2021-01-22

Deformable Image Registration Using Attentional Generative Adversarial Networks

Zhou, Hanchong

UNIVERSITY OF CALGARY

Deformable Image Registration Using Attentional Generative Adversarial Networks

by

Hanchong Zhou

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE

DEGREE OF MASTER OF SCIENCE

GRADUATE PROGRAM IN ELECTRICAL ENGINEERING

CALGARY, ALBERTA

JANUARY, 2021

Abstract

Deformable image registration is a fundamental process that aims to estimate non-linear spatial correspondence between input images. Medical image registration serves wildly on clinical treatment evaluation, monitoring disease and tracking disease. Conventional registration algorithms iteratively optimize a similarity function for each pair of images, which result in long registration time. Learning based registration method typically use convolutional neural networks to learn features automatically during training and register an image pair in one shot. Generative adversarial network is a novel structure that involves a generator and a discriminator, where the discriminator encourages the former to generate better results. Predicting deformation between brain magnetic resonance images is a complicated task because of their high-dimensional non-linear transform. A generative model leveraging is proposed using an attentional mechanism to estimate the complicated deformation field, and train the model using perceptual cyclic constraints. As an unsupervised method, our model dose not need any labels for training. Experimental results show quantitative evidence that the proposed method can predict reliable deformation field at a fast speed.

Acknowledgements

The thesis would not be finished without the support from my supervisor, Dr. Henry Leung. Throughout the time I spent with him, he is always passionate about cutting-edge research and has insightful understanding of various fields, especially on machine learning. His experience in image registration and machine learning is extremely valuable to me and is helpful to this work.

I would like to deliver my sincere gratitude towards my thesis examination committee members: Dr. Ethan MacDonald, Dr. Hadi Hemmati and Dr. Yunyan Zhang for their time reading this thesis. Thank you very much for your work to provided valuable comments and constructive feedbacks on this work.

I would also like to say thanks to department staff that work hard to support every student to finish their dream. I will never forget the timely help and suggestions from friends and colleagues in ICT 424A, our research group provides me with a positive environment that is always open for discussion.

Finally, allow me to express my gratitude to my family, Yin and all my friends for their unfailing care and support that companies me all the time. Thanks to anyone who is interested and reading this thesis.

Deliciated to my parents

Table of Contents

List of Tables

List of Figures and Illustrations

List of Symbols, Abbreviations and Nomenclature

Symbol	Definition
2D	Two Dimensional
3D	Three Dimensional
AE	Auto-Encoder
AG	Attentional Gate
ANTs	Advanced Normalization Tools
BN	Batch Normalization
B-spline	Basis Spline
CC	Cross Correlation
cGAN	Conditional Generative Adversarial Network
CNN	Convolutional Neural Network
Cnt	Count
CPU	Central Processing Unit
CT	Computed Tomography
D	Discriminator
DICOM	Digital Imaging and COmmunications in Medicine
DVF	Dense Vector Field
FA	Fluorescein Angiography
FCN	Fully Convolutional Neural Network
FLAIR	FLuid Attenuated Inversion Recovery
FoW	Field of View
G	Generator

GAN	Generative Adversarial Network
GHz	Giga-hertz
GPU	Graphic Processing Unit
GT	Ground Truth
ICE	Inverse Consistency Rrror
I_F / F	Fixed image/template image/target image
I_M / M	Moving image/floating image/source image
IXI	Information eXtraction from Images
LDDMM	Large Deformation Diffeomorphic Metric Mapping
L-ReLu/LReLu	Leaky Rectified Linear Unit
MI	Mutual Information
MR	Magnetic Resonance
MRI	Magnetic Resonance Imaging
MSE	Mean Squared Error
NCC	Normalized Cross Correlation
NIfTI	Neuroimaging Informatics Technology Initiative
NLP	Natural Language Processing
NMSE	Normalized Mean Squared Error
NN	Nearest Neighbour
OASIS	Open Access Series of Imaging Studies
ReLu	Rectified Linear Unit
RNN	Recurrent Neural Network
RoI	Region of Interest

SCD	Sunnybrook Cardiac Data
SSD	Sum of Squared Difference
STN	Spatial Transformer Network
SyN	Symmetric image Normalization
TFLOPS	Tera-flops
TPS	Thin-plate spline
VM	VoxelMorph
W.O.	Without

1. Introduction

1.1. Image registration overview

Image registration, also known as image alignment, is a high-dimensional optimization process that aims to find an optimal spatial transformation to map the corresponding points of two or more images obtained under different imaging devices, different time, and other conditions to a common space. At present, image registration has a wide range of applications in the fields of medical image processing, text recognition, material mechanics, etc.

In real life, when multiple image data is collected, there are misalignments between each of them in a set of images, because they might be taken from different time, from multiple devices, or they are images of different subjects, or the set of images are taken over a certain time. Image registration is a process in which two, or in general more images of the same scene are aligned.

1.2. Medical image registration and image modality

Medical image registration is an indispensable key step in medical image analysis and has become the pre-requisite for realizing medical image fusion, segmentation, contrast, and reconstruction. Medical images can reflect the anatomical structure, physiological functions, and pathological characteristics of human tissues and organs. Driven by the development of science and technology, various medical imaging devices have different imaging principles, and the resulting medical images contain different types of information features. Therefore, medical images can be divided into anatomical images and functional images.

Anatomical images are mainly used to distinguish different tissues and organs of the human body, which can clearly display their external structure information. For example, computed tomography (CT) is ideal for checking bone and joint, while magnetic resonance imaging (MRI) has better image quality for soft tissue like brain and muscle. Functional images are commonly

used to display the metabolism and blood flow of human body functions, but their spatial resolution is low and cannot distinguish the contour structure information of organs and blood vessels well enough. So more often, functional images are used to quantitatively, locate and accurately determine the local radiopharmaceutical concentration, and are mainly used for judgment of tumor progression.

Based on imaging modalities, image registration is divided into single-modal and multi-modal registration. Both have their own advantages. single -modal medical images have large consistency in pixel intensity, which usually bring in more accurate registration result. Multi-modal medical image registration is possible to utilize the complementary information between images of different modalities.

Single modal image registration is also known as mono- or uni-modal image registration, and it is an important step in clinical practice. Registering images obtained by the same patient during the course of the disease can monitor and track changes in the patient's condition along the treatment process. Registering single model images before and after treatment provides medical evaluation of the treatment effect of the patient. Registering images from different patients can do population studies and assist disease diagnosis in general. In this work, we focus on mono-modal registration problem.

1.3. Past and present of deformable image registration

1.3.1. Conventional registration methods

According to nature of transform, image registration can be grouped into rigid registration methods and deformable registration methods. Rigid registration is usually a global transform, where the number of transform parameters is limited, while deformable registration focuses on local discrepancies, typically with a large number of parameters to describe the transform.

Researchers have made great contribute to rigid image registration over the few past decades and proposed many rigid medical image registration algorithms with universality and high efficiency as it only needs to estimate translation, rotation, and affine transform. A comprehensive study of medical image registration can be found in [1]. However, as the nature of medical image deformation is non-rigid, it is inappropriate to fit a rigid transform model to the complex deformation in tissues and organs, so deformable medical image registration is necessary and become a mainstream active research direction.

Conventionally, deformable image registration methods are mainly classified into intensity-based methods and feature based methods. A conventional registration procedure typically includes feature extraction, feature matching, deformation field estimation, and image transformation. Intensity based methods directly use the similarity measurement between two images, such as sum of squared difference (SSD) and mutual information (MI), and progress iteratively to find the maximum or minimum similarity based on only the information inside the image. Without the need the search the entire image space, feature based registration methods is usually faster than intensity-based methods, but the results highly depend on the quality of chosen features.

Technically speaking, conventional deformable medical image registration methods can achieve relatively high alignment accuracy, but when the reference image is changed, the parameters must be re-adjusted to fit the new pair of images from scratch. As training is a high-dimensional and repetitive iterative optimization problem, conventional methods are inefficient and usually suffered from long registration time.

1.3.2. Registration methods based on neural networks

1.3.2.1. Supervised methods

By imitating the way of information transfer between human brain neurons, deep neural network shows its powerful learning ability that enables it to perform various tasks in the fields of image classification [2], image segmentation [3], image synthesis [4] and more. The increasing popularity of deep learning has driven research in many fields, as it can automatically learn and extract features from a large set of image data. It has made breakthroughs in the fields of medical diagnosis and medical image processing, such as brain tumors detection [5], lung diseases classification [6]. Recently, neural networks are also applied to deformable image registration. The use of deep learning methods separates the training and deploying steps, which enables fast and precise registration.

The current literature of using deep learning for deformable medical image registration can be roughly divided into two categories: an extension to conventional methods and methods that predict transform parameter direction. The methods that serve as the extension of traditional registration methods only use neural networks to extract features or image descriptors for each pixel, and it still use the conventional iteratively optimize the objective function. The second type uses supervised or unsupervised deep learning methods to achieve one-step registration, and directly uses neural networks to predict spatial transformation parameters. The former still needs to be combined with traditional registration methods for iterative optimization. Early research is mostly the first type of method, but it does not give full play to the advantages of deep learning, takes a long time, and is difficult to achieve real-time registration. With the gradual increase in the requirements for registration speed, the research on one-step registration algorithms based on deep learning is also increasing.

Convolutional neural network (CNN) is a good candidate to predict the complicated non-linear deformation. Miao er al. [7] is one of the first to use CNN to train a neural network and predict rigid transform parameters in the end-to-end fashion. Eppenhof et al. [8] showed that it is possible to perform 3D registration with high speed using the CNN approach. In this work, a smaller version of VGG architecture has been adapted to learn transformation parameters between pairs of three-dimensional images for deformable registration task.

Although there are some works that predict rigid registration parameters using deep neural networks [8][7], its advantage to model complex transform is not utilized. As inferred in [9], it is more common to apply deep learning to regress the high-dimensional deformable field to solve deformable registration problem. Simonovsky et al. [10] use a CNN to compute similarity costs and to optimize the transformation parameters of the registration task, CNN was trained with patches of multi-modal image sets such as T1-MRI and T2-MRI. In Yang et al. [11], a deep multi-step encoder-decoder has been used to predict the deformation model through regression. It focused on a Large Deformation Diffeomorphic Metric Mapping (LDDMM) model. This approach is nonparametric and allows the learning of a voxel-wise metric at the same time as it predicts the transformation parameters from the image patches reducing the computational complexity of the registration process.

1.3.2.2. Unsupervised methods

The aforementioned learning-based deformable medical image alignment methods show accurate alignment performance, but these methods require training of supervised models of massive, labeled image data. Unlike natural images, it is extremely costly to acquire such labels for medical image. Therefore, unsupervised learning becomes an important criterion in medical image registration as it removes the needs of labels to train the network.

Spatial transformer network (STN) [12] proposed by Jaderberg et al. simplified the registration from predicting the transform parameter to predicting the deformation field. Nowadays, almost all neural networks for image registration use STN as an end-to-end trainable building block that transforms the input image according to the deformation field. For example, Bob D. de Vos et al. [13] utilized a CNN regressor to compare input images and generate parameters for a local deformation. Then, a neural network estimated a spatial transformation and performed the resampling process with STN. One of the most popular unsupervised network is VoxelMorph (VM) [14], a U-net like neural network for image registration. During training, VM tries to maximize the similarity measure between the fixed image and moving image and learns how to estimate a proper deformation field with large set of data. When deployed, VM can predict the deformation field and directly generate the transformed image in one shot. Most recently, generative adversarial nets (GANs) also become available for image registration [15][16], and experimental results show promising performance using GAN.

1.4. Motivation and thesis outline

Although deep neural networks have been proved to work well on the registration task, most existing methods did not consider the inverse consistency of the registration. In other word, the resulted registration is asymmetric. These methods see the image pair as individual image, and do not put any attention on the topological preservation during that may leads to a degrade on the deformation.

In this thesis, we aim to solve this problem by applying perceptual cyclic loss in an attentional adversarial training, which supports topological preservation by adding a cyclic constraint on the transformation and achieves a better similarity metric by incorporate a perceptual loss term in the objective function. Additionally, to reduce region imbalance, we

design an attentional model to suppress less significant features. Besides, as an unsupervised method, ours do not require any pre-defined ground truth of the transformation.

Chapter 1 introduces the importance of medical image registration and why deep neural nets are fit to solve the deformable image registration problem. The rest of this thesis will be organized as follows:

Chapter 2 gives a formulation of both conventional and learning based deformable image registration problem. First introduction to the key steps in conventional registration framework is given. This chapter also covers a detailed discussion on neural network design for deformable image registration. Finally, a novel design of image registration network is presented, along with the insights to highlights of the design of the registration network.

Chapter 3 provides the discussion and the importance to the generative adversarial nets along with the attentional mechanism. Topic also include a proposed training scheme that utilizes attentional generative adversarial net with cyclic perceptual loss. In this chapter we also discuss the motivation to use such settings and shows such setting could help achieve better prediction as well as preserve topology in the deformation field.

Chapter 4 carries out experiments on both a real-world dataset and a synthetic dataset that are collected from hospitals and research labs. We perform ablation study on each components of the proposed network and make comparison with a top performing algorithm and a state-of-the-art learning-based algorithm.

Chapter 5 concludes the thesis with the findings and summary of this work. The major contributions of this thesis are the novel design of the generative registration network with attentional model integrated and the usage of cyclic perceptual loss. Recommendation for future direction is provided.

2. Modeling image registration with neural networks

2.1. Formulation of image registration problem

Given a pair of a fixed image I_F (also called reference image, template image or target image) and a moving image I_M (also called floating image), the task of image registration is to find the topological spatial mapping that can align the fixed image and the moving image, so that the pixels of the two or more images to be registered are at the same spatial position represents the same anatomical structure. The resulting image is obtained by apply the mapping on the moving image, it is usually known as registered image or warped image. Typically, we tend to register the moving image to fixed image, but in fact we can do either way.

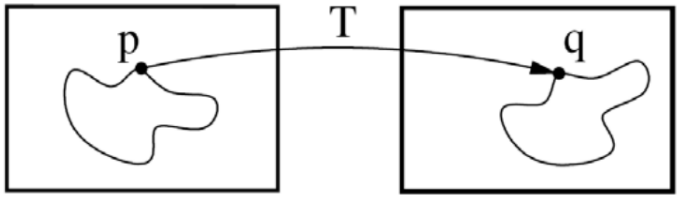


Figure 2.1 Spatial mapping between the images

The idea to solve this problem is to find a best spatial transform $\widehat{\Phi}$ by solving for the following optimization problem,

$$\widehat{\Phi_\mu} = \underset{\Phi}{\operatorname{argmin}}\{J(I_F, (I_M \circ \Phi_\mu)) + \lambda \cdot R(\Phi_\mu)\}$$

where main loss function $J(\cdot)$ is a similarity measurement, the spatial transform Φ_μ is parameterized by its transform parameter μ. The spatial transform Φ is typically modeled by a dense vector field (DVF), also known as deformation field or deformation flow. $(I_M \circ \Phi_\mu)$ represents the wrapping on moving image I_M by the deformation field Φ_μ. R is the regularization on the deformation field, which ensure the smoothness of it. λ is a coefficient to balance the main objective and the transform.

2.2. Key steps in image registration

2.2.1. Generic registration procedure

It is widely acknowledged that a generic image registration pipeline include four key steps [1]: spatial transformation, similarity measurement, optimization algorithm and interpolation algorithm. Image preprocessing include denoising fixed and moving image, pre-align image to the same coordinate so that they have same size as neural network takes inputs of identical shape. A descriptive diagram is shown in Figure 2.2. Spatial transformation is applied based on the feature extraction and matching from the fixed and moving image, followed by the interpolation that complete warp the moving image to the fixed image. Decided by similarity measurement, the optimization may continue the update the transform parameter unless the similarity is maximized or reach a certain threshold. This is typically an iterative process.

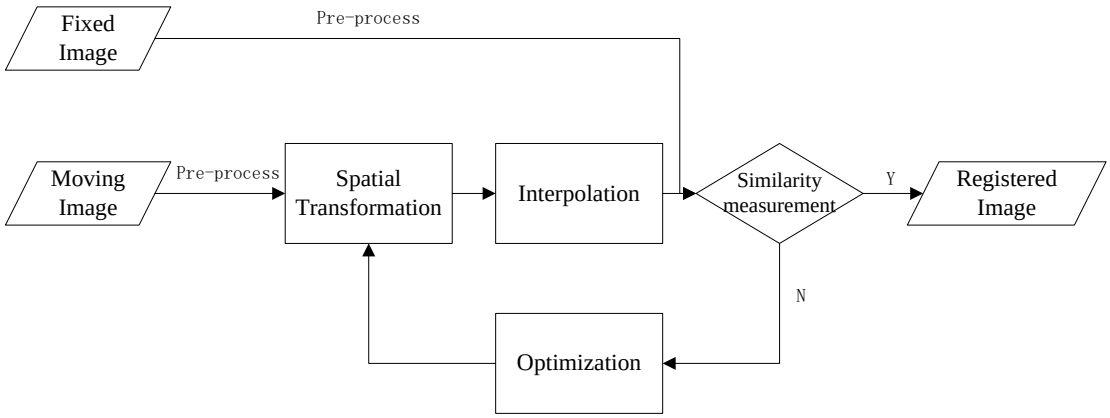


Figure 2.2 Generic image registration diagram

2.2.2. Key steps in image registration

2.2.2.1. Spatial transformation

Spatial transformation refers to a mapping model that associates a moving image with a fixed image. Commonly used geometric transformations include linear

transformations and nonlinear transformations. Linear transformation is composed of rigid transformation, affine transformation, and projective transformation. Linear transformation only includes translation, rotation, mirroring, scaling and projective transform of the image. The property of linear transform is that the straight line remains a straight line after the transform, which is called line preserving. Thus it is inappropriate for medical image because of the local non-linear deformation caused by tissue motion or external interference on medical images [17]. Compared with linear transformation, non-linear transformation has larger capacity to model the non-linear transform, but the complexity of the model increases, and the amount of calculation increases as well.

1) Rigid transform

Rigid transformation is the most basic global linear transformation, which only includes changes in the position (translation) and orientation (rotation) of the coordinate axis. After transformation, the distance between any two points remains unchanged, and the angle between any line segments remains unchanged as well. Denoting an arbitrary point (x, y, z) on the image, the same point (x', y', z') on the transformed image can be obtained by,

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\omega & 0 & 0 \\ -\sin\gamma & \cos\gamma & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\omega & 0 & -\sin\omega & 0 \\ 0 & 1 & 0 & 0 \\ \sin\omega & 0 & \cos\omega & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & \sin\theta & 0 \\ 0 & -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

A typical transform matrix for three-dimensional rigid transform is composed of 6 independent parameters, including 3 translational and 3 rotational parameters. t_x, t_y and t_z indicates the translation displacement while θ, γ and ω represents the

rotation on the three axes. Rigid registration is usually used on satellite images or medical image such as joints or spine, where the global deformation on these images should be paid attention to.

2) Affine transformation

Compared to rigid transformation, affine transformation can describe more complicated linear transformations. Affine transformation is composed of a series of basic transformations. Before and after the transformation, the straightness and parallelism will be maintained, but the length and angle of the straight line are not maintained. Three-dimensional affine transformation is typically described by 12 parameters,

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & t_x \\ a_{21} & a_{22} & a_{23} & t_y \\ a_{31} & a_{32} & a_{33} & t_z \\ 0 & 0 & 0 & s \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

where the abbreviation matrix $A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$ indicates the rotation and shearing, while $T = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}$ indicates the translation. If scaling of the image is needed, *s* can be changed to fit the scaling coefficient and makes it 13 parameters in total. Affine transformation is usually used as a preprocess step for deformable registration as it can shape the inputs to the same size.

3) Projective transformation

As its name indicates, projective transformation projects the image to another plane and establish a one-to-one correspondence between the pixels of the two input planes. In essence, it is a procedure from 2D to 3D and then back to 2D mapping. Straightness is preserved, but the parallelism and perpendicularity are not maintained. Projective transformation is usually used in 2D-to-3D or 3D-to-2D registration.

4) Non-linear transformation

Non-linear transformation can transform a straight line to a curve and vice versa, which make it fit deformable imager registration task most. Non-linear transformation can be divided into non-parametric and parametric model.

Parametric models use function with a small number of parameters to describe the complex dense transformation. Radial basis functions are commonly seen in registration, such as thin-plate spline (TPS) and basis spline (B-spline). B-spline functions have local support. When simulating a non-linear transform, the change on the control points only affects the neighbourhood regions because of it.

Non-parametric models often have physical intuitive. Popular non-parametric models include elastic model [18], viscous fluids [19] and flow model. Elastic model sees the image as elastic materials and formulate the registration procedure as its internal force and external force seeks equality. Viscous fluids see the registration as a process where the fixed image is warped to the moving image by its internal force. Originated from computer vision community, flow model [20] based registration is a more recent approach to image registration. Registration is regarded as the moving image moves to the fixed image.

2.2.2.2. Similarity measurement

The goal of image registration is to make the fixed image and the moving image look similar. The similarity measure is to judge how similar the two images are. The best solution is obtained by iteratively optimizing the similarity measure. The loss function used in medical image registration usually consists of two parts: similarity measure calculated by pixel intensity value or structural similarity, and regularization term. Commonly used similarity measures based on intensity include

mean squared error (MSE), normalized cross-correlation (NCC) and mutual information (MI).

1) Mean squared error (MSE)

$$MSE = \frac{\sum_{i=0}^{I}\sum_{j=0}^{J}[F(i,j) - M'(i,j)]^2}{I \cdot J}$$

where *I*, *J* are the size of the image. MSE is computed as the average squared error over all pixels (*i*, *j*) between the fixed image and the transformed image M'. A good registration should give the minimum error between the fixed image and registered image.

2) Cross-correlation (CC)

$$CC = \frac{\sum_{i=0}^{I}\sum_{j=0}^{J}\left[F(i,j) - \overline{F(i,j)}\right]\left[M'(i,j) - \overline{M'(i,j)}\right]}{\sqrt{\sum_{i=0}^{I}\sum_{j=0}^{J}\left[F(i,j) - \overline{F(i,j)}\right]^2}\sqrt{\sum_{i=0}^{I}\sum_{j=0}^{J}\left[M'(i,j) - \overline{M'(i,j)}\right]^2}}$$

The assumption of using CC and its variant normalized cross-correlation is that the intensity of input pairs should have linear relationship. Thus, it is proved to be more robust than MSE and commonly use in single modal registration.

3) Mutual information (MI)

$$MI = \sum_{F}\sum_{M} p(F, M') \log_x \frac{p(F, M')}{p(F)p(M')}$$

where $p(\cdot)$ denotes the marginal distributions. As mutual information and its variant normalized mutual information (NMI) represents the statistical correlation between the input images, it does not require high similarity in input images. Thus, mutual information is usually applied to solve multi-modal registration problem.

2.2.2.3. Optimization algorithm

The optimization algorithm refers to the iterative search to minimize the loss function and constantly adjust the space transformation parameters to find the optimal

solution, such that the registered images are aligned with the fixed image as much as possible. According to different images and registration requirement, transformation models are usually various. Only if an optimization algorithm fits the transformation model can the model converge quickly and find the optimal transformation parameters through the optimizing procedure. To improve the computational efficiency and stability of medical image registration, determining a reasonable and applicable optimization algorithm is very important in medical image registration.

1) Powell's method

Powell's method [21] is a local search method proposed to solve unconstrained optimization problem. Without the need to compute the gradients of each pixel, it is proved to be computation efficient and robust in many areas. However, Powell's method relies heavily on the initialization. This may be a problem in high dimensional images as it may not converge because of local minima.

2) Gradient descent

As one of the most popular optimization algorithms in machine learning, gradient descent receives its name because the optimizing direction is simply negative of the gradient. By iteratively find the gradient, the optimization will finally converge to the minima. It is worth to mention that gradient descent requires a suitable step size. An oversized step may miss the optimal solution while an undersize step may cause the search to be trapped in a local minimum.

