UNIVERSITÄT ZU LÜBECK
INSTITUT FÜR MEDIZINISCHE INFORMATIK

MRT-Registration

MRT-Registration

Masterarbeit

verfasst am
Institut für Medizinische Informatik

im Rahmen des Studiengangs
Medizinische Informatik
der Universität zu Lübeck

vorgelegt von
Jan Meyer

ausgegeben und betreut von
Prof. Dr. Mattias Heinrich

mit Unterstützung von
Ziad Al-Haj Hemidi, Eytan Kats

Lübeck, den 1. Januar 2025

Eidesstattliche Erklärung

Ich erkläre hiermit an Eides statt, dass ich diese Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

_____

Jan Meyer

Zusammenfassung

Irgendwas über MRT und Registration

Abstract

Something about MRI and registration

Acknowledgements

Danke an Mattias, Ziad und Eytan für die gute Betreuung.

Contents

1

Introduction

Introduction of stuff.

1.1 Contributions of the Thesis

We implemented stuff.

1.2 Related Work

There are many papers on image registration in general, however in the context of medical image registration with deep learning their number is reduced due to the specialized nature of the subject at hand. Yet, there are a couple of papers that give a good overview of the topic. They give a brief overview of registration methods, basics of deep learning with already existing networks for image registration as well as covering potential applications and challenges [1, 2, 3, 4, 5]. A lot of different approaches for medical image registration are based on *VoxelMorph* [6], such as both *Fourier-Net* [7] and its successor *Fourier-Net+* [8].

1.3 Structure of the Thesis

This Thesis contains a lot of stuff in different chapters.

2

Basics

In this chapter the basics of the thesis are explained.

2.1 Magnetic Particle Imaging

Here MRI is described.

2.2 Image Registration

Image registration is a challenging, yet important task for image processing. It can be described as the process of transforming different image datasets into one coordinate system with matched imaging contents [2]. In the medical field this can be used for clinical applications such as disease diagnosis and monitoring, image-guided treatment delivery, and post-operative assessment. Medical image registration is typically used to preprocess data for tasks like object detection (for e.g. tumor growth monitoring) and segmentation (for e.g. organ atlas creation) where variation in spatial resolution is common between modalities like CT and MRI and patients. Thus the performance of these methods is dependent on the quality of image registration [1].
Medical image registration was often done manually by clinicians, however, registration tasks are often challenging and the quality of manual alignments is dependent on the expertise of the user. These manual registrations are thus not only time consuming, but also hardly reproducible leading to high interobserver-variability. The need for automatic registration is very much apparent, but this task remained hard to solve for a long time, requiring a lot of computational power and time for computer algorithms to solve the problem. While neural networks also require a lot of computational power and time to train, they promise fast execution after training. With the rise of deep learning these network gained popularity and now pose a real alternative to conventional algorithms and manual registration [2]. We will discuss these new approaches in the next section, but first we need to formally define our problem.
In pair-wise image registration two images (F and M) are to be aligned, with F denoting the fixed and M the moving image. T is the desired spatial transformation that aligns the

two images. This can be posed as an optimization problem:

$$T' = \arg\max S(F, T(M)), \tag{2.1}$$

with T' being the best transformation that maximizes the similarity S between the two images. This process is done iteratively improving estimates for the desired T, such that the defined similarity in the cost function is maximized [1].
Transformations can be categorized as rigid, affine, and deformable. A rigid transformation consists of rotation and translation; an affine transformation includes translations, rotations, scaling, and sheering; the two kinds of transformations are described as a 2D single matrix. Unlike rigid and affine transformation, deformable transformation is a high-dimension problem that we need to formulate by a 3D matrix for 2D deformable registration i.e., a so-called deformation field. While rigid and affine registration algorithms have already achieved good performance in many applications, deformable registration is still a challenging task due to its intrinsic complexity, particularly when the deformation is large. However, these are also the transformations most likely encountered in clinical practice as it can be utilized to fuse information from different modalities such as MRI and CT [4]. Additionally, deformable image registration can also be utilized for various computer-assisted interventions like biopsy [9] and (MRI-guided) radiotherapy [10, 11].
Intuitively, deformable image registration is an ill-posed problem, making it fundamentally different from other computer vision tasks such as object localization, segmentation or classification. Given two images, deformable image registration aims to find a spatial transformation that warps the moving image to match the fixed image as closely as possible. However, there is no ground-truth available for the desired deformation field and without enforcing any constraints on the properties of the spatial transformation, the resulting cost function is ill-conditioned and highly non-convex. In order to address the latter and ensure tractability, all image registration algorithms regularize the estimated deformation field, based on some prior assumptions on the properties of the underlying unknown deformation [1].
Many methods have been proposed for medical image registration to deal with the complex challenges of this task. Popular conventional registration methods include optical flow [12], demons [13] and many more. However, most of these still lack accuracy and computation speed, which makes newer deep learning approaches all the more interesting [3].

2.3 Deep Learning Architectures

Neural networks, despite the theoretical concepts being around for decades, have seen a meteoric rise in popularity over the last few years as constraints on computational power have been alleviated. Especially deep neural networks, which are often summarized under the term deep learning (DL). Recent years have witnessed an almost exponential growth in the development and use of DL algorithms, sustained thus far by rapid improvements in computational hardware (e.g. GPUs). Consequently, clinical applications requiring image classification, segmentation, registration, or object detection/localization, have

witnessed significant improvements in algorithmic performance, in terms of accuracy and/or efficiency [1]. The following network architecture are widely used for different tasks including medical image registration.
Some basic stuff about network training, testing and different architectures that are relevant for the later Chapters.

