2022

Empirical Study of Deep Neural Network Architectures for Non-Rigid Medical Image Registration

Chuanhui Tian

UNIVERSITY
OF WOLLONGONG
AUSTRALIA

Empirical Study of Deep Neural Network Architectures for Non-Rigid Medical Image Registration

Chuanhui Tian

This thesis is presented as part of the requirements for the conferral of the degree:

Master of Philosophy

Supervisor:
Prof. Philip O. Ogunbona

Co-supervisor:
A/Prof. Wanqing Li

The University of Wollongong
School of Computing and Information Technology

March, 2022

This research has been conducted with the support of an Australian Government Research Training Program Scholarship.

Declaration

I, *Chuanhui Tian*, declare that this thesis is submitted in partial fulfilment of the requirements for the conferral of the degree *Master of Philosophy*, from the University of Wollongong, is wholly my own work unless otherwise referenced or acknowledged. This document has not been submitted for qualifications at any other academic institution.

Chuanhui Tian

August 9, 2022

Abstract

Medical image registration is the alignment of two or more images of the same scene or object, but taken possibly from different viewpoints, at different times or by different sensors. Accurate registration plays an important role in the diagnosis and treatment of diseases. Several factors make the task of medical image registration challenging. The surface curvature of the tissues implies that the medical image registration is non-rigid and non-linear. Additionally, the quality of acquired images could be poor because of noise, inherent pathologies, low overlap area and repeated patterns. Recent development in computer vision and medical image processing has seen the introduction of transformer-based networks in accomplishing various tasks and with notable results. This trend has been seen in medical image registration where the performance of convolutional-based networks is being challenged by transformer-based networks. However, it is unclear that whether the improvement cited for transformer-based networks is due mainly to the architecture or other factors such as scale of transformation fields, dataset characteristics and the guidance of different loss functions. In this study, several deep neural network architectures are critically reviewed from the viewpoint of components of architectures, loss functions, scale of transformation fields and datasets respectively. Experiments involving ablation studies over several architectural options were designed and conducted to reveal the performance differences. Theoretical analyses are provided to interpret results.

Acknowledgments

I would like to express my deepest thanks to my principal supervisor, Professor Philip O. Ogunbona. Many thanks for his guidance and supports during my M.Phil study, especially during the COVID-2019 pandemic. His patience, continuous feedback and suggestions were very helpful when I was designing and conducting my experiments. This continued even when he was sick. With his teaching and regular meeting, I developed my logic and writing skills. He always encouraged me to think critically. Importantly, he showed me a good example to be a good researcher during my M.Phil study. I learned a lot from him, not only the academic knowledge but also the ability to discover and learn new things. I also want to acknowledge my co-supervisor, Associate professor Wanqing Li for his encouragement and providing a sounding board for my ideas.

Also, I would like to show my appreciation to my friends Dr. Lei Qi, Dr. Melody Tan and Christine Zheng for their help and valuable suggestions in respect of designing experiments, writing thesis and preparing presentations. I would also like to thank my colleagues in Advanced Multimedia Research Lab. It was a privilege for me to study in this lab at the end of the research journey. Sharing and discussing with them helped me to broaden horizons in the area of computer science. Further, I am extremely grateful to my friends who encouraged and kept me company when I felt sad and disappointed during my M.Phil study. It is difficult to imagine this research journey without their encouragements. Especially, I would like to thank my dearest Margaret and Jeff Fuller, Dorothy and Reg Piper, Barbara Kennard and other lovely friends in Wollongong, their care and love made my stay enjoyable and peaceful.

Particularly, I would like to extend my deepest gratitude to my parents. This study would not have been possible without their financial support. Special thanks for their

understanding, unconditional love and support without expecting anything in return, so I can pursue and complete my M.Phil without anxiety and stress from finance and family.

Contents

Abstract **iv**

1 Introduction **1**

1.1 Background 1

1.2 Research Gap 6

1.3 Research Questions 7

1.4 Contribution 8

1.5 Publications 9

1.6 Thesis Organization 9

2 Literature Review **11**

2.1 Deep Learning 11

2.2 Medical Image Registration 12

2.3 Performance Evaluation Metrics 14

2.3.1 Key points-based Methods 14

2.3.2 Segmentation map-based Methods 15

2.4 Architectures 16

2.5 Loss Functions 25

2.6 Chapter Summary 28

3 Experimental Design **29**

3.1 Networks 29

3.2 Implementation Settings 30

3.3 Datasets Description 30

3.3.1 Retinal Image Datasets 31
3.3.2 Brain Tumor Datasets 34
3.4 Dataset Preprocessing 36
3.4.1 Dataset Generation 36
3.4.2 Image Resizing 36
3.4.3 Image Pair Generation 37
3.4.4 Dataset Split 38
3.5 Evaluation and Statistical Significance 39
3.6 Chapter Summary 40

4 Results **41**

4.1 Network Architecture Components and Performance 42
4.1.1 Convolutional-based Network: Size vs Performance 42
4.1.2 Transformer-based Network: Size vs Performance 47
4.1.3 Transformer-based Network: Multi-head vs Performance 52
4.2 Further Exploration of Transformer-based Architecture 55
4.2.1 Retinal Image Registration: Scale of Transformation Field vs Performance Differences 56
4.2.2 Brain Image Registration: Scale of Transformation Field vs Performance Differences 58
4.3 Loss Functions and Architectures 61
4.3.1 Retinal Image Registration: Loss functions vs Performance 61
4.3.2 Brain Image Registration: Loss functions vs Performance 64
4.4 Loss Functions and Dataset Characteristics 67
4.4.1 Convolutional-based Network: Dataset Vs Performance 68
4.4.2 Transformer-based Network: Dataset Vs Performance 69
4.5 Chapter Summary 71

5 Discussion **72**

5.1 Network Architecture Components and Performance 72

5.2 Further Exploration of Transformer-based Architecture 76
5.3 Loss Functions and Architectures 79
5.4 Loss Functions and Datasets 80
5.5 Chapter Summary 83

6 Conclusion **85**
6.1 Summary 85
6.2 Further Work 87

Bibliography **88**

List of Figures

1.1 Block diagram of deep learning-based image registration methods 5

2.1 Diagram showing the architecture of method proposed by J. Wang and Zhang (2020) 18

2.2 The overview of the autoencoder network proposed by Krebs et al. (2018) 19

2.3 A U-Net is built to regress deformation field directly (Balakrishnan et al., 2018) 20

2.4 A U-Net is designed to regress intermediate variables (Dalca et al., 2018) 21

3.1 The overview of ViT-V-Net architecture (J. Chen et al., 2021) 30

3.2 Retinal image pairs of FIRE dataset 33

3.3 Five common abnormal findings in diabetic retinopathy, ordered by the increase stage of seriousness (Kauppi et al., 2007): (a) microaneuryms, (b)hemorrhages, (c)hard exudates, (d)soft exudates, (e)neovascularization. 34

3.4 Samples from each retinal image dataset 35

3.5 Some brain image examples 35

3.6 Distributions of each datasets and combination dataset 37

4.1 Experimental setting of Section 4.1.1 to explore the relationship between the size of convolutional-based network and dice score performance . . . 43

4.2 Average dice scores of convolutional-based architectures at different sizes interacted with different loss functions to complete retinal image registration 45

4.3 Average dice scores of convolutional-based architectures at different sizes interacted with scales of transformation fields to complete retinal image registration 46
4.4 Experimental setting of Section 4.1.2 to explore the relationship between the size of transformer-based network and dice score performance 47
4.5 Average dice scores of transformer-based architectures at different sizes interacted with different loss functions to complete retinal image registration 50
4.6 Average dice scores of transformer-based architectures at different sizes interacted with scales of transformation fields to complete retinal image registration 51
4.7 Experimental setting of Section 4.1.3 to explore how the number of heads in self-attention components affects the performance of transformer-based networks in retinal image registration 52
4.8 Average dice scores of transformer-based architectures changed with the size and number of multi-head self-attention components in retinal image registration 55
4.9 Experimental setting of Section 4.2.1 to explore how transformer-based networks address limitation of convolutional-based networks in retinal image registration 56
4.10 The differences of average dice scores between transformer-based and convolutional-based networks at different scales of transformation fields and loss functions 58
4.11 Experimental setting of Section 4.2.2 to explore how transformer-based networks address the limitation of convolutional-based networks in brain image registration 59
4.12 Experimental setting of Section 4.3.1 to explore how loss functions interacting with architectures affect the performance of retinal image registration 62

4.13 Average dice scores of different architectures trained with different loss functions at various scales of transformation fields in retinal image registration 63
4.14 Experimental setting of Section 4.3.2 to explore how loss functions interacting with architectures affect the performance of brain image registration 65
4.15 Average dice scores of different architectures trained with different loss functions at various scales of transformation fields in brain image registration 66
4.16 Experimental setting of Section 4.4.1 to explore how loss functions interacting with dataset characters affect the performance of convolutional-based networks. 68
4.17 Average dice scores of convolutional-based networks trained with different loss functions at different scales of transformation fields in retinal and brain image registration 69
4.18 Experimental setting of Section 4.4.2 to explore how loss functions interacting with dataset characters affect the performance of transformer-based networks. 70
4.19 Average dice scores of transformer-based networks trained with different loss functions at different scales of transformation fields in retinal and brain image registration 70

5.1 An example of a retinal image to capture structural information by subtracting mean intensity 82
5.2 An example of a brain image to capture structural information by subtracting mean intensity 83

List of Tables

3.1 Details of constructing different size architectures 30
3.2 Hyperparmaeters of training stage 30
3.3 Publicly available retinal image datasets 34
3.4 Summary of parameters to limit transformation 38
3.5 Details of splitting retinal images into training, validation and test subset 38
3.6 Details of splitting brain images into training, validation and test subset . 38

4.1 List of used abbreviations 42
4.2 Dice scores for convolutional-based architectures at different sizes and different loss functions to complete retinal image registration at various scales of transformation fields 43
4.3 Dice score differences and p-values for convolutional-based architectures at different sizes to complete retinal image registration at various scales of transformation fields and different loss functions 43
4.4 Dice scores for transformer-based architectures at different sizes and different loss functions to complete retinal image registration at various scales of transformation fields 50
4.5 Dice score differences and p-values for transformer-based architectures at different sizes to complete retinal image registration at various scales of transformation fields and different loss functions 51
4.6 Dice scores for multi-head self-attention components to complete small-displacement retinal image registration with SSIM loss function 54

4.7 p-values for multi-head self-attention components to complete small-displacement retinal image registration with SSIM loss function 54
4.8 Dice score differences and p-values between convolutional-based and transformer-based architectures at different scales of transformation fields and loss functions in retinal image registration 58
4.9 Dice scores for the convolutional-based architecture at different scales of transformation fields and loss functions in brain image registration 60
4.10 Dice scores for the transformer-based architecture at different scales of transformation fields and loss functions in brain image registration 60
4.11 Differences and p-values between convolutional-based and transformer-based architectures at different scales of transformation fields and loss functions in brain image registration 61
4.12 p-values for the convolutional-based architecture at different scales of transformation fields and loss functions in retinal image registration . . . 64
4.13 p-values for the transformer-based architecture at different scales of transformation fields and loss functions in retinal image registration 64
4.14 p-values for the convolutional-based architecture at different scales of transformation fields and loss functions in brain image registration 66
4.15 p-values for the transformer-based architecture at different scales of transformation fields and loss functions in brain image registration 66
4.16 The order of dice score for retinal and brain image registrations at different scales of transformation fields 67

Chapter 1

Introduction

1.1 Background

Medical image registration is the alignment of two or more images of the same scene or object, but taken possibly from different viewpoints, at different times, or by different sensors (Zitova & Flusser, 2003). When the object or scene is non-deformable and planar, the registration task can be accomplished by affine transformations. Deformable and non-planar objects on the other hand must go through a more complex transformation to achieve registration. Medical images are typical non-planar images requiring deformable registration because of the surface curvature of the organ.

