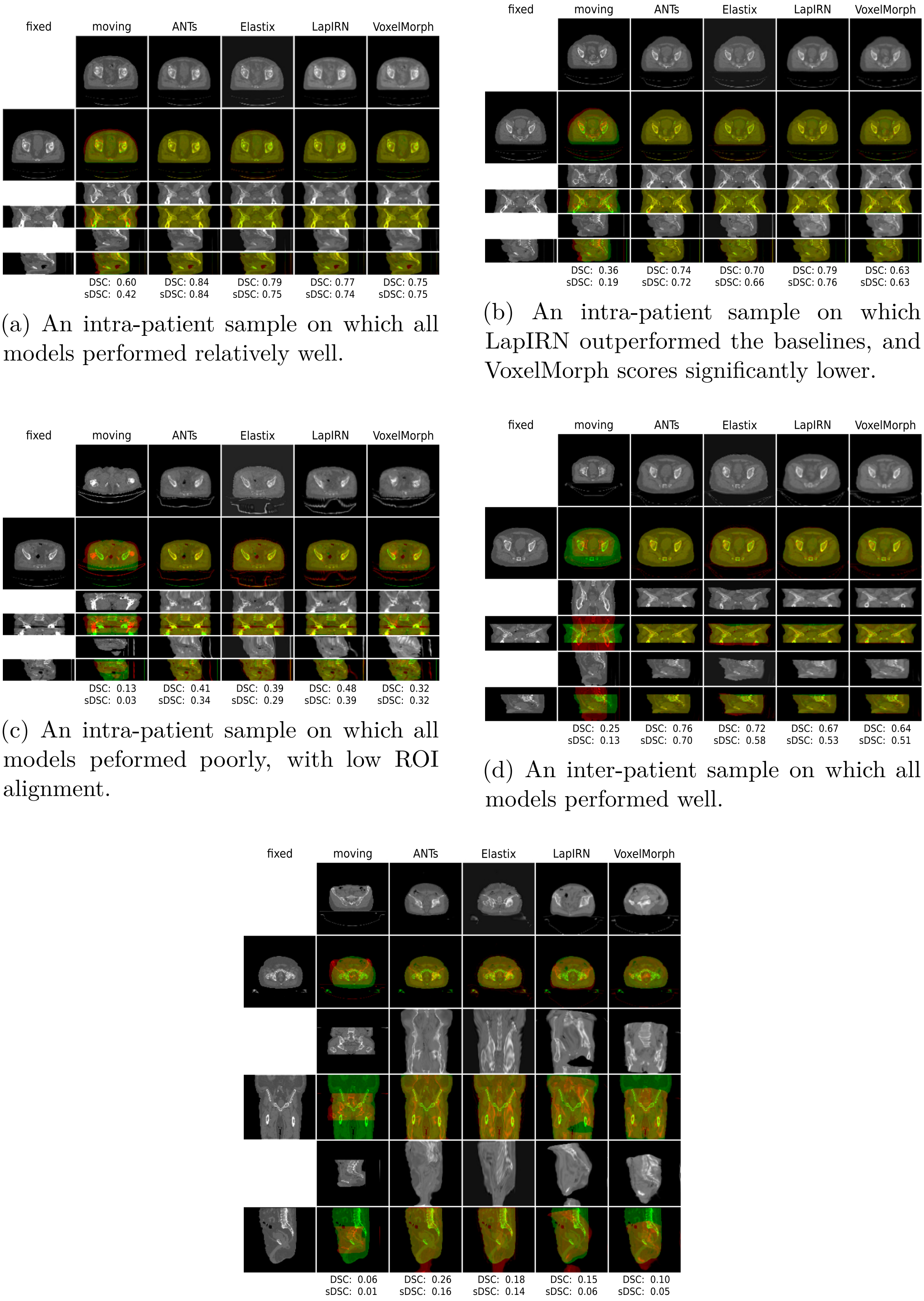


Figure 19: Boxplots of the DSC scores achieved by the deformable models compared to ANTs and Elastix. The mean scores are shown in black.

Dataset	Model	DSC	sDSC	JC	Runtime (s)
Inter-patient	VoxelMorph	0.41 (0.33)	0.33 (0.25)	0.16 (0.16)	**0.35 (0.37)**
	LapIRN	0.48 (0.33)	0.40 (0.26)	0.70 (0.40)	5.00 (6.23)
	ANTs	**0.63 (0.29)**	**0.58 (0.23)**	**0.00 (0.00)**	49.90 (20.40)
	Elastix	0.53 (0.31)	0.43 (0.24)	2.85 (3.88)	9.39 (0.95)
	No reg.	0.14 (0.22)	0.06 (0.08)	-	-
Intra-patient	VoxelMorph	0.72 (0.24)	0.66 (0.21)	0.02 (0.07)	1.80e-2 (2.38-4)
	LapIRN	0.79 (0.23)	0.76 (0.21)	0.33 (0.28)	**9.68e-3 (8.95e-4)**
	ANTs	**0.80 (0.20)**	**0.78 (0.18)**	**0.00 (0.00)**	24.63 (9.77)
	Elastix	0.78 (0.22)	0.73 (0.22)	0.32 (1.27)	8.62 (0.55)
	No reg.	0.43 (0.28)	0.26 (0.17)	-	-

Table 6: Means and standard deviations of DSC, sDSC, and JC scores, and the runtimes of the registrations on the validation sets and test set using the best VoxelMorph and LapIRN models, compared to the deformable registration models of ANTs and Elastix.



(a) An intra-patient sample on which all models performed relatively well.

(b) An intra-patient sample on which LapIRN outperformed the baselines, and VoxelMorph scores significantly lower.

(c) An intra-patient sample on which all models peformed poorly, with low ROI alignment.

(d) An inter-patient sample on which all models performed well.

(e) An inter-patient sample on which all models performed poorly.

Figure 20: Examples of registration results. Axial, coronal, and sagittal center slices and overlays are shown for the fixed scan and the moving scan, as well as the warped moving scans by the three methods. Average DSC and sDSC scores are shown as well.

In terms of runtime, both LapIRN and VoxelMorph are very fast, with average runtimes of less than a second on the test sets, almost 1000 and 500 times faster than ANTs respectively, and both over 1000 times faster than Elastix. This underlines how using DLIR dramatically reduces the time that registration takes. One note is that for the inter-patient test scans and the synthetic validation scans (see Appendix B.2), the runtimes are a bit longer. This can be attributed to the size of some of the scans, necessitating many patches. For example, for one scan in the test set with a depth of 116 voxels, 69 patches would be taken; this scan took LapIRN almost 20 seconds to register. For the DLIR models, the number of patches, and the implementation of the patch creation and aggregation, are the main factors influencing the registration speed. However, even in this case and despite the large patch overlap, the models are both much faster than the baselines.

When analyzing individual scans visually, the performance difference between the different methods can be seen. In figure 20a, an intra-patient sample is shown on which all of the models achieved a relatively good result, with ANTs achieving the highest score. For almost 60% of the scans in the test set, ANTs had the highest average DSC score out of the methods compared. The better performance of ANTs can, for example, be identified in the bladder, which is visibly better aligned in the axial slice. However, on more close inspection, the largest improvement to the DSC and sDSC scores comes from the seminal vesicles (not visible), as LapIRN completely failed to align these.

20b shows an example where LapIRN performs slightly better than the baselines, whose performances were mainly brought down by misalignment of the L4 for Elastix and the bowel bag for ANTs. VoxelMorph is evidently outperformed by the other methods, failing to capture the deformations in the lower bones correctly.

Lastly, Figure 20c shows the intra-patient scan in the test set with the lowest overall performance. As can be seen in the image, it is a difficult case, as the scans are largely misaligned, and the fixed scan suffers from a big scan artifact (e.g. due to an implant). Visually, the scan warped by LapIRN may be the closest to the actual fixed scan, but its alignment is still rather poor.

For the inter-patient scans, Figure 20d shows an example where all models perform quite well. The main differences are visible at the edges of the scan: While ANTs and LapIRN removed virtually all non-overlapping areas, there are still some visible in the Elastix result. Note, however, that this behavior by ANTs and LapIRN is not necessarily desired, as one of the scans may show a larger part of the body than the other scan, which ought to remain visible. An example of this is shown in Figure 20e. In this case, all models warped the moving scan onto parts of the fixed scan that were not visible in the moving scan, e.g. the abdominal region at the top of the fixed scan, or the legs. In this example, ANTs completely distorted the anatomy, with strongly warped bones; Elastix has a similar issue, where pelvic bones are stretched into the legs. LapIRN came up with the worst result, which even seems to feature a gap inside the warped body. Out of the methods, VoxelMorph is the only one that did not stretch the scan into the abdomen, but it still tried to cover the legs. As both of these inter-patient scans show, scans of different sizes still appear to pose a large challenge for all tested methods.

5.3 Influence of key hyperparameters

Various hyperparameters were tested in the search for the most optimal DLIR models, namely different architectures, different training data, and various supervision methods and loss functions. The effect that these parameters were found to have on the model performance is briefly discussed in the next sections. In all sections, tables in which different settings are compared are shown. These tables show the performance on both the validation set (synthetic scans) and the test sets (intra- and inter-patient scans); note, however, that the original comparison and model selection was based only on the scores achieved on the validation set.

5.3.1 Affine models

ResNet architectures For the affine model, the three different ResNet architectures were tested. As can be seen in Table 7, the ResNet-10 model clearly performed the worse, and the ResNet-18 and -34 models reached similar performance, with the former model slightly beating the latter. It must be noted that the ResNet-34 and ResNet-10 models were only trained using supervised learning, while the ResNet-18 model was trained using both supervised and unsupervised learning. It is thus possible that the performance difference would be different with added unsupervised training; however, this was not attempted.

ResNet-	Synthetic		Inter-patient		Intra-patient	
	DSC	sDSC	DSC	sDSC	DSC	sDSC
10	0.34 (0.30)	0.23 (0.18)	0.24 (0.29)	0.14 (0.13)	0.52 (0.29)	0.37 (0.19)
18	**0.40 (0.30)**	0.26 (0.19)	**0.31 (0.30)**	**0.20 (0.15)**	**0.62 (0.28)**	**0.49 (0.21)**
34	**0.40 (0.30)**	**0.28 (0.20)**	0.28 (0.28)	0.17 (0.13)	0.61 (0.29)	**0.49 (0.22)**

Table 7: Average similarity scores on experiments with the ResNet-10, -18, and -34 architectures.