Convolutional Neural Networks

Convolutional neural networks (CNNs) are a type of deep neural networks with regularized multilayer perceptron, which are mainly used for image processing. CNNs use convolution operations instead of general matrix multiplications in typical neural networks. These convolutional filters make CNNs very suitable for visual signal processing. Because of their excellent feature extraction ability, CNNs are some of the most successful models for image analysis. Different variants of CNN have been proposed and have achieved the-state-of-art performances in various image processing tasks. A typical CNN usually consists of multiple convolutional layers, max pooling layers, batch normalization layers, sometimes dropout layers, a sigmoid or softmax layer. In each convolutional layer, multiple channels of feature maps are extracted by sliding trainable convolutional kernels across the input feature maps. Hierarchical features with high-level abstraction are extracted using multiple convolutional layers. These feature maps usually go through multiple fully connected layer before reaching the final decision layer. Max pooling layers are often used to reduce the image sizes and to promote spatial invariance of the network. Batch normalization is used to reduce internal covariate shift among the training samples. Weight regularization and dropout layers are used to alleviate data overfitting [3]. The loss function is often defined as the difference between the predicted and the target output. CNNs are usually trained by minimizing the loss via gradient back propagation using optimization methods like Adam [14].

U-Net

The U-Net [15] architecture is an extension of the typical CNNs structure typically used for image segmentation, however it can also be used for image registration tasks. It adopts symmetrical contractive and expansive paths with skip connections between them. The encoding blocks on the left extract important features from the image using convolution layers and max pooling, which are then stored in the latent space in the middle. From there it is reconstructed using upsampling and convolutions in the decoding blocks on the right. Additionally, skip connections are used to improve the spatial resolution of the segmentation. This architecture allows effective feature learning from a small number of training datasets [3].

Autoencoders

An autoencoder (AE) is a type of CNNs that learns to reconstruct an image from its input without supervision. AEs usually consists of an encoder which extracts the input features, which are stored a low-dimensional latent state space, similar to a U-Net, and a de-

coder which restore the original input from the latent space. To prevent an AE from learning an identity function, regularized autoencoders were invented, which can be used for e.g. denoising AEs. Variational AEs (VAEs) are generative models that learn latent representation using a variational approach, which constrains the variability of the outputs. VAEs can been used for anomaly detection and image generation [3].

Generative Adversarial Networks

Generative adversarial networks (GANs) consist of two competing networks, a generator and a discriminator. The generator is trained to generate artificial data that approximate a target data distribution from a low-dimensional latent space similar to an AE. The discriminator is trained to distinguish the artificial data from actual data. The discriminator encourages the generator to predict realistic data by penalizing unrealistic predictions via learning. Therefore, the discriminative loss could be considered as a dynamic network-based loss term. The generator and discriminator both are getting better during training to reach Nash equilibrium. In medical imaging, GANs have been used to perform image synthesis for inter- or intra-modality, such as MRI to synthetic CT and vise versa. In medical image registration, GANs are usually used to either provide additional regularization or translate multi-modal registration to uni-modal registration [3].

2.4 Deep Learning for Image Registration

Recently, there has been a surge in the use of deep learning based approaches for medical image registration. Their success is largely due to their ability to perform fast inference, and the flexibility to leverage auxiliary information such as anatomical masks as part of the training process. The most effective methods, such as *VoxelMorph* [6], typically employ a U-Net style architecture to estimate dense spatial deformation fields. These methods require only one forward pass during inference, making them orders of magnitude faster than traditional iterative methods. Following the success of *VoxelMorph*, numerous deep neural networks have been proposed for various registration tasks [8]. Other approaches also utilize CNNs, AEs and GANs. Typical strategies are discussed in more detail in the following sections.

Supervised Registration

Supervised registration describes training a network with a ground truth displacement field that is either real (created by hand) or synthetic (generated via traditional iterative registration algorithms). Thus the loss can easily be calculated as the difference in the displacement fields of the network prediction and the ground truth. These methods have achieved notable results with real displacement fields as supervision. However, this approach is very limited by the size and the diversity of the dataset. As the displacement fields are often calculated by conventional algorithms their effectiveness might be limited for difficult problems with which the traditional algorithms struggle. Fully supervised methods are widely studied and have notable results, but the generation of real or

synthetic displacement fields is hard, and these displacements fields might be different from the real ground truth, which can impact the accuracy and efficiency of these kinds of methods [4]. Notable approaches include *BIRNet* [16].

Unsupervised Registration

As the preparation of the ground truth displacement field for supervised methods is inconvenient, limitations in generalizing results in different domains and various registration tasks are inevitable. Thus, unsupervised registration has a more convenient training process with paired images as inputs, but without a ground truth. Generally, unsupervised learning consists of similarity-based and GAN-based methods, where the loss function computes the similarity between the aligned images and the smoothness of the displacement field, rather than the difference to a ground truth [4]. Well known example are *IC-Net* [17], *VoxelMorph* [6], *TransMorph* [18] and *SYMNet* [19].

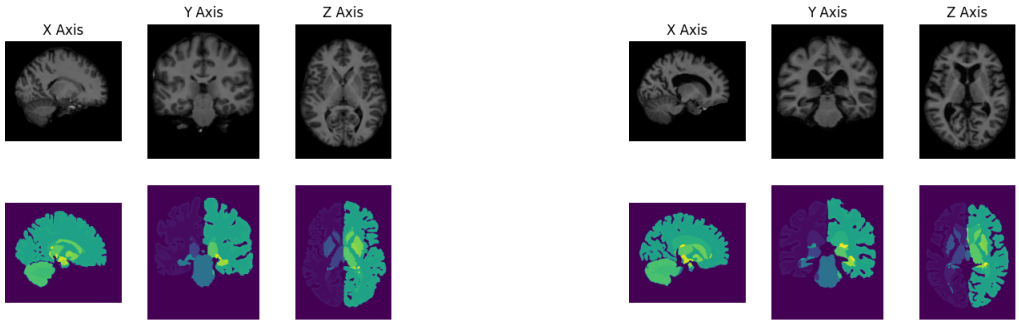
3

Data

Here the datasets are described.