Medical image registration plays an important role in a variety of clinical applications such as disease diagnosis and monitoring, image-guided treatment delivery, telesurgery, and post-operative assessment (X. Chen et al., 2020). By comparing and matching medical images acquired over a period of time, it is possible to determine the severity of diseases. In addition, a wide variety of information from different images is accurately integrated by using the correct image registration method, which makes it easier and more convenient for clinicians to observe symptoms from different viewpoints. At the same time, through the result of medical image registration, clinicians could quantitatively analyze the changes of lesions and organs to make diagnosis and treatment more accurate and reliable. Therefore, a highly accurate image registration algorithm is required to augment clinical decision-making. Basically, medical image registration is a challenging task because of a series of factors, including the low quality of medical images influenced by

noise, variation in spatial resolution, the changing size and shape of organs when taking time-lapsed medical images (X. Chen et al., 2020).

In this thesis, conventional methods indicate those algorithms that predate the popularity of deep learning-based methods. Generally, conventional methods deal with image registration as an iterative optimization task. The iterative optimization process could be expressed as

$$\hat{\tau} = \arg\min_{\tau} DS(I_F, \tau(I_M)), \tag{1.1}$$

where transformation field is denoted by τ, and a dissimilarities metric $DS()$ is used to calculate dissimilarities between fixed image (I_F) and warped moving image ($\tau(I_M)$). The overview of the optimization process is shown in Algorithm 1. The conventional methods are computationally expensive and could be slow to converge. However, clinical applications usually require real-time registration. In addition, the conventional registration method is inefficient, because it measures dissimilarities iteratively for each input pair of images. Well-tuned performance parameters which are optimized by a specific input pair may not be suitable for other input pairs. Moreover, conventional methods are more easily influenced by the quality of images than deep learning methods. For example, some medical images are very faint and blurry, and thus pose difficulties to conventional methods when measuring dissimilarities based on features or intensities. The influence of poor image quality on deep learning methods is provided in a later description.

Algorithm 1: The process of conventional method of image registration

Input : moving image (I_M) and fixed image (I_F)
Output: Transformation field (τ) and moved image ($I_{M'}$)
1. Initialize parameters of transformation field: τ_0
2. Generate moved image by warping moving image on transformation field: $I_{M'} = \tau_0(I_M)$
3. Calculate dissimilarities between fixed and moved image: $DS(I_F, I_{M'})$
while $DS(I_F, I_{M'}) > Constant$ **do**
  $\tau_{i+1} \leftarrow update\ \tau_i$
  $I_{M'} = \tau_{i+1}(I_M)$
end

With the development of deep learning in computer vision, several convolutional neural

network architectures have been applied to image registration tasks. Deep neural networks learn from a large number of image pairs to predict transformation parameters directly. The fact that poor image quality has little effect on deep neural networks, has led to their superior performance over traditional approaches (Géron, 2019). In addition, the large dataset used to train a deep learning network allows it approximately learn its underlying probability distribution. Therefore, a well-trained network could register new unseen image pairs well, especially when the assumption of unseen image pairs being samples of the same distribution as the training set holds. Similarly to conventional methods, a deep network requires long training times but it could register new image pairs very quickly, which is beneficial in realizing real-time clinical applications. From the viewpoint of learning scenarios, deep learning consists of supervised and unsupervised learning. Supervised training needs dataset to be labelled with ground truth. In the case of medical image registration, the ground truth is a known transformation field between input pairs, which is a high dimensional parametric space for deformable transformation. Since there exists a lot of plausible transformation fields between images, it is important to have the precise target transformation. Normally, the method of obtaining ground truth transformation field is through manual annotation by medical experts. Therefore, it is expensive and difficult to acquire reliable ground truth data for supervised training of registration networks. This has motivated the development of unsupervised networks.

The advent of spatial transform network (STN) (Jaderberg et al., 2015) has given rise to unsupervised learning networks that could realize end-to-end medical image registration. Jaderberg et al. (2015) proposed the spatial transformer network (STN) to learn how the same feature is represented after applying various transformations including affine, rigid and deformable transformations, etc. The components of an STN include a localization network, a grid generator and a sampler. To be precise, a localization network predicts transformation parameters, then a grid generator uses the predicted transformation parameters to output the location of each input point in a common coordinate. The sampler interpolates the input image with locations in a common coordinate which is the output from the grid generator to create a new image. Importantly, these three components are

differentiable, thus STN is able to be trained with backpropagation, and it is suitable to combine STN with other architectures.

Therefore, several unsupervised registration networks have been inspired by the STN to warp the moving image, thus making it possible to describe the new image and the fixed image on a common coordinate, and align similar anatomical structures in both images. A typical deep learning-based method (shown in Figure 1.1) can be described as follows. The network takes a pair of images (moving image (I_M) and fixed image (I_F)) to regress a transformation field directly. Due to the non-planarity and curvature of organs captured in medical images, a deformable registration is usually assumed and this requires a dense transformation field to be estimated. This dense transformation is in the form of a dense displacement field representing how each pixel moves from the moving image to the fixed image. The spatial transformation network takes the generated dense field and moving image as input and generates the moved image. A loss function comprising fidelity and regularization terms is minimized to update the parameters of the registration network. The fidelity term of the loss function is a dissimilarity metric that quantifies the difference between the moved and fixed images. In order to constrain the hypothesis space of the transformation field, the regularizer applies smoothness and boundedness constraints to guide the optimization. The loss function can be expressed as

$$\mathscr{L} = \mathscr{L}_{dissimilarity}(I_F, \phi(I_M)) + \lambda \mathscr{L}_{smooth}(\phi), \tag{1.2}$$

where I_F and I_M are fixed image and moving image respectively. The transformation field is denoted by ϕ. The hyper-parameter λ adjusts the influence of the regularizer.

Since transformer network (Vaswani et al., 2017) dominated in natural language possessing (NLP), several works have introduced self-attention layers, which is the most important component of transformer network, into CNN-like architectures in computer vision area. From a theoretical perspective, self-attention layers behave similarly to convolutional layers (Cordonnier et al., 2019). Convolution operation in neural networks is essentially a correlation since it lacks the signal reversal of the mathematical convolution (Goodfellow et al., 2016). It is a dot product operation similar to those typical

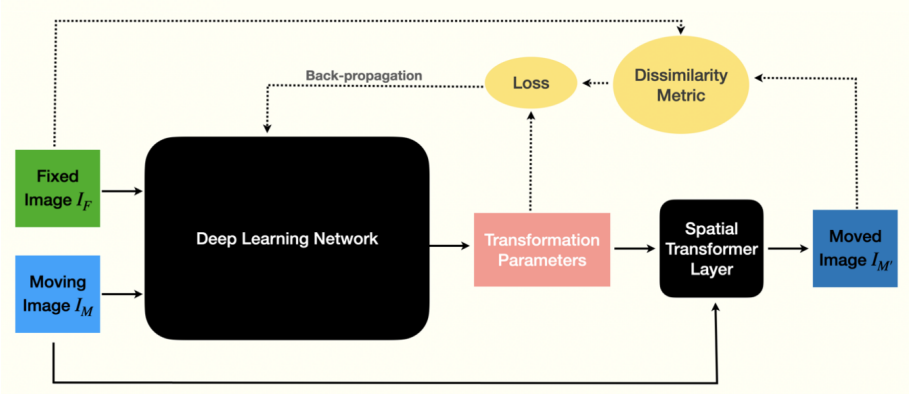


Figure 1.1: Block diagram of deep learning-based image registration methods

of self-attention mechanisms. In self-attention, a dot product-based computation measures the similarity between query and key vectors acquired from a little projection of feature maps. Self-attention mechanism is much more flexible than convolutional operation (Tuli et al., 2021). Unlike convolutional inductive bias, self-attention mechanism can learn global information of images, and it is able to handle the problem of extracting long-distance interactions between features. In the case of medical image registration, we hypothesize that the transformer-based network is able to register larger-displacement image pairs better than convolutional-based network.

Dosovitskiy et al. (2020) introduced a pure Transformer network directly in computer vision to complete large-scale image recognition tasks. The architecture named Vision Transformer (ViT) mimics the transformer network in NLP with few modifications and was able to outperform the state-of-the-art method in the task of image recognition (Dosovitskiy et al., 2020). Since then, more ViT-based architectures have been proposed to complete various tasks in computer vision, including image classification, image segmentation as well as medical image registration. ViT-V-Net (J. Chen et al., 2021) is the first transformer-based network to complete volumetric brain image registration. It improved upon the performance of VoxelMorph (Balakrishnan et al., 2018) which is a convolutional-based network from 0.711 ± 0.135 to 0.726 ± 0.130 based on the dice score. They concluded that introducing a vision transformer network can well serve to improve the performance of medical image registration.

1.2 Research Gap

Although several deep learning-based registration networks have achieved state-of-the-art performance, it is not clear what makes one architecture outperform another. While there has been a variety of review papers (e.g. Boveiri et al., 2020; X. Chen et al., 2020; Fu et al., 2020; Haskins et al., 2020) that summarize different deep learning-based registration networks, they pay more attention to analyze what architectures are used in medical image registration and group these registration works according to typical structures of architectures, objects to be registered, modality of input image pairs and learning algorithm (supervised, weakly-supervised and unsupervised learning, etc). They did not provide performance comparisons of these registration networks, and hence it is not clear what makes one registration work outperform another. It is well known that the performance evaluation of deep learning-based method is influenced by a variety of factors, including data, architecture, loss function and evaluation method (Goodfellow et al., 2016).

In terms of data, a large amount of data is used to train a deep network for a good generalization ability (Chollet, 2018). Good generalization indicates that the network performs well on previously seen data (training data) as well as on unseen data (test data). A dataset can be considered as a collection of examples from an unknown underlying generative distribution. Roughly speaking, unsupervised learning algorithms observe a random part of examples in this dataset, and then it tries to learn the probability distribution of this entire dataset (Goodfellow et al., 2016). Additionally, various datasets have different probability distributions, and it is possible that some datasets possess a more complex distribution, which may be hard for a network to learn (Rahane & Subramanian, 2020). Therefore, it is crucial that when evaluating various networks, the same dataset is used. Another key influencing factor for performance is architecture. In deep learning, we regard the neural network as a function approximation algorithm. Each layer is a function, for example, the first layer is expressed as

$$\mathbf{h}^{(1)} = g^{(1)}(\mathbf{W}^{(1)\top}\mathbf{x} + \mathbf{b}^{(1)}), \tag{1.3}$$

where g denotes the activation function. Respectively, metrics of weights and vector of

bias are denoted by $\mathbf{W}$ and $\mathbf{b}$ in a layer, and input data is denoted by x. Most architectures consist of a stack of layers in a chain structure, thus the second layer is given by

$$\mathbf{h}^{(2)} = g^{(2)}(\mathbf{W}^{(2)\top}\mathbf{h}^{(1)} + \mathbf{b}^{(2)}), \tag{1.4}$$

and so on. Intuitively, architecture represents a combined function of a series of functions. The layered structure and the nonlinearity allow the network to approximate many functions. Given an underlying distribution dataset, there is a target function mapping input to output, and the target function is unknown. All possible functions mapping input to output constitute a hypothesis space. Therefore, the architecture should be designed to represent this approximation function within the hypothesis space. Moreover, a well-designed architecture should easily find the approximation function with a small generalization error (Mohri et al., 2018).