5.3.2 Deformable models

Training data Firstly, training the models using the two different sources of data, either synthetically augmented scans or inter-patient scans, were compared. A test was performed using VoxelMorph trained with NCC loss and L_2-regularization with either training set. A summary of the results from this experiment listed in Table 8, shows higher scores for training on the synthetically augmented scans on all three. This is interesting, as better performance on the inter-patient test set is expected after training on inter-patient scans. An explanation may be sought in the smaller size of the inter-patient training set, compared to the infinitely large synthetic dataset, or the general difficulty of aligning inter-patient scans. An experiment using LapIRN lead to equivalent conclusions, with synthetic training data leading to better registrations.

Data type	Synthetic		Inter-patient		Intra-patient	
	DSC	sDSC	DSC	sDSC	DSC	sDSC
Synth.	**0.60 (0.30)**	**0.53 (0.25)**	**0.38 (0.33)**	**0.31 (0.25)**	**0.73 (0.25)**	**0.67 (0.22)**
Inter.	0.45 (0.32)	0.38 (0.24)	0.31 (0.34)	0.26 (0.25)	0.67 (0.26)	0.62 (0.21)

Table 8: Average similarity scores on the two test sets for the training data experiment.

Unsupervised and supervised learning Various unsupervised loss functions (NCC, MI, and local MI) were compared, as well as supervised training. For the unsupervised training, L_2-regularization was again used. For the supervised method, training on scans that were first aligned using the affine model lead to implementation issues with the resampling of the ground-truth deformation fields. Therefore, instead, it was trained on scans with only very slight affine augmentation, to simulate what scans may be like after being pre-aligned, as the affine pre-alignment is not perfect. For proper comparison, an unsupervised model using NCC loss was trained on this data as well. Of course, in testing, the affine model is still used to pre-align the scans. An overview of the performance of the unsupervised models is shown in Table 9.

As can be seen in the table, of the unsupervised models, the model trained with NCC loss manages to reach slightly higher average scores than those trained with MI. The difference is only slight, however, with the local MI model performing best on the inter- and intra-patient test sets. the NCC model was still chosen, as model selection was done based on the performance on the validation set to avoid overfitting. However, the high performance with local MI indicates that for multi-modal applications, using MI (as is common for this application) need not be a disadvantage. When comparing the supervised model, it is clear that training on scans that are not pre-aligned worsens the result on the intra-patient scans, leading to a reduction of 0.06 and 0.07 in the DSC and sDSC scores for the NCC models. However, it is also clear that the supervised model performs much worse than the NCC model used for comparison. In fact, the supervised model only improves over the affine model for the synthetic scans.

Loss func.	Synthetic		Inter-patient		Intra-patient	
	DSC	sDSC	DSC	sDSC	DSC	sDSC
NCC	**0.60 (0.30)**	**0.53 (0.25)**	0.38 (0.33)	0.31 (0.25)	**0.73 (0.25)**	0.67 (0.22)
MI	0.57 (0.30)	0.50 (0.24)	0.37 (0.33)	0.29 (0.24)	0.72 (0.24)	0.66 (0.21)
MI$_{local}$	0.59 (0.30)	0.52 (0.26)	**0.39 (0.34)**	**0.32 (0.26)**	**0.73 (0.23)**	**0.68 (0.20)**
NCC*	0.58 (0.29)	0.50 (0.22)	0.37 (0.33)	0.30 (0.24)	0.67 (0.26)	0.61 (0.21)
sup.*	0.52 (0.29)	0.41 (0.20)	0.30 (0.31)	0.21 (0.18)	0.58 (0.29)	0.46 (0.20)

Table 9: Average similarity scores on the two test sets for the experiment with different unsupervised losses. $*$ = models trained on scans with smaller affine augmentation that were not pre-aligned.

Regularization The effects of different types of regularization, namely L_2- and BE-regularization, were tested, as well as two different regularization strengths. As can be seen in the overview in Table 10, using stronger L_2 regularization worsens the result over the best results from the previous experiment. However, when the BE loss is used with this larger weight, it leads to higher similarity scores than were achieved with the default L_2 loss. For the test sets, the differences are only slight, making them not visible in the table due to the rounding. Of course, it is expected that regularization does not have huge effects, since it only has a small effect on the loss. The effect of the regularization term is visible in the JC scores, however, with the larger λ bringing down the scores on all three datasets. The worst JC scores are achieved by the BE model with $\lambda = 0.25$. This is to be expected, as the regularization is meant to lead to a smoother result.

Reg. func. (λ)	Synthetic		Inter-patient		Intra-patient	
	DSC	sDSC	DSC	sDSC	DSC	sDSC
L_2 (0.25)	0.60 (0.30)	0.53 (0.25)	**0.38 (0.33)**	**0.31 (0.25)**	**0.73 (0.25)**	0.67 (0.22)
L_2 (2)	0.57 (0.30)	0.50 (0.24)	0.37 (0.33)	0.29 (0.24)	**0.73 (0.24)**	**0.68 (0.22)**
BE (0.25)	0.59 (0.30)	0.52 (0.26)	0.37 (0.34)	0.30 (0.26)	0.72 (0.24)	0.67 (0.21)
BE (2)	**0.61 (0.30)**	**0.54 (0.25)**	**0.38 (0.34)**	**0.31 (0.26)**	**0.73 (0.24)**	**0.68 (0.21)**

Table 10: Average similarity scores on the two test sets for the regularization experiment.

Weak supervision The effect of adding weak supervision to the training was also investigated. The final models for both VoxelMorph and LapIRN were finetuned using weak supervision. However, weak supervision can also already be used from the start of the training, as was attempted in some experiments with good results. In table 11, a summary of an experiment with VoxelMorph is shown, where unsupervised learning from scratch, training solely using weak supervision from scratch, and finetuning a model trained in an unsupervised manner with weak supervision are compared. Although the results are very similar, the combined approach gave the best result. Similar findings were also observed using LapIRN.

Training strategy	Synthetic		Inter-patient		Intra-patient	
	DSC	sDSC	DSC	sDSC	DSC	sDSC
Unsup.	0.61 (0.30)	0.54 (0.25)	0.38 (0.34)	0.31 (0.26)	**0.73 (0.24)**	**0.68 (0.21)**
weakly sup.	0.62 (0.28)	0.55 (0.23)	0.40 (0.33)	**0.33 (0.25)**	0.72 (0.24)	0.66 (0.20)
Both	**0.63 (0.28)**	**0.56 (0.23)**	**0.41 (0.33)**	**0.33 (0.25)**	0.72 (0.24)	0.66 (0.21)

Table 11: Average similarity scores on the two test sets for the experiment with weakly supervised learning.

6 Discussion

6.1 Conclusions

In this project, DL methods for medical image registration were compared on the task of aligning inter- and intra-patient male pelvic FFoV CT scans. The methods used employ a cascade of an affine (global) registration and a deformable (local) registration. For the affine registration step, a 3D ResNet model was used. The two deformable methods that were investigated are VoxelMorph, the most commonly used DLIR framework, and LapIRN, a recent multi-resolution DLIR method and one of the winners of the Learn2Reg 2020 MICCAI Registration Challenge. The two registration steps were trained separately. For the affine registration step, both supervised and unsupervised learning methods were employed. For the deformable step, multiple training methods and loss and regularization functions were compared. The training was done on synthetically augmented CT scans. The results were compared to results obtained with two top-performing iterative image registration frameworks, ANTs and Elastix.

The best-performing DLIR methods performed only slightly worse than the baseline iterative methods on the intra-patient scans, but worse on the inter-patient scans. Especially in the affine registration step, the DLIR methods were largely outperformed. One advantage over the baseline affine methods, however, is that the DLIR method never decreased the alignment, which happened for some of the registrations with the affine baselines. When followed by LapIRN in the deformable step, the DLIR approach managed to perform similarly to the baseline methods, even managing to outperform them on 7 out of 12 ROIs on the intra-patient scans. Furthermore, the methods achieve a very significant time speedup compared to the baseline methods. The multi-resolution LapIRN architecture significantly outperforms VoxelMorph, of which the performance trails far behind the other methods. LapIRN does have a longer runtime and slightly higher JC scores, however.

While the DLIR models were able to achieve moderately good performance on the task of intra-patient registration, the performance was nowhere near good enough for clinical applications. This is also the case for the baseline methods. In the intra-patient registration tasks, none of the methods tested were able to align some ROIs like the bladder and seminal vesicles correctly. Furthermore, the inter-patient registration tasks seemed to be too challenging for the methods. However, this may improve in the future with further improvements of DLIR, which was exemplified by the performance improvement of the LapIRN model over the older VoxelMorph model. In any case, the results show great potential for using DLIR for deformable image registration of pelvic region scans in the future.

6.2 Contributions

As most current research only focuses on deformable registration and works with scans that have previously been rigidly registered, the multistage registration pipeline of both affine and deformable registration using DL that was used in this project is academically relevant. The comparison made in this thesis showed that affine registration with a 3D ResNet followed by deformable registration with the multi-resolution LapIRN architecture performs almost as well as the iterative baseline methods, with the LapIRN architecture clearly outperforming the more widely used VoxelMorph architecture in the deformable registration step. This mirrors the findings of Mok and Chung [26] in their tests with brain MR scans. However, it was shown that while the method is able to give good registration of most ROIs, some ROIs like the bladder and seminal vesicles remain a challenge for all methods tested. Thus, there is a definite need for improvements to achieve the level necessary for clinical applications. Furthermore, the task of registering inter-patient scans, which is less frequently attempted in the literature, was shown to be significantly harder. The models especially struggled with padded scans of which the contents did not fully overlap.

Next to that, this project contributes to the literature by focusing on the registration of CT scans of the male pelvic area, a task rarely explored in DLIR literature. As CT-CT registration for this region has a considerable and growing clinical relevance, it is imperative to test the extent to which methods that have been found to work well for other regions also provide sufficient results in this region.

For affine registration, in this work, a ResNet-based architecture, which are common in a variety of applications, was employed. Unfortunately, this method did not manage to achieve the desired performance, showing that the affine registration is not as simple as often (implicitly) assumed. More research into affine or rigid registration using DLIR is thus advised.