3.1 OASIS

For training and testing the *OASIS-1* dataset [20] was mainly used, which contains T1-weighted MRI brain scans from 454 subjects. These were further pre-processed by [21] for the *Learn2Reg*-Challenge [22]. This enables subject-to-subject brain registration, as all MRI scans were bias-corrected, skull-stripped, aligned, and cropped to the size of $160 \times 192 \times 224$. Examples can be seen in Figures 3.1b and 3.1a with slices from the center of the x-, y- and z-axis.



(a) (b)

Figure 3.1: Example images (upper row) from the *OASIS* dataset with corresponding labels (bottom row).

4

Network Architectures

As a starting point *Fourier-Net* [7] and its successor *Fourier-Net+* [8] were used. These networks, which are explained in the following pages, enable fast and accurate registration while needing less resources compared to similar approaches. But we also did our own stuff...

4.1 Fourier-Net

Fourier-Net is a new unsupervised approach that aims to learn a low-dimensional representation of the displacement field in a band-limited Fourier domain instead of the full field in the spatial domain. This band-limited representation is then decoded by a model-driven decoder to the dense, full-resolution displacement field in the spatial domain. This allows for fewer parameters and computational operations, resulting in faster inference speeds [7]. The architecture is based on the U-Net [15], like most deep registration approaches, but replaces the expansive path with a parameter-free model-driven decoder as mentioned before. The encoder of *Fourier-Net* consists of a CNN and takes two images (fixed and moving) as inputs. The output is a displacement field that is then converted from the spatial domain into the Fourier domain via an discrete Fourier transformation (DFT) and band-limiting by low-pass filtering. From there, this band-limiting representation is padded with zeros to the full resolution of the original displacement field. The field is then recovered by using the inverse DFT (iDFT) to convert it back into the spatial domain. This displacement field is then used to warp the moving image into the fixed image. Additionally, squaring and scaling layers [23] can be added before warping the image in order to encourage a diffeomorphism in final deformation.

Encoder

The encoder of *Fourier-Net* consists of a CNN that generates the displacement field between the two input images. This network is the fully convolutional neural network (FCN) taken from the *SYMNet* [19]. It is very similar to the *U-Net* as seen in Figure 4.2.
The FCN concatenates the inputs images X and Y as a single 2-channel input and estimates two dense, non-linear displacement fields ϕ_{XY} and ϕ_{YX} from X and Y which com-

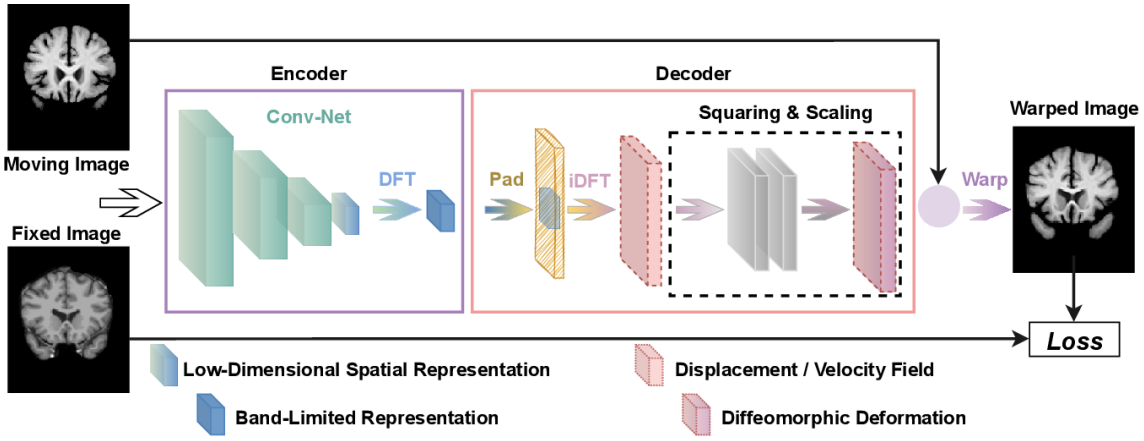


Figure 4.1: Architecture of *Fourier-Net* taken from [7].

bine to form $\mathbb{S}_\phi$. For each level in the encoding part of the FCN, two successive convolution layers are applied, which contain one $3 \times 3 \times 3$ convolution layer with a stride of 1, followed by a $3 \times 3 \times 3$ convolution layer with a stride of 2 to further compute the high-level features between the input images as well as downsample the features by half until the lowest level of the network is reached. For each level in the decoding part of the FCN, we concatenate the feature maps from the encoding part through skip connection and apply $3 \times 3 \times 3$ convolution with a stride of 1 and $2 \times 2 \times 2$ deconvolution layer for upsampling the feature maps to twice of its size. At the end of the decoding part, two $5 \times 5 \times 5$ convolution layers with a stride of 1 are appended to the last convolution layer and generate the displacement fields θ_{XY} and θ_{YX} Each convolution layer in the FCN is followed by a rectified linear unit (ReLU) activation, except for the output convolution layers, where a SoftSign activation function is used [19]:

$$\text{SoftSign}(x) = \frac{x}{1 + |x|}. \tag{4.1}$$

As discussed, the encoder aims to learn a displacement (or velocity field) in the band-limited Fourier domain. Intuitively, this may require convolutions to be able to handle complex-valued numbers, which can be done by using complex-valued CNNs [24], which are suitable when both input and output are complex values, however these complex-valued operations sacrifice computational efficiency. Other approaches like *DeepFlash* [25] tackle this problem by converting the input images to the Fourier domain and using two individual real-valued CNNs to learn the real and imaginary parts separately. This, however, increases training and inference cost. To bridge the domain gap between real-valued spatial images and complex-valued band-limited displacement fields without increasing complexity, *Fourier-Net* uses a DFT layer at the end of the FCN. This is a simple and effective way to produce complex-valued band-limited displacement fields without the network being able to handle complex values itself. The DFT applied to the displacement field ϕ can be defined as follows:

$$[\mathcal{F}]_{k,l} = \sum_{n=0}^{H-1} \sum_{m=0}^{W-1} \phi_{n,m} \cdot \exp\left(i \cdot \left(\frac{2\pi k}{H} \cdot n + \frac{2\pi l}{W} \cdot m\right)\right), \tag{4.2}$$