Generally, it is important to define an appropriate loss function to train networks. The loss function compares predictions to the expectations, giving feedback on how well network predictions match the expected network output. By minimizing the loss function, the network parameters will assume values that iteratively move the prediction closer to expectation. A suitable loss function is crucial to guide the network to arrive at the expected output more easily. Furthermore, a variety of evaluation methods have been used to measure the reported performance of image registration. The choice of evaluation method is straightforward to indicate the performance, but various evaluation methods may vary from what is really measured. This sometimes makes it difficult to compare the methods objectively.

1.3 Research Questions

There is a variety of proposed deep learning networks to complete medical image registration for different organs, such as brain, lung, chest, heart and retina, etc. In order to have a fair comparison, we train a variety of proposed registration networks on the same dataset and then evaluate their performances with the same evaluation method. In order to eliminate performance improvement attributable to randomness, a statistical significance

test is provided as well.

A series of controlled experiments and ablation studies are conducted to comparatively analyze the relationship between performance and architectures, loss functions, data and maximum displacement in image pairs respectively. Visual observation, mathematical and statistical interpretation are utilized to provide insights into what contributes to the observed performance improvement.

In summary, we aim to answer the following research questions:

1. What components and composition of architectures make deep learning-based registration achieve high performance?

2. How does the transformer network address the limitation of convolutional neural network in medical image registration?

3. How do different loss functions interact with different architectures to affect performance?

4. How do different loss functions interact with different datasets and affect registration performance?

1.4 Contribution

In our work, we train convolutional-based and transformer-based networks to complete unsupervised retinal and brain image registration respectively. The contributions are listed as follows:

1. By comparatively analyzing their performances, this work provides empirical insights into the relationship between performance and a series of factors, including architectures, scales of transformation fields, loss functions and data characteristics.

2. Using a series of ablation studies, we explore the machinery of convolutional-based and transformer-based networks relative to their performance in registration tasks.

3. It provides insights into different loss functions in medical image registration. We clarify how different loss functions interact with datasets and network architectures

to influence performance. To be precise, we provide two-factor factorial designs (i.e. loss functions and datasets, loss functions and architectures respectively) to explore their influences on registration performances.

4. In order to address the problem of the shortage of publicly available image registration datasets, we applied realistic transformations with different groups of parameters on an image, the image and its transformation images can be paired to complete image registration task. The groundtruth of transformation field is kept as well. Three retinal fundus image registration datasets and three brain slices image registration datasets are generated.

1.5 Publications

1. Chuanhui Tian, Philip O. Ogunbona and Wanqing Li. Empirical Study of Transformer-based Network for Non-Rigid Medical Image Registration. Submitted to Pattern Recognition.

1.6 Thesis Organization

In this section, we provide an overview of our thesis organization.

In Chapter 1, we introduced the background knowledge including the definition and application of medical image registration and the overview description of the conventional method and deep learning method in the case of medical image registration. In addition, we clarified the research gap in the research area of deep learning in medical image registration. We observed that the generalization ability is influenced by a series of factors, but most medical image registration works paid more attention to improving performance by reconstructing architectures, it is still unclear what makes deep neural networks achieve better performance. Therefore, we proposed four research questions to explore the relationship between performance and a series of factors: architectures, the scale of transformation fields, loss functions and datasets respectively. Also, we conclude with major contributions in this chapter.

Next, we provided a literature review in Chapter 2. We explained roughly the concepts of deep learning including architectures and loss function in the context of medical image registration. Then, we expressed the problem formulation of medical image registration in terms of deep learning algorithm. We regarded medical image registration as a regression task where image data and network parameters are regressors, and the dependent variable is the transformation field. Also, we reviewed several registration works by analyzing their components of architectures, regressor, dependent variables and performance. Further, loss functions and performance evaluation metrics are reviewed in this chapter as well.

In Chapter 3, we described details of the experimental design including dataset description, the construction of network and the setting of hyperparameters in implementation. We proposed a method to generate image pairs by utilizing image segmentation datasets, it is helpful to address the limitation of publicly available image registration datasets. Additionally, we introduced the statistical significance test to determine whether the performance difference is due to chance.

Finally, we designed and conducted a series of controlled experiments and ablation studies based on our four research questions. The results of the experiments are presented in Chapter 4. In Chapter 5, we provided a discussion of the results using some theoretical considerations of deep networks. A conclusion is given in Chapter 6 along with suggestions for further work.

Chapter 2

Literature Review

This chapter briefly provides a presentation of deep learning concepts in the context of medical image registration, then performance evaluation metrics are reviewed. Lastly, comprehensive reviews of medical image registration based on architecture and loss function are provided respectively.

2.1 Deep Learning

In the real world, there is an assumption that there is a target function f^* mapping input x to ground truth output y given an underlying distribution dataset (Goodfellow et al., 2016), which is expressed as:

$$y = f^*(x). \tag{2.1}$$

However, the real underlying distribution dataset is unknown due to hardly collecting all data, thus the target function f^* cannot be found. Therefore, a neural network containing a group of parameters (θ) defines a function f to mimic the behaviour of the target function f^*, mapping input x to output y, which is expressed as:

$$y = f(x, \theta), \tag{2.2}$$

where θ is the parameter set of the network.

Then the goal of training a network is to approximate a target function f^* (Goodfellow et al., 2016), enforcing that defined function f matches to f^*. Let the input and output be denoted by $x \in \mathscr{X}$ and $y \in \mathscr{Y}$ respectively, $\mathscr{X} \subset \mathbb{R}^d$ and $\mathscr{Y} \subset \mathbb{R}$. A function f maps

$(x,y) \in \mathscr{X} \times \mathscr{Y}$ to $\mathscr{L}(f(x),y)$, where $\mathscr{L}$ is an arbitrary loss function with a non-negative value, $\mathscr{L} \subset \mathscr{Y} \times \mathscr{Y} \to \mathbb{R}$. The target function $f^* : \mathscr{X} \to \mathbb{R}$ is derived by minimizing the expected risk ($I[f]$), which is denoted by Rosasco et al. (2004):

$$\min_{f \in \mathscr{F}} I[f], \tag{2.3}$$

with

$$I[f] = \int_{\mathscr{X} \times \mathscr{Y}} \mathscr{L}(f(x),y)p(x,y)dxdy, \tag{2.4}$$

where $\mathscr{F}$ is the space of measurable functions, $p(x,y)$ is the probability of the pair (x,y). The real value of $p(x,y)$ is unknown in the real world, which further explained why the target function f^* cannot be found. By assuming that examples (x_i, y_i) in a finite dataset are independent and identically distributed, an empirical risk can be calculated as:

$$I_{emp}[f] = \frac{1}{n}\sum_{n}^{i=1} \mathscr{L}(f(x_i),y_i), \tag{2.5}$$

where n is the number of examples in dataset. These errors from each example in datasets are aggregated to generate an average loss over the dataset. The scalar value roughly represents the distance between approximation function (f) behavior and target function (f^*) behavior. The approximation function (f) is selected by a minimizer:

$$\min_{f \in \mathscr{H}} \frac{1}{n}\sum_{n}^{i=1} \mathscr{L}(f(x_i),y_i), \tag{2.6}$$

where $\mathscr{H}$ is hypothesis space, each element f in $\mathscr{H}$, $f : \mathscr{X} \to \mathscr{Y}$. Therefore, the desired function f is a coarse approximation of target function f^*. By minimizing equation 2.6, these parameters are updated to generate the desired function f, which mimics the behavior of target function f^*, resulting in a good generalization ability.

2.2 Medical Image Registration

The goal of image registration task is to find a transformation mapping that aligns a given pair of images. The alignment could extend to more than two images (or series of images) wherein the images are captured from the same subject (X) but possibly from different

fields of view or at different times.

Therefore, the transformation mapping (τ) can be defined as:

$$\tau : X_{I_M} \mapsto X_{I_F} \leftrightarrow \tau\left(X_{I_M}\right) = X_{I_F}, \tag{2.7}$$

where X_{I_M} and X_{I_F} are a set of all positions describing $x \in X$ (subject) in moving image (I_M) and fixed image (I_F) respectively.

When considering the characteristics of the mapping, the number of parameters and formats depend on the type of transformation. More specifically, six parameters suffice to describe rigid transformation including translation and rotation within the global area. Affine transformation is a global transformation including scaling and sheering transformation. It requires 12 parameters to model this transformation. However, deformable transformation is a local transformation and generally applies different transformations to each pixel. It is more complex and requires a dense displacement field to describe the movement of consecutive pixels along each dimension. In the case of deformable medical image registration, we denote a dense transformation field (ϕ) to describe the transformation mapping (τ). The input of a network is a pair of images which are concatenated into a single input x, indicated as

$$f : ((I_M, I_F), \theta) \rightarrow \phi. \tag{2.8}$$

Equation 2.8 denotes a regression task where the image data and the network parameters are the regressors and the transformation field is the dependent variable. We have more to say about regression tasks and loss functions in section 2.4 and section 2.5.

In our case of medical image registration, the predicted output (i.e. predicted transformation field) is denoted by $\hat{\phi}$. Thus the loss function can be expressed as $\mathscr{L}(\hat{\phi}, \phi)$. In practice, the true transformation field is hard and expensive to assess. Therefore, we define the loss function indirectly by warping moving image on true and predicted transformation respectively. Based on Equation 2.7, the fixed image is actually derived by

warped moving image on the true transformation field, denoted by

$$I_F = \phi(I_M). \tag{2.9}$$

So, we warp moving image (I_M) on the predicted transformation field ($\hat{\phi}$) to get a new image, denoted by moved image ($I_{M'}$):

$$I_{M'} = \hat{\phi}(I_M). \tag{2.10}$$

Then the indirect loss function $\mathscr{L}(\hat{\phi}(I_M), \phi(I_M))$ is denoted by $\mathscr{L}(I_{M'}, I_F)$, which calculates the dissimilarities between moved image and fixed images. Equation 1.2 shows the overview definition of the general loss function in unsupervised medial image registration.

2.3 Performance Evaluation Metrics

Several evaluation metrics have been devised to investigate the performance of image registration methods. In this thesis, the choice of evaluation metric is particularly important to explore the performance tradeoff achievable with different combination of network architectures and loss functions. Some of the metrics commonly used in the literature are briefly described in this subsection.