For the deformable registration step, the architectures used, VoxelMorph and LapIRN, had already been used in registration literature. In this work, however, they were explored further with novel loss functions and training methods. In previous literature, supervised learning had not been used with either, and weakly supervised learning had not been used with LapIRN. Supervised learning with synthetic augmentations did not lead to good results, compared to those achieved with unsupervised learning. Adding weak supervision, however, improved the registration performance in both methods. Furthermore, the MI loss function (in its global and local formulation) had not been previously used with LapIRN, and neither architecture had been tested with BE loss functions. The effects of the different loss functions for unsupervised training did not appear to be large; Bigger gains could potentially be made in improvements in the architecture or training setup. The local MI loss was shown to perform only slightly worse than NCC loss in this particular task. Adding BE regularization was shown to lead to an improvement over training with L_2 regularization.

6.3 Limitations

In this section, some limitations of the current research project are discussed.

Data In this project, training was done on synthetically augmented scans of inter-patient scans. As the differences between these pairs of scans do not closely resemble realistic augmentations, this may have hindered the learning of the model. For example, in the synthetically augmented scans, bones routinely get warped, while this is something that would never occur in intra-patient scans. Therefore, if possible, a "smarter" model could potentially be trained on intra-patient training data, if this were available. As was shown, synthetically augmented scans seem to provide a good alternative, but it would be good to confirm this through an experiment.

Another limitation of the current research is that it uses private data for both training and evaluation. Unfortunately, for the chosen task of CT-CT intra-patient registration of the male pelvic region, no commonly used public dataset is available. However, if possible, using public datasets for at least evaluation would have facilitated future comparison.

Preprocessing In this work, only downsampled scans were used, for both training and evaluation. While some small experiments (not included in the result section) showed that evaluating on downsampled scans does not lead to largely different similarity scores, the literature shows that training on full-resolution images can lead to slightly better registration results [78].

Next to that, the scans were padded to match each other. As was found, the models and baseline methods struggled most with padded scans of which the contents that did not fully overlap. Different ways of loading the scans that do not require padding could thus be explored. Secondly, in the format the scans were saved in, the spacing of the scans was not encoded. It was assumed that as the spacing was approximately the same in all scans, the model would learn to deal with this by itself. However, for iterative registration methods, saving the scan in a format where the original spacing is present (e.g. NIFTI) is preferred. In many DLIR publications, scans are isotropically resampled, so that spacing does not have an effect at all. This difference in preprocessing makes it harder to compare the results to previous work on both iterative or DL-based image registration, and likely negatively affected the performance of the baseline methods.

Model and training In this research, the model architectures were not changed, e.g. the number of layers and number of filters per layer were kept constant. However, the architecture is one of the key ingredients for getting a good performance, as is evident from the performance difference between VoxelMorph and LapIRN. Adding more layers, more filters per layer, more skip connections, or different residual blocks (such as ResNeXt or Wide ResNet blocks) would have more than likely improved the performance. Additionally, further hyperparameter optimization for the methods used is recommended. The latter would likely improve the performance slightly.

On a similar note, the training schemes used in this research likely could have been tweaked as well, for more optimal results. For example, the training was not done in an end-to-end manner, so training of both an affine and a deformable model simultaneously, which usually leads to better results. Furthermore, testing the models in a manner where affine and deformable transformations are combined so that the scan is only re-sampled once would likely improve the resulting images visually.

One surprising finding is the very poor performance of the deformable models trained with supervised training, which is not consistent with the literature. The fact that due to programmatic limitations this training schema could not be used with pre-aligned scans seems to have affected the performance. However, this probably does not explain the poor performance in itself. Therefore, more experiments with supervised training are warranted.

Evaluation Another limitation of the research is the quality of the evaluation. The evaluation was mainly done using various scores: DSC, sDSC, and JC. sDSC, which is an improvement of the DSC in a sense, is not commonly used in the image registration literature, which mainly sticks to the regular DSC. Using this metric thus already provided an improvement compared to the evaluation seen in many publications. However, still, the metrics used do not directly correspond to the practical usability of the models. Good values for these metrics do not always mean that a transformation is realistic, as the scores are merely based on a select set of ROIs and the transformation smoothness.

An additional limitation for the evaluation is that for the intra-patient test scans, the segmentation masks were generated automatically, and were not checked by experts. While it is unlikely that the segmentation software made large mistakes, it would have been better if this was adequately assessed.

6.4 Suggestions for future work

Data Based on the first limitation, using intra-patient scans for training would likely improve the models. While obtaining a large intra-patient dataset would be challenging, adding a set of intra-patient scans to a training dataset with synthetic and/or inter-patient scans can lead to stronger models.

Because the registration task does not differ greatly per region of the body (unlike, for example, medical image segmentation, where different ROIs need to be identified in each part), it would be interesting to test the same model set-up for different regions of the body. One approach that can be tried in future research is to train a "universal" model using scans from multiple regions; such a model could be applied to similar scans from any region.

On a similar note, moving to other modalities would also be interesting. For the male pelvic region, the existing model could also be tested for CT-CBCT registration, as these modalities are relatively comparable in terms of intensity. Moreover, CBCT scanning is more often used for rescans than CT scanning, so moving to CBCT would be more clinically relevant for intra-patient registration. Next to that, the models could also be trained for CT-MR registration, which is also clinically important for the pelvic region. However, as this task is much more challenging, this would likely warrant various changes, e.g. to the loss functions used.

Preprocessing In future research, resampling the scans isotropically would be a useful step, as is used in many publications and can potentially improve performance. Furthermore, using full-resolution scans would likely lead to improved results. Experiments could also be done with different patch sizes. Next to that, as the ROIs with the worst alignment were generally soft-tissue ROIs, the intensity could be clamped to to [-200, 200] HU (instead of [-500, 800] HU), as this is the range where the intensities of soft tissues lie, to make these ROIs stand out more.

Next to that, as the models mainly struggled with padded scans of which the contents did not fully overlap, the synthetic data augmentations should likely be changed. Currently, in the synthetic training data, parts of a scan can only not be visible in the other scan if they fall out of the frame. Randomly leaving out parts of images could, for example, be of help for this.

Model Firstly, end-to-end training of the affine and deformable models would likely lead to improvements, as was demonstrated by Zhao, Dong, Chang, *et al.* [59]. This could either be done using completely synthetic augmentations, where the augmentations are known, or based on results from Elastix, ANTs, or another traditional method. The latter option is an avenue that would be interesting to explore for the deformable model, having successfully been applied by Sentker, Madesta, and Werner [51].

For the affine model, the model used in this research was not taken directly from previous work. Although it managed to get moderately good results, different models could be tested in the future. One addition that was observed in many papers in the literature is using multiple fully connected layers at the end of the network. Furthermore, many authors have chosen to separate the values for rotation, translation, scaling, etc., rather than predicting the transformation matrix directly. Another potential improvement that was mentioned in the literature is removing the maximum pooling layers, as these can obscure small movements [117].

For VoxelMorph, the calculated DVF was only half the size of the output image, which improved training. With a progressive training approach, perhaps a full resolution VoxelMorph model would be able to improve the result. However, more than VoxelMorph, the multi-resolution approach of LapIRN showed clear potential. This multi-resolution approach could also be combined with other base architectures in future work. The base architecture of both VoxelMorph and LapIRN consists of rather simple blocks, compared to architectures commonly used in image segmentation and other related tasks; Adding these established blocks into registration networks could likely improve the performance.

Next to that, using multiple cascading networks of the same resolution could also be explored. Of course, besides DLIR methods, the combination of iterative and DL-based methods could also be explored further. For example, DLIR can be used for the initial registration that is followed by iterative finetuning. However, as this would lead to a compromise in terms of runtime, it is not preferred.

Lastly, the unsupervised training of the affine model and un- or weakly supervised training of LapIRN may also benefit from the bidirectional loss calculation used for VoxelMorph.

Evaluation It is recommended to find ways to compare the results from this paper to other publications. This could be done by training other models on this same dataset. However, that may not be desirable. Instead, the current models could be trained on a task for which benchmark datasets or competitions are available, e.g. for CT-CT inspiration-expiration lung registration (DIR-Lab), as results of various other works are available for these tasks. Alternatively, more extensive evaluation could be done using more similarity metrics, e.g. Hausdorff distance or absolute volume difference. This would also allow for easier comparison to other works.

Evaluation in a clinical setting would be useful. Evaluating models with experts will likely lead to more clinically relevant evaluation results, e.g. with regards to what misalignments are acceptable and which are problematic, if the deformable transformation is smooth enough, etc. Furthermore, testing the model performance when using the model as a part of a larger pipeline, e.g. to transfer dose maps (similar to Elmahdy, Jagt, Zinkstok, *et al.* [84]), would be meaningful.

References

[1] M. A. Viergever, J. A. Maintz, S. Klein, K. Murphy, M. Staring, and J. P. Pluim, "A survey of medical image registration–under review," *Medical image analysis*, vol. 33, pp. 140–144, 2016.

[2] G. Haskins, U. Kruger, and P. Yan, "Deep learning in medical image registration: A survey," *Machine Vision and Applications*, vol. 31, no. 1, pp. 1–18, 2020.

[3] H. R. Boveiri, R. Khayami, R. Javidan, and A. Mehdizadeh, "Medical image registration using deep neural networks: A comprehensive review," *Computers & Electrical Engineering*, vol. 87, p. 106 767, 2020.

[4] J.-G. Lee, S. Jun, Y.-W. Cho, H. Lee, G. B. Kim, J. B. Seo, and N. Kim, "Deep learning in medical imaging: General overview," *Korean journal of radiology*, vol. 18, no. 4, p. 570, 2017.

[5] Y. Fu, Y. Lei, T. Wang, W. J. Curran, T. Liu, and X. Yang, "Deep learning in medical image registration: A review," *Physics in Medicine & Biology*, vol. 65, no. 20, 20TR01, 2020.

[6] G. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio, F. Ciompi, M. Ghafoorian, J. A. Van Der Laak, B. Van Ginneken, and C. I. Sánchez, "A survey on deep learning in medical image analysis," *Medical image analysis*, vol. 42, pp. 60–88, 2017.

[7] N. J. Tustison, B. B. Avants, and J. C. Gee, "Learning image-based spatial transformations via convolutional neural networks: A review," *Magnetic resonance imaging*, vol. 64, pp. 142–153, 2019.

[8] D. L. Hill, P. G. Batchelor, M. Holden, and D. J. Hawkes, "Medical image registration," *Physics in medicine & biology*, vol. 46, no. 3, R1, 2001.

[9] A. Sotiras, C. Davatzikos, and N. Paragios, "Deformable medical image registration: A survey," *IEEE transactions on medical imaging*, vol. 32, no. 7, pp. 1153–1190, 2013.

[10] G. C. Pereira, M. Traughber, and R. F. Muzic, "The role of imaging in radiation therapy planning: Past, present, and future," *BioMed research international*, vol. 2014, 2014.