2.2.2.4. Interpolation

In the medical image registration, the coordinates of the corresponding pixels in the moving image after spatial transformation are not necessarily mapped on the grid point of the fixed image, which results in a failure to create the warped image. The principle of the interpolation algorithm is to use the spatial correlation between pixels

to assign values to coordinate points that are not on the grid according to the intensity of surrounding pixels and the distance of the spatial position. Optimization is an iterative process that involves thousands or tens of thousands step of spatial transformations and interpolation operations on the image. Therefore, choosing a suitable and computation efficient interpolation algorithm can speed up the registration process and reduce the distortion to some extents.

1) Nearest neighbour (NN) interpolation

As the simplest interpolation algorithm, nearest neighbor interpolation method searches for the neighboring grid points around a point *p* that is not on the grid in the moving image after spatial transformation, and directly selects the intensity value of the point closest to the point *P* as its intensity value. Although this method can save a lot of calculation time, it completely ignores the influence of the intensity of the remaining adjacent points. This may introduce distortion, mosaic, and aliasing in the registered image.

2) Bilinear interpolation

Bilinear interpolation is one of the most popular interpolation algorithms in image registration. Bilinear interpolation considers the intensity of 4 neighbourhood points and aggregates them according to ratio of the distance between the neighbourhood point and point *P* itself. Compare to the nearest neighbour interpolation, bilinear results in an image of much better quality while increase the computation burden not by much. Denoting the point to be estimated as $p(i+u, j+v)$, the bilinear interpolated intensity of p' is obtained by,

$$p' = (1-u)(1-v)f(i,j) + u(1-v)f(i,j+1) + v(1-u)f(i+1,j) + uvf(i+1,j+1)$$

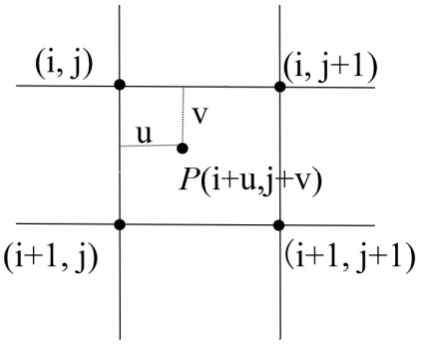


Figure 2.3 Bilinear interpolation

There is more advanced method that use 16 neighbourhood points to obtain the interpolated value, which is known as cubic interpolation. Cubic interpolation reduces the loss of high frequency component so that it can achieve better results. However, we do not use cubic interpolation in practice as it is extremely computational heavy.

2.3. Image registration using neural networks

Nowadays, registration methods are experiencing an important revolution from traditional methods to deep learning methods. Recent studies have shown that deep learning methods can be used for non-rigid medical image registration. Deep learning methods automatically learn various and complicated information in images related to the registration task. As deep learning training introduce significantly fewer local minima problem, it converges faster and make reliable prediction compared to using conventional method. In addition, deep learning methods are highly parallelizable that it be implemented on GPUs. It is computation efficient to process large amounts of data and to predict multiple data at the same time.

2.3.1. Key steps in Image registration using neural networks

2.3.1.1. Image preprocessing

A sample clinical brain MRI data is shown in Figure 2.4. Human body is anatomically divided into three planes according to the view of an imaginary scanner, namely axial/transverse plane (scanner moves vertically), coronal plane (scanner

moves back and forth) and sagittal plane (scanner moves horizontally). The T1-weighted brain MRI data contains multiple tissue information including spinal cord, bone, and fat, as well as anatomical structures such as skull, neck, and eyes. In Figure 2.4, top left: Axial/Transverse plane, top right: coronal plane, bottom left: sagittal plane, bottom right: Anatomical planes of body.

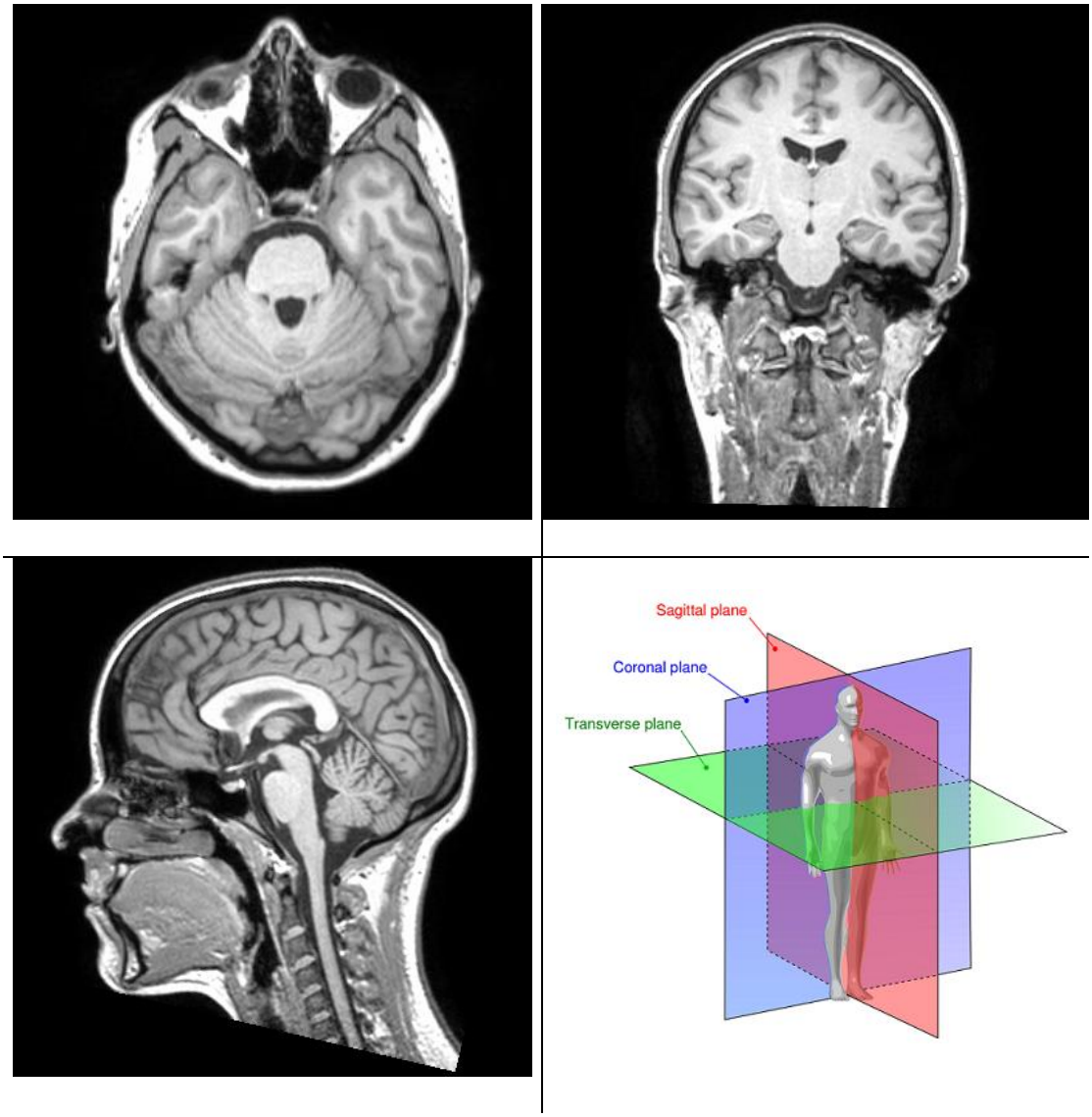


Figure 2.4 Clinical brain MRI data

In brain registration task, organs are not relevant and would become a distraction to the registration task. Therefore, a series of standard pre-processing steps would be necessary before the data is fed into the registration network. Standard

image preprocessing for brain images includes brain extraction, affine registration, cropping and normalization.

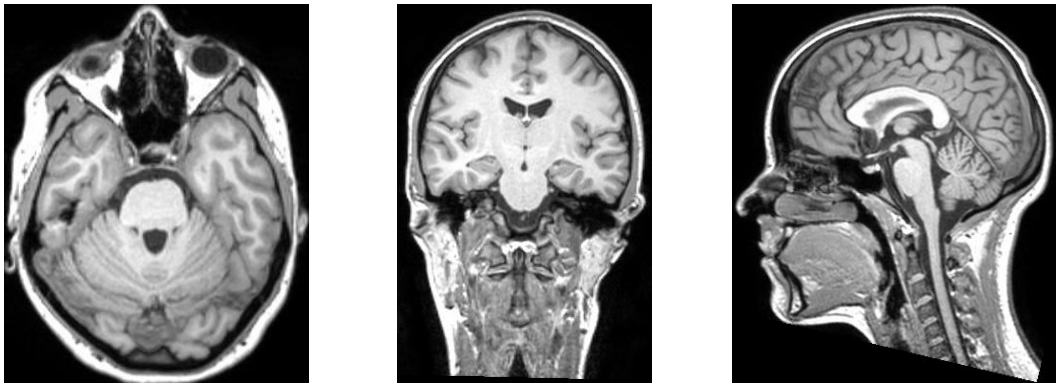
2.3.1.1.1. Brain extraction

Extracting brain part in the raw MRI image requires lots of professional skill. As our work focuses on the registration process, here we use FreeSurfer [22] for convenience to perform brain extraction to the our data. FreeSurfer is one of the most world-known and freely available medical image processing software. With its powerful core command recon-all, which contains motion correction, neck removal, skull stripping and other 26 systematic steps in total 3 phases. that is necessary enough for a full cortical reconstruction process.

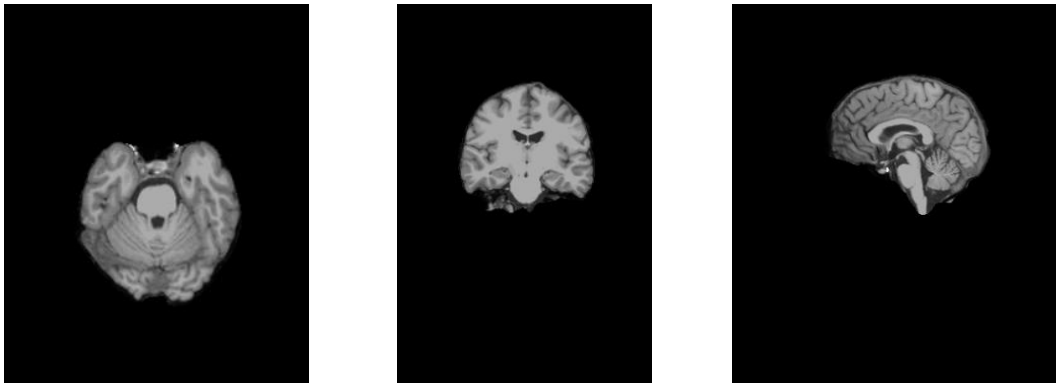
As the whole cortical reconstruction may take around 14~16 hours per image file, here we only use its functions for neck removal, skull stripping and resolution resampling. Resolution resampling ensures that the pixel has the same physical spacing despite different settings of capturing machines. The command we use for these is,

recon-all -parallel -i *filename* -autorecon1

To speed up the process time, we add -parallel option to force it use multi-threads. -autorecon1 option indicates it only carry out steps in phase 1 as these three steps are included in phase 1. After the above command, the raw image data is resampled to 1x1x1 mm^3 resolution by default. A sample processed image is shown in Figure 2.5 below. The first row shows the middle slice in a raw 3D image in the dataset, the second row shows the corresponding view of it after the brain extraction step.



a) Raw image data



b) Extracted brain

Figure 2.5 Brain extraction (Skull stripping)

Additionally, since there many data format in medical image file system, such as Neuroimaging Informatics Technology Initiative (NIfTI), Digital Imaging and Communications in Medicine (DICOM) and Massachusetts General Hospital NMR Center format (MGH), we save all other formats as NIfTI file with file extension nii.gz by using the build-in command mri_convert in FreeSurfer to ensure the resulting data is in the same file format and the same coordinate system.

2.3.1.1.2. Affine registration

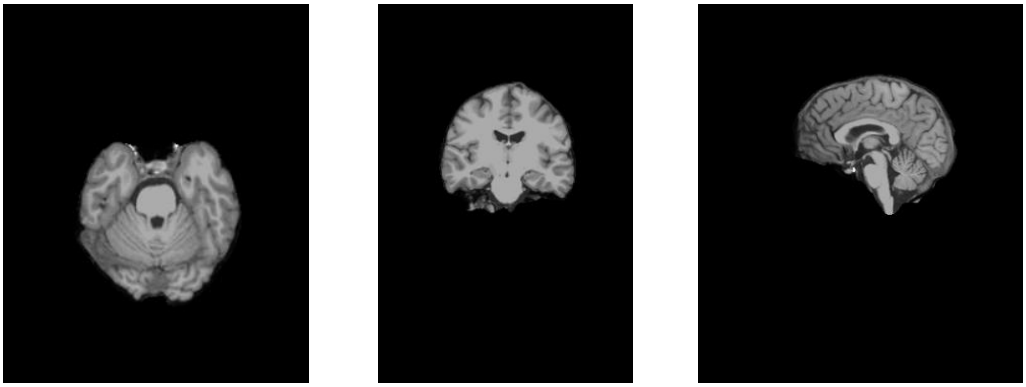
The goal of this work is to estimate the non-linear transformation between the moving image and the fixed image that makes up the deformable registration of medical images. Therefore, all data performs affine registration before the deformable registration. As discussed before, affine registration is a combination of a series of

transforms including translation, scaling, rotation and shearing. It is a linear transformation that maintains the parallelism and straightness of lines. This work performs affine registration on the image pre-processed by the aforementioned brain extraction to complete the pre-alignment of the image. Till this step, the moving image and the fixed image have been aligned globally, and the only misalignment factor comes from the non-linearity between the image pairs.

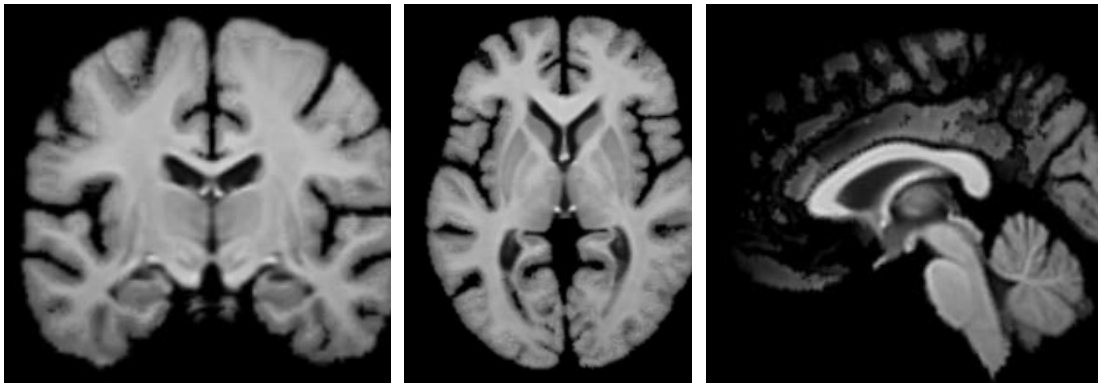
We use best open-source registration tool Advanced Normalization tools (ANTs) for Affine registration. ANTs are popularly considered as state-of-the-art registration and segmentation packages for medical image processing. The command we use to perform affine registration is,

antsRegistrationSyNQuick.sh -d *dimension* -f *fixedimg* -m *movingimg*-o *outputimg*-t a

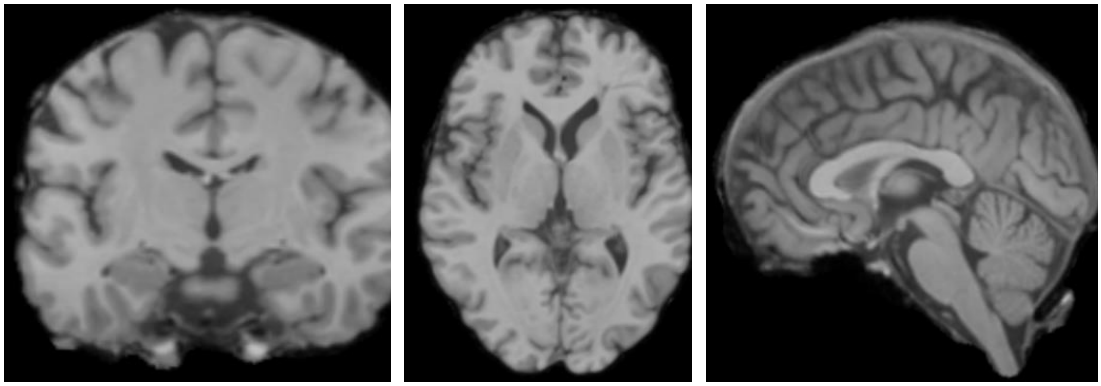
where dimension refers to the dimension of the data, 2 for 2D data and 3 for 3D data. *fixedimg*, *movingimg* and *outimg* specify the fixed image, moving image and affinely registered image, respectively. The option -t taking a as parameter indicates this only involve affine transform in the procedure. We affinely register images coming from different sources to one “atlas” image as reference. A pair of sample image data (showing middle slice of the 3D data) are shown in Figure 2.6. The first row shows the moving image, the second row show the atlas image, the third row shows the registered image with affine transform only. One can see that the rotation (especially in column 3) and the middle slice (in column 1 and 2) are changed significantly because of the affine registration.



a) Moving image



b) Atlas image



c) Affinely registered image (with cropping and normalization)

Figure 2.6 Affine registration

It is worth to mention that the SyN algorithm [23] which is also one of the best conventional deformable registration methods is included in ANTs, and we use it as a baseline method to compare with our result.

2.3.1.1.3. Cropping

In order to ensure the input images of the neural network are of the same size, all images will be crop to the same size after brain extraction and affine registration to ensure that all images have the same shape. Also, cropping image to a smaller size can reduce the burden of the network as well as fit the training in the GPU memory.

In additional, we have to make sure that the cropped image will contain the whole structure while eliminating the blank area. This will be done by cropping around the center of mass. The center of mass is essentially the pixel where the sum of image intensity along *x*, *y* and *z* axes are the same (only *x* and *y* for 2D image). Thus, the center of mass of the image is defined as,

$$x_0 = \frac{\sum_{p_i} x_i}{\sum p_i}, y_0 = \frac{\sum_{p_i} y_i}{\sum p_i}, z_0 = \frac{\sum_{p_i} z_i}{\sum p_i},$$

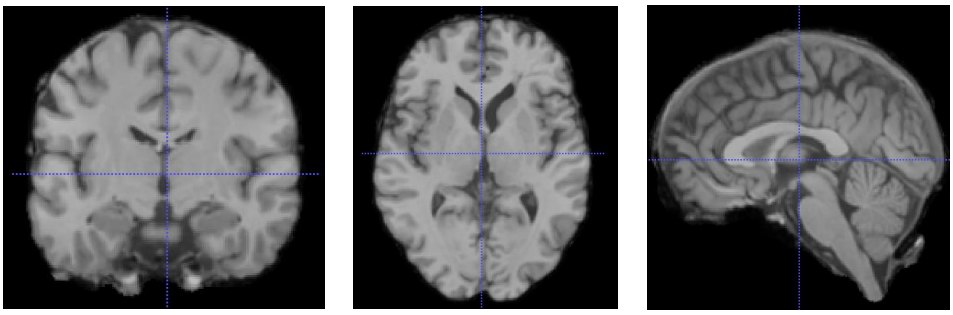


Figure 2.7 Center of mass in the scan

As indicated by the intersection of the two blue dotted lines, the Figure 2.7 above shows the massive point calculated by the above equation. Find the massive point of the image and crop a space of certain size out around this massive point. In our experiment, we empirically set it to be 160x192x224.

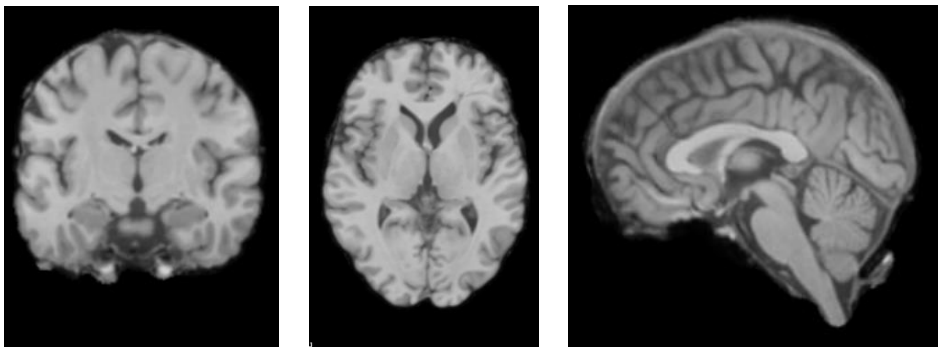
2.3.1.1.4. Normalization

Since the image data comes from different people, at different image acquisition times or by different image acquisition equipment, the collected data may have huge differences in intensity distribution, and there may be problems such as

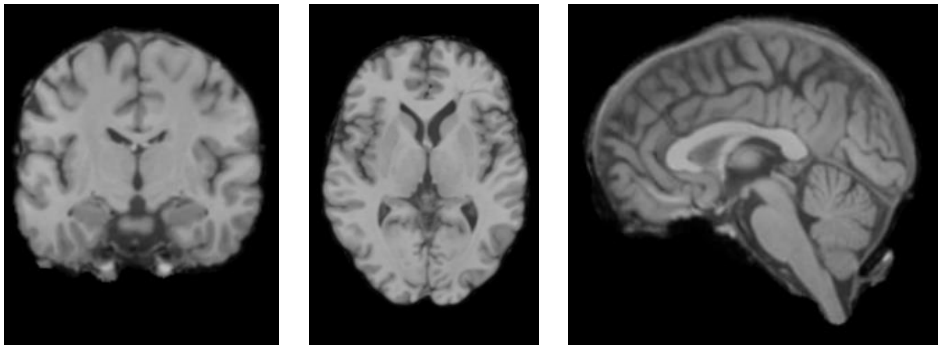
concentration of intensity distribution. The intensity of MRI image reflects the non-absorbed radio frequency energy from the external magnetic field, thus different machines may give different intensity value even for the same person. Simultaneously, most neural networks constraint their output to range [0, 1] because of the activation function output limits. These problems will affect the feature extraction and recognition of neural networks. To avoid these problems, we apply normalization to the cropped data and scale the intensity to [0, 1] by,

$$i'(x,y) = \frac{i(x,y) - i_{min}}{i_{max} - i_{min}}$$

where $i(x,y)$ and $i'(x,y)$ represent the intensity of pixel (x,y) before and after normalization respectively, i_{min} and i_{max} are the minimum and maximum intensity of whole dataset, respectively. The i_{min} and i_{max} in our brain dataset are 124 and 212.



a) Before normalization



b) After normalization

Figure 2.8 Normalization

The above Figure 2.8 shows the difference before and after the normalization.

Till now it finalizes the pre-processed image that we can use as our training or testing data. Figure 2.6 row c shows an example of a fully pre-processed image. The main part of the neural network takes the moving image and fixed image as input, and predicts the deformation field as the transform parameter for deformable image registration. The deformation field is a dense, pixel-level vector that indicates the transform from moving image to the fixed image.

2.3.1.2. Supervised and unsupervised learning for image registration

As its name indicates, supervised methods for image registration methods require the corresponding ground truth deformation fields in training. A diagram for supervised learning framework is shown in Figure 2.9. Usually, the network takes a pair of images (moving and fixed image) as input, the output of the network is the transform parameters (either deformation field or rigid transform parameters). The loss is the MSE between the predicted parameter and the ground truth parameter.

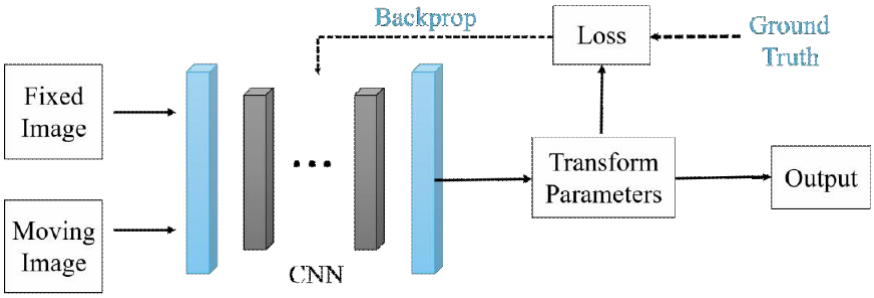


Figure 2.9 Supervised learning framework for image registration

Typically, deep learning models often face the problem of if the training data is insufficient, which makes the model perform poorly on the test set. People usually increase the training data amount to solve this problem. However, it is difficult for medical images to get a large number of real deformation field. Thus, supervised training is not feasible for medical image registration.