2.3.1 Key points-based Methods

Root Mean Square Error (RMSE)

Root Mean Square Error (RMSE) calculates the Euclidean distance between corresponding points in fixed and moved images. A comparatively small value indicates a better performance. The RMSE is computed as,

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} \left[(x_i - x_i')^2 + (y_i - y_i')^2\right]}{N}}, \tag{2.11}$$

where N is the number of corresponding points, (x_i, y_i) and (x_i', y_i') are coordinates of corresponding points in fixed and moved images respectively.

Target registration error (TRE)

There are pairs of ground-truth corresponding landmarks provided by experts. This metric computes the average distance measured in pixels, it is similar to RMSE but computed for given points. TRE is computed as,

$$TRE = \frac{1}{N}\sqrt{(x_2 - x_1)^2 - (y_2 - y_1)^2}, \tag{2.12}$$

where N is the number of manual corresponding landmarks, (x_1, y_1) and (x_2, y_2) are coordinates of landmarks in fixed and moved images respectively. A small TRE value indicates a better registration performance.

2.3.2 Segmentation map-based Methods

Pixel Accuracy (PA)

This metric calculates the ratio of the same pixels in segmentation maps of fixed and moved images. Comparatively higher values indicate better performance. Pixel accuracy is defined as

$$PA = \frac{\sum_{i=0}^{k} p_{ii}}{\sum_{i=0}^{k}\sum_{j=0}^{k} p_{ij}}, \tag{2.13}$$

where p_{ii} is the number of the same pixels in both segmentation maps, $\sum_{i=0}^{k}\sum_{j=0}^{k} p_{ij}$ indicates the total number of pixels in two segmentation maps. In the task of medical image registration, it is an estimate of the probability of accurate prediction of the transformation field.

Dice Score (DS)

Dice Score is the dominant metric used to calculate similarities between two segmentation maps. It calculates the ratio of overlap area to the sum of segmentations of fixed and moved images. Comparatively higher values indicate better alignment.

$$DS = \frac{2 \times |S_F \bigcap S_{M'}|}{|S_F| + |S_{M'}|}, \tag{2.14}$$

where $|S_F|$ and $|S_{M'}|$ are the segmentations of fixed and moved images respectively.

2.4 Architectures

Goodfellow et al. (2016) proposed that the design of architecture is a key consideration for neural networks. They explained the universal approximation theorem (Cybenko, 1989) wherein it was shown that a large network is able to represent any function on a closed and bounded subset of $\mathbb{R}^n$, but it is not able to specify how large enough the network should be. Therefore, it becomes partly science, partly art to design a proper architecture to approximate the target function. In this section, we review several core components of network architecture and some convolutional-based and transformer-based medical image registration networks respectively.

A typical convolutional block consists of several convolutional layers followed by a maxpooling layer. There are two major functions of convolutional operators: feature aggregation and feature transformation. In terms of feature aggregation, kernels slide on all the locations within the image to extract features, and convolutional operations combine all features to output a feature map. In addition, a series of linear transformation and non-linear activation functions are utilized to realize feature transformation.

When it comes to maxpooling layers, remaining invariant to small translations is the key consideration to insert pooling layers in architecture. After extracting and aggregating features through successive convolutional layers, maxpooling layer provides a summary statistic of the nearby outputs. Maxpooling layer is an efficient component to reduce the spatial size of feature maps without any trainable parameters. In addition, it is beneficial to enlarge the effective receptive field (Le & Borji, 2017).

Some architectures only consist of several convolutional blocks as their backbone to complete a complex image registration task. Instead of regressing a dense deformation field directly, this kind of architecture is more likely to regress some dependent variables within a lower-dimensional space to parametrize deformation field. For example, De Vos et al. (2019) built a fully-convolutional architecture to complete cardiac cine MRI image registration. This architecture takes a pair of images to go through three convolutional blocks and additional convolutional layers to regress B-spline control points directly, then the final deformation field is generated by resampling these estimated control points with

B-spline interpolation. However, this architecture does not seem to improve the performance compared with conventional image registration. The method yielded a dice score of 0.87 ± 0.18 on registered cardiac cine MRI image pairs. When compared to a conventional method using SimpleElastix (Marstal et al., 2016) which yielded a dice score of 0.86 ± 0.18, this is hardly a performance improvement. Besides, the performance is largely influenced by the user-chosen B-spline grid spacing, and B-spline may not be able to describe a dense deformation field precisely.

Another example is that J. Wang and Zhang (2020) set up a dual-net architecture to regress an initial velocity field, which is able to determine a transformation field according to the Fourier representation. Similarly, each pipeline consists of only several convolutional blocks, but the regressor and dependent variables are different. A high-dimensional image can be decoupled into a real part and an imaginary part in the Fourier space, realized by complex-valued operations and functions. A complex-valued convolution can be defined as:

$$H * \widetilde{X} = H * \mathscr{R}(\widetilde{X}) + iH * \mathscr{I}(\widetilde{X}). \tag{2.15}$$

As shown in Figure 2.1, the real part and imaginary part decoupled from an image are fed into two separate pipelines (R_{net} and I_{net}) respectively. In this case, the regressor is the concatenation of corresponding parts derived from moving image and fixed images. The outputs of R_{net} and I_{net} are a real part and an imaginary part of the velocity field respectively, these predicted parts are combined to obtain the initial complex-value velocity field, then used to yield a deformation field between moving and fixed images. Compared to VoxelMorph (Balakrishnan et al., 2018) achieved a 0.774 dice score, this dual-net architecture achieved a 0.780 dice score on 2D brain image registration. Therefore, a simple architecture is able to achieve somewhat superior performance for a complex image registration when the regressor and dependent variables are designed suitably. In addition, training architecture in a low dimensional bandlimited space is helpful to reduce computational requirements and speed up the training time.

Furthermore, several convolutional blocks are used to construct the encoder path in

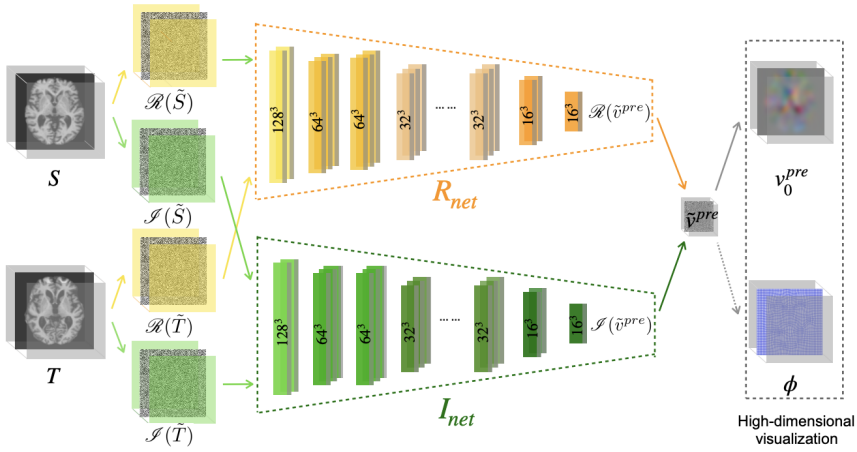


Figure 2.1: Diagram showing the architecture of method proposed by J. Wang and Zhang (2020)

an autoencoder network. Since image registration is also modelled as a reconstruction problem, a conditional variational autoencoder network (Krebs et al., 2018) is built to reconstruct the fixed image by warping moving image with transformation field. As shown in Figure 2.2, an encoder is trained to predict a latent variable to approximately describe posterior registration probability ($p_\theta(z \mid M;F)$) which is defined as:

$$p_\theta(z \mid M;F) = \mathscr{N}(\mu(F,M), \sigma(F,M)). \tag{2.16}$$

In Equation 2.16, the set of trainable encoder parameters is denoted by θ; encoder output is the latent vector denoted by z; mean and diagonal covariance are denoted as $\mu(F,M)$ and $\sigma(F,M)$ respectively; and moving and fixed images are denoted as M and F respectively. Additionally, the moving image as conditioning data and the output of encoder z are fed into the decoder to reconstruct fixed image. The decoder is naturally defined as distribution $p_\gamma(F \mid z;M)$ with trainable parameters γ. The decoder is constructed with several deconvolutional layers to upsample the latent vector to a velocity field with the same dimension as the input. Subsequent processing in a convolutional Gaussian layer is used to smoothen the velocity field explicitly before yielding the transformation field. The final differentiable exponentiation layer integrates the velocity field to yield a smooth transformation field. This autoencoder network achieved a dice score of 0.783 on MRI

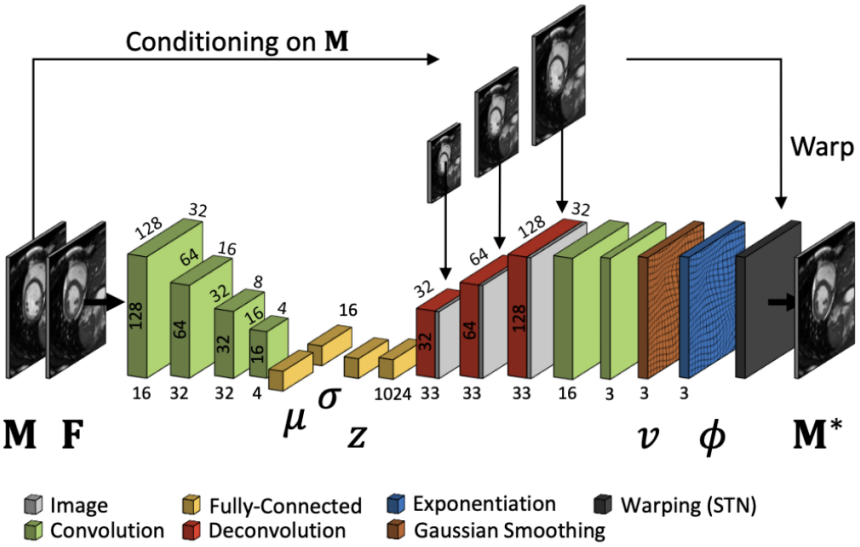


Figure 2.2: The overview of the autoencoder network proposed by Krebs et al. (2018)

cardiac image registration compared to non-diffeomorphic VoxelMorph (Balakrishnan et al., 2018) with 0.775.

Stergios et al. (2018) built an encoder-decoder network and trained image pairs to regress the deformation field and affine transformation parameters directly. Rather than incorporating maxpooling layers in the encoder stage to enlarge the receptive field, this encoder utilized dilated convolutional kernels to extract useful features within a large receptive field. Additionally, instead of generating a lower-dimensional latent vector in the encoder, the output of this encoder is concatenated input images along with all feature maps generated from five layers in the encoder respectively. There are two separate pipelines in the decoder, one pipeline only adopts global average operation to reduce those feature maps to 12-parameter parametrized affine transformation. The other pipeline is built with a squeeze excitation block (Hu et al., 2018) followed by several convolutional layers with non-dilated kernels. The squeeze excitation block is used to weigh the most important features to generate a deformation field. The final transformation field is retrieved from the composition of affine and deformation transformation. Compared to the performance of Symmetric Normalization (SyN) (Avants et al., 2008) implemented in the ANTs software package, the encoder-decoder network improved the performance from 0.838 ± 0.060 to 0.914 ± 0.022 on dice score to complete 3D MRI lung image registration.