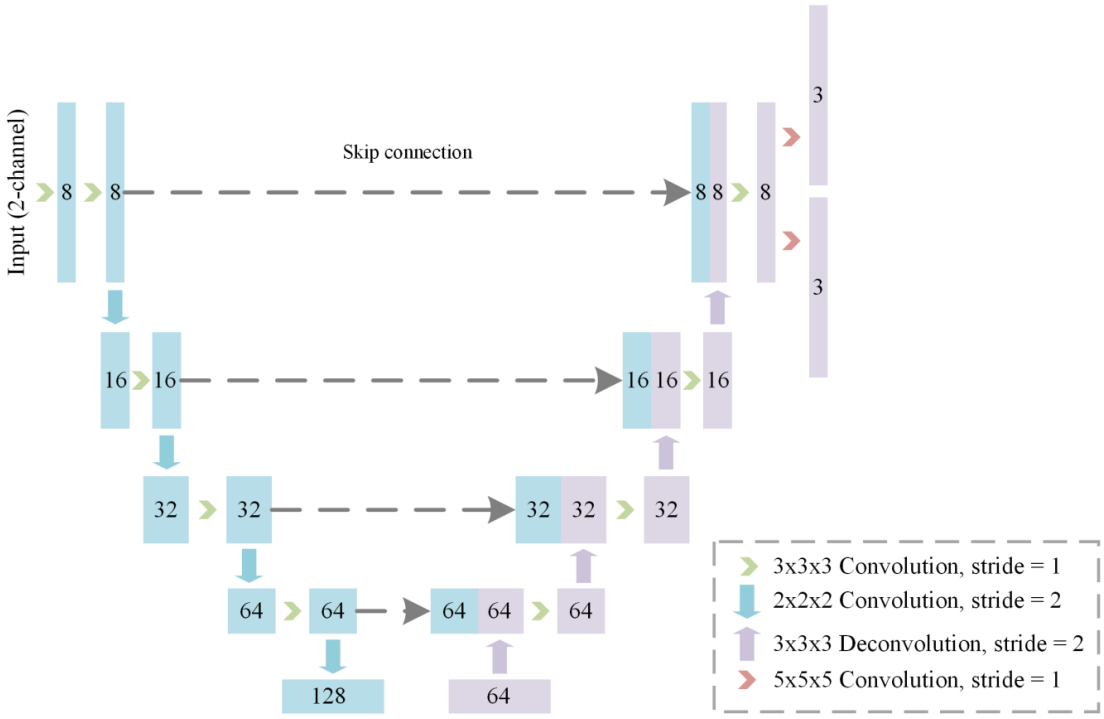


Figure 4.2: Architecture of the FCN from *SYMNet* [19].

where ϕ has size $H \times W$, $n \in [0, H-1]$ and $m \in [0, W-1]$ are the discrete indices in the spatial domain, and $k \in [0, H-1]$ and $l \in [0, W-1]$ are the discrete indices in the frequency domain with i being the imaginary unit. However ϕ in this equation is actually a low-pass filtered displacement field coming from the FCN, which can be formulated as follows:

$$\mathbb{S}_\phi = \mathrm{FCN}(M, F; \Theta), \tag{4.3}$$

with M being the moving and F the fixed image, as well as Θ representing the parameters of the FCN. Thus, the encoder can be defined mathematically as:

$$\mathbb{B}_\phi = \mathcal{F}(\mathbb{S}_\phi) = \mathcal{F}(\mathrm{FCN}(M, F; \Theta)), \tag{4.4}$$

with the DFT layer $\mathcal{F}$, full-resolution spatial displacement field ϕ and the complex band-limited displacement field $\mathbb{B}_\phi$.
For a $H \times W$ sized sampling mask $\mathcal{D}$ whose entries are zeros if they are on the positions of high-frequency signals in ϕ and ones if they are on the low-frequency positions, thus low-pass filtering the displacement field. With $\mathcal{D}$, the displacement field ϕ from equation (4.2) can be recovered via the iDFT as follows [7]:

$$\phi_{n,m} = \frac{1}{HW} \sum_{k=0}^{H-1} \sum_{l=0}^{W-1} \mathcal{D}_{k,l} [\mathcal{F}(\phi)]_{k,l} \cdot \exp\left(i \cdot \left(\frac{2\pi n}{H} \cdot k + \frac{2\pi m}{W} \cdot l \right) \right). \tag{4.5}$$

Decoder

The decoder contains no learnable parameters, instead the usual expansive path is replaced with a zero-padding layer, an iDFT layer, and an optional squaring and scaling layer. The output from the encoder is a band-limited representation $\mathbb{B}_\phi$ of the displacement field. To recover the full-resolution displacement field ϕ in the spatial domain, we first pad the patch $\mathbb{B}_\phi$ containing mostly low frequency signals to the original image resolution with zeros. We then feed the zero-padded complex-valued coefficients to an iDFT layer consisting of two steps: shifting the Fourier coefficients from centers to corners and then applying the iDFT to convert them into the spatial domain. For this, the low-frequency signals are shifted to a center patch with size $\frac{H}{a} \times \frac{W}{b}$ with $a = 2 \cdot Z_a, b = 2 \cdot Z_b$ where $Z_a, Z_b \in \mathbb{Z}^+$, which is then center-cropped to get $\mathbb{B}_\phi$, and reconstructed using the iDFT, thus modifying equation (4.5) to:

$$[\mathbb{S}_\phi]_{n,m} = \frac{ab}{HW} \sum_{k=1}^{\frac{H}{a}-1} \sum_{l=1}^{\frac{W}{b}-1} [\mathbb{B}_\phi]_{k,l} \cdot \exp\left(i \cdot \left(\frac{2\pi an}{H} \cdot k + \frac{2\pi bm}{W} \cdot l\right)\right), \tag{4.6}$$