To overcome this, the trend comes up with unsupervised registration approaches. Compared with supervised methods, the image pairs themselves are the only input to the network, without the need of the corresponding deformation field. Similarly, the network takes a pair of images as input, the output of the network is the transform parameters. Then the output transform parameters are connected a differentiable spatial transform network [12], which warp and transform the moving image with the predicted deformation field. The final output of the network is the registered image, which fulfills an end-to-end fashion. Typically, the loss used in this unsupervised learning framework is similar to the objective function in the conventional registration network,

$$L = L_{sim} + \alpha L_{reg}$$

where L_{sim} is the similarity loss (particularly CC or MI) between the predicted image and the fixed image, while L_{reg} is the regularization on the deformation field. α is to balance the penalty on image similarity and deformation field regularization.

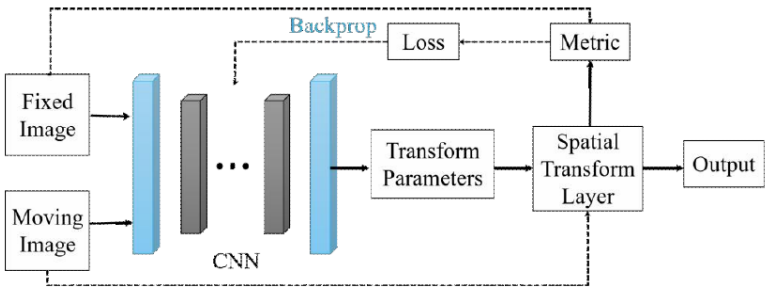


Figure 2.10 Unsupervised learning framework for image registration

Unsupervised image registration does not require labeled data for training, and this is similar to the idea of traditional medical image registration. The model directly learns what it needs to predict a best registration by minimizing the loss function of the fixed image and the deformed moving image. By backpropagating the errors, it optimizes the network parameters to estimate the deformation field. The trained

network model can predict the spatial correspondence between test images in one shot.

2.3.1.3. Generic design of unsupervised neural network for image registration

Figure 2.10 shows a generic design of unsupervised, end-to-end neural network for image registration. In this sub-section, we focus on the novel design of a registration network.

a) Fully convolutional neural network

Currently, most literature for deformable image registration are built on convolutional neural networks (CNNs) and fully convolutional neural networks (FCNs). FCN is an extension to CNN, replacing the fully connected layers to fully convolutional neural network [24]. Typically, the fully convolutional layer uses 1x1 convolution to convolve with the previous layer. The use of convolutional layer instead of fully connected layer resolve the issue that linear connection of fully connected layers may cause spatial information incomplete, which is especially important in registration as it focus every single pixel in the image.

The design of fully convolutional neural network, the network is similar to auto-encoder (AE) network design. It can be divided into two parts, encoding layers and decoding layers. Encoding layers are typically convolutional layer that extract local features that largely reduce size of features, while in decoding layers, the features will be used to reconstruct the feature map at various resolutions and use transposed convolutional layer to reconstruct the image. Normally each layer consists of convolutional layer, activation layer and pooling layer (Figure 2.11).

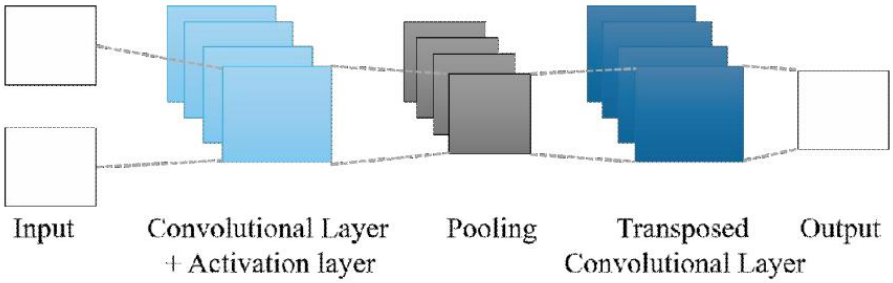


Figure 2.11 Fully convolution networks

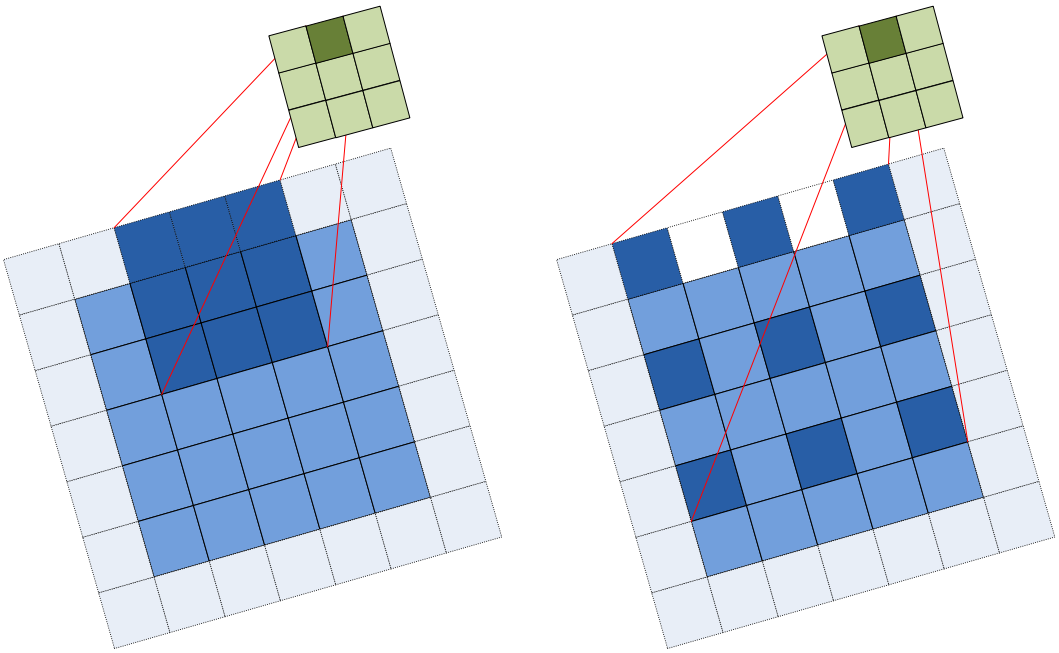
b) Convolutions

The convolutional layer contains several convolution kernels. The feature map between layers is extracted from these convolution kernels through the operation named convolution. The result of the convolution operation is connected to an activation function that limits the output for a purpose. The convolutional kernels, also known as neurons, are essentially sliding windows the moves at a certain step size by perform dot products on the local image area. The advantages of using convolutional layers are local receptive fields and shared weights and biases. Using local receptive fields ensures that only partial information in the image will be transfer to the next layers, which helps to learn only effective features. Shared weights and biases significantly reduce the size of trainable parameters, and it also help with overfitting problem.

The result from the convolution is called feature maps. The convolutional kernels may be different size in order to extract features from different resolutions. Larger convolutional kernel indicates a larger receptive field. But people start to use 3x3 neurons as building block design since VGGnet [25] proved that larger size neurons can be decomposed to a series of smaller size neurons while saving a lot of computation power. For example, a 5x5 kernel can be decomposed into 2 stacked 3x3 kernel with non-linearity in between. Additionally, this design reduces number of

parameters by 28% compared to a 5x5 kernel (#kernel_5x5=5^2=25, #kernel_3x3=2×3^2=18).

Dilated convolution [26], also known as atrous convolution, which can be viewed as convolution with holes, become famous in a segmentation network called DeepLab. The purpose of using dilated convolution is to increase the receptive field while reducing the number of kernels. Compared to conventional convolution, dilated convolution uses one hyper parameter called dilation rate, which refers to the spacing between the kernel convolves with. In particular, the dilation rate conventional convolution is 1. Figure 2.12 shows the comparison between conventional convolution and dilated convolution with dilation rate = 2. The receptive is indicated by the block included in the red line. One can see that dilated convolution with dilation rate =2 has a receptive field of 5x5 area while normal convolution only covers 3x3 area. Compared to VGGnet design, a single 3x3 dilated convolutional kernel serves as well as two stacked 3x3 conventional convolutional kernel in terms of receptive field.



Conventional convolution Dilated convolution

Figure 2.12 Receptive field of conventional convolution and dilated convolution

c) Activations functions

Neural networks mainly use convolution with weights and biases to capture features. This operation is linear, no matter how many layers of the neural network is stacked, the output is still a linear combination of inputs. Nowadays however, computer vision tasks, most samples are not linearly separable. To tackle with this problem, activation functions come to the play.

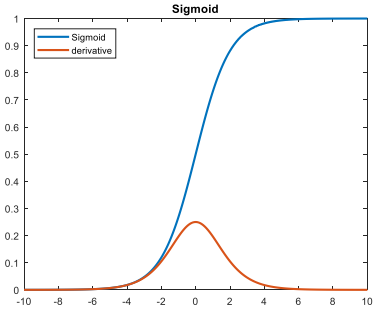
The activation function is important in all neural networks, as it introduces non-linearity between the neurons of the neural network. The linear features are non-linearly transferred through the activation function. Therefore, the output becomes a non-linear mapping of the input, which can better fit the data and improve the registration result of the model.

The most popular activation functions in compute vision related tasks are sigmoid, Tanh, ReLu and its variants.

- Sigmoid

$$sigmoid(x) = \frac{1}{1 + e^{-x}}$$

$$sigmoid'(x) = sigmoid(x)(1 - sigmoid(x))$$

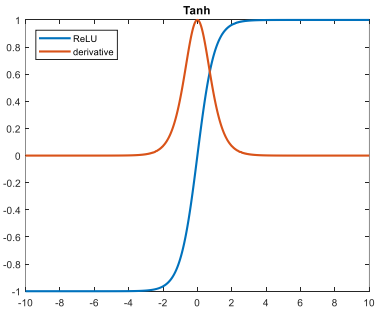


As errors backpropagate from the back to the front, the gradients keep decreasing. The gradient of Sigmoid function peaks in the range of [-3,3], while it is relatively flat in other regions. On one hand, this may cause a problem called vanishing gradient, which makes the network harder to train or even untrainable. On the other hand, sigmoid function uses exponential computation, which is computational heavy for computers. Especially for deep neural networks, it dramatically increases the training time. Additionally, the mean of sigmoid is not zero-centered. That is, the gradient of sigmoid will always be positive. The update of parameter will always be positive, causing a slow training speed.

- Tanh

$$tanh(x) = \frac{1 + e^{-xx}}{1 + e^{2x}}$$

$$tanh'(x) = 1 - \tanh^{2}(x)$$

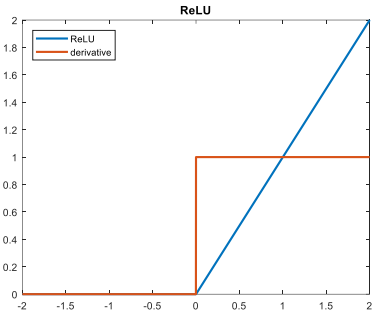


$tanh(x)$ is the revised version of $sigmoid(x)$, as it resolves non zero-centered problem of sigmoid function. Plus, it is steeper than sigmoid function in around 0, therefore it converges faster than sigmoid.

- Rectified linear unit (ReLU) and its variants

$$ReLU(x) = \max(0, x)$$

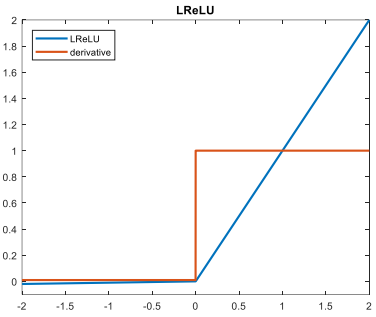
$$ReLU'(x) = \begin{cases} 0, x < 0 \\ 1, x \geq 0 \end{cases}$$



First of all, ReLU function is extremely simple that it speeds up the computation time by a large extent compared to those that involve exponential computation. However, as half of the function remains zero, an inappropriate learning rate may easily cause a large number of neurons to inactive and makes the network untrainable.

$$LReLU(x) = \max(\alpha x, x)$$

$$LReLU'(x) = \begin{cases} \alpha, x < 0 \\ 1, x \geq 0 \end{cases}$$



The design of leaky rectified linear units (LReLU) is an improved version of ReLU activation. It keeps its linearity in positive range while remains active (not zero) in the negative range. ELU resolves the problem of dead neuron in ReLU function. Thus, it becomes the design choice of our proposed network design for deformable image registration.

2.3.2. Proposed neural network design for deformable image registration

In the previous parts, we have discussed the main novel component used in our design. In the following, we will give a detailed description of the image registration network design in this thesis.

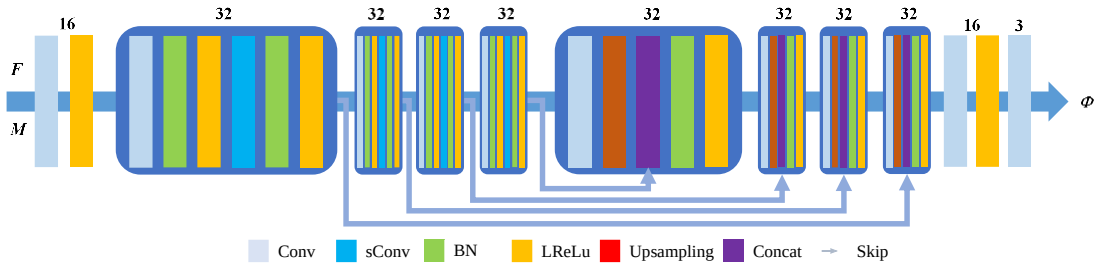


Figure 2.13 Proposed network for image registration

Inspired by design of residual blocks [27] and U-Net [28], we define a novel image registration network that can be used for 2D and 3D image registration with simple adjustment. The generator concatenates moving image M and fixed image F in the first place before feeding into the following convolutional layers. Similar to [14][28], we use skip connections to capture local and global anatomical features and ensure low level features can be propagated forward.

In the first half, each block consists of two convolutional layers with batch normalization (BN) and Leaky rectified linear unit. We use dilated convolution that is learnable in the second convolution to reduce spatial size in the feature map by a factor of two. The resolution of feature maps keeps decreasing until the it reaches 1/16 which composes high-level semantic features.

In the second half is symmetric to the first half, where the resolution gradually recovers to the original input size by the up-sampling layers in the convolutional blocks. Skip connections associate the early stage features with the current features, which makes them available for final alignment. It also facilitates gradients to flow back in the training stage.

Eventually, we use two convolutional layers to refine the prediction. Note that the last layer use linear activation on purpose since the displacement in the deformation field should not be restricted by the neuron outputs. It is also worth to mention that pooling layer is not used in the proposed network as pooling will easily led to loss of

information [29]. Please note that this image registration network will be used in the thesis and incorporates with the other network called discriminator, forming a generative adversarial net.

2.4. Chapter summary

In this chapter, the formulation of conventional image registration methods is given, along with an introduction and comparison on the key steps in image registration including spatial transformation, similarity measurement, optimization algorithm and interpolation method. A design overview of image registration network is presented, with a detailed discussion on the preprocess steps, the difference between supervised and unsupervised registration frameworks and the generic design of a registration network. Finally, a novel design of registration network is presented. This chapter demonstrates the ability to use neural networks for image registration, and the proposed image registration network will be served as the generator in the generative adversarial net design.

3. Attentional generative adversarial nets

A generic unsupervised learning framework for image registration and a novel registration convolutional neural network that builds on dilated residual blocks are introduced in chapter 2. However, the aforementioned method along with most current methods do not consider the reverse consistency of the registration. In other words, the resulted registration is asymmetric and may be unrealistic. Such methods see the image pair as individual image, and do not put any attention on the topological preservation during that may leads to a degrade on the deformation. To overcome these disadvantages, we apply perceptual cyclic loss in a generative adversarial setting, which is expected to support topological preservation by adding a cyclic constraint on the transformation and aims for a better similarity metric by incorporate a perceptual loss term in the objective function. Moreover, to improve the overall accuracy, attentional model is integrated in the network by focusing on the necessary features while suppressing the unrelated ones.

3.1. Generative adversarial nets and its variants

The concept of generative adversarial nets (GANs) was introduced by Dr. Goodfellow [30] in 2014 and quickly gained attention. GAN is originated from the two-player zero-sum game, where each player's gain is precisely balanced by his opponent's loss, such that the sum of gains is even out at zero. The two players are realized as the generator model (G) and discriminator model (D) in GAN.

3.1.1. Vanilla generative adversarial nets

The vanilla generative adversarial net [30] constitutes of one generator model and one discriminator model. The basic idea is to train a generator such that it generates image as realistic as possible, while the discriminator is trained such that it distinguishes whether an image is from the dataset (considered as real) or is produced

by the generator (considered as fake). As the core component, the generator is ultimately expected to produce images that are like a real image, by when the discriminator is no longer able to tell fake images from ones. As introduced by [31], the basic framework for vanilla GAN is shown in Figure 3.1.

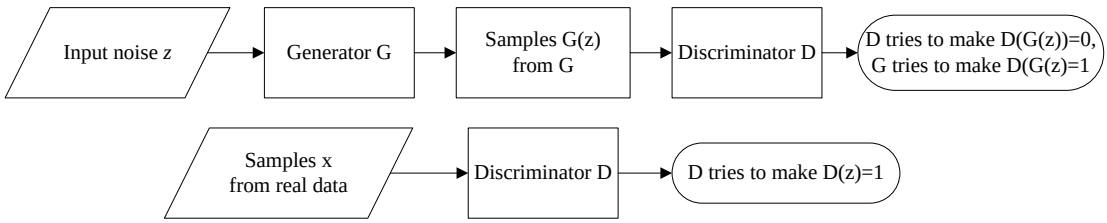


Figure 3.1 Vanilla GAN framework (two scenarios)

As indicated in figure 3.1, there are two scenarios take place simultaneously. In the first scenario, generator G randomly samples from a prior latent variable *z* and produce the output G(*z*), and then the discriminator take the fake sample G(*z*) created by the generator as input. In this situation, the discriminator tries to make fake sample D(G(*z*)) closed to 0 while generator tries to fool the discriminator by making its fake output D(G(*z*)) closed 1. The generator and discriminator will reach the Nash equilibrium where the generator can produce realistic output G(*z*) that has the same distribution as the real data sample *x* and the discriminator cannot tell G(*z*) from *x*. In the second scenario, discriminator only takes samples *x* from real data in training. In fact, half of the inputs to discriminator comes from the real sample *x* while the other half is created by the generator. Thus, the goal of discriminator is to ensure D(*x*) approaches 1. Denoting p_{data} as real data distribution and p_z as prior noise distribution, a vanilla GAN framework could be viewed as an min-max optimization process,

$$G^* = \min_G \max_D E_{x\sim pdata(x)}[\log D(x)] + E_{z\sim p_z(z)}[\log(1 - D(G(z)))]$$

In GAN, both the generator and the discriminator are defined as differentiable functions so that they both have trainable parameters θ_g and θ_d. Neural networks are popular in computer vision related tasks for their ability to capture spatial information in the image. Therefore, GANs modeled with neural networks have shown great power in many computer vision tasks, such as style transfer [32][33], super-resolution [34] and image generation [32].

3.1.2. Conditional generative adversarial nets

The vanilla GAN take random noise as input, and learns how to produce an output that has similar distribution as training data. This setting is unconstraint, and it would be considered correct as long as its generative content fits the distribution. However, it is more general to expect the generator can produce an output the meets the given information. The conditional generative adversarial nets (cGAN) [35] is an extension to the vanilla GAN with additional auxiliary information and alleviates the mode collapse problem to some extent.

cGAN takes two inputs, the noise *z* and the information *y*. *y* could be in any form in general, such as digit labels, texts, or images. The generator in cGAN is expected to produce the output conditional on this information *y*, as is shown in figure 3.2. Such process could be expressed as,

$$G^* = \min_G \max_D E_{x\sim pdata(x)}[\log D(x|y)] + E_{z\sim p_z(z)}[\log(1 - D(G(z|y)))]$$

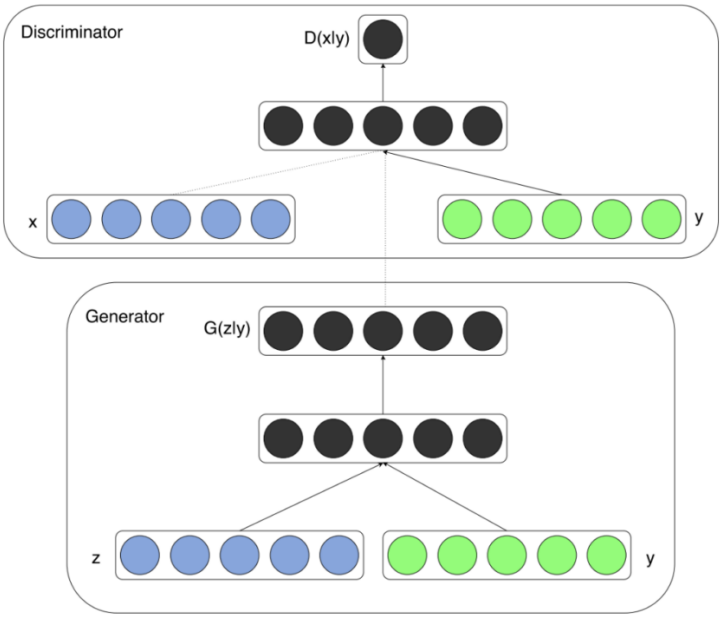


Figure 3.2 Conditional GAN

3.1.3. Image-to-image translation with GANs

Translation commonly refers to express the meaning or writing in another language. Accordingly, image-to-image translation refers to transform one image to another. For example, style transfer that could reconstruct an aerial image to a Google-map style image [32], super-resolution that could enhance the resolution of an image [34]. A popular GAN structure named pix2pix [32] is proposed to unify such works on image-to-image translation. Pix2pix works on pixel-level prediction, whose purpose is similar to image registration task mentioned in chapter 2.

The generator in Pix2pix use U-net [36] encoder-decoder architecture as backbone. The encoder down-samples and encode lower dimension features from input image, while the decoder decodes this information and reconstruct the desired image output. Skip connections are used between former parts and the later parts in the network to ensure the lower-level features are propagated all the way through the

later part and reduce information loss during down-sampling. With the objective of pix2pix is given by,

$$G^* = \min_G \max_D E_{x,y}[\log D(x|y)] + E_x\left[\log\left(1 - D(G(x))\right)\right] + \lambda L_1(G)$$

with an additional L1 distance loss $L_1(G) = E_{x,y}[\|y - G(x)\|_1]$ to ensure pixel-wise similarity. Note that noise vector *z* is no longer needed as the output image $G(x)$ is only generated from *x* since it is inefficient to introduce noise to increase the variety of generation in such a one-to-one translation task. Noise can be introduced in many forms in the training phases, for example pix2pix use dropout to introduce noise. In this thesis, we use a trick called label flipping which will be discussed in the robustness design in this chapter.