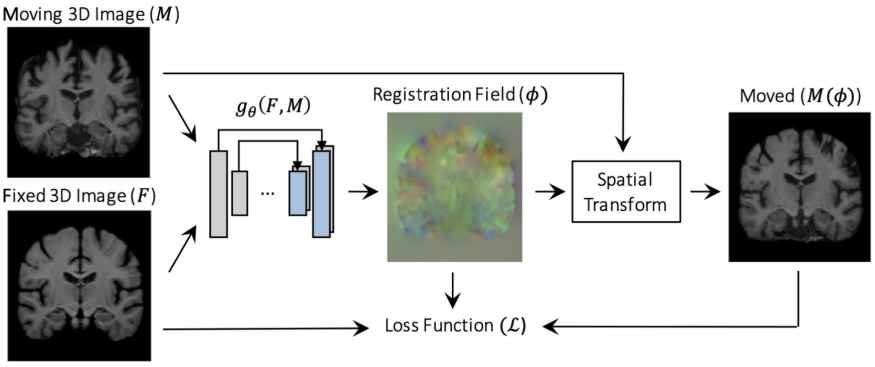


Figure 2.3: A U-Net is built to regress deformation field directly (Balakrishnan et al., 2018)

Naturally, applying skip connections in an autoencoder architecture generates a U-Net architecture, which has been proposed to complete medical image segmentation tasks. By replacing the classification layer applied with a sigmoid activation function at the end of U-Net with a regression layer without any activation function, U-Net has been used in medical image registration tasks to regress some dependent variables describing the deformation field. For example, Balakrishnan et al. (2018) proposed a U-Net architecture named VoxelMorph (Figure 2.3) to take a pair of images and regress deformation field directly. Several convolutional blocks are used to extract features from the concatenated fixed and moving images, which generated a latent vector. This operation produced efficient information representation in the encoder path. In the decoder path, the latent vectors along with upsampling produced output of similar size as the input. Skip connections are used to provide some features information in the symmetric path to predict a deformation field. A regression layer is appended at the end of the decoder to predict a dense deformation field. VoxelMorph achieved comparable results to Symmetric Normalization (SyN) (Avants et al., 2008) implemented in ANTs software package; dice scores of 0.750 ± 0.137 and 0.749 ± 0.135 respectively to complete 3D MRI lung image registration.

Similarly, Kuckertz et al. (2020) proposed a four-level U-Net to regress the deformation field directly. In terms of regressors, there are three types of inputs fed into architecture, including the pair of fixed and moving images, the pair of images with a segmentation map derived from fixed image, and the pair of images with their corresponding segmenta-

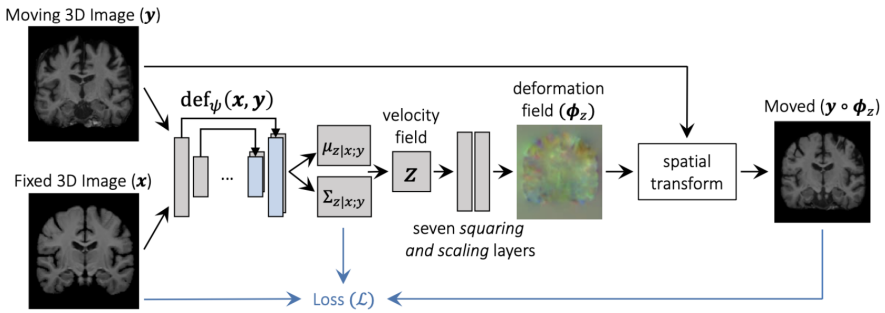


Figure 2.4: A U-Net is designed to regress intermediate variables (Dalca et al., 2018)

tion maps. Incorporating segmentation maps is beneficial to guide the network to extract useful features based on this additional structure information. The pair of images with a pair of segmentation maps achieved the best dice score of 0.91 $\pm$ 0.08 in a multi-modal pelvis image registration task. The additional segmentation map derived from fixed image was helpful to improve the performance (0.80 $\pm$ 0.15) compared to the performance (0.76 $\pm$ 0.15) obtained from only taking image pairs as a regressor.

Instead of regressing the deformation field directly, constructing U-Net architecture is aimed to regress intermediate variables to describe the deformation field. Dalca et al. (2018) built the same architecture as their previous work (Balakrishnan et al., 2018). As shown in Figure 2.4, the U-Net is trained to regress two dependent variables (i.e. the velocity field mean $\mu_{z|x;y}$ and the velocity field variance $\Sigma_{z|x;y}$), and used to describe the posterior registration probability ($p(z \mid x;y)$) according to Equation 2.16. Then, the most likely velocity field is obtained for unseen image pairs by estimating the posterior registration probability. Next, seven squaring and scaling layers are used to realize diffeomorphic integration, integrating velocity field over time to obtain the final deformation field. Compared with their own previous work (Balakrishnan et al., 2018), the performance of 3D MRI brain image registration on dice scores is improved from 0.750 $\pm$ 0.137 to 0.753 $\pm$ 0.137. This work provides a novel method and incorporates diffeomorphic integration to complete image registration as a regression problem.

Mok and Chung (2020) completed symmetric image registration without assumption of the transformation direction. In their work, the approximate function is denoted as $f_\theta(X,Y) = (\phi_{XY}^{(1)}, \phi_{YX}^{(1)})$. A five-level U-Net with skip connections is designed to regress a

velocity field and its corresponding reverse direction velocity simultaneously. Thus, two convolutional layers are appended at the end of the decoder path to output these two velocity fields respectively, including the velocity field from image X to image Y (v_{XY}) and the inverse velocity field from image Y to image X (v_{YX}). Then these two predicted velocity fields are integrated over time to generate corresponding transformation fields via scaling and squaring layers. The symmetric image registration method is helpful to preserve the topology of transformation field, thus it contributes to improving the performance of 3D brain image registration. The dice score performance of this symmetric registration work achieved 0.743 ± 0.113, while the performance of VoxelMorph (Dalca et al., 2018) is 0.693 ± 0.132.

With the advent of the transformer network (Vaswani et al., 2017), multi-head self-attention mechanism as a major component has been incorporated into the design of image registration architecture. We have mentioned that convolution operation is essentially decoupled into two major functions: feature aggregation and feature transformation. Multi-head self-attention mechanism and a series of layers with non-linear functions are able to replace convolution operations to realize these two functions respectively (Zhao et al., 2020). To be precise, feature aggregation combines feature neighbourhood regions. The convolution operation as implemented in deep network is essentially cross-correlation within a small neighbourhood around (i, j), which is defined as:

$$S(i,j) = (I * K)(i,j) = \sum_m \sum_n I(i+m, j+n) K(m,n), \tag{2.17}$$

where I and K indicate input of convolutional layer and kernel respectively. Similarly to convolution operation, a single head self-attention mechanism computes a scaled dot-product in a neighbourhood around a centre location (i, j), and is defined as:

$$y_{i,j} = \sum_{a,b \in \mathcal{N}(i,j)} softmax\left(\frac{q_{ij} k_{ab}^{\top}}{\sqrt{d_k}}\right) v_{ab}, \tag{2.18}$$

where queries, keys and values are denoted as q, k, v respectively, they are computed as linear transformations from input with different trainable matrices W_q, W_k and W_v. Location in the neighbourhood is denoted as (a, b), and a scaling factor $\frac{1}{\sqrt{d_k}}$ of keys dimension

is used to reduce large gradient input to the softmax and possible exploding gradients. Self-attention mechanism calculates the similarities between queries and keys; an operation similar to the convolution operation. Unlike convolution operations, these similarities are normalized through softmax function to output a weight matrix with values in the interval $[0, 1]$. This weight matrix is used to give a different degree of attention to features in different locations. When queries and keys are more similar, a higher weight value is applicable. Weighting the most important features to the output is similar to the behavior of a squeeze excitation block which was used in Stergios et al. (2018). Rather than only measuring the attention once, multi-head self-attention mechanism is designed to calculate attentions several times in parallel. Queries, keys and values are generated from different matrices W_q, W_k and W_v in each head. In essence, this form of attention efficiently extracts different features from different representation subspaces. The final attention is obtained by concatenating attentions estimated from multiple heads.

In the case of image registration, Z. Wang and Delingette (2021) trained a transformer network to regress the deformation field between moving and fixed images. Fixed image and moving image are split into several patches to go through the encoder and decoder respectively. The encoder takes fixed image patches to estimate similarities among each patch and outputs a relationship map. Moving image patches are fed into the decoder under the guidance of the similarities relationship map to adjust these locations of patches, then displacements are calculated between corresponding patches of moving and fixed images. Although a qualitative result shows that moved images look similar to fixed images after registration through transformer network, Z. Wang and Delingette (2021) trained MNIST dataset (Deng, 2012) to complete image registration, which is a simple dataset compared to medical image datasets. Medical image datasets normally contain many subtle features, that are often similar in appearance, and the convolutional layer is better at capturing local features such as edges. Thus, it might be better to combine convolutional layers and self-attention layers to complete medical image registration. Additionally, capturing long-range interactions is a big challenge for convolutional layers because of the receptive field. Convolutional kernels are only able to extract features within a small

neighbourhood area depending on the size of the kernel, and hence it is difficult to extract global information. In medical image registration, the larger receptive field is necessary to help find similar features and bring them to alignment in the pair of images. Since self-attention mechanism was advised to help memorize long-range information, it should be advantageous in medical image registration. J. Chen et al. (2021) introduced multi-head self-attention mechanisms into VoxelMorph network (Balakrishnan et al., 2018) to regress a deformation field directly. Several multi-head self-attention layers were constructed to connect the encoder and decoder, thus playing a role to capture similarities within a global area of feature maps. The output of self-attention components is attention maps providing similarities among features in different locations. Rather than only feeding feature maps to the decoder, this relationship information is beneficial to help the decoder to generate a deformation field. Multi-head self-attention components contribute to achieving superior performance on 3D brain image registration giving a dice score of 0.726 ± 0.130 compared to the performance of VoxelMorph (0.711 ± 0.135). The result quoted by the authors was subject to a statistical test of significance because of the slim margin of difference. It was found that the improvement was not statistically significant relative to the null hypothesis. Our expectation is that when small displacements are considered attention mechanisms may not hold an advantage over convolutional networks.

In conclusion, a series of components such as convolutional blocks, skip connections, and self-attention layers are combined to construct an architecture in different ways. These architectures we reviewed in this section are designed to complete image registration as a regression task. In terms of regressors, registration architectures commonly take a pair of images in a high-dimension spacing as input to extract features by several convolutional blocks, outputting a latent low-dimension vector or several feature maps to describe input. Sometimes additional segmentation maps are incorporated with image pairs to provide structural information to regress a deformation field. In addition, lower-dimensional features derived from image pairs by some external functions are fed into architecture as well. As far as dependent variables, a variety of architectures are normally trained to regress the deformation field directly. The velocity field which is derived

from differentiating deformation field is another optimal option to be regressed by many architectures. Furthermore, some architectures are trained to regress some intermediate variables to parametrize the velocity field, and additional layers are constructed to yield the final deformation field by integrating the velocity field.