[11] M. Maspero, M. H. Savenije, T. C. van Heijst, J. J. Verhoeff, A. N. Kotte, A. C. Houweling, and C. A. van den Berg, "CBCT-to-CT synthesis with a single neural network for head-and-neck, lung and breast cancer adaptive radiotherapy," *arXiv preprint arXiv:1912.11136*, 2019.

[12] I. Maund, R. Benson, J. Fairfoul, J. Cook, R. Huddart, and A. Poynter, "Image-guided radiotherapy of the prostate using daily CBCT: The feasibility and likely benefit of implementing a margin reduction," *The British journal of radiology*, vol. 87, no. 1045, p. 20 140 459, 2015.

[13] K. K. Brock, S. Mutic, T. R. McNutt, H. Li, and M. L. Kessler, "Use of image registration and fusion algorithms and techniques in radiotherapy: Report of the AAPM Radiation Therapy Committee Task Group No. 132," *Medical physics*, vol. 44, no. 7, e43–e76, 2017.

[14] J. M. Fitzpatrick, D. L. Hill, C. R. Maurer, *et al.*, "Image registration," *Handbook of medical imaging*, vol. 2, pp. 447–513, 2000.

[15] E. Saiti and T. Theoharis, "An application independent review of multimodal 3D registration methods," *Computers & Graphics*, vol. 91, pp. 153–178, 2020.

[16] P. Cattin, "Basics of image registration," Slides, 2016.

[17] X. Ouyang, X. Liang, and Y. Xie, "Preliminary feasibility study of imaging registration between supine and prone breast CT in breast cancer radiotherapy using residual recursive cascaded networks," *IEEE Access*, 2020.

[18] K. A. Eppenhof, M. W. Lafarge, M. Veta, and J. P. Pluim, "Progressively trained convolutional neural networks for deformable image registration," *IEEE transactions on medical imaging*, vol. 39, no. 5, pp. 1594–1604, 2019.

[19] A. Wong and W. Bishop, "Efficient least squares fusion of MRI and CT images using a phase congruency model," *Pattern Recognition Letters*, vol. 29, no. 3, pp. 173–180, 2008.

[20] Y. Fu, Y. Lei, T. Wang, K. Higgins, J. D. Bradley, W. J. Curran, T. Liu, and X. Yang, "LungRegNet: An unsupervised deformable image registration method for 4D-CT lung," *Medical physics*, vol. 47, no. 4, pp. 1763–1774, 2020.

[21] G. Janssens, L. Jacques, J. Orban de Xivry, X. Geets, and B. Macq, "Diffeomorphic registration of images with variable contrast enhancement," *International journal of biomedical imaging*, vol. 2011, 2011.

[22] L. Mansilla, D. H. Milone, and E. Ferrante, "Learning deformable registration of medical images with anatomical constraints," *Neural Networks*, vol. 124, pp. 269–279, 2020.

[23] J. Ashburner, "A fast diffeomorphic image registration algorithm," *Neuroimage*, vol. 38, no. 1, pp. 95–113, 2007.

[24] X. Chen, A. Diaz-Pinto, N. Ravikumar, and A. Frangi, "Deep learning in medical image registration," *Progress in Biomedical Engineering*, 2020.

[25] V. Mani and S. Arivazhagan, "Survey of medical image registration," *Journal of Biomedical Engineering and Technology*, vol. 1, no. 2, pp. 8–25, 2013.

[26] T. C. Mok and A. C. Chung, "Large deformation diffeomorphic image registration with laplacian pyramid networks," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer, 2020, pp. 211–221.

[27] B. Zitova and J. Flusser, "Image registration methods: A survey," *Image and vision computing*, vol. 21, no. 11, pp. 977–1000, 2003.

[28] S. Klein, M. Staring, and J. P. Pluim, “Evaluation of optimization methods for nonrigid medical image registration using mutual information and B-splines,” *IEEE transactions on image processing*, vol. 16, no. 12, pp. 2879–2890, 2007.

[29] F. Maes, A. Collignon, D. Vandermeulen, G. Marchal, and P. Suetens, “Multimodality image registration by maximization of mutual information,” *IEEE transactions on Medical Imaging*, vol. 16, no. 2, pp. 187–198, 1997.

[30] A. Nazib, C. Fookes, and D. Perrin, “A comparative analysis of registration tools: Traditional vs deep learning approach on high resolution tissue cleared data,” *arXiv preprint arXiv:1810.08315*, 2018.

[31] D. Rueckert, L. I. Sonoda, C. Hayes, D. L. Hill, M. O. Leach, and D. J. Hawkes, “Nonrigid registration using free-form deformations: Application to breast MR images,” *IEEE transactions on medical imaging*, vol. 18, no. 8, pp. 712–721, 1999.

[32] M. Modat, G. R. Ridgway, Z. A. Taylor, M. Lehmann, J. Barnes, D. J. Hawkes, N. C. Fox, and S. Ourselin, “Fast free-form deformation using graphics processing units,” *Computer methods and programs in biomedicine*, vol. 98, no. 3, pp. 278–284, 2010.

[33] C. K. Guo, “Multi-modal image registration with unsupervised deep learning,” Ph.D. dissertation, Massachusetts Institute of Technology, 2019.

[34] S. Klein, M. Staring, K. Murphy, M. A. Viergever, and J. P. Pluim, “Elastix: A toolbox for intensity-based medical image registration,” *IEEE transactions on medical imaging*, vol. 29, no. 1, pp. 196–205, 2009.

[35] M. Staring, S. Klein, J. H. Reiber, W. J. Niessen, and B. C. Stoel, “Pulmonary image registration with elastix using a standard intensity-based algorithm,” *Medical Image Analysis for the Clinic: A Grand Challenge*, pp. 73–79, 2010.

[36] D. Mahapatra and Z. Ge, “Training data independent image registration with GANs using transfer learning and segmentation information,” in *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*, IEEE, 2019, pp. 709–713.

[37] Z. Jiang, F.-F. Yin, Y. Ge, and L. Ren, “A multi-scale framework with unsupervised joint training of convolutional neural networks for pulmonary deformable image registration,” *Physics in Medicine & Biology*, vol. 65, no. 1, p. 015 011, 2020.

[38] B. B. Avants, N. Tustison, and G. Song, “Advanced Normalization Tools (ANTS),” *Insight j*, vol. 2, no. 365, pp. 1–35, 2009.

[39] X. Cao, J. Yang, L. Wang, Z. Xue, Q. Wang, and D. Shen, “Deep learning based inter-modality image registration supervised by intra-modality similarity,” in *International workshop on machine learning in medical imaging*, Springer, 2018, pp. 55–63.

[40] J. Krebs, H. Delingette, B. Mailhé, N. Ayache, and T. Mansi, "Learning a probabilistic model for diffeomorphic registration," *IEEE transactions on medical imaging*, vol. 38, no. 9, pp. 2165–2176, 2019.

[41] N. Mogadas, T. Sothmann, T. Knopp, T. Gauer, C. Petersen, and R. Werner, "Influence of deformable image registration on 4d dose simulation for extracranial SBRT: A multi-registration framework study," *Radiotherapy and Oncology*, vol. 127, no. 2, pp. 225–232, 2018.

[42] F. Altaf, S. M. Islam, N. Akhtar, and N. K. Janjua, "Going deep in medical image analysis: Concepts, methods, challenges, and future directions," *IEEE Access*, vol. 7, pp. 99 540–99 572, 2019.

[43] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, pp. 1097–1105, 2012.

[44] S. Sun, J. Pang, J. Shi, S. Yi, and W. Ouyang, "Fishnet: A versatile backbone for image, region, and pixel level prediction," *arXiv preprint arXiv:1901.03495*, 2019.

[45] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[46] A. Veit, M. Wilber, and S. Belongie, "Residual networks behave like ensembles of relatively shallow networks," *arXiv preprint arXiv:1605.06431*, 2016.

[47] C. Zhang, P. Benz, D. M. Argaw, S. Lee, J. Kim, F. Rameau, J.-C. Bazin, and I. S. Kweon, "ResNet or DenseNet? introducing dense shortcuts to ResNet," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2021, pp. 3550–3559.

[48] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[49] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*, Springer, 2015, pp. 234–241.

[50] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu, "Spatial transformer networks," *arXiv preprint arXiv:1506.02025*, 2015.

[51] T. Sentker, F. Madesta, and R. Werner, "GDL-FIRE$^{\text{4D}}$: Deep learning-based fast 4d ct image registration," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer, 2018, pp. 765–773.

[52] Y. Sun, A. Moelker, W. J. Niessen, and T. van Walsum, "Towards robust CT-ultrasound registration using deep learning methods," in *Understanding and Interpreting Machine Learning in Medical Image Computing Applications*, Springer, 2018, pp. 43–51.

[53] M. Hoffmann, B. Billot, J. E. Iglesias, B. Fischl, and A. V. Dalca, "Learning image registration without images," *arXiv preprint arXiv:2004.10282*, 2020.

[54] A. Kori and G. Krishnamurthi, "Zero shot learning for multi-modal real time image registration," *arXiv preprint arXiv:1908.06213*, 2019.

[55] G. Balakrishnan, A. Zhao, M. Sabuncu, J. Guttag, and A. V. Dalca, "An unsupervised learning model for deformable medical image registration," 2018, pp. 9252–9260.

[56] C. Tanner, F. Ozdemir, R. Profanter, V. Vishnevsky, E. Konukoglu, and O. Goksel, "Generative adversarial networks for mr-ct deformable image registration," *arXiv preprint arXiv:1807.07349*, 2018.

[57] B. D. De Vos, F. F. Berendsen, M. A. Viergever, H. Sokooti, M. Staring, and I. Išgum, "A deep learning framework for unsupervised affine and deformable image registration," *Medical image analysis*, vol. 52, pp. 128–143, 2019.

[58] Y. Hu, M. Modat, E. Gibson, N. Ghavami, E. Bonmati, C. M. Moore, M. Emberton, J. A. Noble, D. C. Barratt, and T. Vercauteren, "Label-driven weakly-supervised learning for multimodal deformable image registration," in *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)*, IEEE, 2018, pp. 1070–1074.

[59] S. Zhao, Y. Dong, E. I. Chang, Y. Xu, *et al.*, "Recursive cascaded networks for unsupervised medical image registration," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 10 600–10 610.