The training of pix2pix requires paired data, such as the pre-aligned image pair of aerial image and Google-map style image. This is difficult to collect and may require a lot of human labelling, not to mentioned that it is almost impossible in medical imaging. Inspired by dual learning, cycleGAN [4] incorporates adversarial loss and cycle consistency loss to reduce the need for paired training data when training the network. The dual learning in cycleGAN refers to two structurally identical generators that will be trained at the same for forward translation $G: X \to Y$ and backward translation $F: Y \to X$ respectively, and two structurally identical discriminators D_X that distinguish *x* from F(*y*) and D_Y that distinguish *y* from G(*x*). Except for adversarial loss, cycleGAN introduces cycle consistency loss, which is defined as,

$$L_{cyc} = E_x\left[\left\|F\left(G(x)\right) - x\right\|_1\right] + E_y\left[\left\|G\left(F(y)\right) - y\right\|_1\right]$$

The formulation above states that the original image *x* is expected to be alike with itself after a forward transform $G(\cdot)$ and a backward transform $F(\cdot)$ or vice versa. As two pairs of generator and discriminator are involved in this process, the adversarial loss comes in pairs as well,

$$L_{xy} = E_y[\log D_Y(y)] + E_x[\log(1 - D_Y(G(x)))]$$

$$L_{yx} = E_x[\log D_X(x)] + E_y[\log(1 - D_X(F(y)))]$$

Let $L_{adv} = L_{xy} + L_{yx}$, the total loss function of cycleGAN is,

$$L = L_{adv} + \lambda L_{cyc}$$

while the optimal solution is given by,

$$G^* = \min_{G,F} \max_{D_X,D_Y} L$$

3.2. Design of generative adversarial network for deformable image registration

3.2.1. Structural design of generative adversarial network

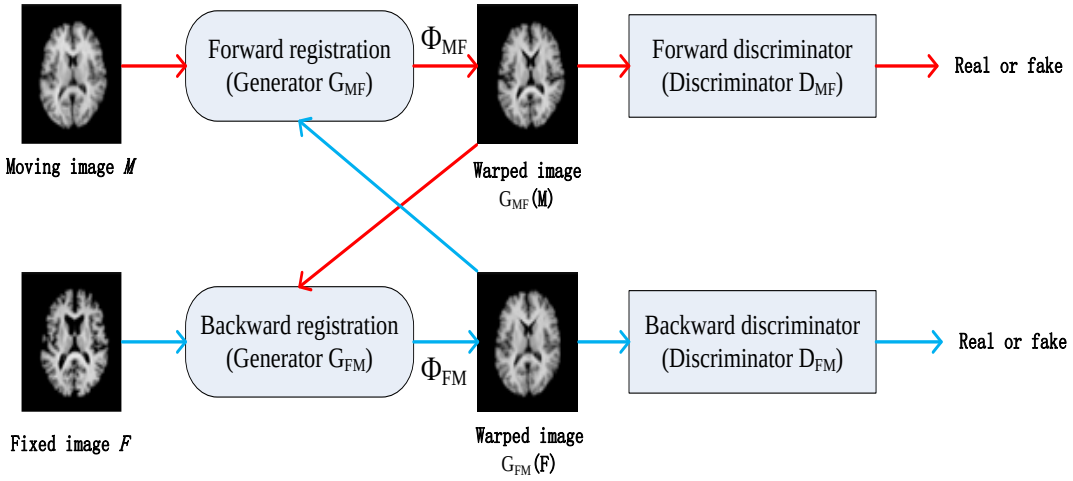


Figure 3.3 Proposed generative adversarial network

The general design of the proposed generative adversarial network shown in figure 3.3. The first one (indicated by red arrows) is used to register moving image to fixed image, just as normal image registration network G_{MF}: *moving->fixed*, while the second one (indicated by the blue arrows) is used to reversely register fixed image to moving image G_{FM}: *fixed->moving*. Each of them has their corresponding discriminators D_{MF} and D_{FM} that are used to distinguish the corresponding generated

image from real samples. Besides going under tested by the discriminator, the generated image will also pass to be the input of the other registration network.

Although in such a dual registration setting, the concept of moving image and fixed image have been weakened, we still use moving image and fixed image for convenience. Inspired by [4], the proposed structure also consists of two identical sub-GAN models that formulate a cyclic loop. The cyclic loop refers to two opposite registrations, G_{MF}: *M*->*F* and G_{FM}: *F*->*M*. Intuitively, we expect them to be the same or practically similar during a cyclic loop:

$$M \rightarrow G_{MF}(M) \rightarrow G_{FM}(G_{MF}(M)) \approx M$$

$$F \rightarrow G_{FM}(F) \rightarrow G_{MF}(G_{FM}(M)) \approx F$$

The cyclic loop is essentially a reconstruction of the input image. The discriminator D_{MF} is expected to distinguish the generated image $G_{MF}(M)$ and $G_{MF}(G_{FM}(M))$ from real sample F, while the discriminator D_{FM} is expected to distinguish the generated image $G_{FM}(F)$ and $G_{FM}(G_{MF}(M))$ from real sample M. The whole structure as depicted in figure 3.3 makes up a dual loop that the encourages it to predict a closed-loop transformation.

3.2.2. Generator design

As discussed in chapter 2, a generic registration network accepts a pair of moving image and fixed image, and outputs the deformation field that is used to warp the moving image to fixed image by a spatial transform network. The details of the generator used in this thesis have been discussed in chapter 2.3.2.

Technically speaking, the generator network for image registration can build upon any existing registration network. However, the dual design that utilize 2 generator and 2 discriminator doubles the memory usage in training. Therefore, limited by the GPU memory, we design the network that only contains 4 encoder

blocks and 4 decoder blocks. With the current design, the GPU memory usage has already taken up to 93.55% on a 16GB NVIDIA Tesla V100. It is worth mentioning that the performance could be improved by replacing a more efficient registration network. Figure 3.4 below shows a practical design of the registration network that could fit in current setting. The generator takes a pair of fixed (F) and moving image (M) as input, followed by a concatenate layer. The concatenate layer basically linked a pair of $H \times W \times D$ image together and fuse them as an $H \times W \times D \times 2$ volume data to feed into the neural network. Since this is not a learnable layer, it is not depicted in Figure 3.5. The learnable parts are encoder and decoder. The decoder consists of 4 successive convolutional layers (dark blue colored), while the decoder consists of another 5 successive convolutional layers (yellow colored). The number in the center of rectangles show the numbers of channels used in the layers while the number underneath depicts the size compared to the input image. The convolutional layer in light blue is a refined layer that introduce more non-linearity before the final convolutional layer that colors green. The solid arrow indicates a common connection between two successive layers in the neural net, while the dotted arrow denotes the skip connection between two corresponding layers.

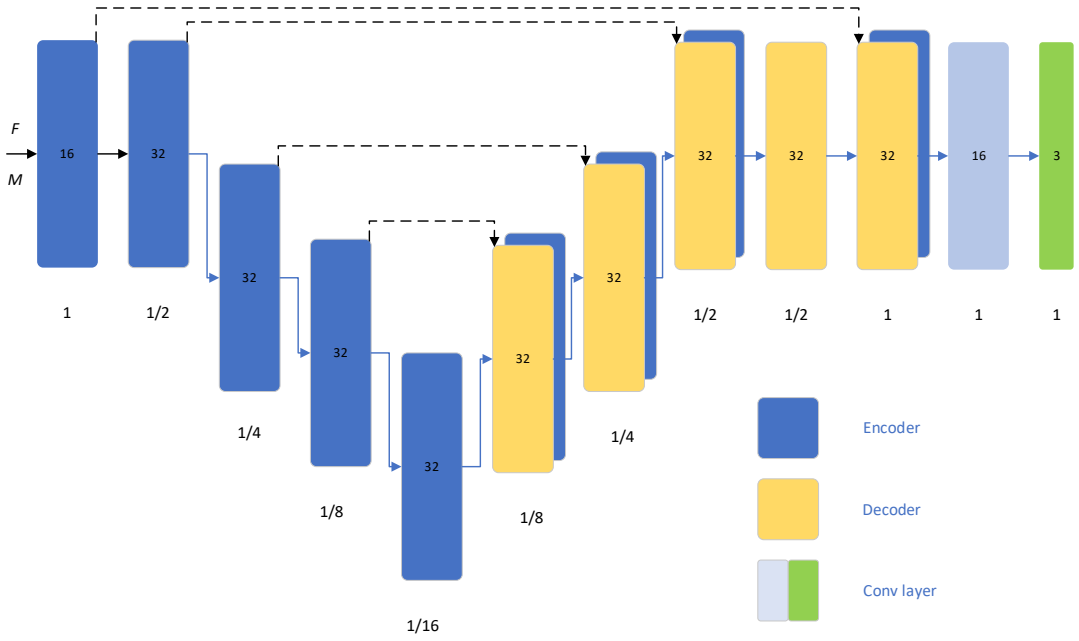


Figure 3.4 Generator design

3.2.2.1. Encoding part

In each encoder blocks, convolutions are followed by a leaky ReLu activation. The size of the convolution kernel is 3x3x3. The stacked convolutional layers acquire hierarchical features that are important for the registration task. The convolution used in the encoder is dilated convolutions. The advantages of using dilated convolutions have been discussed in detail before (Chapter 2.3.1.3). The main purposes of using dilated convolutions are to maintain the receptive field while saving decent amount of computing power. This is extremely helpful in our design since the cyclic setting consumes a lot more computing power than common CNNs or GANs.

It is also worth mentioning that instead of using pooling layers, we set the strides of convolution layers to be 2 following the most recent trend [37][28][38]. The usage of max pooling in neural networks is simply historical [39]. The reason for early literatures is to decrease the spatial dimensions of feature map. For instance, 2x2 max-pool layers only record the feature that is the maximum of a 2x2 neighborhood and lose the others, resulted in a feature map shrink to be only 1/4 of the original

feature map size. However, pooling is a fixed operation that is not trainable by the error backpropagation. Therefore, we remove the pooling layers and use convolutions with strides=2 to reduce the size by half in the encoder. The consecutive convolution captures coarser representations and learns higher level representations of the inputs, until the minimum size of the feature map is reached. The minimum size in the setting is $10 \times 12 \times 14$, 1/16 of the input image ($160 \times 192 \times 224$). It is not recommended to be anyway smaller since it would be more difficult to recovers size in the decoder. Additionally, deformation registration focuses on local difference of the images, thus global representation of the inputs may be less significant for the task.

3.2.2.2. Decoding part

In each decoder block, upsampling layers are used to enlarge the feature map, followed by the convolutions with stride=1 and leaky ReLu activation. Since registration is a pointwise (point-to-point) prediction task, the spatial dimension of the feature map is gradually increased and finally becomes the same as the original input. After each upsampling, there is a skip-connection between the previous layer that has the same size. Since the upsampling layer utilizes linear interpolations that only use very limited neighborhood information, skip connections is of great importance as it directly pass the features learnt in the encoding procedures to the decoder.

The stack of decoder blocks gradually recovers the spatial dimension back to the original size, which make it possible to do precise point-to-point alignment. With the help of skip connections, decoder decodes the coarse features to a finer scale. Kindly note that before the final output layer (green colored in figure 3.4), there are two layers that aims to introduce additional non-linearity while remains the same spatial dimension of the feature map, which follows the original design of U-net [36]. Finally, the depth of the output layer is 3, which is consistent with the shape of the

deformation field of a 3D registration task $(H, W, D, 3)$. The 3 vectors represent the 3-dimensional movement (x, y, z) from the moving image to the fixed image.

3.2.3. Attentional Mechanism integration

3.2.3.1. Intuitive of attentional mechanism

When confronted with a large amount of information, human brain is always able to grab the most important part and concentrate on it, either using hearing system or vision system. For example, when reading a newspaper, instead of reading it from top to bottom, we pay attention to only certain things, such as bolded title or the vivid figures. Similarly, the attention mechanism in neural networks aims to simulate this human ability by adding weights to each feature and selecting only key information to process. In short, attention mechanism is about how to automatically generates a weight mask for the feature (as shown in figure 3.5).

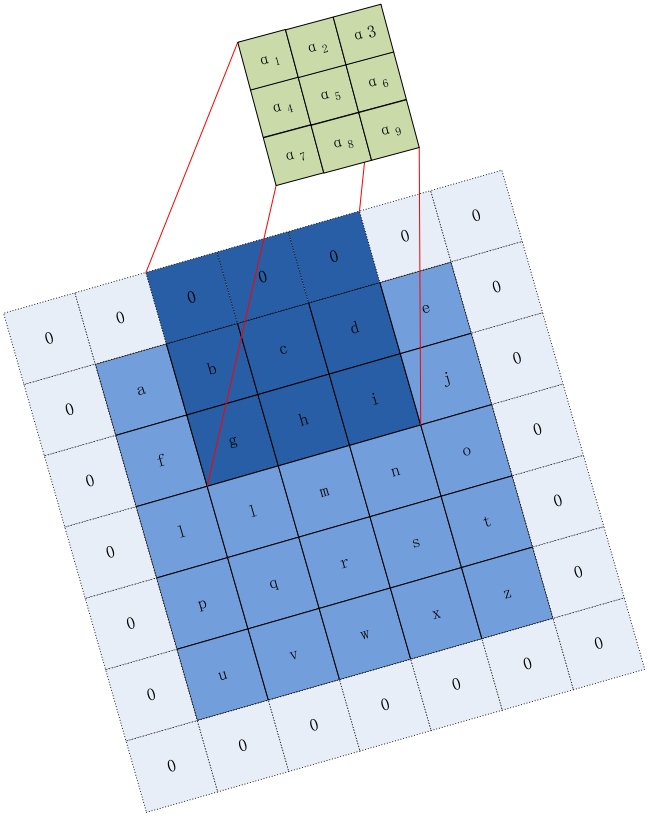


Figure 3.5 Attention as a weight mask on feature map

It is more intuitive to apply attention on natural language processing (NLP) tasks. Due to the fact that speech and text is sequential, the attentional mechanism is usually used in recurrent neural networks (RNNs), with applications in natural language processing such as machine translation [40]. The word sequential means that it would change or even lost their meaning if the sequence is changed, especially in machine translation task. Xu et al. [41] is one of the first works to introduce attentional mechanism in image processing, where they use attentional mechanism to establish a relationship from a word to a specific area in an image to export the image captions. Almost in the same time, teams from Google applied attentional mechanism on RNN model for image classification [42] and machine translation [43], after which their impressive results draw the eyeballs worldwide.

3.2.3.2. Attention gate

Practically, the attentional mechanism is typically used as an additional module that builds a branch from the backbone network, and then fuses information with the backbone network after passing the attentional mechanism module. The attentional module, also well-known as attention gate, can select certain parts of the input information and learn to assign different weights to each pixel of the feature map in a way that generates a weighted map. It serves to as feature filtering and reinforcing certain features of the input feature map.

In terms of outputs, attentional mechanism can be classified into soft attention and hard attention [41]. Hard attention, also refers to stochastic hard attention, computes a one-hot vector for the hidden layers with argmax. This is fast inference, but typically untrainable since it is not differentiable, errors cannot be backpropagated through the network. Soft attention, which computes the probability of each hidden

neurons, is differentiable and trainable. In short, soft attention takes coefficient from [0,1] while hard attention takes coefficients either 0 or 1.

In terms of how the attention is learnt, the attention model can be grouped into focus attention and saliency attention [41]. Focus attention usually gains its coefficient from a supervised task, such as classification and segmentation, where the targeted output is deterministic and unique. The attention model is then learnt from the ground truth and subjective to the supervised information. For instance, in cat or dog classification task, if ear shape has large influence on recognition accuracy, then features on how to detect ear shape would gain high coefficient. On contrast, saliency attention, also known as passive attention, is more self-motivated. The attention coefficient is learnt by how much it is triggered by the tasks. For instance, in a task to find what are essential to human, there might be thousands of thing human cannot liver without. If human cannot live without air, then the factor "air" may gain a high coefficient.

The work in this thesis to develop an unsupervised method for image registration, thus we use soft saliency attention. As discussed in chapter 1.3.2, although image registration can be formulated as supervised task, but more commonly it is formulated as a supervised task since it is difficult to obtain the ground truth. It is worth to mention that we also provide performance analysis with ground-truth in Chapter 4.2. Given such setting, attention mechanism for our registration network can be formulated as,

$$y = f(x) \odot x'$$

where $f(x)$ is the attention coefficient depending on the corresponding feature map *x'*, and $\odot$ is dot product of coefficient $f(x)$ and feature map *x*. $f(x)$ could model as arbitrary methods, such as hidden layers in RNN network [40]. Previously, the

attention is learnt through multi-stage network [44][45], where people add a separate localization network upon the current network and the localization gradually learns to suppress features that corresponds to irrelevant background regions. However, the training of the localization network is not easy since there is no guaranteed training strategy. Later on, Oktay et al. [38] proposed a attention gate (AG) for his attention U-net model that could be integrated in any U-net like networks without the need of having sub-networks. as shown in figure 3.6.

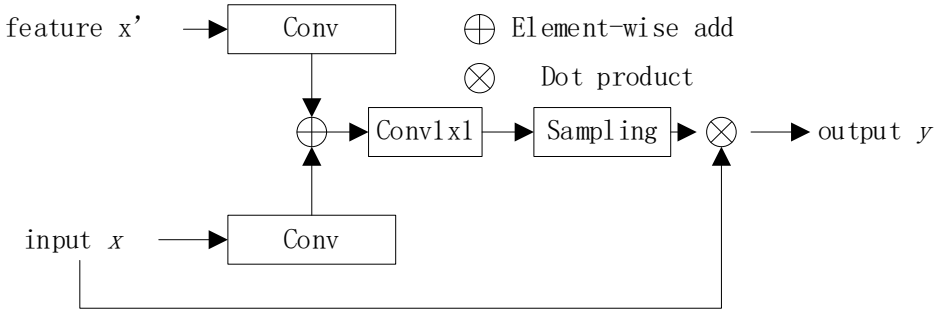


Figure 3.6 Attention gate

The current feature *x* first makes element-wise addition with the corresponding feature map *x*', which is typically the symmetric part in the encoder-decoder setting (or arbitrary feature map that requires additional sampling to make them share the same shape). The use of channel-wise 1x1x1 convolution can reduce the need for computing power as well as alter the output channel to map the chosen feature *x*' to the same shape of the input feature map *x*, which yields the attention coefficient. The output of attention gate is the dot product of the attention coefficient and the input *x*, typically followed by an activation function that limit its output to a desired range. The output y is the attention coefficient ranged in [0, 1] with the help of sigmoid, which is expected to stress on the salient image regions that have significant influence on the final registration results.

3.2.3.3. Attentional mechanism integrated generator

Thanks to the work of attention U-Net [38], attentional mechanism can be integrated into our generator network without the need of a separate localization network. The overall framework is similar to the designed shown in figure 3.5. The encoder part remains unchanged, while we integrate the attentional mechanism by replacing the decoder blocks with attentional decoder blocks. Figure 3.8 shows the framework of the attentional generator.

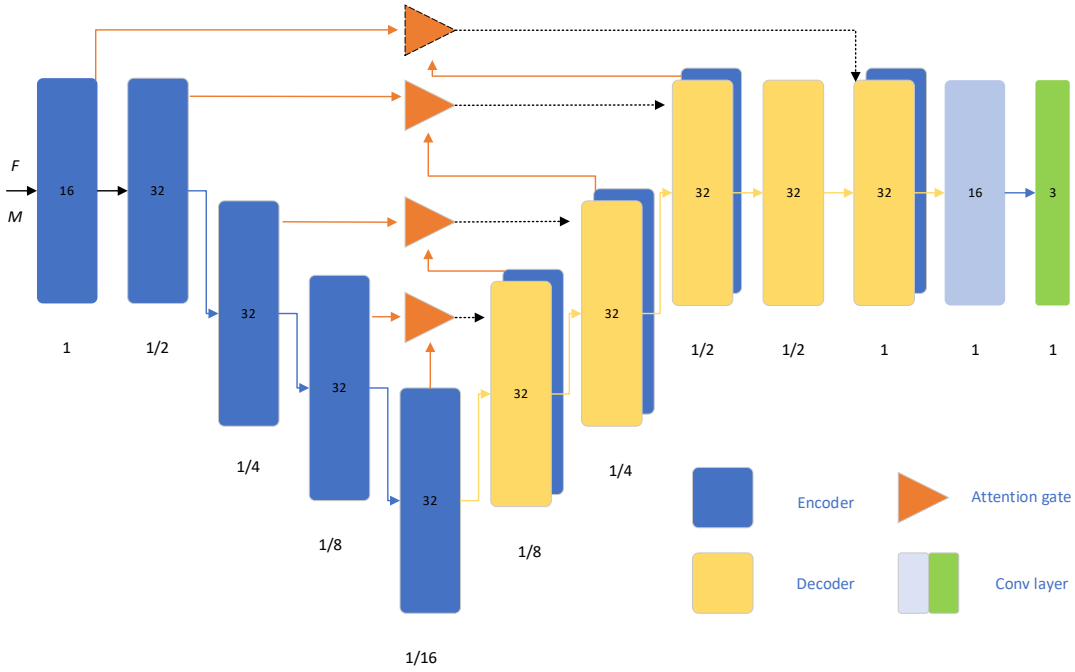


Figure 3.7 Attention integrated generator

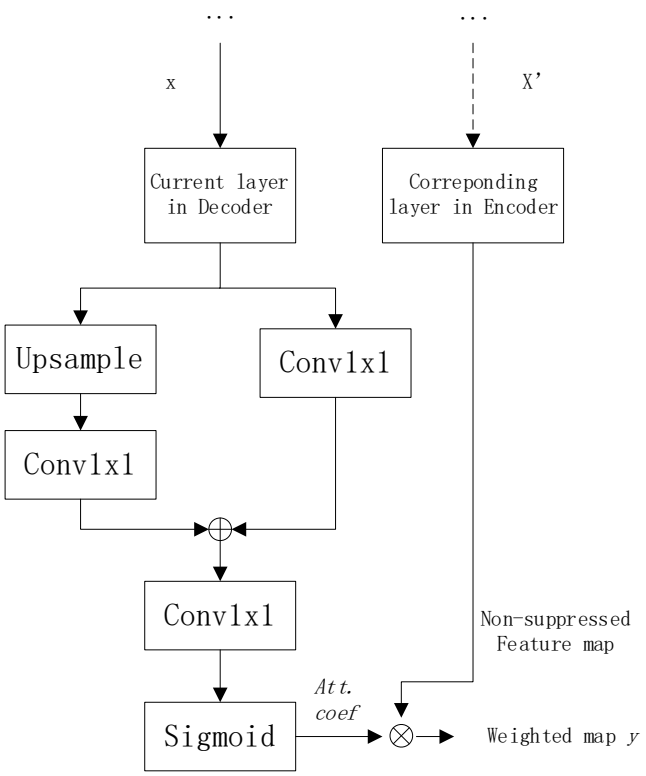


Figure 3.8 Attentional gate in the generator

The change in the generator mostly reflects in the additional attentional gate in the decoder part. The decoder fuse multi-resolution features with the attentional gate, which reassigns the attention coefficient on these features. A higher coefficient indicates that the features are significant to the registration while a lower coefficient indicates the neurons is less efficient for certain regions and should be suppressed. The incitement and the suppression are automatically learnt during the training procedure.