with $n \in [0, \frac{H}{a} - 1]$ and $m \in [0, \frac{W}{b} - 1]$ being the indices of the spatial domain, while $k \in [0, \frac{H}{a} - 1]$ and $l \in [0, \frac{W}{b} - 1]$ are the indices of the frequency domain with i being the imaginary unit. The output from *Fourier-Net* is thus a full-resolution spatial displacement field as $\mathbb{S}_\phi$ contains all necessary information from ϕ:

$$[\mathbb{S}_\phi]_{n,m} = ab \cdot \phi_{an,bm}. \tag{4.7}$$

Because both padding and iDFT layers are differentiable, *Fourier-Net* can be optimized via back-propagation. For *Diff-Fourier-Net* extra squaring and squaring layers are needed in the decoder turning the displacement field into a stationary velocity field. Typically seven scaling and squaring layers are used to impose such diffeomorphism [7, 23].

Diffeomorphism

Describe what a diffeomorphism is. Describe Squaring and Scaling layers.
A diffeomorphic deformation is defined as a smooth and invertible deformation, thus the output of the iDFT layer can be regarded as a stationary velocity field denoted by v instead of the displacement field ϕ. In group theory, v is a member of Lie algebra, and it can be exponentiated to obtain the diffeomorphic deformation. This specific version is then called *Fourier-Net Diff* A diagram of this can be seen in Figure 4.1.

SpatialTransformer

The warping layer of *Fourier-Net* utilizes the *Spatial Transformer* [26], which allows for spatial image manipulation within the network. This is a differentiable and learnable module for neural networks which applies a spatial transformation to a feature map during a single forward pass. The spatial transformer mechanism is split into three parts as seen in Figure 4.3. First is the localization network, which takes the input and outputs the parameters for the transformation. These are then used to create a sample grid using the

grid generator. Lastly, the sampler produces the output feature map based on the input at the grid points.
From the input feature map $U \in \mathbb{R}^{H \times W \times C}$ with width W, height H and channels C the localization network f_{loc} computes the parameters $\theta = f_{\text{loc}}(U)$ of the transformation $\mathcal{T}_\theta$ which is later applied to the feature map. Thus the size of θ varies depending on the transformation. The localization network function can both be implemented as a fully-connected network or as a CNN, but should include a final regression layer to produce the transformation parameters [26].
In order to warp the input feature map, each output pixel is computed by applying a sampling kernel centered at a particular location in the input feature map. The output pixels are defined to lie on a regular grid $G = G_i$ of pixels, forming an output feature map $V \in \mathbb{R}^{H' \times W' \times C}$, where H' and W' are the height and width of the grid with C again being the number of channels, which is the same for input and output.
In order to perform a spatial transformation of the input feature map U, the sampler must take the set of sampling points $\mathcal{T}_\theta(G)$, along and produce the sampled output feature map V. Each coordinate (x_i^s, y_i^s) in $\mathcal{T}_\theta(G)$ defines the spatial location in the input where a sampling kernel is applied to get the value at a particular pixel in the output:

$$V_i^c = \sum_n^H \sum_m^W U_{nm}^c k(x_i^s - m; \Phi_x) k(y_i^s - n; \Phi_y), \tag{4.8}$$

with Φ_x and Φ_y being the parameters for a generic sampling kernel k that defines the image interpolation, U_{nm}^c is the value of the input feature maps at location (n, m) in the channel $c \in [1, \ldots, C]$ and V_i^c is the value for every pixel $i \in [1, \ldots, H'W']$ for the output feature map. Any sampling kernel can be used, as long as (sub-)gradients can be defined with respect to (x_i^s, y_i^s) to allow the loss gradients to flow back not only to the input feature map, but also to the sampling grid coordinates and therefore back to the transformation parameters ϕ and localization network, thus enabling back-propagation.

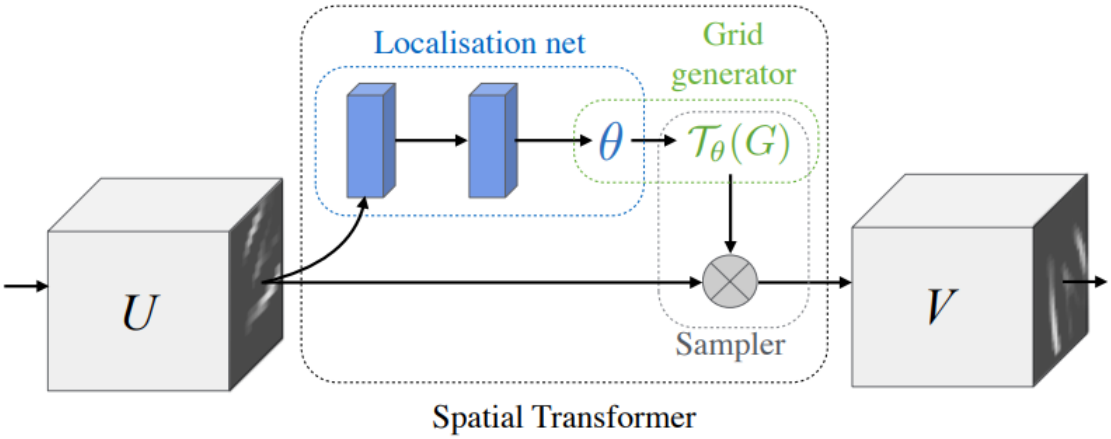


Figure 4.3: Architecture of the *Spatial Transformer* taken from [26].