2.5 Loss Functions

The choice of loss function is very crucial in deep learning. Intuitively, loss function is designed to guide network to optimize the set of parameters of architecture to approximate a target function from hypothesis space. Several authors have explored the nature of loss function and its relationship with generalization abilities of networks. Rosasco et al. (2004) investigated the theoretical behaviours of different loss functions by analyzing how estimation error bounds change with loss functions. And the derivation of estimation error bounds is obtained by convergence rates according to a covering number and an explicit value. These two optimal variables vary from loss functions, thus different loss functions result in different generalization abilities. A faster convergence rate indicates a better generalization ability. Additionally, the nature of loss function is explained by the flatness/sharpness of minima as well. Several works (Keskar et al., 2016; Swirszcz et al., 2016) have suggested that the flat minima lead to a better generalization ability. However, Dinh et al. (2017) reparametrized a network with flat minima to an equivalent network with sharp minima, and the pair of equivalent networks have the same generalization ability. Therefore, the flatness/sharpness of minima cannot be used to explain generalization ability alone. Furthermore, the nature of loss function has been directly visualized through a loss landscape, and how the underlying landscape of loss function affects generalization ability was explored by Li et al. (2018). Loss function is closely related to the parameters of architecture, and there exist a lot of parameters in architecture, thus the loss function is usually a high-dimensional function. A deeper depth of layers in the architecture has been noted to lead to a more chaotic landscape, which in turn degrades the generalization ability. Incorporating skip connections into a deep neural network is helpful to prevent the loss landscape from becoming chaotic. Thus, architecture with skip

connections is more likely to result in better generalization.

Furthermore, the loss function is a natural mathematical formulation of the informal aim and domain knowledge of the research (Hennig & Kutlukaya, 2007). In medical image registration, the aim of researchers is to align the same objects in moving images and fixed images, thus the mathematical formula of loss function is to compute the dissimilarities between a pair of images based on the same objects. Additionally, the loss function should be able to indicate how human observers register a pair of images. Instead of comparing pixel-wise similarities, human observers are likely to compare the similarities between a pair of images based on the salient features. And the prominent features vary from medical images containing various tissues. Importantly, the choice of loss function depends on input dataset as well. Given an image, it is conceivable that either structure, texture or objects are prominent features. A loss function tailored to a salient feature of the image will most likely lead to faster registration.

There are three loss functions commonly used in image registration, mean squared error (MSE), normalized cross-correlation (NCC) and structural similarity index metric (SSIM). To be precise, MSE computes the similarities between two images only depending on the pixel-by-pixel difference, which is defined as:

$$\mathscr{L}_{MSE} = \frac{1}{n}\sum_{i}^{n}(F_{p_i} - M'_{p_i})^2, \tag{2.19}$$

where n is the number of pixels, F_{p_i} and M'_{p_i} indicate pixel value at position p_i in fixed and moved image respectively.

Additionally, the texture tends to show repeated patterns in an image. It is reliable to calculate the similarities between two images based on corresponding patches. Therefore, NCC could be regarded as a similarity metric for texture properties of the image, which is defined as:

$$\mathscr{L}_{NCC} = -\sum_{p\in\Omega}\left(\frac{\sum_{p_i}(F_{p_i} - \overline{F_p})(M'_{p_i} - \overline{M'_p})}{\sqrt{\sum_{p\in\Omega}(F_{p_i} - \overline{F_p})^2 \sum_{p\in\Omega}(M'_{p_i} - \overline{M'_p})^2}}\right)^2, \tag{2.20}$$

where F and M' are fixed and moved images respectively, Ω is the domain of patches in

the image. A patch consists of a center pixel (p) and the surrounding pixels (p_i) thereof. $\overline{F}$ and $\overline{M'}$ are the local intensity means over a patch domain in fixed and moved images respectively.

SSIM is used to calculate the image similarities based on edges and other structural landmarks. In retinal images, thick and thin branches of blood vessels are usually dominant structures. The definition is shown as follows:

$$\mathscr{L}_{SSIM} = -\frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \tag{2.21}$$

with

$$\sigma_{xy} = \frac{1}{N}\sum_{i=0}^{N}(x_i - \mu_x)(y_i - \mu_y), \tag{2.22}$$

where x and y are moved and fixed images respectively, μ and σ indicate the mean intensity and the standard deviation of an image respectively. The derivation of the overlap part between moved and fixed images is denoted as σ_{xy}. C_1 and C_2 are constants to avoid the denominator being close to 0.

In conclusion, the choice of loss function is important to ensure a good generalization ability, and the design of loss function should consider the model design and the input dataset as well. Most registration works are likely to choose one specific loss function to complete registration while ignoring the characteristics of tissues in images. For example, Balakrishnan et al. (2019) trained the same architecture with MSE and NCC to complete 3D brain image registration respectively, arriving at the dice score of 0.727 ± 0.146 and 0.737 ± 0.139, but the performance improvement might not be statistically significant ($p\text{-}value$ = 0.8904). In addition, Mahapatra et al. (2018) combined three different loss functions with the same and fixed weights to complete retinal and cardiac image registration respectively. In terms of performances, the combination loss function trained on a GAN network achieves 0.946 and 0.85 compared to a convolutional-based DIRNet (0.91 and 0.80) on retinal and cardiac image registrations. Because the architecture and loss function are different in these two works, it is difficult to tell whether the superior performance comes from the change of architecture or loss function.

2.6 Chapter Summary

This chapter reviews background knowledge of deep learning and problem formulation of medical image registration. The definition of deep learning indicates that the design of architectures and loss functions are crucial. Also, we regarded image registration as a regression task. Next, we briefly reviewed several performance evaluation metrics, which are categorized into key point-based evaluation methods and segmentation map-based evaluation methods.

Importantly, comprehensive reviews of the state-of-the-art research in medical image registration approaches from the perspective of architectures and loss functions are provided in Section 2.4 and Section 2.5 respectively. From the perspective of the regression task, deep learning-based image registration methods use image data and the network parameters as independent variables to regress the transformation fields. Therefore, we reviewed the components of architecture from simple to complex, which are convolutional blocks, auto-encoder, U-Net, and transformer-based architectures. Then, we analyzed how architecture components perform differently in terms of the regressors (i.e. a pair of images or indirect representations such as feature maps) and dependent variables (i.e. deformation field or its variants such as velocity field).

Also, this chapter emphasizes the importance of loss functions in medical image registration. This review describes the relationship between loss functions and generalization abilities. It suggests that loss function should be able to indicate how human observes register a pair of images, thus we hypothesized that neural networks achieve better performances when loss functions match the characteristics of images.

In conclusion, this chapter provides a theoretical background and comprehensive analysis of current registration works. Since it is still unclear how architectures, loss functions and datasets affect registration performance, this thesis sets out to design and conduct a series of experiments to explore the answer.

Chapter 3

Experimental Design

In this chapter, we describe details of the experiments conducted in our work, including the design and training of networks, generation of datasets, and evaluation of the performance of the registration achieved. Additionally, a description of the statistical significance test conducted to determine the extent to which the observed improvement between different experimental settings could be due to chance.

3.1 Networks

This work aims to explore the contributory factors of performance improvement attributable to network architectural components. Specifically, we explore the factors responsible for the relative performance of transformer-based networks in comparison with convolutional neural networks. To this end, we train ViT-V-Net (J. Chen et al., 2021) and CNN examples. Figure 3.1 shows the overview of ViT-V-Net architecture. The backbone of architecture is a five-level U-Net. Unlike a convolutional-based network, it incorporates self-attention mechanism components to connect the encoder and decoder of U-Net. Therefore, we remove the self-attention mechanism components to construct a pure convolutional-based network.

In order to explore how architecture sizes affects the performance of registration, we build a family of transformer-based networks with different sizes and corresponding convolutional-based networks. Table 3.1 shows the number of neurons in each layer of different architectures. We build large, mid and small size of convolutional-based and transformer-based

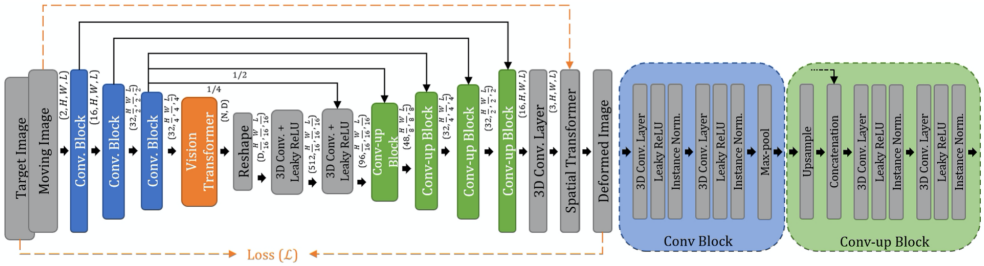


Figure 3.1: The overview of ViT-V-Net architecture (J. Chen et al., 2021)

networks respectively. Additionally, we build three different heads of multi-head self-attention mechanisms for transformer-based networks, we set the number of heads to 12, 9, and 6 respectively in each size of the transformer-based network.

Table 3.1: Details of constructing different size architectures

	Small size				Mid size				Large size			
	Conv-based	Transformer-based			Conv-based	Transformer-based			Conv-based	Transformer-based		
Encoder	(8,16,16)				(9,18,18)				(16,16,32)			
Head	-	6	9	12	-	6	9	12	-	6	9	12
MLP-dimension	-	1536			-	2304			-	3072		
Hidden latent vector	-	126			-	189			-	252		
Connect layer	256				384				516			
Decoder	(48,24,16,16,8)				(72,36,18,18,9)				(96,48,32,32,16)			
Parameters	0.22M	3.33M	4.69M	5.74M	0.44M	7.25M	10.42M	12.78M	0.88M	13.27M	18.69M	24.12M

3.2 Implementation Settings

Our work is implemented on the Pytorch (Paszke et al., 2019) framework. In order to have a fair comparison, we train convolutional-based and transformer-based networks by following the training procedures provided by the original work (J. Chen et al., 2021). Table 3.2 shows the details of hyperparameters setting during training networks.

Table 3.2: Hyperparmaeters of training stage

Optimizer	Learning rate	Learning rate decay	Dropout	Epochs	Regularization	Batch Size
ADAM	$1e^{-4}$	Polynomial(0.9)	0.1	500	0.02	2

3.3 Datasets Description

Given that one of our research questions (as stated in Section 1.3) is to explore how different loss functions interact with different datasets to affect registration performance, we created two different image datasets with various objects (i.e. retina and brain) for

use in a series of experiments. There is a large number of publicly available retinal and brain image datasets, but most of them do not provide image pairs information. Therefore, they are not suited to the development and evaluation of registration algorithms. In our work, we create our own datasets by combining several image datasets and generating image pairs. Section 3.4 provides more detail about the motivation and process. This section gives a brief description of the salient features of publicly available retinal and brain datasets used in our work respectively.

3.3.1 Retinal Image Datasets

The following are brief descriptions of eight retinal image datasets used in this thesis.

HRF Dataset

High-Resolution Fundus (HRF) dataset (Budai et al., 2013) consists of 45 images, including 15 images acquired from healthy subjects, 15 images with diabetic retinopathy and 15 images acquired from glaucomatous patients. The size of the image is 3504 $\times$ 2336, captured at 45° field of view.