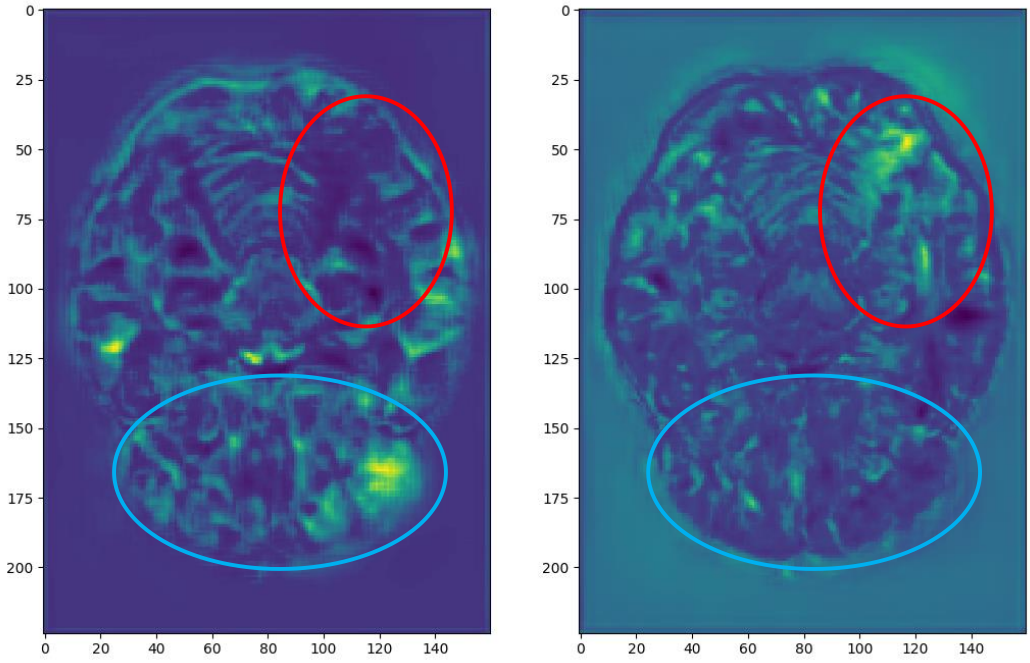


Figure 3.9 Neuron responses with attentional mechanism

Two sample neuron responses from the attentional generator are shown in figure 3.10. In particular, image on the left is from Deconv4_15 and the image on the right is from Deconv4_16 after the attention gate. Since the convolution is 3D convolution, we show the middle slice from the coronal plane and zoom-out to the same shape of the input image. Highlighted area indicates a higher filter response. The red-circle area refers to the cerebellum white matter area in the input images, while the blue-circled area indicates the cerebral cortex. Filter #15 in the deconv4 tends to put more attention on the cerebral cortex while filter #16 tends to put more attention cerebellum white matter area. Figure 3.10 suggests that attention mechanism forces the neurons to learn to focus only on a subset of potential target structures and proves the effectiveness of integrating attentional mechanism.

3.2.4. Discriminator design

3.2.4.1. Network design

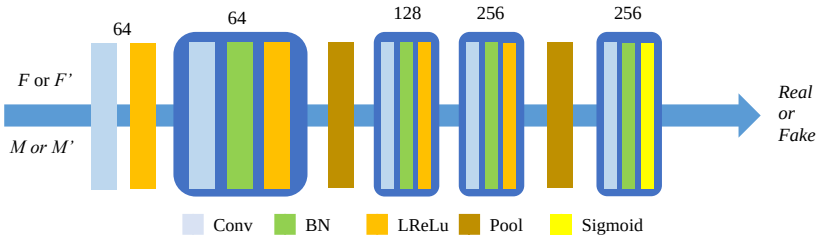


Figure 3.10 Discriminator network

The architecture of discriminator is shown in figure 3.11 above. The input of the discriminator is either the samples from real data *F* and *M* or samples from the output of the generator *F'* and *M'*. The overall structure in the discriminator is built upon the building blocks as well.

Each building block consists of a convolutional layer, batch normalize layer and an L-ReLU activation function. Except for the convolution in the first block operates at kernel size=3 with strides=2, all other convolution operates at kernel size=3 with strides=1. Pooling layer are max pooling with step size =2. L-ReLU activation is used in all blocks except for the last output layer.

The output of the discriminator is typically 0 indicating generated samples (fake) and 1 indicating samples from real data (real). To exploit the maximum capability, we make some adjustment compared to the convention discriminator design, namely patch discriminator design to ensure pixel-level accuracy, soft labeling, and anti-noise design to improve the robustness. These will be introduced in the following paragraphs.

3.2.4.2. Patch discriminator

Patch discriminator design, also known as PatchGAN is first used by [32]. Instead of taking the whole image as input and only estimate one single number from

it, the discriminator receives $N \times N$ blocks that are cropped out of the image and determine whether these blocks are sampled from real data. Such setting makes the final result in a vector form, and the decision is determined by the average of the scores from all blocks.

The traditional way that only estimates one single number could be viewed as the overall judgement of the image, while patch discriminator outputs a score vector with size $N \times N$, typically implemented by using convolution with strides=2 or pooling layer with step size=2. Each number in the vector indicates a patch of the input image. As shown in figure 3.12, the size of a projective patch is the field of view (FoW) of the discriminator.

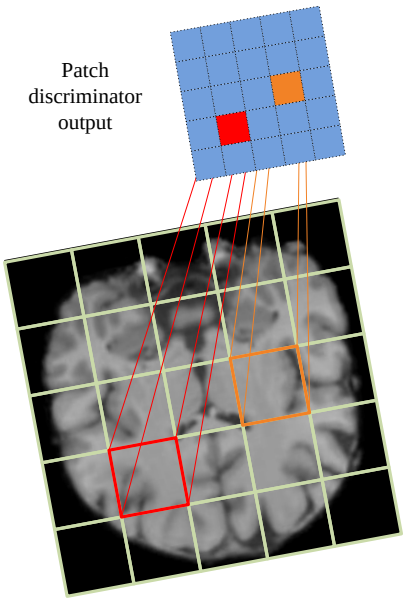


Figure 3.11 Field of view in the patch discriminator

3.2.4.3. Robustness designs

The robustness design mainly considers two adjustments: soft label and label flipping.

Soft label refers to replace hard labels (0 for fake, 1 for true) to a smoother boundary with a value slightly less than 1 for true labels, such as 0.9. As discussed in

[31], this could prevent fierce inference behavior in the discriminator. The discriminator will be penalized if it is too confident to predict extremely large logits that will be converted to a probability approaching 1 for some inputs. Simultaneous, this helps the discriminator to learn more efficiently as it keeps the activation function operates at a more active range (solution to $sigmoid(x) = 0.9$ is 2.197, active range of sigmoid is [-3,3]). It is important to know that the label of fake samples should never been smoothed as the discriminator will reinforce the incorrect behavior of the generator [31].

Another technique used is occasionally flip the labels to introduce noise *z*. Traditional GAN provides a noise vector *z* to increase the variety of outputs [32]. However, it is found to be ineffective since after some epochs, the generator simply learned to ignore the noise. Thus, we set the probability of randomly flipping the label as 0.9, which we found to be most effective in our experiment.

3.3. Objective function

Objective function, also known as loss function, is an estimation of distance between the network prediction $\hat{y}$ and the true value y, which can be backpropagated through the network to update the network parameters. In this work, the objective function consists of three parts that are designed for different purposes. In particular, we use a combination of local cross-correlation (CC), cyclic loss and perceptual loss during training.

3.3.1. Statistical similarity constraint

Cross-correlation (CC) is a statistical similarity measurement in many registration methods [23][14]. As shown in chapter 2, the computation of CC requires time-consuming operations such the squared error for all pixels. This may slow down the

training process, especially for the proposed dual training scheme. Instead, we usually use a fast compute of cross-correlation in practice, named local cross-correlation [14]. Let $\tilde{F} = \frac{1}{n^2}\sum_{p_{ij}} F(p_{ij})$ where p_{ij} goes over an n^2 space around center point $p(i,j)$. Likewise, $\widetilde{M'} = \frac{1}{n^2}\sum_{p_{ij}} M'(p_{ij})$ denote the local mean around of the center point p in the transformed image.

$$local\ CC = \frac{\sum_{i=0}^{I}\sum_{j=0}^{J}[F(i,j) - \tilde{F}(i,j)]\left[M'(i,j) - \widetilde{M'}(i,j)\right]}{\sqrt{\sum_{i=0}^{I}\sum_{j=0}^{J}[F(i,j) - \tilde{F}(i,j)]^2}\sqrt{\sum_{i=0}^{I}\sum_{j=0}^{J}\left[M'(i,j) - \left[M'(i,j) - \widetilde{M'}(i,j)\right]\right]^2}}$$

Just like the normal cross correlation, a higher local CC indicates a better registration. The loss of statistical similarity is then given by the negative of local CC:

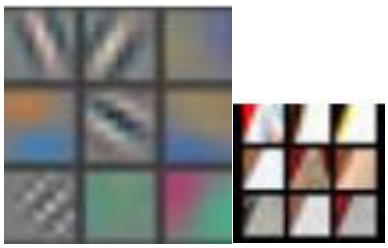
$$L_{sim} = -local\ CC$$

3.3.2. Machine vision constraint

How machine perceives images is different from how human interprets them. For example, cross correlation is a commonly seen metrics to quantify the level of alignment in image registration. However, it should be helpful to train the network with additional perceptive information from the artificial neural network itself. The way that simulates how machine perceives images can be called machine vision. The technique to apply such loss during training is called perceptual loss.

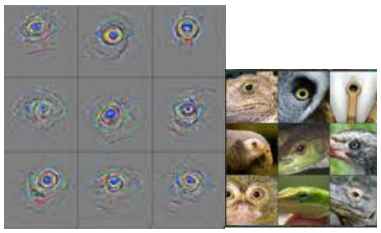
Perceptual loss is first introduce for style transfer mission in [46] where the author introduce leant features into image synthesis tasks. It is proven that texture features are encoded in the trained layers. In the later work from [47], it shows that lower level features, such as edge and color information, are stored in the lower level layers; while higher level features, such as certain patterns and combined shape, can be memorized by higher level layers.

As a naïve example, as VGG16 network is trained on a COCO dataset (a large scale image dataset of natural image), some of the neural activation is shown in the following [48]. One can see that usually neurons at earlier layers fire to low-level features while the neurons at the back layers fire to more specific or high-level patterns. Lower-level features is closer to normal filters that are artificially designed, as they are intuitively plausible and easy to define mathematically. But higher-level filters, which propagate through many neurons and layers could find complex patterns that are related to the tasks. For example, figure 3.13 below shows some feature activation at different layers from a trained AlexNet [2],



Feature activation at 2nd layer

(low-level features)



Feature activation at 5th layer

(high-level patterns)

Figure 3.12 Exemplified features in AlexNet

Here we propose to use U-net [28] as our "perception" to the machine vision. U-net is commonly used fully convolutional architecture in biomedical image segmentation. There are mainly two part of U-net architecture, namely encoder to extract features and decoder for to recovers the resolution for medical image segmentation. Compared with the FCN [24] which simple adds up the features of different level, U-net introduce a new feature fusion technique by concatenating features from different level to make a "wider" feature maps for later convolutional layers. Also, the long skip-connections from the first few layers to the last few layers allow low-level and high-level feature to be combined together for a better prediction.

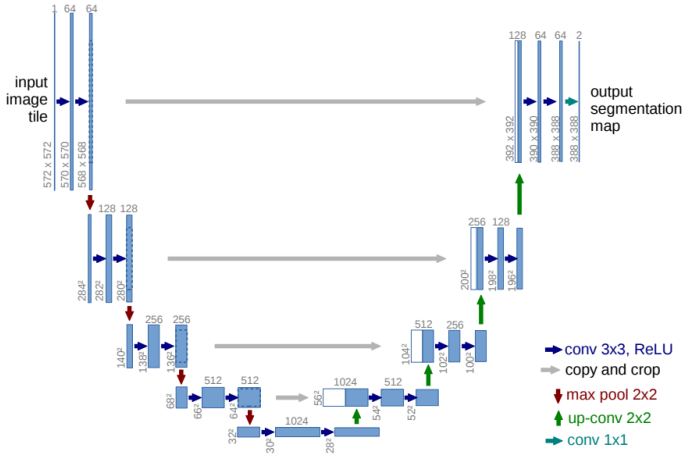


Figure 3.13 U-net architecture [36]

For these reasons, U-net is the best candidate feature extraction for medical images and thus we use the difference between the activation map of fixed image and registered image as our perceptual loss. In other words, perceptual loss is defined as feature distance of a pretrained network.

Specifically, denoting $\phi_i(x)$ as the feature map of ith layer of a pre-trained network, given the input image x, if the ith layer has a feature map of shape (H_i, W_i, D_i, C_i), the perceptual loss of essentially the Euclidean distances between some chosen feature maps given fixed image F and registered image $G(F, M)$,

$$L_{Perc} = \frac{1}{H_i W_i D_i C_i} \|\phi_i(F) - \phi_i(G(F, M))\|_2$$

We chose U-net as the pre-trained network to compute perceptual loss. To fit the perceptual network in the limited GPU memory, we simplified the original U-net structure [28] as follows to speed up the training. The number of channels in the layers is halved; the maximum depth of layers is set to be 256. The memory efficient perceptual network is shown in figure 3.15. Kindly note that blue block in figure 3.14 represents the feature map with its depth shown above, while blue block in figure 3.15 is a convolutional layer with the number of channels shown on top, so that the legend in this thesis is consistent.

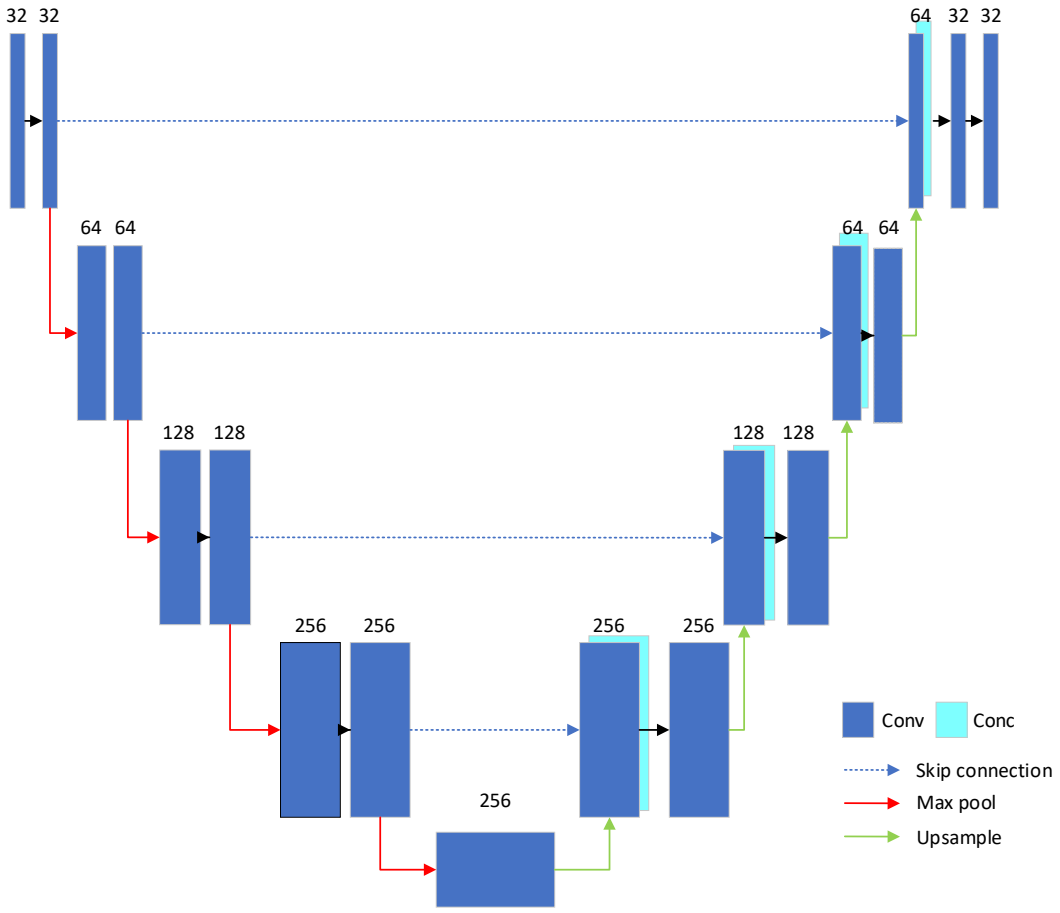


Figure 3.14 Memory efficient U-Net

Since the original U-Net structure is too big to fit in memory, we cannot use transfer learning to load the pre-trained weights in this memory efficient U-Net. Therefore, we trained this U-Net on the multi-modal brain tumor segmentation challenge (BraTS19) dataset*. Since the shape of scans various from each other, we crop or pad the data volume to 160x192x224 to be consistent with our data. We initialized the light U-net with BraTS19, and then fine-tuned on our own scans.

* BraTS19 dataset: https://www.med.upenn.edu/cbica/brats2019/registration.html

BraTS19 provides brain tumor MRI scans of modality from T1-weighted, post-contrast T1-weighted, T2-weighted and T2 fluid attenuated inversion recovery (FLAIR) from different institutions. Labels of the tumor are annotated by experienced neuro-radiologists, which comprise 3 different categories, namely whole tumor, tumor

core and enhancing tumor core as described in [49]. The data has been pre-registered to the same anatomical template, interpolated to the same resolution (1 mm^3) and skull-stripped, whose preprocessing is consistent with ours as discussed in Chapter 2. Top tumor core segmentation accuracy on this dataset can go up to around 0.82~0.85. For example, Wang et al [50] proposed a multi-stage cascade CNN for the task, which contains 3 subnets for to detect whole tumor, tumor core and enhancing tumor core respectively. The reported average dice score is 0.8173. Chen et al. [51]used a U-Net like network and replace the convolutional block with inception modules [52], yielding an average score of 0.8243. Compared to their customized network for the tumor segmentation task, our modified U-net trained on T1 image only got on an average dice score of 0.7674, which is slightly lower than these two methods. The main purpose of training such a light U-net is to compute the perceptual loss of image pairs, the performance is acceptable in this sense.

To find out the most useful layers, we prepare 10 validation pairs of unregistered scans and registered scans and feed them the pre-trained network to compute the feature distance.

Layers	unregistered pair diff.	registered pair diff.	feature map shape	Abs. Diff. (|%|)
ϕ_{1_2}	1220686.2	1232070.8	(160,192,224,32)	0.0092
ϕ_{2_2}	535796.560	528814.440	(80,96,112,64)	0.0132
ϕ_{3_2}	71379.930	66672.470	(40,48,56,128)	0.0706
ϕ_{4_2}	1932.306	2031.912	(20,24,28,256)	0.0490
ϕ_5	315.193	297.586	(10,12,14,256)	0.0592
ϕ_{6_2}	1796.238	1272.645	(20,24,28,256)	0.4114

ϕ_{7_2}	64126.746	50100.477	(40,48,56,128)	0.2800
ϕ_{8_2}	93984.445	34050.625	(80,96,112,64)	1.7601
ϕ_9	11220939.3	10568027.2	(160,192,224,64)	0.0618

Table 3.1 Feature difference between unregistered pairs and registered pairs

Table 3.1 shows the feature distance between unregistered pairs and registered pairs. The memory efficient U-Net shares the same framework with the original U-Net. Both of them have 9 convolutional layers, and in each of layer there are 2 successive convolution blocks. For example, ϕ_{i_2} denotes the second convolutional output in the i^{th} layer, which is connected to the next i+1 convolution layer. The pair difference (abbreviated to diff.) is computed as the sum of difference of the filter response on the corresponding layer, absolute difference is given by $\left|\frac{\text{registered pair diff.}-\text{unregistered pair diff.}}{\text{unregistered pair dif.}}\right|$, which indicates how much these two values different from each other. From the table, one can tell the most significant feature difference can be obtained by ϕ_{8_2} with shape of only 1/2 size of the input volume. In other words, if the network can be trained to minimize the feature distance between the moving and fixed image pairs, it also minimizes the machine perceptual difference between the two. The literature also shows that shallow features from the very beginning layers often produce results that are hardly distinguishable from the input image [53], indicating lower-level features are less useful than higher-level features, which can achieve a better image reconstruction result and are better choice for perceptual loss function. Our results are consistent with the finding, where higher level feature maps are more useful in this case. Therefore, we chose feature map ϕ_{8_2} from the pre-trained U-Net as our perceptual loss.

3.3.3. Topology preservation constraint

The majority of either conventional algorithms or learning-based algorithms did not put focus on generating a symmetric transformation. There are some work trying to achieve more realistic deformation in the literature. Typical solutions include estimate diffeomorphic transformation, reinforce anti-folding in the deformation field and cyclic training techniques. For example, Dalca et al. [54] use a CNN to learn multivariate Gaussian distributions that forms the velocity fields in the training data, and samples from the learnt distribution to compute the diffeomorphic transformation in the inference phase. Zhang et al. [55] proposed an ICNet to prevent folding in the deformation field in the deformation field by directly applying penalty on the gradient of the folding voxels.

Another way to achieve more realistic deformation is to use the discriminative ability to distinguish bad transformation, which is proposed in this work. For example, Fan et al. [15] proposed an adversarial approach for brain MRI image registration, where the conditional discriminator was trained to tell whether an image pair of warped and fixed images was well-aligned. Mahapatra et al. [16] designed a generative adversarial network that estimates multimodal registration field on retinal fundus images and fluorescein angiography (FA) images.

Typically, these methods do not estimate the reverse transformation. This might introduce a biased registration result on the choice of the target domain [55]. In this work, we encourage inverse consistency by using cyclic training as described in cycleGAN [4]. Preserving inverse consistency tackles the problem by simultaneously estimating forward and backward transformation with two identical subnetworks pairs (G_{MF}/D_{MF}, G_{FM}/D_{FM}). The dual network design is shown in figure 3.15, where G_{MF}/D_{MF} predicts forward transform register moving image to fixed image like

normal registration, while predicts G_{FM}/D_{FM} backward transform from the fixed image to the moving image. In the most ideal situation, the forward and backward transform is expected to be the inverse of each other.

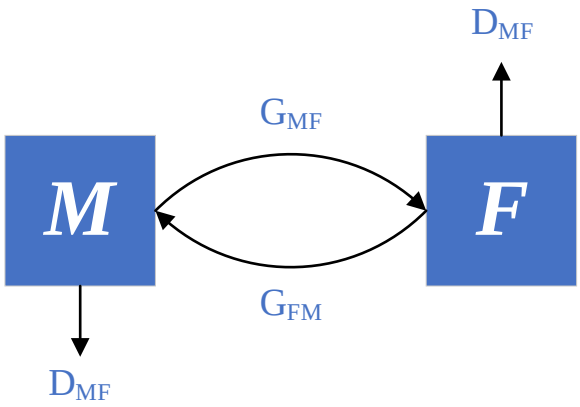
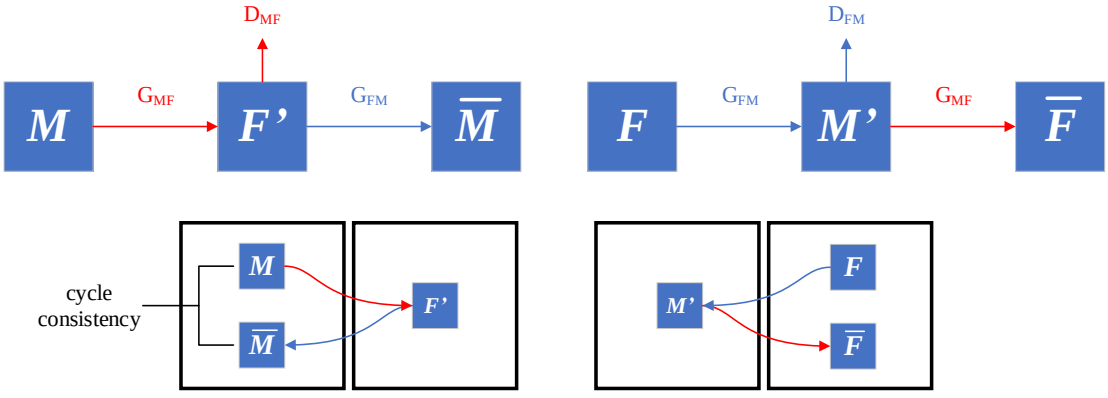


Figure 3.15 Dual network

The cyclic loop is essentially a reconstruction of the input image. Although in such a dual registration setting, the concept of moving image (M) and fixed image (F) have been weakened, we still use moving image and fixed image for word consistency in this thesis.



(a) Forward transform (b) Reverse transform

Figure 3.16 Cycle consistency in registration

The intuitive of using identical dual setting in training is that, for each image from different domain (either moving image or fixed image), the reversed transform should be able to bring it back to the original domain,

$$M \rightarrow G_{MF}(M) \rightarrow G_{FM}(G_{MF}(M)) \approx M$$

$$F \rightarrow G_{FM}(F) \rightarrow G_{MF}(G_{FM}(F)) \approx F$$

Figure 3.16 shows this dual transform. Specially, for moving image M, it first goes through a forward transform $G_{MF}(M)$ that is expected to be a registered image F', followed by the reversed registration network $G_{FM}(F')$ that is expected to be the same as itself $\bar{M}$. Similarly, fixed image F is registered with the forward transform $G_{FM}(F)$ which is expected to be a registered image M', followed by the reversed registration network $G_{MF}(M')$ which is expected to be the same as itself $\bar{F}$.