Loss Function

The loss function consists of two parts to enable unsupervised learning, which are balanced using the scalar parameter λ. The first, $\mathcal{L}_1$, measures the similarity between the fixed image and the moving image after warping, while the second, $\mathcal{L}_2$, ensures a smooth displacement field. Thus, the unsupervised loss $\mathcal{L}$ can be calculated as follows:

$$\begin{aligned}\mathcal{L}(\Theta) &= \min\Big(\mathcal{L}_1(\phi(\Theta)) + \lambda \cdot \mathcal{L}_2(\phi(\Theta))\Big) \\ &= \min\Big(\mathcal{L}_1(v(\Theta)) + \lambda \cdot \mathcal{L}_2(v(\Theta))\Big),\end{aligned} \tag{4.9}$$

for both displacement fields ϕ and velocity fields v. The first part of the loss function consists of:

$$\mathcal{L}_1(\phi(\Theta)) = \frac{1}{N}\sum_{i=1}^{N} \mathcal{L}_{Sim}(M_i \circ (\phi_i(\Theta) + \mathrm{Id}) - F_i), \tag{4.10}$$

where $\circ$ denotes the warping operation, N the number of training pairs with moving images M_i and fixed images F_i, Θ the network parameters, ϕ_i the displacement field, Id the identity grid. $\mathcal{L}_{Sim}$ determines the similarity between warped moving images and fixed images via MSE or NCC, and the second term of the unsupervised loss, $\mathcal{L}_2$, defines the smoothness regularization function that controls smoothness of the displacement fields:

$$\mathcal{L}_2(\phi(\Theta)) = \frac{1}{N}\sum_{i=1}^{N} \|\nabla\phi_i(\Theta)\|_2^2, \tag{4.11}$$

with ∇ denoting the first order gradient and $\|\cdot\|_2^2$ denoting the squared L_2-Norm.
When using the squaring and scaling layers, thus making the deformation of the moving image diffeomorphic, the loss needs to be modified by replacing the displacement field θ with the velocity field v. Thus, both parts of the of the loss function need to be changed:

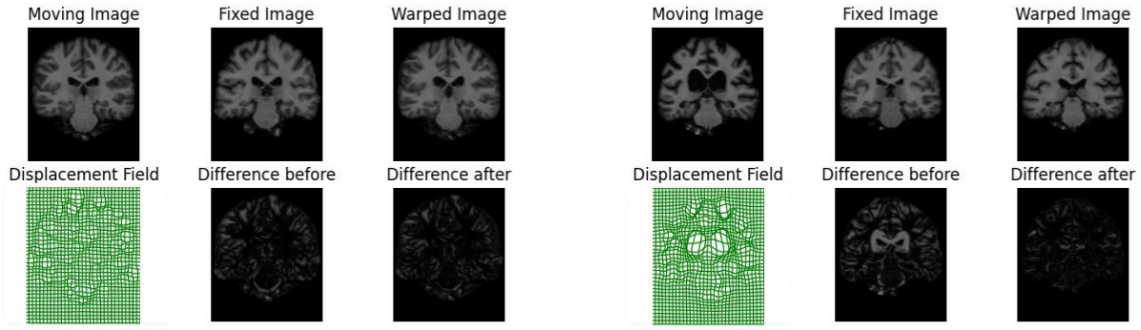
$$\mathcal{L}_1(v(\Theta)) = \frac{1}{N}\sum_{i=1}^{N} \mathcal{L}_{Sim}(M_i \circ Exp(v_i(\Theta) - F_i), \tag{4.12}$$

$$\mathcal{L}_2(v(\Theta)) = \frac{1}{N}\sum_{i=1}^{N} \|\nabla v_i(\Theta)\|_2^2 \tag{4.13}$$

Training

Describe how to reproduce results from the paper…
Fourier-Net can be used with both 2D as well as 3D inputs, however the latter are harder to visualize. As the *OASIS* dataset provides 3D volumes with annotations we can slice these to get 2D image data with matching labels (see Figure 3.1 for the calculation of the Dice-Score. Thus we can nicely visualize the training success of the 2D network by locking at the difference between images before and after registration in addition to the Dice-Score,

which can also be calculated for 3D data. This can be seen in Figure 4.4 with two examples from different stages of the training process. Despite the example in Figure 4.4b being the far harder example to align the network performs better than before (see Figure 4.4a) due to the training progress.



(a) Training progress after 800 iterations.

(b) Training progress after 29600 iterations.

Figure 4.4: Differences between moving and fixed images before and after registration.

4.2 Fourier Net+

Fourier-Net+, as the name suggests, is an extension of Fourier-Net which takes the band-limited spatial representation of the images as input, instead of their original full-resolution counterparts. This leads to further reduction in the number of convolutional layers in the contracting path of the network, resulting in a decrease of parameters, memory usage, and computational operations. This makes *Fourier-Net+* even more efficient than its older predecessor [8].
As seen in Figure 4.5, the network architecture is almost the same as for *Fourier-Net* (see Figure 4.1 for reference). However, while the decoder, and thus the loss function, remain the same, the encoder is slightly altered to make the network even more efficient. For this, similarly to the decoder, a DFT is used, however this time the idea is applied to the input images. These are first transformed into the Fourier domain, then low-pass filtered and finally reconstructed from their band-limited representation back into the spatial domain via an iDFT. The two images, now compressed, are the input for the encoder of *Fourier-Net*, meaning the CNN and following DFT. However, due to the band-limiting before the CNN, the latter can be made much more light-weight, thus reducing computational cost. This is visualized in Figure 4.6. Thus, *Fourier-Net+* too is overall lighter than the baseline *Fourier-Net* in terms of the number of parameters and computations. However, such a light network may face limitations in accurately capturing complex deformations. To counter this potential weakness, the authors propose a cascaded version of *Fourier-Net+*, which uses multiple versions of *Fourier-Net+* cascaded one after the other to achieve a better overall displacement field [8]. A schematic for this can be seen in Figure 4.7.