[60] S. Zhao, T. Lau, J. Luo, I. Eric, C. Chang, and Y. Xu, "Unsupervised 3d end-to-end medical image registration with volume tweening network," *IEEE journal of biomedical and health informatics*, vol. 24, no. 5, pp. 1394–1404, 2019.

[61] H. Qiu, C. Qin, A. Schuh, K. Hammernik, and D. Rueckert, "Learning diffeomorphic and modality-invariant registration using B-splines," in *Medical Imaging with Deep Learning*, 2021.

[62] A. V. Dalca, G. Balakrishnan, J. Guttag, and M. Sabuncu, "Unsupervised learning of probabilistic diffeomorphic registration for images and surfaces," *Medical Image Analysis*, vol. 57, pp. 226–236, 2019.

[63] J. Krebs, T. Mansi, B. Mailhé, N. Ayache, and H. Delingette, "Unsupervised probabilistic deformation modeling for robust diffeomorphic registration," in *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, Springer, 2018, pp. 101–109.

[64] V. Arsigny, O. Commowick, X. Pennec, and N. Ayache, "A log-euclidean framework for statistics on diffeomorphisms," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer, 2006, pp. 924–931.

[65] A. V. Dalca, G. Balakrishnan, J. Guttag, and M. R. Sabuncu, "Unsupervised learning for fast probabilistic diffeomorphic registration," in *International*

Conference on Medical Image Computing and Computer-Assisted Intervention, Springer, 2018, pp. 729–738.

[66] B. Kim, J. Kim, J.-G. Lee, D. H. Kim, S. H. Park, and J. C. Ye, "Unsupervised deformable image registration using cycle-consistent CNN," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer, 2019, pp. 166–174.

[67] M. Blendowski and M. P. Heinrich, "Combining MRF-based deformable registration and deep binary 3D-CNN descriptors for large lung motion estimation in copd patients," *International journal of computer assisted radiology and surgery*, vol. 14, no. 1, pp. 43–52, 2019.

[68] K. A. Eppenhof, M. W. Lafarge, P. Moeskops, M. Veta, and J. P. Pluim, "Deformable image registration using convolutional neural networks," in *Medical Imaging 2018: Image Processing*, International Society for Optics and Photonics, vol. 10574, 2018, 105740S.

[69] M. D. Foote, B. E. Zimmerman, A. Sawant, and S. C. Joshi, "Real-time 2D-3D deformable registration with deep learning and application to lung radiotherapy targeting," in *International Conference on Information Processing in Medical Imaging*, Springer, 2019, pp. 265–276.

[70] J. O. Onieva, B. Marti-Fuster, M. P. de la Puente, and R. S. J. Estépar, "Diffeomorphic lung registration using deep CNNs and reinforced learning," in *Image Analysis for Moving Organ, Breast, and Thoracic Images*, Springer, 2018, pp. 284–294.

[71] L. Zhao and K. Jia, "Deep adaptive Log-Demons: Diffeomorphic image registration with very large deformations," *Computational and mathematical methods in medicine*, vol. 2015, 2015.

[72] E. Castillo, R. Castillo, J. Martinez, M. Shenoy, and T. Guerrero, "Four-dimensional deformable image registration using trajectory modeling," *Physics in Medicine & Biology*, vol. 55, no. 1, p. 305, 2009.

[73] R. Castillo, E. Castillo, R. Guerra, V. E. Johnson, T. McPhail, A. K. Garg, and T. Guerrero, "A framework for evaluation of deformable image registration spatial accuracy using large landmark point sets," *Physics in Medicine & Biology*, vol. 54, no. 7, p. 1849, 2009.

[74] D. Toth, S. Miao, T. Kurzendorfer, C. A. Rinaldi, R. Liao, T. Mansi, K. Rhode, and P. Mountney, "3D/2D model-to-image registration by imitation learning for cardiac procedures," *International journal of computer assisted radiology and surgery*, vol. 13, no. 8, pp. 1141–1149, 2018.

[75] Z. Lu, G. Yang, T. Hua, L. Hu, Y. Kong, L. Tang, X. Zhu, J.-L. Dillenseger, H. Shu, and J.-L. Coatrieux, "Unsupervised three-dimensional image registration using a cycle convolutional neural network," in *2019 IEEE International Conference on Image Processing (ICIP)*, IEEE, 2019, pp. 2174–2178.

[76] H. Sokooti, B. De Vos, F. Berendsen, B. P. Lelieveldt, I. Išgum, and M. Staring, "Nonrigid image registration using multi-scale 3D convolutional neural networks," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer, 2017, pp. 232–239.

[77] M. P. Heinrich, "Closing the gap between deep and conventional image registration using probabilistic dense displacement networks," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer, 2019, pp. 50–58.

[78] B. Cabrera Gil, "Deep learning based deformable image registration of pelvic images [Master's thesis, KTH Royal Institute of Technology]," Ph.D. dissertation, 2020. [Online]. Available: http://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-279155.

[79] R. Liao, S. Miao, P. de Tournemire, S. Grbic, A. Kamen, T. Mansi, and D. Comaniciu, "An artificial agent for robust image registration," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 31, 2017.

[80] K. A. Eppenhof and J. P. Pluim, "Pulmonary CT registration through supervised learning with convolutional neural networks," *IEEE transactions on medical imaging*, vol. 38, no. 5, pp. 1097–1105, 2018.

[81] T. Fechter and D. Baltas, "One-shot learning for deformable medical image registration and periodic motion tracking," *IEEE transactions on medical imaging*, vol. 39, no. 7, pp. 2506–2517, 2020.

[82] V. S. Deshpande and J. S. Bhatt, "Bayesian deep learning for deformable medical image registration," in *International Conference on Pattern Recognition and Machine Intelligence*, Springer, 2019, pp. 41–49.

[83] L. Hansen and M. P. Heinrich, "Tackling the problem of large deformations in deep learning based medical image registration using displacement embeddings," *arXiv preprint arXiv:2005.13338*, 2020.

[84] M. S. Elmahdy, T. Jagt, R. T. Zinkstok, Y. Qiao, R. Shahzad, H. Sokooti, S. Yousefi, L. Incrocci, C. Marijnen, M. Hoogeman, *et al.*, "Robust contour propagation using deep learning and image registration for online adaptive proton therapy of prostate cancer," *Medical physics*, vol. 46, no. 8, pp. 3329–3343, 2019.

[85] X. Cheng, L. Zhang, and Y. Zheng, "Deep similarity learning for multimodal medical images," *Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization*, vol. 6, no. 3, pp. 248–252, 2018.

[86] S. Sun, J. Hu, M. Yao, J. Hu, X. Yang, Q. Song, and X. Wu, "Robust multimodal image registration using deep recurrent reinforcement learning," in *Asian Conference on Computer Vision*, Springer, 2018, pp. 511–526.

[87] S. Miao, S. Piat, P. Fischer, A. Tuysuzoglu, P. Mewes, T. Mansi, and R. Liao, "Dilated FCN for multi-agent 2D/3D medical image registration," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[88] J. Zheng, S. Miao, Z. J. Wang, and R. Liao, "Pairwise domain adaptation module for CNN-based 2-D/3-D registration," *Journal of Medical Imaging*, vol. 5, no. 2, p. 021 204, 2018.

[89] M. Simonovsky, B. Gutiérrez-Becker, D. Mateus, N. Navab, and N. Komodakis, "A deep metric for multimodal registration," in *International conference on medical image computing and computer-assisted intervention*, Springer, 2016, pp. 10–18.

[90] Q. Liu and H. Leung, "Tensor-based descriptor for image registration via unsupervised network," in *2017 20th International Conference on Information Fusion (Fusion)*, IEEE, 2017, pp. 1–7.

[91] Y. Hu, M. Modat, E. Gibson, W. Li, N. Ghavami, E. Bonmati, G. Wang, S. Bandula, C. M. Moore, M. Emberton, *et al.*, "Weakly-supervised convolutional neural networks for multimodal image registration," *Medical image analysis*, vol. 49, pp. 1–13, 2018.

[92] S. Van Kranen, T. Kanehira, R. Rozendaal, and J. Sonke, "Unsupervised deep learning for fast and accurate CBCT to CT deformable image registration," *Radiotherapy and Oncology*, vol. 133, S267–S268, 2019.

[93] M. Blendowski, N. Bouteldja, and M. P. Heinrich, "Multimodal 3d medical image registration guided by shape encoder–decoder networks," *International journal of computer assisted radiology and surgery*, vol. 15, no. 2, pp. 269–276, 2020.

[94] A. Hering, S. Kuckertz, S. Heldmann, and M. P. Heinrich, "Memory-efficient 2.5 d convolutional transformer networks for multi-modal deformable registration with weak label supervision applied to whole-heart ct and mri scans," *International journal of computer assisted radiology and surgery*, vol. 14, no. 11, pp. 1901–1912, 2019.

[95] M. P. Heinrich and L. Hansen, "Highly accurate and memory efficient unsupervised learning-based discrete CT registration using 2.5D displacement search," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer, 2020, pp. 190–200.

[96] S. S. M. Salehi, S. Khan, D. Erdogmus, and A. Gholipour, "Real-time deep pose estimation with geodesic loss for image-to-template rigid registration," *IEEE transactions on medical imaging*, vol. 38, no. 2, pp. 470–481, 2018.

[97] H. Kataoka, T. Wakamiya, K. Hara, and Y. Satoh, "Would mega-scale datasets further enhance spatiotemporal 3D CNNs?" *arXiv preprint arXiv:2004.04968*, 2020.

[98] K. Hara, H. Kataoka, and Y. Satoh, "Can spatiotemporal 3D CNNs retrace the history of 2D CNNs and ImageNet?" In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 6546–6555.

[99] K. Ma, J. Wang, V. Singh, B. Tamersoy, Y.-J. Chang, A. Wimmer, and T. Chen, "Multimodal image registration with deep context reinforcement

learning," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer, 2017, pp. 240–248.

[100] G. Haskins, J. Kruecker, U. Kruger, S. Xu, P. A. Pinto, B. J. Wood, and P. Yan, "Learning deep similarity metric for 3D MR–TRUS image registration," *International journal of computer assisted radiology and surgery*, vol. 14, no. 3, pp. 417–425, 2019.

[101] E. Chee and Z. Wu, "AIRNet: Self-supervised affine registration for 3d medical images using neural networks," *arXiv preprint arXiv:1810.02583*, 2018.

[102] Y. Zhang, R. Liu, Z. Li, Z. Liu, X. Fan, and Z. Luo, "Coupling principled refinement with bi-directional deep estimation for robust deformable 3D medical image registration," in *2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI)*, IEEE, 2020, pp. 86–90.