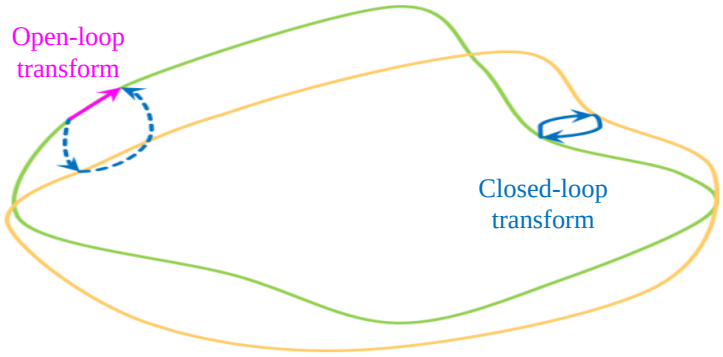


Figure 3.17 Closed-loop and open-loop transform

The transform that M->$G_{MF}(M)$->$G_{FM}(G_{MF}(M)) \approx M$ and F->$G_{FM}(F)$->$G_{MF}(G_{FM}(M)) \approx F$ can be called as a closed-loop transform (as indicated by the solid blue arrow in figure 3.17. In a close-loop transform, the vector sum of the forward transform Φ_{MF}: M->F and backward transform Φ_{MF}: F->M should be zero: $\Phi_{MF} + \Phi_{MF} = 0$. Ensuring a close-loop registration can enhance the registration accuracy [56] as well as ensure a diffeomorphic transform, which allows some desired properties such as maintaining topology and invertibility. If the transform is open loop, the vector sum of two transform will not be zero, which results in an error vector as indicated by the pink arrow on the left in figure 3.17.

The discriminator D_{MF} is expected to distinguish the generated image $G_{MF}(M)$ and the reconstructed image $G_{MF}(G_{FM}(M))$ from real sample F, while the discriminator D_{FM} is expected to distinguish the $G_{FM}(F)$ and $G_{FM}(G_{MF}(M))$ from real sample M.

The two subnetworks are expected to learn simultaneously by minimizing the L1 distance between the original image and the double transformed image,

$$L_{cyc} = \left|G_{FM}\left(G_{MF}(M)\right) - M\right|_1 + \left|G_{MF}\left(G_{FM}(F)\right) - F\right|_1$$

3.3.4. Full objective function

Accordingly, the objective function to train our proposed network is formulated as follows,

$$L = L_{sim} + \alpha L_{perc} + \beta L_{cyc}$$

where α and β are hyperparameters to apply penalties on perceptual loss and cyclic loss.

3.4. Chapter summary

In this chapter, the background and formulation of generative adversarial nets are provided, as long with its variants such as conditional generative adversarial nets, pix2pix and cycleGAN. The main contribution of this thesis is the design of the proposed attentional generative adversarial networks that consist of two identical sub-registration networks. The details in network design are also discussed. The proposed deep neural networks are trained using a combined loss that is designed to maintain statistical similarity, machine perceptual similarity and close-loop transformation. Details of reasons behind and how to design such loss terms are well elaborated.

4. Experimental results

4.1. Datasets

We jointly use two MRI datasets, namely (Open Access Series of Imaging Studies) OASIS dataset [57] and (Information eXtraction from Images) IXI dataset [58]. OASIS dataset is first publicly available in 2007 and now is released as the 3rd version (OASIS-3). Aiming to establish a public dataset for neuroimaging, the OASIS brain project involves more than 1000 participants and provides more than 2168 MRI sessions set including T1, T2, FLAIR and more. MMR scans are collected by Knight ADRC department at Washington University. Due to license permit, we have only access to 1564 scans from OASIS dataset. IXI dataset provides MRI images obtained from each subject including T1, T2 and PD-weighted images, MRI images and diffusion-weighted images. Scans are collected from three major hospitals in London using either Philips 3T system or GE 1.5T system. 581 Image sets are provided in NIfTI format.

We perform standard medical image preprocess on all available T1 weighted scan following the suggestion by [14]. Skull removal and brain extraction for all scans, then resampled to be 1 mm^3 isotropic voxels for IXI scans, cropped and affinely registered into 160x192x224 around the brain to remove irrelevant areas and to fit the memory constraints. We perform the intensity normalization and automatic anatomical segmentation. All pre-processing and automatic segmentation are performed using FreeSurfer as discussed in chapter 2. After the pre-process, we jointly use visual inspection and simple detection skills to remove data with bad quality. We manually select 2 scans that are well processed as reference scans, and compute the available segmentation overlap of the rest with them. We remove 268 scans that has less than 50% overlap with the reference scan.

With the combination of OASIS and IXI dataset, we split the data into approximately 80%/10%/10% for training/validation/testing. To verify the proposed methods is applicable to a new dataset, we use all scans from OASIS as training set and IXI for validation and test set. Finally, our training set contains 1564 scan, validation set contains 156 scans and test set contains 157 scans. This combined dataset becomes the main train and test dataset in this work. Test pair is randomly generated from the test set. Additionally, we report the performance of the model that is trained and test on OASIS dataset only.

4.2. Implementation details and baseline methods

4.2.1. Platforms and hardware information

All deep learning models are implemented in Keras with Tensorflow backend, and experiments are accelerated with GPU. We use Adam optimizer to train all deep learning models at the learning rate of 1e-4 and halved after 200 epochs and again after 300 with a batch size of 2. The maximum training phases lasts for 500 epochs and may stop if the loss is converged (loss is not dropped for 50 iterations). The models are trained with remote sessions on the server, the detailed information of our hardware description is shown in Figure 4.1.

CPU: Intel Xeon E5-2699A	
Cores / Threads	22 / 44
Base frequency / Boost frequency	2.40 GHz / 3.60 GHz
GPU: Nvidia Tesla V100	
Tensor cores / CUDA cores	640 / 5120
Single- / Double precision performance	7 TFLOPS / 14 TFLOPS
GPU memory	16 GB

Figure 4.1 Hardware description on the server

4.2.2. Baseline methods

For conventional algorithm, we choose symmetric image normalization (SyN) [23] as our baseline method, which is one of the top performing registration

algorithms [59]. We use SyN implementation available in ANTs package [60], with the cross correlation similarity measure. To our best knowledge, there is no implementation of SyN using GPU, we report their run time on CPU. We found that the default setting in SyN is suboptimal, therefore we ran a light parameter swap to obtain a better solution. We set the gradient step to be 0.15, increase the number of convergence stages to 4, and decrease the converge threshold to be 1e-9. The rest of specification are listed below.

antsRegistration --verbose 1 -d 3 -o [path/to/output]

-t SyN [0.15,3.0,0.0] -m CC [path/to/fixed/image, path/to/moving,1,4]

-c [100x100x100x50,1e-9,15] -s 3x2x1x0 -f 6x4x2x1

For learning based algorithm, we choose VoxelMorph [14] as our baseline. VM is a state-of-the-art learning-based registration algorithm. VM is an end-to-end unsupervised network to estimate deformable transformations by maximizing the image similarity between an image pair, without the need of ground-truth deformations. The original implementation is available on their GitHub page. The original model is trained on a large dataset with almost 8000 T1 weighted MR scans. For a fair comparison, we retrained their network on our dataset for 400 epochs and search for the best-performing one from the last 50 epochs as the reported value.

4.3. Simulated deformation

The study on registration techniques includes the development of registration algorithms and assessment of registration algorithms. Currently, there is no so-called "gold standard evaluation metrics" for various methods. The only gold standard to evaluate a registration result is to compare the deformation field between the ground truth and the prediction. Therefore, we also verify our models on a simulated dataset.

Typically, simulated deformation is generated by altering the amplitude on the identical deformation field, aiming to synthesize an artificial point-to-point displacement. The simulated deformation can be computed from a set of designated displacement [61] or sampling from random distribution [62]. However, we find it less meaningful to apply random numbers on the displacement since the number generated may not be continuous on the edge of anatomical tissues. At the same time, random displacement on the tissue may not be able to represent a realistic deformation on clinical studies.

To synthesize more realistic deformations, we make use of existing deformations generated by SyN algorithm. Specifically, we apply SyN on a test image pair [*fixed image*, *moving image*] to get the deformation field and smooth the deformation field using a Gaussian kernel with a standard deviation of $\sigma = 20$. Deformed image is generated by applying the synthetic deformation field on the fixed image. Finally, to allow more accurate simulation and variety of real images obtained from the physical devices, we add a Gaussian noise with standard deviation of 5 on the deformed image. Examples for synthetic data are shown in Figure 4.2. Images on the top row are the fixed image, moving image and the synthetic image that generated from the fixed and moving image. Images on the bottom row are the corresponding deformed segmentation of them, which clearly shows the deformation of the RoIs.

Fixed image Moving image Synthetic image

Figure 4.2 Synthetic data

We repeat this synthesis procedure on all our test data and yields 127 synthetic test data. Kindly note that the reported models do not involve an additional training for the synthetic data, which indicates that our model also works for synthetic data without future fine-tuning.

4.4. Evaluation metrics

Evaluation on the performance on registration could be different according to the task. Commonly used evaluation metric on medical image registration includes dice coefficient [8][13][15][16][56][61][63], number of negative Jacobian determinant [54][55][63], mean squared error [13][16][63], target registration error [61][62][64]. Recent paper using GAN also reports their results of discriminator score [64], which can be measure to indicate the registration quality of the registration algorithm.

It is challenging to determine performance of a deformable registration algorithm by visual inspection due to the nature that deformable registration algorithms take inputs that are already affinely registered. As there is no gold standard to evaluate the registration algorithm, to compare the registration performance of various approaches rigorously, we proposed a comprehensive evaluation metrics in the aspect of image similarity, structural similarity, and deformation field analysis.

Specially, we measure the image similarity with normalized mean squared error between the fixed image and the predicted image. We measure the structural similarity using the dice coefficient between the RoIs in the image pair. We compute the number of non-positive Jacobian determinant and the inverse consistency error in the deformation field to analyze the invertibility of it. We also report the run time of the algorithms.

4.4.1. Normalized mean squared error (NMSE)

The normalized mean squared error measures the pixel-level difference. NMSE between image *A* and image *B* is given by,

$$NMSE = \frac{MSE(A, B)}{\max(A) - min(A)}$$

where *A* is defined as the reference image, it refers to the fixed image in our case. Essentially NMSE is a similar measure to MSE. But since our data comes from different sources and has been normalized to [0, 1], NMSE is be more suitable metric in this sense. A lower NMSE indicates a better registration results in terms of pixel-wise image similarity.

4.4.2. Dice coefficient

Dice coefficient measures the overlapping of structures. Dice coefficient between image *A* and image *B* is given by,

$$Dice = 2\frac{|A \cap B|}{|A| + |B|}$$

The intuitive of using Dice coefficient is that if the moving image is registered to the fixed image, RoIs in moving image should be warped to be aligned with the corresponding RoIs in the fixed image. Higher dice coefficient indicates a better registration performance. The mean Dice coefficient is simply the average of Dice

coefficients over all RoIs and dice refers to mean dice coefficient in the following paragraphs.

4.4.3. Jacobian determinant

The deformation field demonstrate the point-to-point movement from the moving image to the fixed image. The deformation of a realistic registration algorithm must be diffeomorphic [23], which is related to the Jacobian matrix of the deformation field. Non-positive values in the determinant of the Jacobian matrix (abbreviated to Jacobian determinant) of the deformation field Φ represents a non-invertible deform, such as folding [55]. Figure 4.3 shows the Jacobian determinant map and the zoom-in deformation grid, which visualizes the warping the deformation field [65]. Position where the determinants are non-positive are shown in red while the corresponding zoom-in deformation grid with folding is shown on the right.

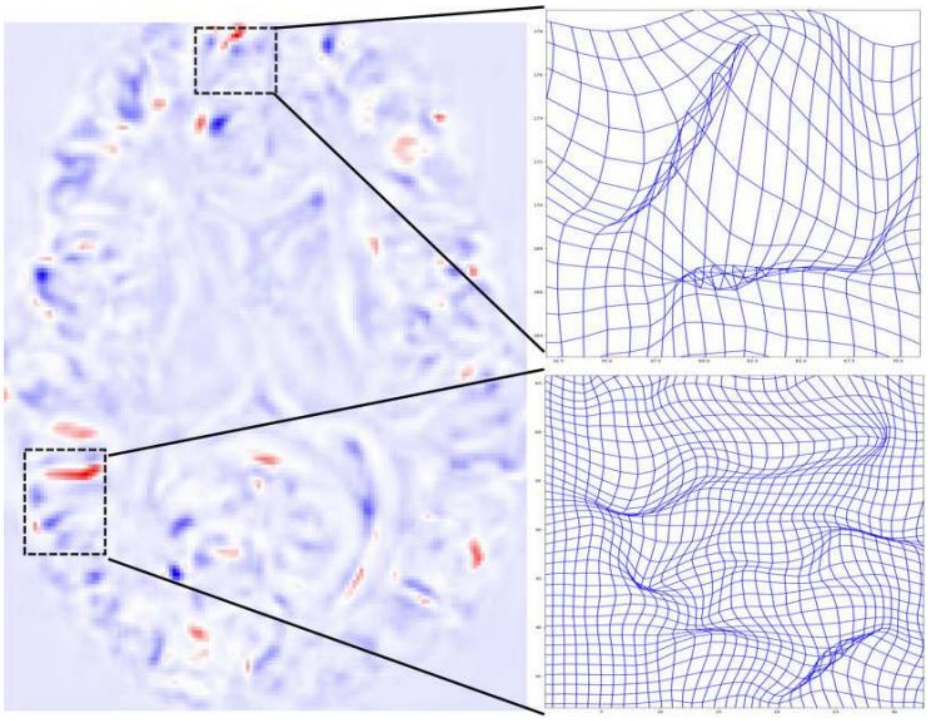


Figure 4.3 Jacobian determinant map and the warped grid

The tissue is enlarged at positions where the Jacobian determinants is greater than 1, while the tissue shrinks at positions where the Jacobian determinants is less than 1. According to the property of Jacobian determinant of the deformation field Φ, the deformation is diffeomorphic if the Jacobian determinants are all positive. Therefore, the percentage of non-positive Jacobian determinant indicates how different the registration is from a diffeomorphic registration. The percentage of non-positive Jacobian determinants defined as,

$$|\nabla Jac(\Phi)|\% = \frac{\#|\nabla Jac(\Phi)| \leq 0}{\mathrm{H} \times \mathrm{W} \times \mathrm{D}} \times 100\%$$

where $\#|\nabla Jac(\Phi)| \leq 0$ represents the number of non-positive Jacobian determinants, H, W and D are the height, width, and depth of the 3D image.

4.4.4. Inverse consistency error (ICE)

Inverse consistency error is proposed by [66] that measures the inverse consistency between the forward deformation and the inverse deformation. Figure 4.4 shows the inverse consistency error. Given image A and image B, let image A registers to image B and becomes image B', and reversely registers image B' back to image A with the same algorithm. Ideally, the vector sum of the forward transform and the inverse transform should be zero, otherwise it is biased [55]. A smaller ICE indicated a better inverse consistency.

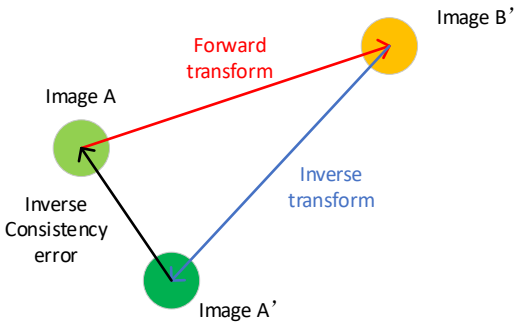


Figure 4.4 Inverse consistency error

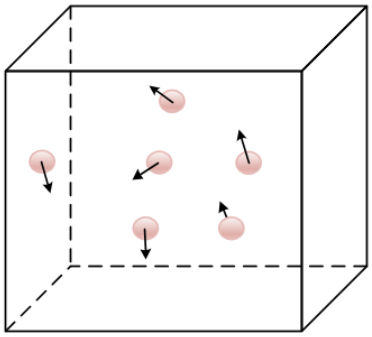


Figure 4.5 deformation field for 3D images

The inverse consistency error between image A and image B is defined as the MSE of the corresponding points in image A and image B. The deformation field at position (x, y, z) is represented by the flow of three scaler value (Φ_x, Φ_y, Φ_x) for 3D images, as indicated by Figure 4.5. Denoting the inverse flow as $(\Phi'_x, \Phi'_y, \Phi'_x)$, the inversed position (x', y', z') is given by,

$$(x', y', z') = (x, y, z) \oplus (\Phi_x, \Phi_y, \Phi_x) \oplus (\Phi'_x, \Phi'_y, \Phi'_x)$$

where $\oplus$ represents the vector sum operation. Finally, the inverse consistency error is formulated as the MSE of all corresponding points,

$$ICE = \frac{\sqrt[2]{(x' - x)^2 + (y' - y)^2 + (z' - z)^2}}{\mathrm{H} \times \mathrm{W} \times \mathrm{D}}$$

4.5. Experimental results

4.5.1. Ablation studies

The proposed approach builds an attentional generative adversarial network and train with cyclic perceptual loss. To verify the influence of each component, we perform ablation studies on the three key contributions of this thesis, namely perceptual loss, cyclic loss, and the attentional mechanism integrated in the generator.

4.5.1.1. Perceptual cyclic loss

Recall that full objective function is given by,

$$L_{sim} + \alpha L_{perc} + \beta L_{cyc}$$

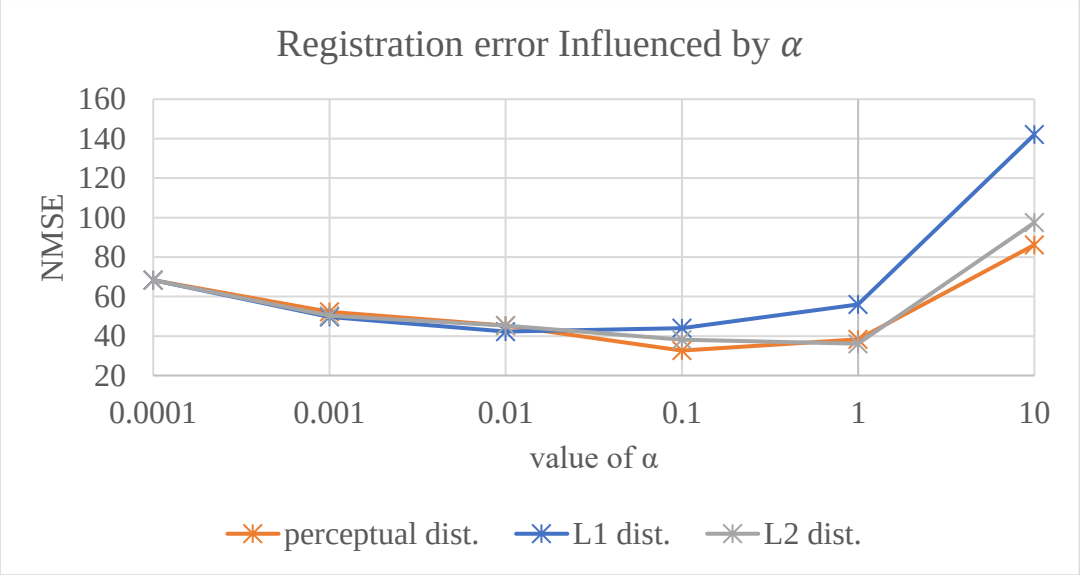
where α and β are hyperparameters to penalty strength on perceptual loss and cyclic loss, respectively. We find $\beta = 0.5$ works well in our setting, so we use $\beta = 0.5$ for further experiments if cyclic loss is involved in training. To study individually on the influence of α, we fixed $\beta = 0$ and perform a wide parameter sweep for hyperparameter searching for α.

α	0.0001	0.001	0.01	0.1	1	10
perc dist.	68.3185	52.2925	45.3637	32.6553	38.3595	86.1861
L1 dist.	68.3185	49.6106	42.2836	44.0200	56.0128	142.0736
L2 dist.	68.3185	50.3299	45.2887	38.1464	36.1369	97.4836

Table 4.1 shows the average NMSE value when training the network with perceptual distance as introduced in chapter 3.3.1, with L1 distance or L2 distance between the fixed image and predicted image.

α	0.0001	0.001	0.01	0.1	1	10
perc dist.	68.3185	52.2925	45.3637	32.6553	38.3595	86.1861
L1 dist.	68.3185	49.6106	42.2836	44.0200	56.0128	142.0736
L2 dist.	68.3185	50.3299	45.2887	38.1464	36.1369	97.4836

Table 4.1 Influence of α



*Note that 0 is not displayable on a log-scale axis, so we manually put the corresponding NMSE value at the position of 0.0001.

Figure 4.6 Influence of α

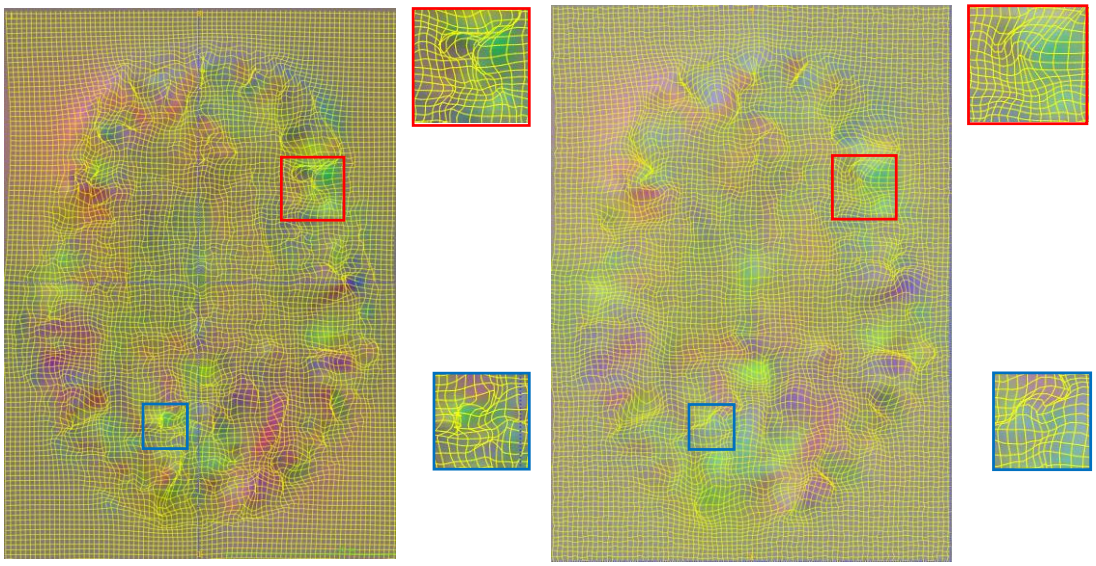
Figure 4.6 suggests that in a certain range, with the increasing amount of penalty on the machine vision loss, the network learns to minimize the pixel level difference. Compared with L1 distance and L2 distance, perceptual distance achieves lower NMSE on the test set among them all. The best result for perceptual distance occurs when $\alpha = 0.1$. In summary, we use $\alpha = 0.1, \beta = 0.5$ for experiments in the following thesis if perceptual or cyclic loss is involved in training.

Table 4.2 shows the experimental results for the ablation study. GAN-only refers to the proposed GAN training with L_{sim} only, GAN (perc only) refers to the GAN training with $L_{sim} + \alpha L_{perp}$, GAN (cyc only) refers to the GAN structure training with $L_{sim} + \beta L_{cyc}$, and GAN (perc + cyc) refers to training the GAN with the proposed term in the loss function. Kindly note that we remove the attentional model in all the aforementioned GAN structures as it will be discussed individually in the next section.