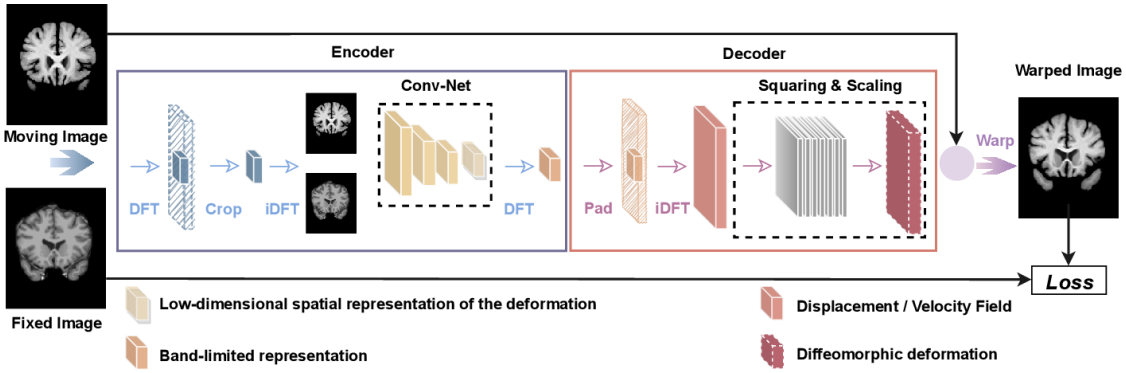


Figure 4.5: Architecture of *Fourier-Net+* taken from [8].

Changes to the Encoder

What changes compared to *Fourier-Net*?

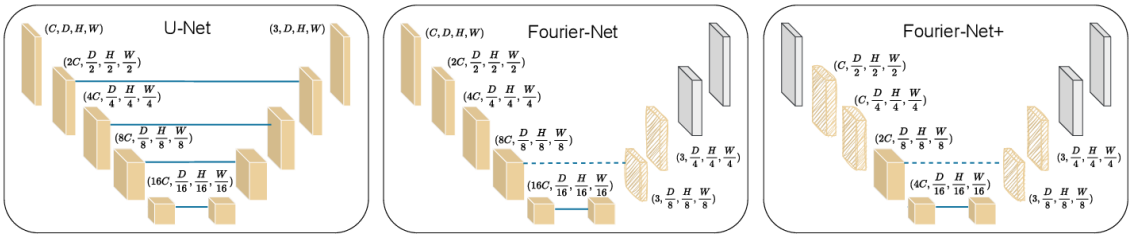


Figure 4.6: Architecture of the CNN for a typical U-Net, *Fourier-Net* and *Fourier-Net+* taken from [8].

Effects of Cascading

Why use cascaded version of the network?

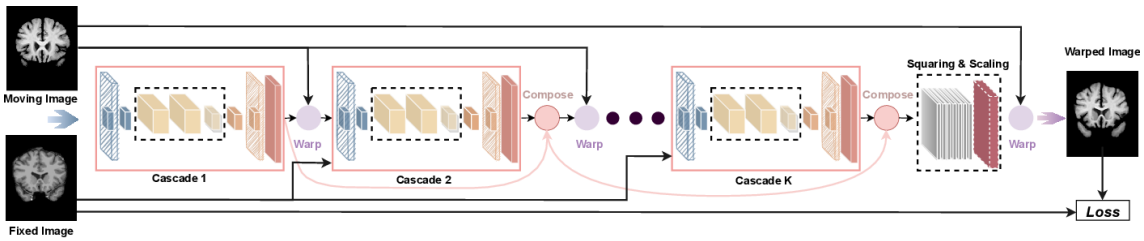


Figure 4.7: Cascaded version of *Fourier-Net+* taken from [8].

5

Experiments

Describe the experiment and evaluation methods being used.

6

Results and Discussion

Here go the results with the discussion.

7

Conclusion

Summery of all stuff…

Bibliography

[1] Chen, X., Diaz-Pinto, A., Ravikumar, N., and Frangi, A. Deep learning in medical image registration. In: *Progress in Biomedical Engineering*, Dec. 2020. ISSN: 2516-1091. DOI: 10.1088/2516-1091/abd37c.

[2] Haskins, G., Kruger, U., and Yan, P. Deep learning in medical image registration: a survey. In: *Machine Vision and Applications* 31(1–2), Jan. 2020. ISSN: 1432-1769. DOI: 10.1007/s00138-020-01060-x.

[3] Fu, Y., Lei, Y., Wang, T., Curran, W. J., Liu, T., and Yang, X. Deep learning in medical image registration: a review. In: *Physics in Medicine & Biology* 65(20):20TR01, Oct. 2020. ISSN: 1361-6560. DOI: 10.1088/1361-6560/ab843e.

[4] Zou, J., Gao, B., Song, Y., and Qin, J. A review of deep learning-based deformable medical image registration. In: *Frontiers in Oncology* 12, Dec. 2022. ISSN: 2234-943X. DOI: 10.3389/fonc.2022.1047215.

[5] Chen, J., Liu, Y., Wei, S., Bian, Z., Subramanian, S., Carass, A., Prince, J. L., and Du, Y. *A survey on deep learning in medical image registration: new technologies, uncertainty, evaluation metrics, and beyond*. 2023. DOI: 10.48550/ARXIV.2307.15615.

[6] Balakrishnan, G., Zhao, A., Sabuncu, M. R., Guttag, J., and Dalca, A. V. VoxelMorph: A Learning Framework for Deformable Medical Image Registration. In: *IEEE Transactions on Medical Imaging* 38(8):1788–1800, Aug. 2019. ISSN: 1558-254X. DOI: 10.1109/tmi.2019.2897538.

[7] Jia, X., Bartlett, J., Chen, W., Song, S., Zhang, T., Cheng, X., Lu, W., Qiu, Z., and Duan, J. Fourier-Net: Fast Image Registration with Band-Limited Deformation. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 37. 1. 2023, pp. 1015–1023.

[8] Jia, X., Thorley, A., Gomez, A., Lu, W., Kotecha, D., and Duan, J. *Fourier-Net+: Leveraging Band-Limited Representation for Efficient 3D Medical Image Registration*. 2023. DOI: 10.48550/ARXIV.2307.02997.