DIARETDB1 Dataset

DIARETDB1 dataset (Kauppi et al., 2007) consists of 89 images including 85 images with diabetic retinopathy such as microaneurysms and 4 normal retinal images. The size of the image is 1500 $\times$ 1152 pixels, captured at 50° field of view.

DRIVE Dataset

Digital Retinal Image for Vessel Extraction (DRIVE) dataset (Staal et al., 2004) was generated from 400 diabetic retinopathy subjects who are 25-90 years old. The size of the image is 768 $\times$ 584 pixels, 8 bits per pixel and captured at a 45° field of view. There are 40 images randomly selected to establish the DRIVE dataset. This dataset consists of 7 images with mild early diabetic retinopathy and 33 images without any signs of diabetic retinopathy.

MESSIDOR Dataset

Methods to Evaluate Segmentation and Indexing Techniques in the field of Retinal Ophthalmology (Messidor) dataset (Decencière et al., 2014) consists of 1,200 eye fundus colour images including 800 images with pupil dilation and 400 images without dilation. These images are captured at 45° field of view. Each image is labelled with medical diagnosis information including retinopathy grade and risk of macular edema. This dataset is divided into 12 subsets, each containing 100 images. The sizes of images are varied: 1440 × 960 pixels, 2240 × 1488 pixels and 2304 × 1536 pixels. It includes images captured from healthy subjects and images with different stages of the seriousness of lesions such as microaneurysms, exudates and hemorrhages, etc.

E-ophtha Dataset

E-ophtha Dataset (Decencière et al., 2013) contains 463 images. It is divided into two datasets according to the different diabetic retinopathy lesions: exudates and microaneurysms. All lesions have been manually outlined by ophthalmologists. One dataset, e-ophtha EX with exudate lesions, includes 47 images with pathology and 35 images without pathologies. The other, e-ophtha MA with microaneurysms, includes 148 images with pathology and 233 healthy images. These images are captured at 40° field of view. The sizes of images are varied: 2544 × 1696, 2048×1360, 1400×960 and 1504 × 1000.

CHASE_DB1 Dataset

CHASE_DB1 Dataset (Fraz et al., 2012) was generated by the Child Heart and Health Study in England (CHASE). This dataset demonstrates that early cardiovascular disease could cause retinal vessel tortuosity. It contains 28 images with the size of 999 × 960, captured at 30° field of view.

Longitudinal Diabetic Retinopathy Screening Dataset

Longitudinal diabetic retinopathy screening (LDRS) dataset (Adal et al., 2015) was taken from 70 diabetic patients. It contains 1120 images with a resolution of 2000 × 1320 pixels, captured at a 45° field of view. In this dataset, four fundus images are captured

from each eye with different fields respectively; these fields are macula-centred, optic nerve-centered, superior and temporal regions. These images with smaller overlap are normally used in mosaicing applications. They are registered to generate a fundus mosaic with a larger field of view.

FIRE Dataset

Fundus Image Registration (FIRE) Dataset (Hernandez-Matas et al., 2017) is relevant for retinal image registration studies. It consists of 129 retinal images taken from 39 patients. These retinal images are arranged in 134 pairs, each image pair has been labelled with ground truth correspondence (see sample images in Figure 3.2). The size of the image is 2912×2912 pixels, captured at a 45° field of view. The database is divided into three categories according to overlap area and the situation of anatomical changes. Category S consists of 71 image pairs that have a large overlap area ($> 75\%$), and they lack anatomical changes. Category P consists of 49 image pairs in which the overlap area is smaller than 75 % and they also lack anatomical changes. Category A consists of 14 image pairs that have large overlap areas ($> 75\%$) and anatomical changes, such as vessel tortuosity, microaneurysms, cotton-wool and spots.

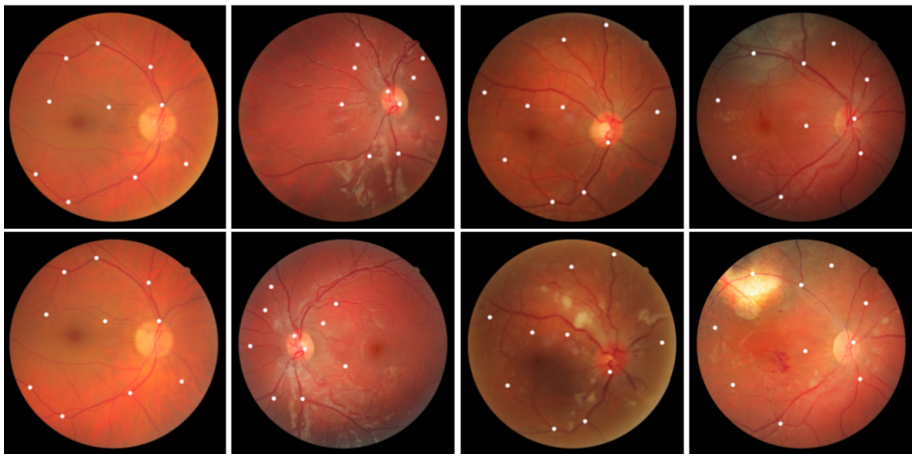


Figure 3.2: Retinal image pairs of FIRE dataset

Summary of Retinal Image Datasets

In conclusion, Table 3.3 presents a summary of eight datasets we used in our work. The last two datasets are specifically designed for registration task, while the other are com-

monly used in vessel segmentation tasks and do not provide image pair information. Figure 3.3 represents five common abnormal findings in retinal images. These abnormal findings are different from normal fundus parts in color and brightness. Figure 3.4 shows a sample image from each dataset.

Table 3.3: Publicly available retinal image datasets

Dataset	Images	Resolution	Field of View	Lesions
HRF	45	3504 × 2336	45°	Diabetic retinopathy, Glaucoma
DIARETDB1	89	1500 × 152	50°	Diabetic retinopathy (such as Microaneurysms)
DRIVE	40	565 × 584	45°	Mild early diabetic retinopathy
MESSIDOR	1200	1440 × 960, 2240 × 1488, 2304 × 1536.	45°	Microaneurysms, exudates, hemorrhages.
E-ophtha	463	2544×1696, 2048×1360, 1400×960, 1504×1000.	40°	Exudates, microaneurysms.
CHASE_DB1	28	999 × 960	30°	Vessel tortuosity
LDRS	1120	2000 ×1320	45°	Diabetic retinopathy
FIRE	129	2912 × 2912	45°	Vessel tortuosity, microaneurysms, cotton-wool, spots

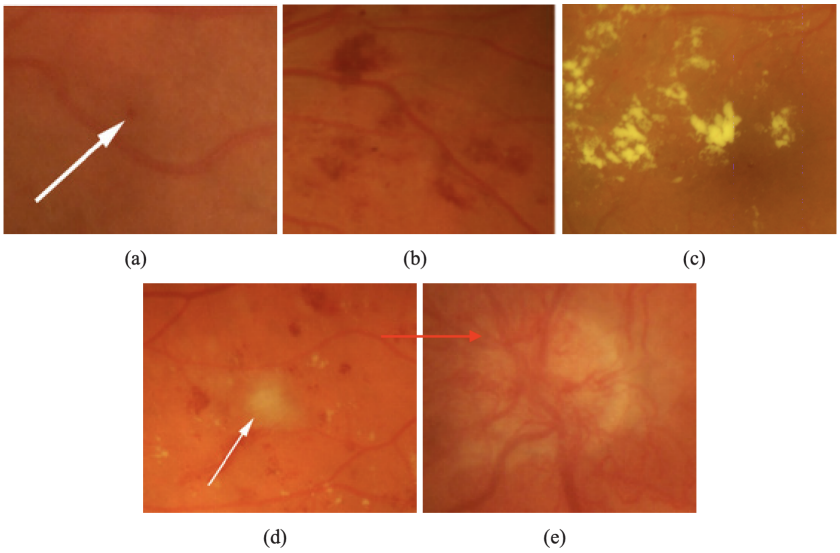


Figure 3.3: Five common abnormal findings in diabetic retinopathy, ordered by the increase stage of seriousness (Kauppi et al., 2007): (a) microaneuryms, (b)hemorrhages, (c)hard exudates, (d)soft exudates, (e)neovascularization.

3.3.2 Brain Tumor Datasets

There are 3,267 MRI slice images in the brain tumor dataset (Bhuvaji et al., 2020). It is actually a classification dataset. The brain tumor dataset is divided into four categories

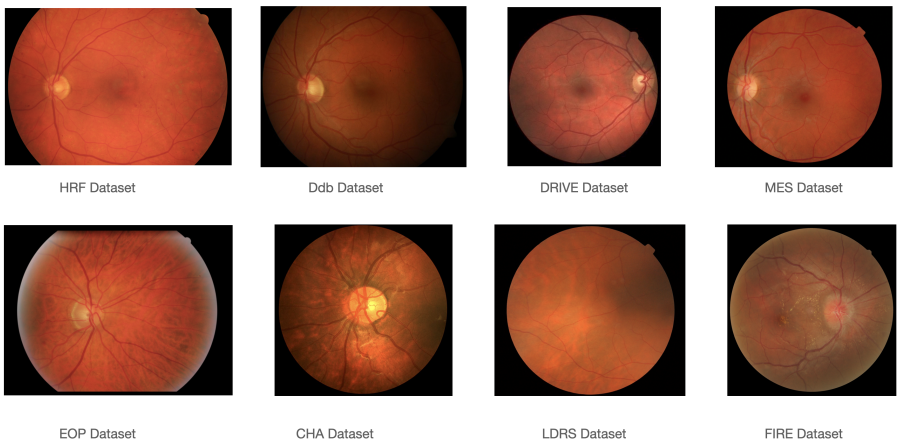
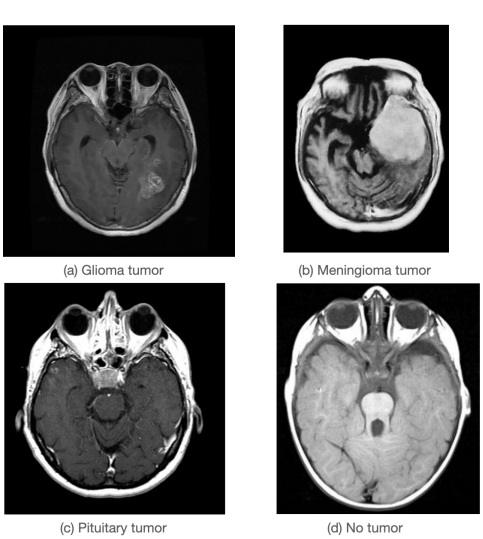
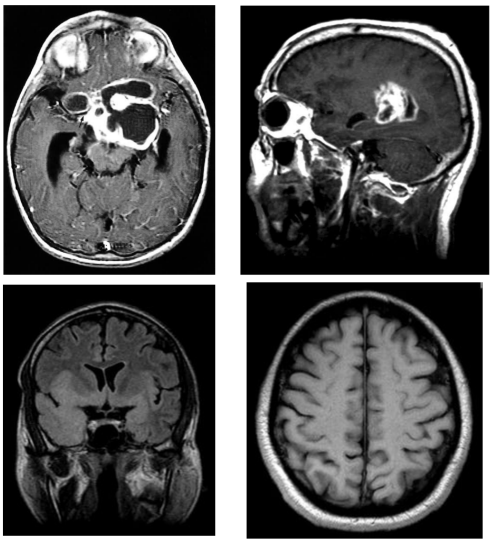


Figure 3.4: Samples from each retinal image dataset

according to the type of tumor, including 926 glioma tumor images, 937 meningioma tumor images, 901 pituitary tumor images and 500 healthy brain images. Figure 3.5a shows examples from different categories. The resolution of these images varies from 167×175 to 1446×1375; in total 440 different resolutions. In addition, there exists a variety of viewpoints in this brain image datasets; Figure 3.5b shows some examples.