Method	Dice	NMSE	$\|J(\Phi)\| \leq 0$	$\|J(\Phi)\| \leq 0\%$
VoxelMorph	0.8075	44.4864	28263.77	0.004%
GAN-only	0.7060	68.3185	20509.34	0.003%
GAN (perc only)	0.7512	32.6553	27678.50	0.004%
GAN (cyc only)	0.8272	37.0883	2.18	3.17e-5%
GAN (perc + cyc)	0.8098	37.1902	2.23	3.24e-5%

Table 4.2 Ablation studies on perceptual cyclic loss

As indicated by Table 4.2, cyclic loss has great influence on the deformation field generated by the registration algorithm. The neural network learnt to predict deformation field with less folding under this circumstance. Figure 4.7 shows the deformation field generated by the network trained with cyclic loss and without cyclic loss. The deformation field generated by a network trained without the cyclic loss may result in a lower accuracy, because of the folding effect as highlighted on the left figure of Figure 4.7. A network trained with cyclic loss can avoid the folding in the deformation field. As highlighted on the right in Figure 4.7.



Axial deformation field trained WITHOUT cyclic loss

Axial deformation field trained WITH cyclic loss

Figure 4.7 Deformation field influenced by cyclic loss

4.5.1.2. Attentional mechanism

Table 4.3 shows the value of dice coefficient for anatomical structures. The list of anatomical structure can be found in Appendix A section. Figure 4.8 is a graphical representation of Table 4.3, the data is ordered by the average size of structures from largest to smallest. For example, largest labels refer to cerebral cortex covers 1.20% of all brain voxels, while smallest structure 3rd ventricle and 4th ventricle only take up ~1080 voxels, around 0.015%.

The model without attentional mechanism and the model with attentional mechanism is trained with the perceptual cyclic loss. Compared with the model without attentional mechanism (Figure 3.4), SyN achieves much better performance in ventricle DC, Amygdala, Pallidum. But with the help of attentional design (Figure 3.7), the proposed structure improves them and achieves comparable results in these three structures, while maintain its excellent performance on the rest.

In conclusion, the attention-integrated structure re-assigns the coefficient on important features, forming a diversity of focus while suppressing insignificant features during training.

Anatomical structures	**SyN**	**w.o Att**	**with Att**
Cerebral-Cortex	0.633279	0.746201	0.779084
Cerebral-White-Matter	0.75353	0.828005	0.862669
Cerebellum-Cortex	0.866448	0.864582	0.865265
Brain-Stem	0.916273	0.933231	0.923693
Cerebellum-White-Matter	0.805205	0.782867	0.792504
Lateral-Ventricle	0.89328	0.919245	0.921086
Thalamus-Proper	0.913864	0.886497	0.909667
Putamen	0.890451	0.854129	0.887386
Hippocampus	0.823682	0.82024	0.848952
VentralDC	0.883333	0.839854	0.866731
Caudate	0.866335	0.863871	0.886847
choroid-plexus	0.551375	0.564884	0.570998
Amygdala	0.848734	0.733799	0.832444

Pallidum	0.884799	0.825337	0.858922
CSF	0.712674	0.70766	0.710535
3rd-Ventricle	0.756972	0.834862	0.814035
4th-Ventricle	0.727195	0.76112	0.755043

Table 4.3 Values of dice coefficient for anatomical structures (attention model)

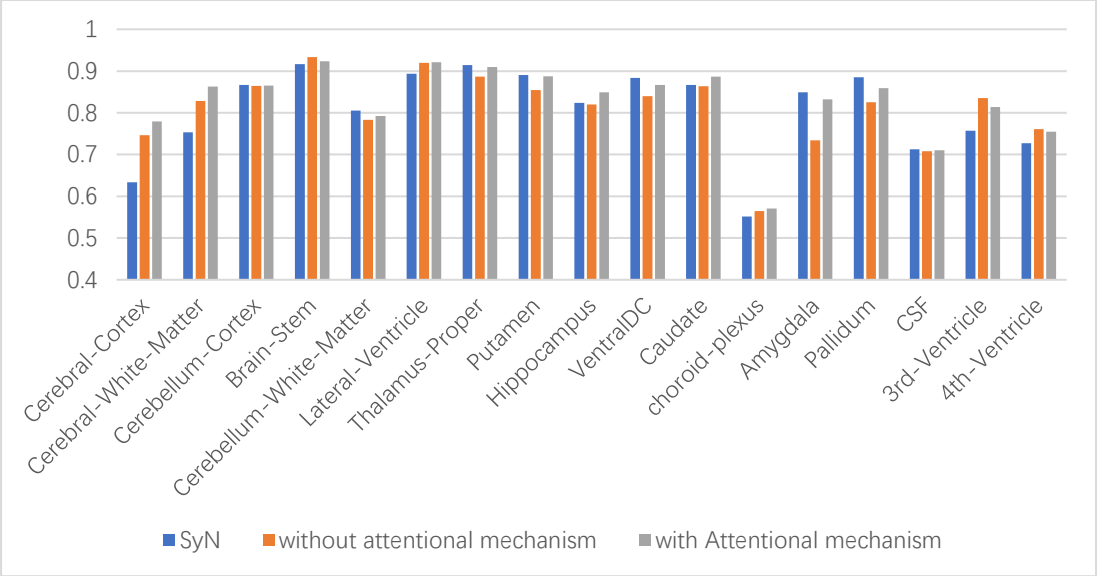


Figure 4.8 Box chart of dice coefficient for anatomical structures (attention)

4.5.2. Registration performance analysis

Figure 4.9 shows a pair of fixed and moving image (subject IXI536-Guys-1059 as fixed image, subject IXI536-Guys-1059 as moving image). Since the MR scan originally comes in a 3D format, we show the middle slice after affine alignment for demonstration. The first column shows the coronal slice, the second column shows the sagittal slice, and the third column shows the axial slice. The goal is to register the moving image (row 2) on the fixed image (row 1).

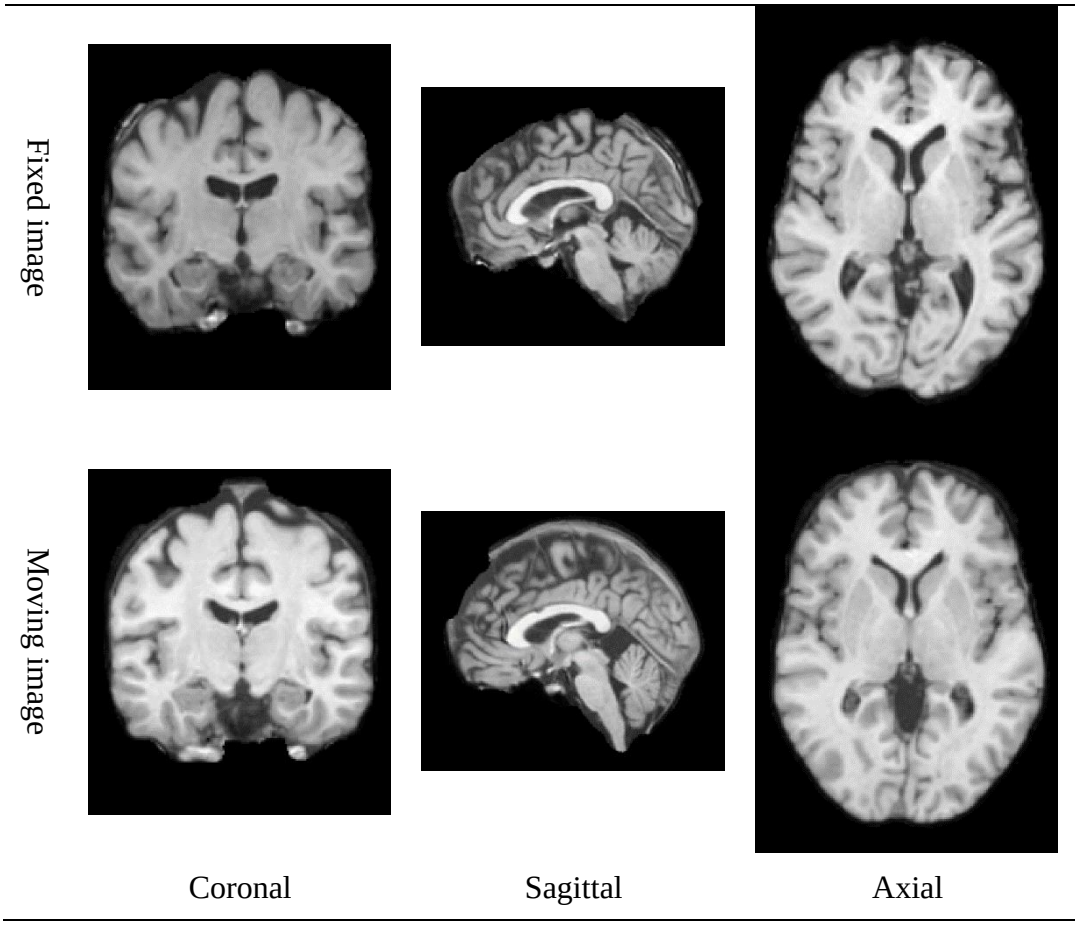


Figure 4.9 Fixed and moving image

Figure 4.10 shows the registration results from the top-performing conventional algorithm SyN, the state-of-the-art learning-based method VoxelMorph and the proposed attentional GAN approach trained with cyclic perceptual loss. One can see that compared to SyN and VoxelMorph, the proposed method looks visually more like the fixed image, especially on the sagittal view (row 2) and axial view (row 3). Table 4.4 gives accurate comparison that summarizes the performance of proposed method and competitor methods. There are 30 RoIs computed in each scan so we do not superimpose them one-by-one, the dice is computed by taking the average of them.

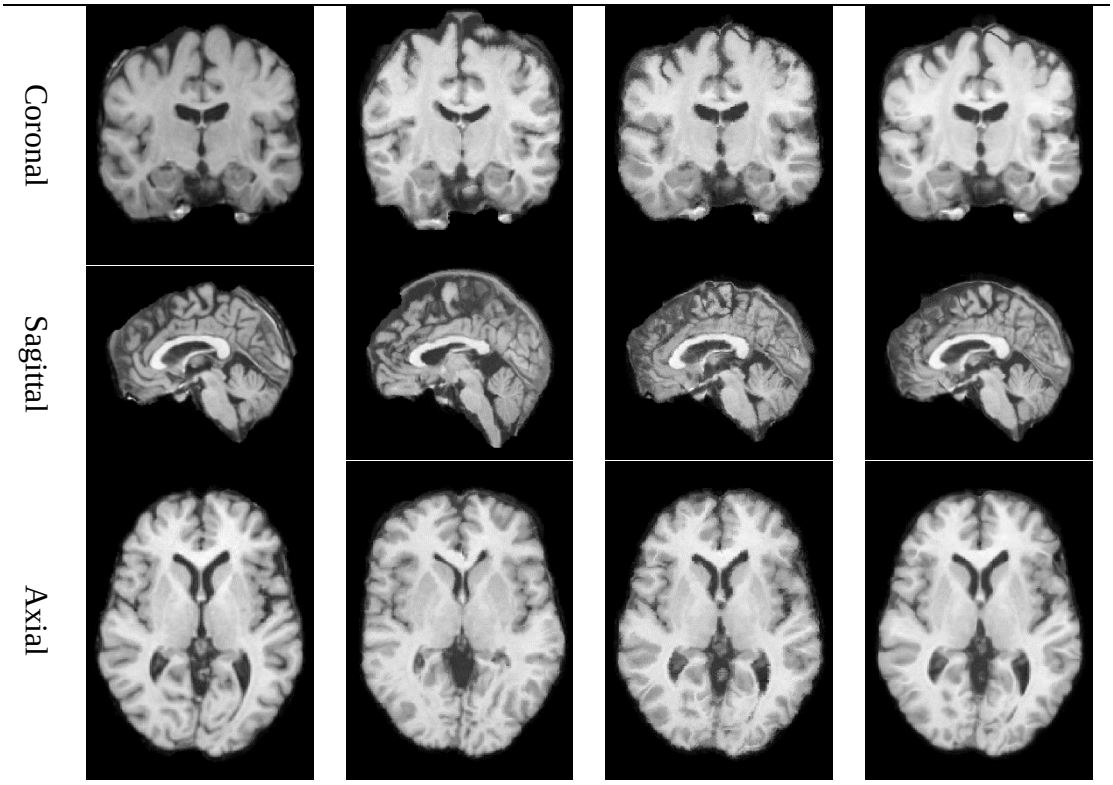


Figure 4.10 Registration results comparison

Method	**NMSE**	**Dice (avg.±std.)**	**$\lvert J(\Phi)\rvert \leq 0$% (cnt., %)**	**ICE (avg.±std.)**	**Runtime (s)**	
					GPU	**CPU**
Affine Reg.	209.7584	0.4726	-	-	-	-
SyN	39.5733	0.8123 (0.104)	1009.94 (0.00015%)	0.010 (0.004)	-	761.8
VoxelMorph	44.4864	0.8075 (0.089)	28263.77 (0.004%)	0.125 (0.032)	0.426	10.47
Proposed	36.9815	0.8323 (0.087)	2.31 (≈ 0%)	0.08 (0.001)	0.564	13.27

Table 4.4 Summary of registration performance

The average dice coefficient for unregistered pairs is 0.47. We report the best available results on the test set. According to Table 4.4, SyN shows its robustness on our test dataset and surpasses the learning method VoxelMorph. Since we have lower NMSE, higher dice score, less percentage of non-positive Jacobian determinant and less ICE, our proposed model outperforms the conventional methods and VoxelMorph. Also, it is worthwhile mentioning that our approach has significantly less non-positive Jacobian determinant in the deformation, which is influenced by the

cyclic training. Even with deeper architecture than VoxelMorph, our model can achieve near 2 fps performance on test scans, which on almost comparable to the inference speed of VoxelMorph. A detailed dice score over all selected structure is shown in Table 4.5 and Figure 4.11.

Anatomical structures	SyN	VoxelMorph	Proposed
Cerebral-Cortex	0.633279	0.744751	0.779084
Cerebral-White-Matter	0.75353	0.840363	0.862669
Cerebellum-Cortex	0.866448	0.868342	0.865265
Brain-Stem	0.916273	0.92157	0.923693
Cerebellum-White-Matter	0.805205	0.80035	0.792504
Lateral-Ventricle	0.89328	0.894383	0.921086
Thalamus-Proper	0.913864	0.900759	0.909667
Putamen	0.890451	0.883246	0.887386
Hippocampus	0.823682	0.839285	0.848952
VentralDC	0.883333	0.855122	0.866731
Caudate	0.866335	0.861037	0.886847
choroid-plexus	0.551375	0.519714	0.570998
Amygdala	0.848734	0.809022	0.832444
Pallidum	0.884799	0.844436	0.858922
CSF	0.712674	0.701361	0.710535
3rd-Ventricle	0.756972	0.771712	0.814035
4th-Ventricle	0.727195	0.753507	0.755043

Table 4.5 Values of dice coefficient for anatomical structures

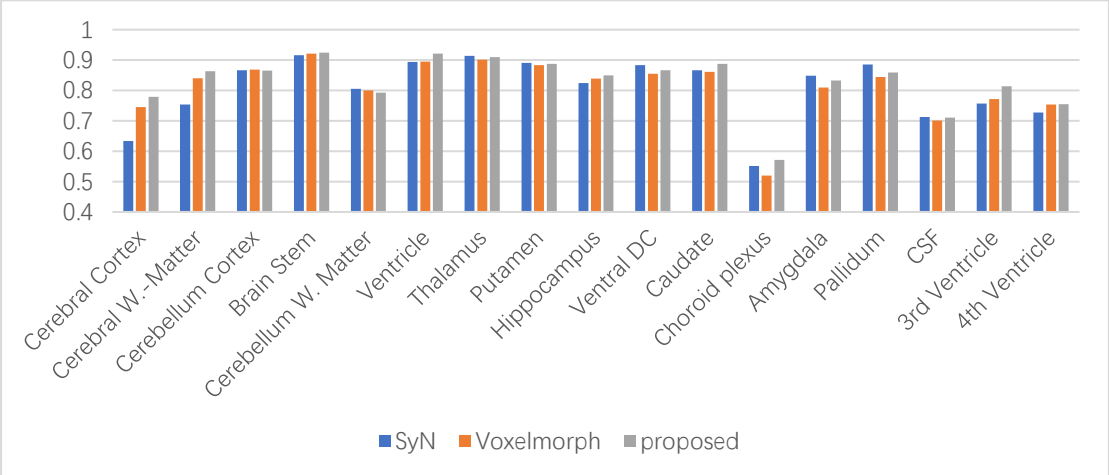


Figure 4.11 Box chart of dice coefficient for anatomical structures

As Table 4.4 summarizes the performance of the model that is trained and test on scans that are collected from dataset, it suggests that our proposed model work has

desirable generalization ability. We also run an experiment where only OASIS dataset is used. We randomly chosen a subset of 100 OASIS data for training, and spilt the data into 80%/10%/10% for training/validation/testing. Each learning-based model is trained for 150k iterations on this training set, approximately 24 epochs. This test set contains 20 original MR scans and forms a total of 380 test pairs.

Method	**NMSE**	**Dice (avg.±std.)**	**$\lvert J(\Phi)\rvert \leq 0$% (cnt., %)**	**ICE (avg.±std.)**	**Runtime (s)**	
					GPU	**CPU**
Affine Reg.	267.6318	0.4138	-	-	-	-
VoxelMorph	62.2056	0.7677 (0.113)	19467.61 (0.003%)	0.344 (0.084)	0.425	10.47
Proposed	51.9341	0.7840 (0.115)	24.82 (3.6e-4%)	0.078 (0.026)	0.568	13.30

Table 4.6 Performance on single dataset

Table 4.6 summarizes the performance on this dataset, we achieved consistent performance given that the training set is not as big as the previous one. Compared with the combined dataset, the number of negative Jacobian determinant drops in Voxelmorph. We speculate that this is because the deformation is limited by the smoothness regularization. Besides, one can observe that the proposed method still outperforms Voxelmorph in terms of NMSE, Dice coefficient and ICE.

4.5.3. Validation on simulation data

Figure 4.12Figure 4.9 shows a pair of fixed and a synthesized. We show the same fixed image from subject IXI536-Guys-1059 as consistent to experiment using real data (Figure 4.9) for comparison. This deformation field comes from SyN algorithm, which yields the registered image in column 2, Figure 4.10., and we apply the deformation field on fixed image to the synthesized data. The deformation field predicted by SyN algorithm is used as ground truth deformation for compute the MSE. Although we use the very deformation field the produce images in column 2, Figure 4.10, one can tell the synthesized imaged (row 2, Figure 4.12) is different. This

is because the SyN is not a perfect registration with a 100% registration accuracy, otherwise these two sets of images will be the same.

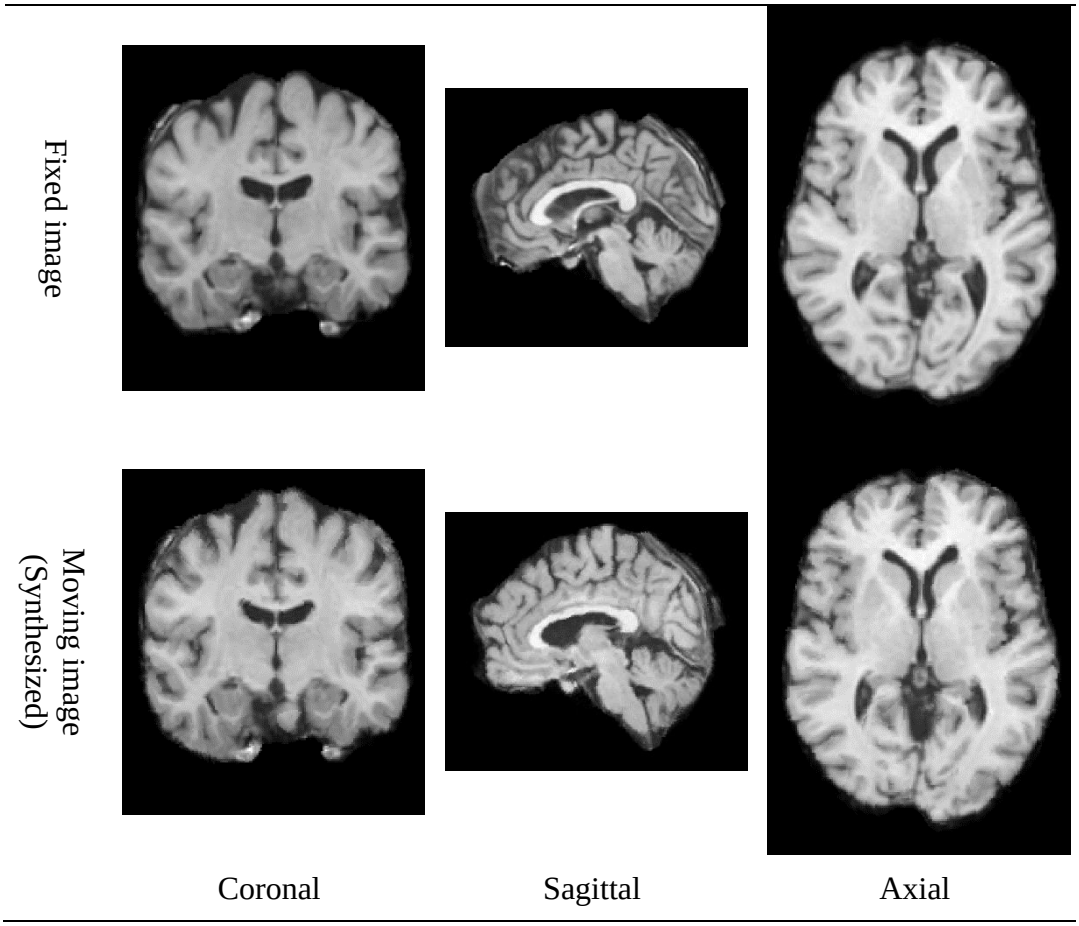


Figure 4.12 Fixed and synthesized image

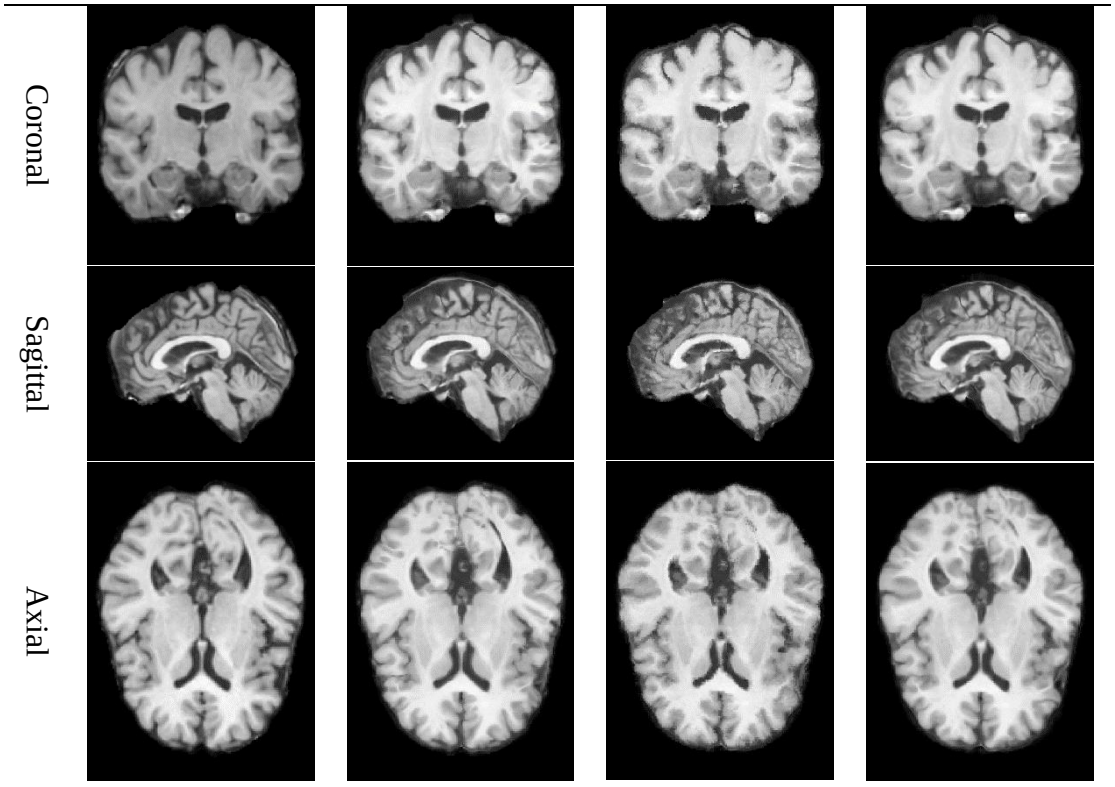


Figure 4.13 Registration results comparison on synthesized data

Figure 4.13 shows the registration results from SyN, the state-of-the-art VoxelMorph and the proposed approach. Our approach yields visually decent similarly with the fixed image. Table 4.7 gives accurate comparison that summarizes the performance of proposed method and competitor methods.