[103] A. V. Dalca, E. Yu, P. Golland, B. Fischl, M. R. Sabuncu, and J. E. Iglesias, "Unsupervised deep learning for bayesian brain mri segmentation," in *MICCAI: Medical Image Computing and Computer Assisted Intervention, LNCS*, vol. 11766, 2019, pp. 356–365.

[104] A. Nazib, C. Fookes, and D. Perrin, "Dense deformation network for high resolution tissue cleared image registration," *arXiv preprint arXiv:1906.06180*, 2019.

[105] G. Balakrishnan, A. Zhao, M. R. Sabuncu, J. Guttag, and A. V. Dalca, "VoxelMorph: A learning framework for deformable medical image registration," *IEEE transactions on medical imaging*, vol. 38, no. 8, pp. 1788–1800, 2019.

[106] Z. Xu, C. P. Lee, M. P. Heinrich, M. Modat, D. Rueckert, S. Ourselin, R. G. Abramson, and B. A. Landman, "Evaluation of six registration methods for the human abdomen on clinically acquired ct," *IEEE Transactions on Biomedical Engineering*, vol. 63, no. 8, pp. 1563–1572, 2016.

[107] A. L. Simpson, M. Antonelli, S. Bakas, M. Bilello, K. Farahani, B. Van Ginneken, A. Kopp-Schneider, B. A. Landman, G. Litjens, B. Menze, *et al.*, "A large annotated medical image dataset for the development and evaluation of segmentation algorithms," *arXiv preprint arXiv:1902.09063*, 2019.

[108] W. R. Crum, T. Hartkens, and D. Hill, "Non-rigid image registration: Theory and practice," *The British journal of radiology*, vol. 77, no. suppl_2, S140–S153, 2004.

[109] Y. R. Rao, N. Prathapani, and E. Nagabhooshanam, "Application of normalized cross correlation to image registration," *International Journal of Research in Engineering and Technology*, vol. 3, no. 5, pp. 12–16, 2014.

[110] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *Journal of Big Data*, vol. 6, no. 1, pp. 1–48, 2019.

[111] S. Nikolov, S. Blackwell, A. Zverovitch, R. Mendes, M. Livne, J. De Fauw, Y. Patel, C. Meyer, H. Askham, B. Romera-Paredes, *et al.*, "Deep learning

to achieve clinically applicable segmentation of head and neck anatomy for radiotherapy," *arXiv preprint arXiv:1809.04430*, 2018.

[112] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "PyTorch: An imperative style, high-performance deep learning library," *Advances in neural information processing systems*, vol. 32, pp. 8024–8035, 2019.

[113] MONAI Consortium, *MONAI: Medical Open Network for AI*, Mar. 2020. DOI: 10.5281/zenodo.4323058. [Online]. Available: https://github.com/Project-MONAI/MONAI.

[114] F. Pérez-García, R. Sparks, and S. Ourselin, "TorchIO: A Python library for efficient loading, preprocessing, augmentation and patch-based sampling of medical images in deep learning," *Computer Methods and Programs in Biomedicine*, p. 106 236, 2021, ISSN: 0169-2607. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0169260721003102.

[115] K. Marstal, F. Berendsen, M. Staring, and S. Klein, "SimpleElastix: A user-friendly, multi-lingual library for medical image registration," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2016, pp. 134–142.

[116] N. J. Tustison, P. A. Cook, A. J. Holbrook, H. J. Johnson, J. Muschelli, G. A. Devenyi, J. T. Duda, S. R. Das, N. C. Cullen, D. L. Gillen, *et al.*, "The ANTsX ecosystem for quantitative biological and medical imaging," *Scientific Reports*, vol. 11, no. 1, pp. 1–13, 2021.

[117] J. M. Sloan, K. A. Goatman, and J. P. Siebert, "Learning rigid image registration-utilizing convolutional neural networks for medical image registration," 2018.

[118] J.-P. Thirion, "Image matching as a diffusion process: An analogy with Maxwell's demons," *Medical image analysis*, vol. 2, no. 3, pp. 243–260, 1998.

[119] E. Garyfallidis, M. Brett, B. Amirbekian, A. Rokem, S. Van Der Walt, M. Descoteaux, and I. Nimmo-Smith, "Dipy, a library for the analysis of diffusion mri data," *Frontiers in neuroinformatics*, vol. 8, p. 8, 2014.

[120] J. D. Hunter, "Matplotlib: A 2D graphics environment," *Computing in Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007. DOI: 10.1109/MCSE.2007.55.

[121] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *arXiv preprint arXiv:1406.2661*, 2014.

[122] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1125–1134.

A Appendix: Implementation details

A.1 Synthetic deformation

The synthetic data augmentations were applied using the MONAI framework [113] in Python, using the random affine transform, and a custom version of the random elastic transform.

Affine deformation A random affine transformation was applied to all scans for training the affine model, and with a probability of 0.50 for the deformable model. The settings are shown in Table A1. Shearing was not used.

Parameter	z-direction	x- and y-directions
Translation (fraction of the scan size)	$U(-0.20, 0.20)/U(-0.30, 0.30)$	$U(-0.10, 0.10)/U(-0.30, 0.30)$
Rotation (degrees)	$U(-10, 10)$	$U(-15, 15)/U(-5, 5)$
Scaling (factor)	$U(0.70, 1.30)$	$U(0.75, 1.25)$

Table A1: Parameters for the random affine transformation applied for the deformable model, with a probability of 0.5. The settings for the affine model, with a probability of 1.0, are shown after the slash if different.

Elastic deformation For the random 3D elastic transform, MONAI first applies random offsets on every voxel or grid point. These are then smoothed with a Gaussian kernel. The random offsets were sampled from $U(0, 5000)$, with the standard deviation of the Gaussian kernel sampled from $U(15, 25)$ voxels. For training the affine model, smaller deformations were used: Random offsets sampled from $U(100, 150)$ with the standard deviation of the smoothing kernel sampled from $U(4, 7)$.

A.2 Additional data augmentation

Next to synthetic deformations, various other augmentations were used: Flipping, applying an affine transformation to both the moving and fixed image, switching the moving and fixed image, and gamma augmentation, noise, and blurring. The first two are applied to both images simultaneously; The rest is applied to the individual images separately. The latter three were implemented using TorchIO [114].

The settings can be found in Table A2. For the affine model, the noise, gamma, and blurring augmentations were not always used.

A.3 Baseline (ANTs and Elastix) settings

Both ANTs and Elastix have various parameters that can be changed to improve the performance, such as the similarity metric used or the number of iterations. As

Augmentation and operation	Value	Probability
Gamma, $\gamma = \exp(\beta)$	$\beta \sim U(-0.5, 0.5)$	0.5
Gaussian blurring with std. $\sigma_x, \sigma_y, \sigma_z$	$\sigma_z \sim U(-0.5, 0.5)$, $\sigma_x, \sigma_y \sim U(-1.25, 1.25)$	0.5
Gaussian noise with mean μ and std. σ	$\mu = 0, \sigma \sim U(0, 0.03)$	0.2
Flipping	-	0.2
Switching moving and fixed scans	-	0.5
Affine transformation (for both scans)	degrees $\sim U(-10, 10)$, scaling: $\sim U(0.7, 1.3)$	0.5

Table A2: Parameters for the additional data augmentations performed on the training scans.

iterative methods can require some tuning to achieve good performance, various settings were tested for both tools.

For Elastix, parameters are mainly taken from the Elastix Model Zoo[22], which contains parameters from previous experiments. The settings that are changed are the type of registration (affine, deformable using B-Splines, or both) number of resolutions in the image pyramid (4 (default) or 1), the similarity metric (MI (default) or CC), adding a regularization term (BE regularization or no regularization (default)) the number of iterations per pyramid level (256 (default) or 1000), and the grid spacing downsampling factors for the different pyramid levels (the default for 4 levels $8 \times 8 \times 8$, $4 \times 4 \times 4$, $2 \times 2 \times 2$, $1 \times 1 \times 1$; In some experiments, this is replaced with $4 \times 8 \times 8$, $2 \times 4 \times 4$, $1 \times 2 \times 2$, $1 \times 1 \times 1$, inspired by Staring, Klein, Reiber, *et al.* [35], as the scans have a smaller resolution in the z-direction).

For ANTs, various default settings can be tried, and parameters can be tweaked as well. Just like for Elastix, for ANTs the type of registration was changed (affine only or translation/affine followed by deformable using B-Splines), the number of iterations (default is (41, 21 and 1 for the three default pyramid levels; I changed this to 100 for each level, copying Nazib, Fookes, and Perrin [30]); the similarity metric (MI (default), CC, or Demons similarity[23]) the gradient step size (0.2 (default) or 0.25 (the setting for the "SyNAggro" default)).

A.4 Visualization

Plotting scans was done using code based on DIPY [119]. The overlays visible in the plots showing examples of scans were created as a color image of two grayscale slices plotted on top of each other using the red channel for the first volume and the green channel for the second one. Visualizing examples of DVFs in the model diagrams was done using Matplotlib [120].

[22] https://elastix.lumc.nl/modelzoo/

[23] Demons similarity refers to the similarity calculation used in the Demons algorithm presented by Thirion [118] as implemented in ITK.