[9] Tam, A. L., Lim, H. J., Wistuba, I. I., Tamrazi, A., Kuo, M. D., Ziv, E., Wong, S., Shih, A. J., Webster, R. J., Fischer, G. S., et al. Image-Guided Biopsy in the Era of Personalized Cancer Care: Proceedings from the Society of Interventional Radiology Research Consensus Panel. In: *Journal of Vascular and Interventional Radiology* 27(1):8–19, Jan. 2016. ISSN: 1051-0443. DOI: 10.1016/j.jvir.2015.10.019.

[10] Chen, A. M., Hsu, S., Lamb, J., Yang, Y., Agazaryan, N., Steinberg, M. L., Low, D. A., and Cao, M. MRI-guided radiotherapy for head and neck cancer: initial clinical experience. In: *Clinical and Translational Oncology* 20(2):160–168, June 2017. ISSN: 1699-3055. DOI: 10.1007/s12094-017-1704-4.

[11] Rigaud, B., Simon, A., Castelli, J., Lafond, C., Acosta, O., Haigron, P., Cazoulat, G., and Crevoisier, R. de Deformable image registration for radiation therapy: princi-

ple, methods, applications and evaluation. In: *Acta Oncologica* 58(9):1225–1237, June 2019. ISSN: 1651-226X. DOI: 10.1080/0284186x.2019.1620331.

[12] Yang, D., Li, H., Low, D. A., Deasy, J. O., and Naqa, I. E. A fast inverse consistent deformable image registration method based on symmetric optical flow computation. In: *Physics in Medicine and Biology* 53(21):6143–6165, Oct. 2008. ISSN: 1361-6560. DOI: 10.1088/0031-9155/53/21/017.

[13] Vercauteren, T., Pennec, X., Perchant, A., and Ayache, N. Diffeomorphic demons: Efficient non-parametric image registration. In: *NeuroImage* 45(1):S61–S72, Mar. 2009. ISSN: 1053-8119. DOI: 10.1016/j.neuroimage.2008.10.040.

[14] Kingma, D. P. and Ba, J. *Adam: A Method for Stochastic Optimization*. 2014. DOI: 10.48550/ARXIV.1412.6980.

[15] Ronneberger, O., Fischer, P., and Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*. Springer International Publishing, 2015, pp. 234–241. ISBN: 9783319245744. DOI: 10.1007/978-3-319-24574-4_28.

[16] Fan, J., Cao, X., Yap, P.-T., and Shen, D. BIRNet: Brain image registration using dual-supervised fully convolutional networks. In: *Medical Image Analysis* 54:193–206, May 2019. ISSN: 1361-8415. DOI: 10.1016/j.media.2019.03.006.

[17] Zhang, J. *Inverse-Consistent Deep Networks for Unsupervised Deformable Image Registration*. 2018. DOI: 10.48550/ARXIV.1809.03443.

[18] Chen, J., Frey, E. C., He, Y., Segars, W. P., Li, Y., and Du, Y. TransMorph: Transformer for unsupervised medical image registration. In: *Medical Image Analysis* 82:102615, Nov. 2022. ISSN: 1361-8415. DOI: 10.1016/j.media.2022.102615.

[19] Mok, T. C. and Chung, A. C. Fast Symmetric Diffeomorphic Image Registration with Convolutional Neural Networks. In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, June 2020. DOI: 10.1109/cvpr42600.2020.00470.

[20] Marcus, D. S., Wang, T. H., Parker, J., Csernansky, J. G., Morris, J. C., and Buckner, R. L. Open Access Series of Imaging Studies (OASIS): Cross-sectional MRI Data in Young, Middle Aged, Nondemented, and Demented Older Adults. In: *Journal of Cognitive Neuroscience* 19(9):1498–1507, Sept. 2007. ISSN: 1530-8898. DOI: 10.1162/jocn.2007.19.9.1498.

[21] Hoopes, A., Hoffmann, M., Fischl, B., Guttag, J., and Dalca, A. V. HyperMorph: Amortized Hyperparameter Learning for Image Registration. In: *Information Processing in Medical Imaging*. Springer International Publishing, 2021, pp. 3–17. ISBN: 9783030781910. DOI: 10.1007/978-3-030-78191-0_1.

[22] Hering, A., Hansen, L., Mok, T. C. W., Chung, A. C. S., Siebert, H., Hager, S., Lange, A., Kuckertz, S., Heldmann, S., Shao, W., et al. Learn2Reg: Comprehensive Multi-Task Medical Image Registration Challenge, Dataset and Evaluation in the Era of Deep Learning. In: *IEEE Transactions on Medical Imaging* 42(3):697–712, Mar. 2023. ISSN: 1558-254X. DOI: 10.1109/tmi.2022.3213983.

[23] Dalca, A. V., Balakrishnan, G., Guttag, J., and Sabuncu, M. R. Unsupervised Learning for Fast Probabilistic Diffeomorphic Registration. In: *Lecture Notes in Computer Science*. Springer International Publishing, 2018, pp. 729–738. ISBN: 9783030009281. DOI: 10.1007/978-3-030-00928-1_82.

[24] Trabelsi, C., Bilaniuk, O., Zhang, Y., Serdyuk, D., Subramanian, S., Santos, J. F., Mehri, S., Rostamzadeh, N., Bengio, Y., and Pal, C. J. *Deep Complex Networks*. 2017. DOI: 10.48550/ARXIV.1705.09792.

[25] Wang, J. and Zhang, M. *DeepFLASH: An Efficient Network for Learning-based Medical Image Registration*. 2020. DOI: 10.48550/ARXIV.2004.02097.

[26] Jaderberg, M., Simonyan, K., Zisserman, A., and Kavukcuoglu, K. Spatial Transformer Networks. In: *ArXiv* abs/1506.02025, 2015. URL: https://api.semanticscholar.org/CorpusID:6099034.