On synthetic test set, the average dice coefficient for unregistered pairs is 0.68. The overall performance on for all algorithm seems better than the test on real data. According to Table 4.7, SyN still outperforms the learning method VoxelMorph. However, VoxelMorph has lower number of non-negative Jacobian determinant on its deformation field, which suggests it learns well from the training data. The proposed model outperforms the conventional methods and VoxelMorph. Since we have the ground truth deformation field, we are able to compute the MSE between the predicted deformation and the ground truth. Our approach achieves the least MSE

compared to SyN and VoxelMorph. Also notice that both learning methods are not re-trained again on the synthetic test set, which proves the leant features are also meaningful for synthetic data.

Mth	**NMSE**	**Dice (avg. ±std.)**	**$\lvert J(\Phi)\rvert \leq 0\%$ (cnt., %)**	**ICE (avg.±std.)**	**MSE**	**Runtime (s)**	
						GPU	**CPU**
Aff. Reg.	91.3182	0.6840	-	-	-	-	-
SyN	14.4203	0.8809 (0.098)	2444.29 (0.0004)	0.037 (0.005)	1.244	-	761.8
VM	19.2207	0.8427 (0.071)	1416.05 (0.0002%)	0.098 (0.029)	1.386	0.426	10.47
Prop.	12.5138	0.8950 (0.062)	0.00 (≈ 0%)	0.004 (<0.001)	1.165	0.564	13.27

Table 4.7 Summary of registration performance on synthetic set

Table 4.8 and Figure 4.14 show the dice coefficient for anatomical structures. Since the deformation field that is used to warp the image comes from SyN, SyN wins most structure than VoxelMorph. Our proposed method still shows comparable or better performance over the selected structures.

Anatomical structures	SyN	VoxelMorph	Proposed
Cerebral-Cortex	0.863723	0.853743	0.85929
Cerebral-White-Matter	0.914664	0.905019	0.932601
Cerebellum-Cortex	0.9212	0.897126	0.927097
Brain-Stem	0.941535	0.932338	0.937485
Cerebellum-White-Matter	0.876982	0.794695	0.860227
Lateral-Ventricle	0.917529	0.926704	0.934441
Thalamus-Proper	0.916693	0.8973	0.941781
Putamen	0.869778	0.856336	0.902058
Hippocampus	0.89519	0.86515	0.930771
VentralDC	0.879805	0.854543	0.86061
Caudate	0.864529	0.862086	0.892865
choroid-plexus	0.586329	0.559628	0.698304
Amygdala	0.874701	0.850536	0.881028
Pallidum	0.833798	0.814981	0.820827
CSF	0.789078	0.760494	0.826377
3rd-Ventricle	0.846571	0.872219	0.872593
4th-Ventricle	0.858633	0.837377	0.853112

Table 4.8 Values of dice coefficient for anatomical structures on synthetic set

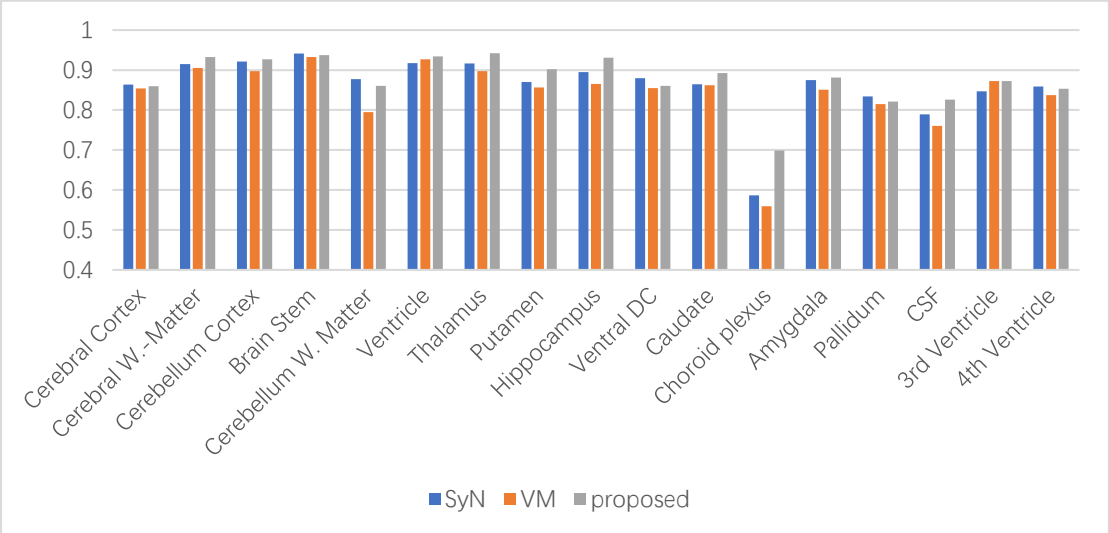


Figure 4.14 Box chart of dice coefficient for anatomical structures on synthetic set

4.6. Chapter summary

This chapter provided detailed information of the datasets and development platform involved in this thesis. Instead of applying random meaningless displacement, a realistic test set is simulated by reusing the predictions from the

existing algorithm. Discussion on evaluation metrics is also given, this work is evaluated with a comprehensive set of metrics in terms of image similarity, structural similarity, and analysis on deformation field. Detailed experimental results are presented.

5. Conclusion

5.1. Summary

Deformable image registration is an important pre-processing step that estimates an optimal spatial transform that aligns the corresponding point between two images. It shows its significance in many key fields such as image fusion and segmentation. Researchers have proposed many algorithms for it by iteratively optimize the similarity measures. These algorithms are robust but typically slow. Rapid development of deep neural network introduces great opportunity for deformable image registration.

This thesis focuses on designing a generative network to predict the non-linear transform field that captures complicated deformation at a fast speed. The proposed architecture is built on a novel attentional generative adversarial net and trained with the perceptual cyclic loss. Attentional mechanism incites useful features while suppressing the insignificant ones during training. The novel perceptual cyclic loss utilized transfer learning and significantly reduce the folding effect on the deformation field. A combined metrics is used to evaluate the proposed algorithm, experimental results show great performance on the 3D brain MRI dataset in terms of accuracy and run time.

5.2. Future scope of work

Due to the limited data that is publicly available, we did not test our method on multimodal data. With necessary modification, the proposed structure should be capable of multimodal image registration. For example, our proposed network can be easily modified for 2D image registration on both medical images. Just for instances, we apply the network on 2D Sunnybrook Cardiac Data (SCD) [67].

SCD dataset consists of 45 short-axis cardiac cine-MRI scans obtained with a 1.5T GE MRI scanner. Each scan contains 20 slices that is acquired at different phases during a breath-hold, which covers the whole heart cycle. Each slice is 256x256 with pixel resolution of 1.28x1.28 mm^2, thickness of 8 mm and spacing of 8 mm. SCD has already been split into train, validation, and test dataset.

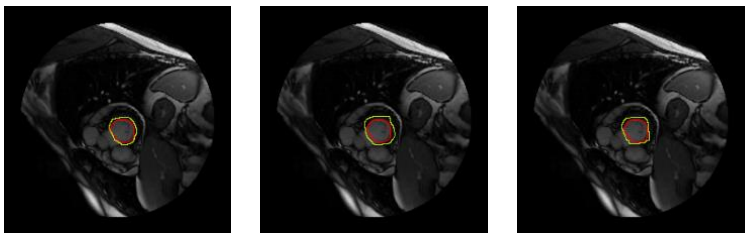


Figure 5.1 Sample results for SCD dataset

Figure 5.1 shows some sample images with superimposed ground truth and prediction, where expert annotation and deformed segmentation mask are both superimposed on the fixed image. Expert annotation is shown in red and the warped contour predicted by registration algorithm is shown in green. The averages dice coefficient for test set in SCD data using the proposed methods is 0.813 compared to only 0.63 before registration. These preliminary experiments suggest two extended applications from our work. The first is its application on 2D medical images. The only modification necessary to do this is just change 3D convolution neurons to 2D convolution, and then retrain the network on the corresponding dataset.

Since the performance of deep learning models rely highly on the quality of data, the actual application on applying deep neural nets for clinical purpose is still under discussion, but we have seen a lot of promising results of it. However, there are few problems that need to be tackled in the future. Firstly, the question goes to the approach is ready for multimodal registration. We have not experimented on multimodal data due to the limitation of publicly available dataset. It is possible to use

NMI as main loss component as indicated in recent literatures. Secondly, as discuss earlier, there is no “gold standard” to evaluate a registration network because of the limitation of labels. Due to the restriction on the share on medical data, the availability for both the raw data itself for training and the ground truth for validation are insufficient. It is urgent to develop a standard database to evaluate registration performance for all different methods.

Appendix A: RoI lookup table

We use FreeSurfer for automatic subcortical segmentation for brain MRI images as introduced in chapter 2. The software provided over 1300 pre-defined labels for various segmentation procedures. As we use its functions for neck removal, skull stripping and resolution resampling, which is defined as phase 1, we obtain over 60 labels. Table A.1 shows all labels we obtained from FreeSurfer, excerpted from here. However, we do not include all the label for to compute their RoIs as some of them are too small to be accurately determined by FreeSurfer [68]. We follow a similar procedure as [13] that only include anatomical structures that are at least 100 voxels in volume for all subjects, resulting in 30 structures, as highlighted green in Table A.1.

Label	Name	R	G	B
0	Unknown	0	0	0
1, 40	Cerebral-Exterior	70	130	180
2, 41	Cerebral-White-Matter	245	245	245
3, 42	Cerebral-Cortex	205	62	78
4, 43	Lateral-Ventricle	120	18	134
5, 44	Inf-Lat-Vent	196	58	250
6, 45	Cerebellum-Exterior	0	148	0
7, 46	Cerebellum-White-Matter	220	248	164
8, 47	Cerebellum-Cortex	230	148	34
9, 48	Thalamus	0	118	14
10, 49	Thalamus-Proper*	0	118	14
11, 50	Caudate	122	186	220
12, 51	Putamen	236	13	176
13, 52	Pallidum	12	48	255
14	3rd-Ventricle	204	182	142
15	4th-Ventricle	42	204	164
16	Brain-Stem	119	159	176
17, 53	Hippocampus	220	216	20
18, 54	Amygdala	103	255	255
19, 55	Insula	80	196	98
20, 56	Operculum	60	58	210
21	Line-1	60	58	210
22	Line-2	60	58	210
23	Line-3	60	58	210

24	CSF	60	60	60
25, 57	Lesion	255	165	0
26, 58	Accumbens-area	255	165	0
27, 59	Substancia-Nigra	0	255	127
28, 60	VentralDC	165	42	42
30, 62	Vessel	160	32	240
31, 63	Choroid-plexus	0	200	200

* Thalamus-Proper is the correct segmentation for the Thalamus.

Table A.1 FreeSurfer look up table

References

[1] B. Zitová and J. Flusser, “Image registration methods: a survey,” *Image Vis. Comput.*, vol. 21, no. 11, pp. 977–1000, Oct. 2003.

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks,” *Adv. Neural Inf. Process. Syst.*, vol. 60, no. 6, pp. 84–90, 2017.

[3] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, “Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation,” *Proc. Eur. Conf. Comput. Vis.*, pp. 801–818, Feb. 2018.

[4] J. Y. Zhu, T. Park, P. Isola, and A. A. Efros, “Unpaired image-to-image translation using cycle-consistent adversarial networks,” in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2223–2232.

[5] H. Dong, G. Yang, F. Liu, Y. Mo, and Y. Guo, “Automatic brain tumor detection and segmentation using U-net based fully convolutional networks,” in *annual conference on medical image understanding and analysis*, 2017, pp. 506–517.

[6] M. Anthimopoulos, S. Christodoulidis, L. Ebner, A. Christe, and S. Mougiakakou, “Lung Pattern Classification for Interstitial Lung Diseases Using a Deep Convolutional Neural Network,” *IEEE Trans. Med. Imaging*, vol. 35, no. 5, pp. 1207–1216, 2016.

[7] S. Miao, Z. J. Wang, Y. Zheng, and R. Liao, “Real-time 2D/3D registration via CNN regression,” in *Proceedings - International Symposium on Biomedical Imaging*, 2016, vol. 2016-June, pp. 1430–1434.

[8] M. W. Lafarge, P. Moeskops, M. Veta, J. P. W. Pluim, and K. A. . J. Eppenhof, “Deformable image registration using convolutional neural networks,” in

Medical Imaging 2018: Image Processing, 2018, vol. 10574, p. 27.

[9] G. Haskins, U. Kruger, and P. Yan, "Deep learning in medical image registration: a survey," *Mach. Vis. Appl.*, vol. 31, no. 1, pp. 1–18, Jan. 2020.

[10] 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*, 2016, pp. 10–18.

[11] X. Yang, R. Kwitt, M. Styner, and M. Niethammer, "Quicksilver: Fast predictive image registration – A deep learning approach," *Neuroimage*, vol. 158, pp. 378–396, Sep. 2017.

[12] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu, "Spatial Transformer Networks," *Adv. Neural Inf. Process. Syst.*, vol. 25, pp. 2017–2025, Jun. 2015.

[13] B. D. de Vos, F. F. Berendsen, M. A. Viergever, M. Staring, and I. Išgum, "End-to-End Unsupervised Deformable Image Registration with a Convolutional Neural Network," in *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, 2017, pp. 204–212.

[14] G. Balakrishnan, A. Zhao, M. R. Sabuncu, J. Guttag, and A. V. Dalca, "An Unsupervised Learning Model for Deformable Medical Image Registration," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 9252–9260.

[15] J. Fan, X. Cao, Q. Wang, P. T. Yap, and D. Shen, "Adversarial learning for mono- or multi-modal registration," *Med. Image Anal.*, vol. 58, p. 101545, Dec. 2019.

[16] D. Mahapatra, B. Antony, S. Sedai, and R. Garnavi, "Deformable medical

image registration using generative adversarial networks," in *Proceedings - International Symposium on Biomedical Imaging*, 2018, vol. 2018-April, pp. 1449–1453.

[17] L. Mercier, V. Fonov, C. Haegelen, R. F. Del Maestro, K. Petrecca, and D. L. Collins, "Comparing two approaches to rigid registration of three-dimensional ultrasound and magnetic resonance images or neurosurgery," *Int. J. Comput. Assist. Radiol. Surg.*, vol. 7, no. 1, pp. 125–136, 2012.

[18] S. K. Kyriacou, C. Davatzikos, S. James Zinreich, and R. Nick Bryan, "Nonlinear elastic registration of brain images with tumor pathology using a biomechanical model," *IEEE Trans. Med. Imaging*, vol. 18, no. 7, pp. 580–592, 1999.

[19] G. E. Christensen, R. D. Rabbitt, and M. I. Miller, "Deformable templates using large deformation kinematics," *IEEE Trans. Image Process.*, vol. 5, no. 10, pp. 1435–1447, 1996.

[20] B. K. P. Horn and B. G. Schunck, "Determining optical flow," *Artif. Intell.*, vol. 281, pp. 319–331, 1981.

[21] M. J. D. Powell, "An efficient method for finding the minimum of a function of several variables without calculating derivatives," *Comput. J.*, vol. 7, no. 2, pp. 155–162, 1964.

[22] B. Fischl, "FreeSurfer," *NeuroImage*, vol. 62, no. 2. Academic Press, pp. 774–781, 15-Aug-2012.

[23] B. B. Avants, C. L. Epstein, M. Grossman, and J. C. Gee, "Symmetric diffeomorphic image registration with cross-correlation: Evaluating automated labeling of elderly and neurodegenerative brain," *Med. Image Anal.*, vol. 12, no. 1, pp. 26–41, Feb. 2008.

[24] 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.

[25] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in *arXiv preprint arXiv:1409.1556.*, 2014.

[26] L. C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, “DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, 2017.

[27] 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.

[28] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2015, vol. 9351, pp. 234–241.

[29] A. Ruderman, N. C. Rabinowitz, A. S. Morcos, and D. Zoran, “Pooling is neither necessary nor sufficient for appropriate deformation stability in CNNs,” *arXiv preprint arXiv:1804.04438*. 2018.

[30] I. Goodfellow *et al.*, “Generative Adversarial Nets,” *Adv. Neural Inf. Process. Syst. 27*, pp. 2672–2680, 2014.

[31] I. Goodfellow, “NIPS 2016 Tutorial: Generative Adversarial Networks,” Dec. 2016.

[32] P. Isola, J. Y. Zhu, T. Zhou, and A. A. Efros, “Image-to-image translation with conditional adversarial networks,” in *Proceedings - 30th IEEE Conference on*

Computer Vision and Pattern Recognition, CVPR 2017, 2017, vol. 2017-January.

[33] A. Radford, L. Metz, and S. Chintala, "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks," *arXiv preprint arXiv:1511.06434.*, 2015. [Online]. Available: http://arxiv.org/abs/1511.06434.

[34] C. Ledig *et al.*, "Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 4681–4690.

[35] M. Mirza and S. Osindero, "Conditional Generative Adversarial Nets," *arXiv preprint arXiv:1411.1784.*, 2014. [Online]. Available: http://arxiv.org/abs/1411.1784.

[36] 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*, 2015, pp. 234–241.

[37] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4510–4520.

[38] O. Oktay *et al.*, "Attention U-Net: Learning where to look for the pancreas," *arXiv*, 2018. [Online]. Available: https://arxiv.org/abs/1804.03999.

[39] R. K. Srivastava, J. Masci, F. Gomez, and J. Schmidhuber, "Understanding locally competitive networks," *arXIV*, 2014. [Online]. Available: https://arxiv.org/abs/1410.1165.

[40] D. Bahdanau, K. H. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv*, 2014. [Online]. Available:

https://arxiv.org/abs/1409.0473.

[41] K. Xu *et al.*, “Show, attend and tell: Neural image caption generation with visual attention,” in *32nd International Conference on Machine Learning, ICML 2015*, 2015, pp. 2048–2057.

[42] V. Mnih, N. Heess, A. Graves, and K. Kavukcuoglu, “Recurrent models of visual attention,” *Adv. Neural Inf. Process. Syst.*, vol. 27, pp. 2204–2212, 2014.

[43] A. Vaswani *et al.*, “Attention is all you need,” in *Advances in Neural Information Processing Systems*, 2017, pp. 5998–6008.

[44] M. Khened, V. A. Kollerathu, and G. Krishnamurthi, “Fully convolutional multi-scale residual DenseNets for cardiac segmentation and automated cardiac diagnosis using ensemble of classifiers,” *Med. Image Anal.*, vol. 51, pp. 21–45, 2019.

[45] H. R. Roth *et al.*, “Hierarchical 3D fully convolutional networks for multi-organ segmentation,” *arXiv*, 2017. [Online]. Available: https://arxiv.org/abs/1704.06382.

[46] L. Gatys, A. S. Ecker, and M. Bethge, “Texture Synthesis Using Convolutional Neural Networks,” *Adv. Neural Inf. Process. Syst.*, vol. 28, pp. 262–270, 2015.

[47] J. Johnson, A. Alahi, and L. Fei-Fei, “Perceptual Losses for Real-Time Style Transfer and Super-Resolution,” in *European conference on computer vision*, 2016, pp. 694–711.

[48] M. D. Zeiler and R. Fergus, “Visualizing and Understanding Convolutional Networks,” in *European conference on computer vision*, 2013, pp. 818–833.

[49] B. H. Menze *et al.*, “The Multimodal Brain Tumor Image Segmentation Benchmark (BRATS),” *IEEE Trans. Med. Imaging*, vol. 34, no. 10, pp. 1993–2024, 2015.

[50] G. Wang, W. Li, S. Ourselin, and T. Vercauteren, “Automatic brain tumor segmentation using cascaded anisotropic convolutional neural networks,” in *International MICCAI brainlesion workshop*, 2017, pp. 178–190.

[51] W. Chen, B. Liu, S. Peng, J. Sun, and X. Qiao, “S3D-UNET: Separable 3D U-Net for brain tumor segmentation,” in *International MICCAI Brainlesion Workshop*, 2018, pp. 358–368.

[52] C. Szegedy *et al.*, “Going Deeper with Convolutions,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[53] A. Mahendran and A. Vedaldi, “Understanding Deep Image Representations by Inverting Them,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 5188–5196.

[54] A. V. Dalca, G. Balakrishnan, J. Guttag, and M. R. Sabuncu, “Unsupervised learning for fast probabilistic diffeomorphic registration,” in *Lecture Notes in Computer Science*, 2018, vol. 11070 LNCS, pp. 729–738.

[55] J. Zhang, “Inverse-Consistent Deep Networks for Unsupervised Deformable Image Registration,” *arXiv*, 2018. [Online]. Available: https://arxiv.org/abs/1809.03443.

[56] S. Zhao, T. Lau, J. Luo, E. I., C. Chang, and Y. Xu., “Unsupervised 3D End-to-End Medical Image Registration with Volume Tweening Network,” *EEE J. Biomed. Heal. informatics*, vol. 24, no. 5, pp. 1394–1404, Feb. 2019.

[57] D. S. Marcus, T. H. Wang, J. Parker, J. G. Csernansky, J. C. Morris, and R. L. Buckner, “Open Access Series of Imaging Studies (OASIS): Cross-sectional MRI data in young, middle aged, nondemented, and demented older adults,” *J. Cogn. Neurosci.*, vol. 19, no. 9, pp. 1498–1507, Sep. 2007.

[58] “IXI Dataset – Brain Development.” [Online]. Available: https://brain-

development.org/ixi-dataset/.

[59] A. Klein *et al.*, “Evaluation of 14 nonlinear deformation algorithms applied to human brain MRI registration,” *Neuroimage*, vol. 46, no. 3, pp. 786–802, Jul. 2009.

[60] B. B. Avants, N. J. Tustison, G. Song, P. A. Cook, A. Klein, and J. C. Gee, “A reproducible evaluation of ANTs similarity metric performance in brain image registration,” *Neuroimage*, vol. 54, no. 3, pp. 2033–2044, Feb. 2011.

[61] H. Sokooti, B. de Vos, F. Berendsen, B. P. F. 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*, 2017, pp. 232–239.

[62] S. Miao, Z. J. Wang, and R. Liao, “A CNN Regression Approach for Real-Time 2D/3D Registration,” *IEEE Trans. Med. Imaging*, vol. 35, no. 5, pp. 1352–1363, May 2016.

[63] 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,” *Med. Image Anal.*, vol. 52, pp. 128–143, Feb. 2019.

[64] P. Yan, S. Xu, A. R. Rastinehad, and B. J. Wood, “Adversarial image registration with application for MR and TRUS image fusion,” in *International Workshop on Machine Learning in Medical Imaging*, 2018, pp. 197–204.

[65] D. Kuang, “On Reducing Negative Jacobian Determinant of the Deformation Predicted by Deep Registration Networks,” *arXiv*. 2019.

[66] G. E. Christensen and H. J. Johnson, “Consistent image registration,” *IEEE Trans. Med. Imaging*, vol. 20, no. 7, pp. 568–582, 2001.

[67] “Evaluation Framework for Algorithms Segmenting Short Axis Cardiac MRI.,”

MIDAS Journal-Cardiac MR Left Vent. Segmentation Chall., vol. 49, 2009.

[68] P. A. Filipek, C. Richelme, D. N. Kennedy, and V. S. Caviness, “The young adult human brain: An MRI-based morphometric analysis,” *Cereb. Cortex,* vol. 4, no. 4, pp. 344–360, 1994.