B Appendix: Additional results

B.1 Affine models

B.1.1 sDSC plot

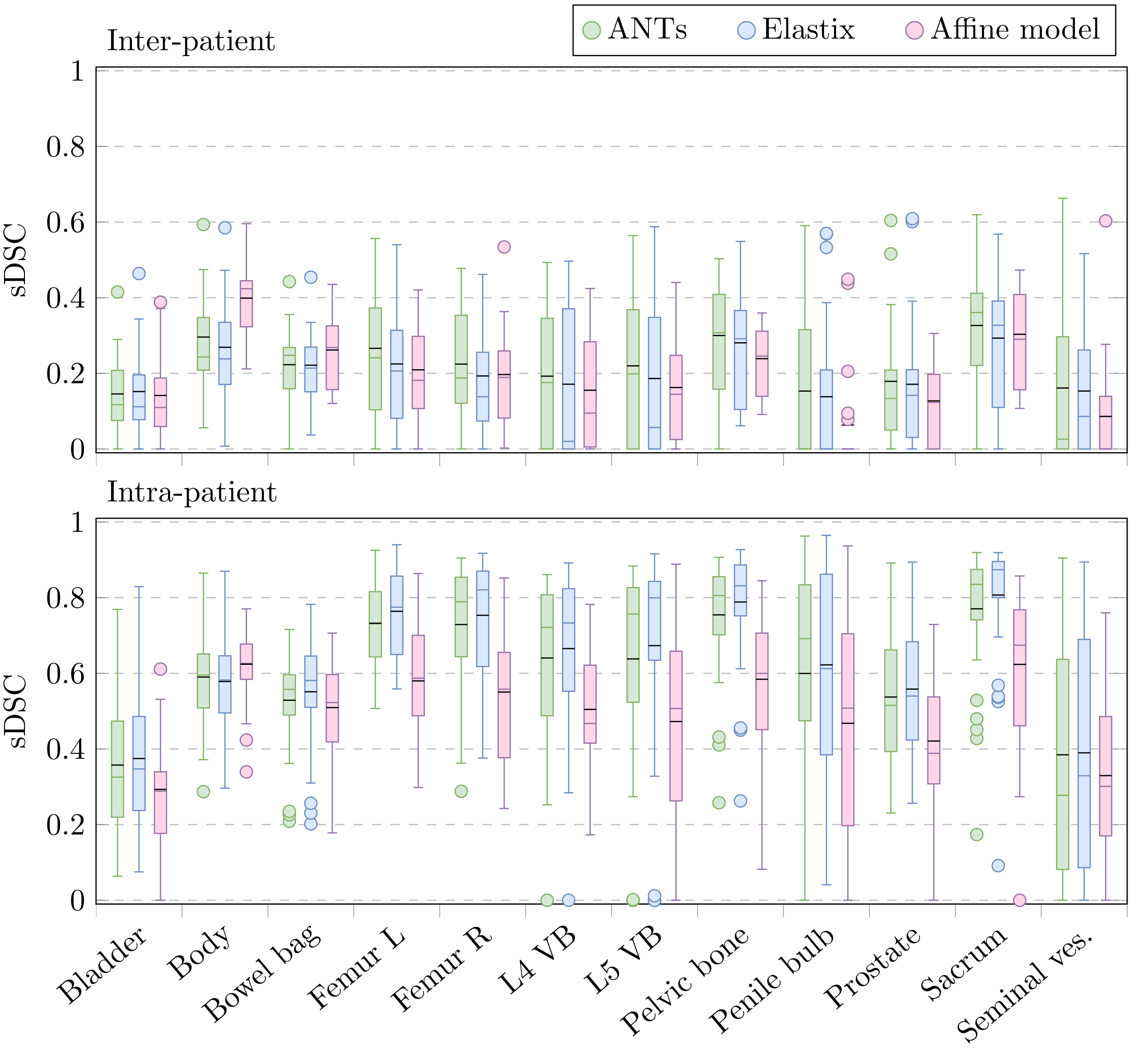


Figure B1: Boxplots of the sDSC scores achieved by the best affine model compared to ANTs and Elastix on the two test sets. The mean scores are shown in black.

B.1.2 Results on the validation set

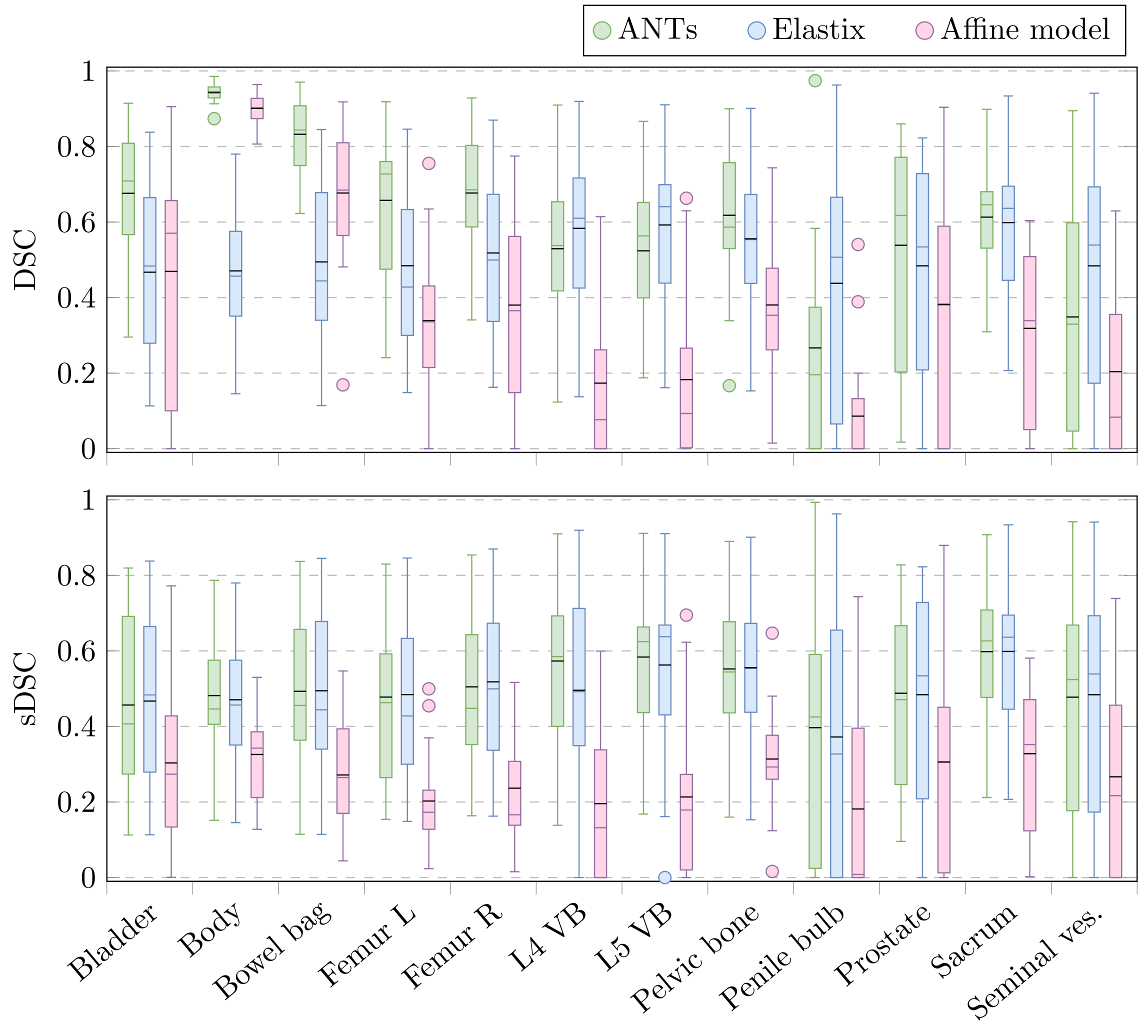


Figure B2: Boxplots of the DSC and sDSC scores achieved by the best affine model compared to ANTs and Elastix on the synthetic validation set. The mean scores are shown in black.

Model	DSC	sDSC	Runtime (s)
3D ResNet	0.38 (0.30)	0.26 (0.19)	**2.41e-3 (1.96e-4)**
ANTs	**0.61 (0.26)**	**0.51 (0.21)**	0.51 (0.13)
Elastix	0.59 (0.28)	0.50 (0.23)	1.24 (0.18)
No reg.	0.12 (0.22)	0.06 (0.06)	-

Table B1: Means and standard deviations of DSC and sDSC scores and the runtimes of the registrations of the best affine model on the synthetic validation set, compared to the affine registration models of ANTs and Elastix.

B.2 Deformable models

B.2.1 sDSC plot

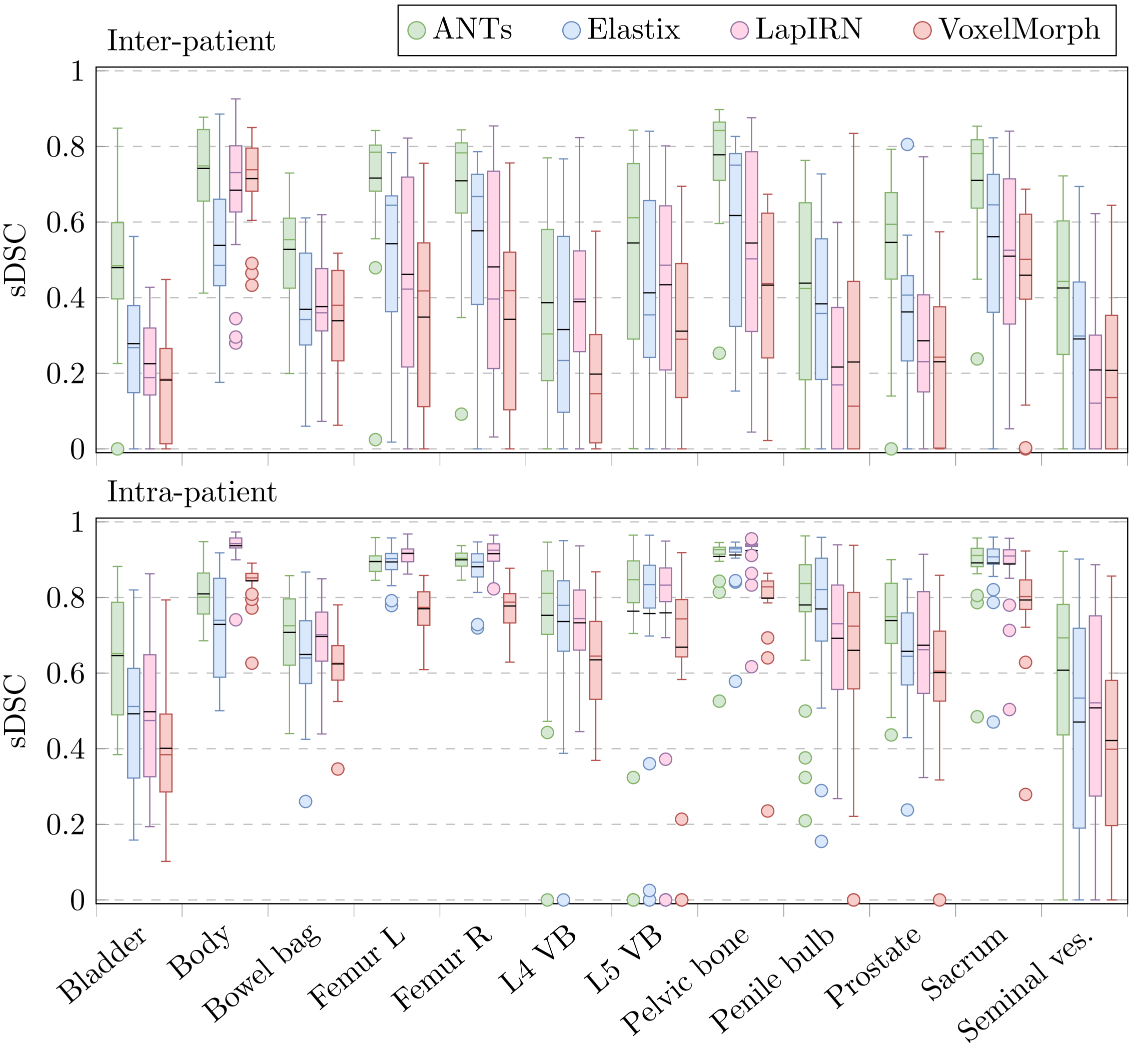


Figure B3: Boxplots of the sDSC scores achieved by the best deformable models compared to ANTs and Elastix on the two test sets. The mean scores are shown in black.

B.2.2 Results on the validation set

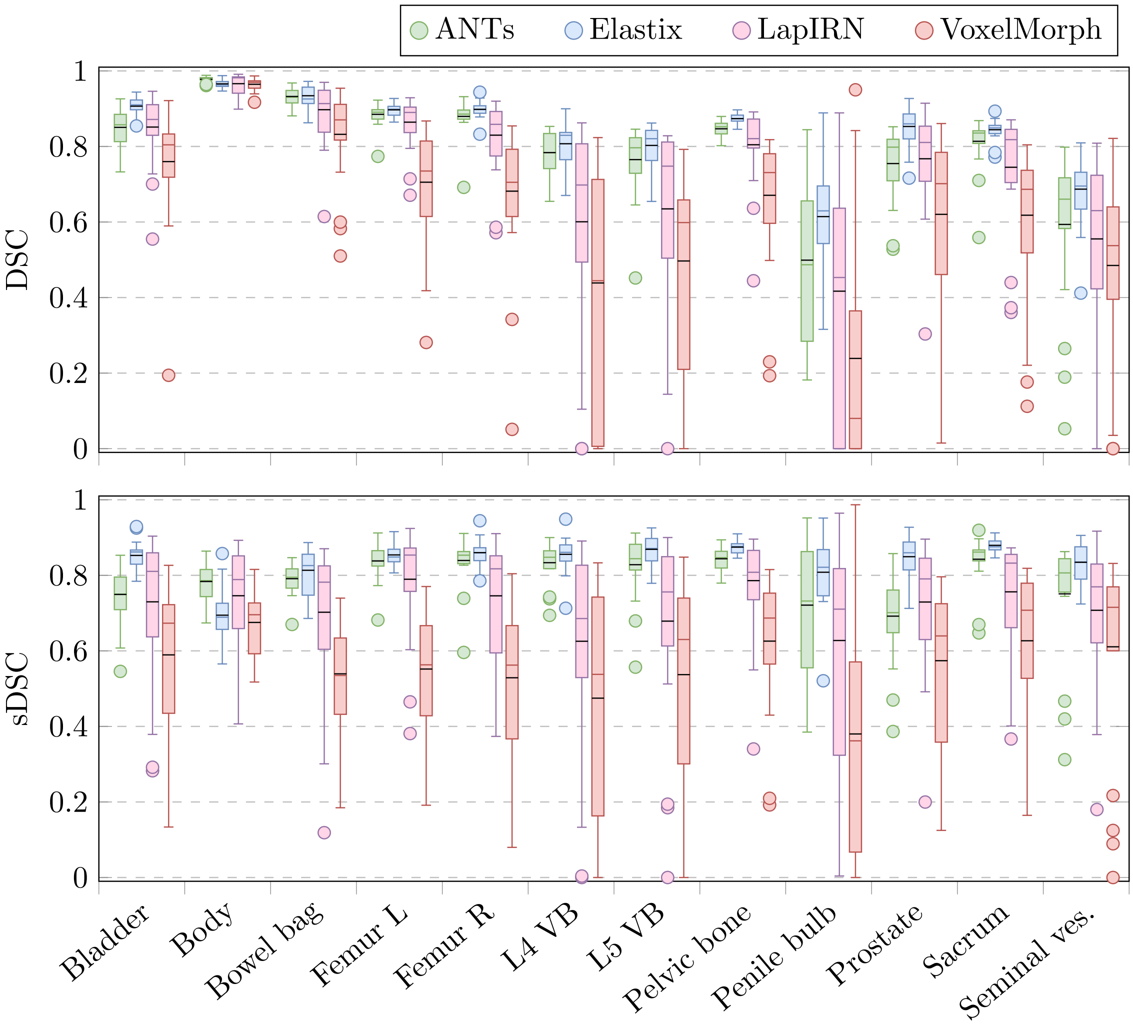


Figure B4: Boxplots of the DSC and sDSC scores achieved by the best deformable models compared to ANTs and Elastix on the synthetic validation set. The mean scores are shown in black.

Model	DSC	sDSC	JC	Runtime (s)
VoxelMorph	0.63 (0.28)	0.56 (0.23)	0.08 (0.14)	**0.19 (0.31)**
LapIRN	**0.75 (0.23)**	**0.72 (0.20)**	0.82 (0.44)	2.49 (5.04)
ANTs	0.63 (0.29)	0.58 (0.23)	**0.00 (0.00)**	49.90 (20.40)
Elastix	0.53 (0.31)	0.43 (0.24)	2.85 (3.88)	9.39 (0.95)
No reg.	0.14 (0.22)	0.06 (0.08)	-	-

Table B2: Means and standard deviations of DSC, sDSC, and JC scores, and the runtimes of the registrations on the synthetic validation set using the best VoxelMorph and LapIRN models, compared to the deformable registration models of ANTs and Elastix.

B.3 Baseline models experiments

B.3.1 ANTs

Model	Synthetic		Inter-patient		Intra-patient	
	DSC	sDSC	DSC	sDSC	DSC	sDSC
Affine	0.61	0.51	0.34	0.22	0.70	0.60
SyN (default; Affine + SyN)	0.79	0.76	0.51	0.42	0.76	0.70
SyN, max iter. 100	0.79	0.78	0.54	0.48	0.78	0.74
SyN, max iter. 100, Demons	**0.80**	**0.79**	**0.63**	**0.58**	**0.80**	**0.78**
SyNOnly	0.66	0.62	0.41	0.32	0.74	0.68
SyNOnly, max iter. 100, Demons	0.76	0.75	0.61	0.55	0.80	0.77
SyNAggro	0.78	0.76	0.55	0.46	0.76	0.71
SyNCC	0.79	0.78	0.55	0.49	0.77	0.72
SyNCC, max iter. 100	0.79	0.78	0.55	0.48	0.76	0.72

Table B3: Mean values across all ROIs and scans for different models that were tested with ANTs. The names (before the comma) refer to the model names in ANTs; after the comma, settings that were changed are denoted. The highest values achieved on the three sets and the best overall model are bolded.

B.3.2 Elastix

Model	Synthetic		Inter-patient		Intra-patient	
	DSC	sDSC	DSC	sDSC	DSC	sDSC
Affine	0.59	0.50	0.32	0.21	0.71	0.63
Affine, 1 res	0.55	0.46	0.27	0.16	0.71	0.62
Affine + def.	**0.82**	**0.81**	0.54	0.43	0.78	0.73
Affine + def., 1 res.	0.79	0.78	0.47	0.36	0.78	0.73
Affine + def., 3 res.	**0.82**	**0.81**	0.53	0.43	0.78	0.73
Affine + def., 1000 iter.	**0.82**	**0.81**	0.53	0.44	0.78	0.73
Affine + def., 1 res, 1000 iter.	0.81	0.80	0.49	0.39	**0.79**	**0.74**
Affine + def., different pyramid	0.80	0.78	0.46	0.34	0.75	0.69
Affine + def., CC	0.77	0.76	0.51	0.43	0.74	0.70
Def.	0.81	0.80	**0.57**	**0.47**	0.78	0.73
Def., different pyramid	0.81	0.80	**0.57**	0.47	0.78	0.73

Table B4: Mean values across all ROIs and scans for different affine and deformable models that were tested with Elastix. The model part after the first comma indicates changed settings (number of iterations, resolutions in the image pyramid, or the "different pyramid" as described in Appendix A). The highest values achieved on the three datasets and the best overall model are bolded.

C Appendix: Other DLIR methods

While the focus of this thesis are (un)supervised methods, other methods have also started to gain more prominence for image registration. For completeness, two of these upcoming categories, RL and GANs, are shortly discussed in this appendix.

Deep reinforcement learning Deep RL is a paradigm where agents learn behavior in an attempt to maximize their reward. In DLIR, this reward is the similarity between the moving and fixed image. While for the previously discussed methods learning was done by backpropagating a loss, which gives direction to the changes of the weights, in Deep RL the agent is generally more free to change its behavior (in this case: setting the transformation parameters) in an attempt to reach a higher reward. For deformable registration, there are many parameters for the model to predict (i.e. transformation parameters in all the grid points). Because this makes for a very large search space, Deep RL is typically used for rigid transformations, where there are only few parameters to train [2]. For deformable transformations, low-resolution deformation maps can be used to restrict the search space. Haskins, Kruger, and Yan [2] expect that, as DRL is only a relatively new field, it may be able to overcome the hurdle of the large search space associated with high-resolution deformations in the future, and receive more attention within registration research.

Generative adversarial networks A GAN, originally proposed in [121], is a type of generative network. Typically, it consists of a generator, a network trained to generate artificial data, and a discriminator, a network trained to discriminate the artificial data from real data. The two networks are trained competitively, with the discriminator encouraging more realistic data generation from the generator, and the discrimination task becoming harder as the generated data becomes more realistic, until an equilibrium is reached.

GANs can be used in training in two auxiliary ways [5]. Firstly, using a model that outputs transformation or a transformed image as a generator, the discriminator network can be used to provide additional regularization of the predicted transformation. If the output is a transformation/DVF, this can be done by testing if the parameters are/DVF is realistic (e.g. that there are no folds or irregularities that would not occur in ground truth DVFs); If the transformed image is outputted, the adversarial regularization can be implemented by testing if the transformed image can be distinguished from a real image, or if the transformed image is well-aligned with the fixed image. Secondly, GANs are popularly used for image translation [122], which can aid multi-modal image registration tasks by converting the images to only one modality[24].

Although GANs are currently only used in a fraction of DLIR literature, their usage is growing rapidly [5].

[24] Note how, similarly, multi-modal registration models can be used to provide ground-truth data for the training of image translation models.