(a) Examples from different categories



(b) Examples from different viewpoints

Figure 3.5: Some brain image examples

3.4 Dataset Preprocessing

In order to address the problem of the paucity of publicly available image registration datasets, we generated our own retinal and brain datasets respectively. The overview of the process is described as follows. First, we combined several image datasets to create more images. Then, we applied realistic transformations to these images to generate corresponding transformation images, which can be paired to complete the image registration task. Lastly, we split these images based on stratified sampling into training, validation and test datasets.

3.4.1 Dataset Generation

For retinal image registration, we combine HRF, DIARETDB1, DRIVE, MESSIDOR, E-ophtha, CHASE_DB1 and LDRS datasets, resulting in a total of 2,985 images. Figure 3.6 shows data distributions of each dataset and combination dataset. There exist domain differences among these datasets because of different abnormal findings, different fields of view and different acquisition devices, etc. These datasets represent various data distributions. Combining datasets is beneficial to create a robust retinal dataset containing rich information for training. This combination dataset actually represents a more robust data distribution. It covers abundant underlying data patterns in retinal fundus images, including retinal vessel structures, abnormal findings and structural lesions, etc. Given unseen data, it is more likely to predict a closer value to desired output in the robust probability distribution of the combined dataset. In order to have a fair and reasonable comparison, we chose the same amount of brain images to complete brain image registration.

3.4.2 Image Resizing

Since our implementation is to train ViT-V-Net (J. Chen et al., 2021) and pure CNN examples, we followed the input size of ViT-V-Net, which is $160 \times 192 \times 224$ for 3D images. Therefore, we need to resize 2D retinal and brain images to the size of 192×224. In order to avoid introducing unexpected deformations, we kept the ratio of the original resolution by setting the same resize factor on width and height. Then we extracted $192 \times$

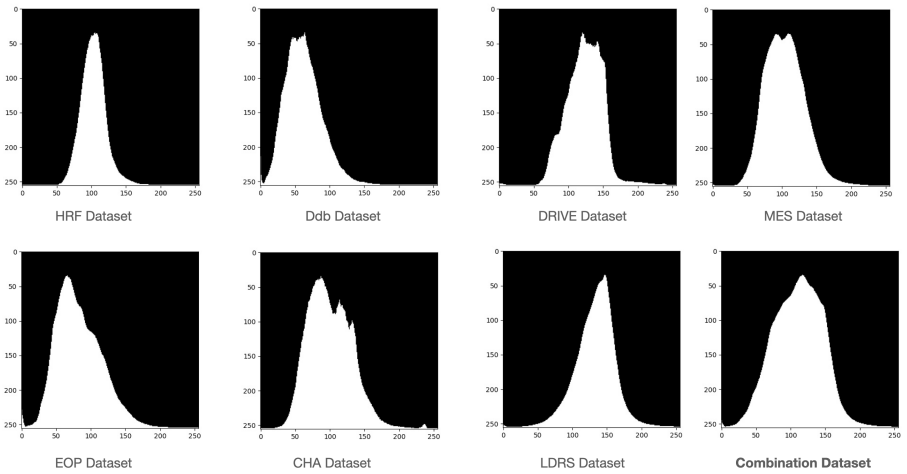


Figure 3.6: Distributions of each datasets and combination dataset

224 regions of interest (RoI) on these resized images.

3.4.3 Image Pair Generation

In adult humans, the entire retina is approximately 72% of a sphere about 22mm in diameter. The maximum misalignments in clinical images is in the range of ± 1.2 mm (De Silva et al., 2021), therefore we calculate the maximum movement pixels (px_{max}) as follows:

$$px_{max} = \frac{N}{22} \times 1.2, \tag{3.1}$$

where N is the number of pixels in diameter of the retinal image.

We generated three various datasets containing small-, mid- and large-displacement image pairs respectively to explore how transformer-based network performs on different scales of transformation field. Let us take an example of how to generate small-displacement image pairs. In order to constrain unrealistic transformation in the real world, the average of maximum movement pixels in these datasets is 15. In terms of translation, the displacement is below five pixels in each direction. As for rotation, the rotation center is limited to the 192×224 regions of interest (RoI), and rotation degree is limited to 0.1 rad. In terms of elastic transformation, it is a local transformation, the displacements for each pixel are different. Thus, we randomly choose a series of values from a Gaussian distribution, and these values are regarded as displacements for each pixel. In addition, we randomly set standard deviation (σ) and mean (μ) to generate dif-

ferent Gaussian distributions. The value of σ is 0.08 to 0.1 times the width of the RoI, and the value of μ is limited to 1 to 1.1 times the width of the RoI. More transformation parameters are presented in Table 3.4. Therefore, we generated three retinal fundus image registration datasets and three brain slices image registration datasets respectively.

Table 3.4: Summary of parameters to limit transformation

Parameters	Small displacement	Medium displacement	Large displacement
Translation limitation(pixels)	[-5,5]	[-10,10]	[-10,10]
Rotation degree	[-5.73, 5.73]	[-11.46, 11.46]	[-11.46, 11.46]
σ	[192 $\times$ 0.08, 192$\times$ 0.1]		
μ	[192, 192$\times$ 1.1]		
maximum displacement (pixels)	15	20	25

After acquiring corresponding transformation images, we extract regions of interest (RoI) with the size of 192×224 pixels from the original images and transformation images respectively.

3.4.4 Dataset Split

In order to ensure that the training, validation and test subsets have a similar distribution, stratified sampling is used to split the combination dataset into three subsets. According to the ratio of 7:2:1, image pairs are split into training, validation and test subsets respectively. Table 3.5 and Table 3.6 show details about splitting retinal and brain datasets into subsets according to stratified sampling respectively.

Table 3.5: Details of splitting retinal images into training, validation and test subset

Dataset	HRF	Ddb	DRIVE	MES	E-ophtha	CHA	LDRS	Total
Training	32	62	28	840	324	20	784	2,090
Validation	9	18	8	240	92	6	224	597
Test	4	9	4	120	47	2	112	298

Table 3.6: Details of splitting brain images into training, validation and test subset

Dataset	Glioma	Meningioma	Pituitary	Health	Total
Training	636	592	625	237	2,090
Validation	181	169	178	67	595
Test	92	84	90	34	300

3.5 Evaluation and Statistical Significance

In our work, we use Dice Score to evaluate registration performance. Statistical significance test is used to query the improved performance or otherwise obtained in the experiments. This strategy has been adopted because the performance difference is usually small in magnitude. The dice score used in medical image registration ranges in value from 0 to 1. To this end, under the hypothesis that the mean performance under two experimental conditions is the same, a *t-test* and a selected significance level can be used to assert the possibility of any observed difference being due to chance (Swinscow, Campbell, et al., 2002). The resultant *t-value* is expressed as (Devore, 2008),

$$t = \frac{\bar{x}_1 - \bar{x}_2}{\sqrt{\left(s^2\left(\frac{1}{n_1} + \frac{1}{n_2}\right)\right)}}, \tag{3.2}$$

where t is the *t-value*, $\bar{x}_1$ and $\bar{x}_2$ are the means of two groups' performances, and n_1 and n_2 indicate the number of each group of performances respectively. s^2 is the pooled standard error which is calculated by the standard deviation of each group. Equation 3.2 indicates that the *t-value* is related to three key components: the mean difference between the performance of two groups of experiments, the standard deviation of each group, and the number of performances recorded in each group. The large *t-value* means a large difference between the two groups. Furthermore, a *p-value* (Thiese et al., 2016) is the probability of rejecting the null hypothesis. To estimate *p-value*, we can use the t-distribution table to find the corresponding value based on the t-value. We specify a non-directional (two-tailed) test at a significance level of 0.05. Thus, at a $p\text{-}value \leq 0.05$ we reject the null hypothesis of equal means since there is insufficient evidence to admit it and assert that the observed difference is statistically significant. However, we add the caveat that the difference could have arisen because of other factors that experimental design may not have taken into consideration.

3.6 Chapter Summary

Based on our research questions in Section 1.3, we designed a series of experiments to explore how performances are affected by architectures, loss functions and datasets respectively. This chapter provides details about the general experimental settings including network components and implementation, dataset description and preprocessing.

In terms of networks, we explored the relative performance of transformer-based networks in comparison with convolutional networks. Therefore, we chose ViT-V-Net (J. Chen et al., 2021) as the backbone of transformer-based networks, then we removed multi-head self-attention components as our convolutional-based network. Moreover, in order to explore how the size of networks affects performance, we built small, medium and large sizes of architectures in respective transformer-based and convolutional-based networks. Additionally, we built 6-head, 9-head and 12-head self-attention components in transformer-based networks to investigate the relationship between the number of multi-head self-attention components and performances.

In terms of datasets, we generated our own retinal and brain registration datasets respectively to explore the relationship between loss functions and datasets. Although there is a large number of publicly available medical images, most of them do not provide image pair information. In order to address the paucity of registration datasets, we collected 2,985 images from different datasets of two human organs (i.e. retina and brain). These combined images represent a robust data distribution, which is beneficial to contain rich information for training. Then we applied realistic transformations to derive transformed images, and we paired corresponding original images and transformed images as our registration datasets. Furthermore, in order to explore how transformer-based networks address the limitation of convolutional-based networks in the task of image registration, we generated three different datasets consisting small-, mid-, and large-displacement image pairs.

Chapter 4

Results

In this thesis, a series of controlled experiments and ablation studies are conducted to explore the relationship between the registration performance of architectures, loss functions, datasets and maximum displacements of image pairs respectively. This chapter presents the experimental results and their comparative analysis. In essence, this chapter provides answers to the research questions listed in section 1.3 respectively.

For clear descriptions, before describing the result of each group of experiments, we reviewed specific experimental settings based on four basic parts of controlled experiments: independent variable, dependent variable, constants and control group. In addition, there are multiple independent variables in some of our experiments, represented in a factorial design table showing multiple independent variables and their corresponding levels. Besides, some experiments do not have constants and control groups.

For convenience, the abbreviations used in this chapter are given in Table 4.1. In addition, the symbol "$*$" suffixed after a *p-value* indicates that the improved performance is statistically significant. Considering that most performance differences between two groups of experiments are statistically significant, we only pointed out those performance differences which fail to reject the null hypothesis for the sake of avoiding cumbersome description.

Table 4.1: List of used abbreviations

Abbr.	abbreviation
CNN-based	Convolutional-based network
TF-based	Transformer-based network
CNN-{Small- /Mid- /Large}	Small- /Mid- /Large Convolutional-based network
TF-{Small- /Mid- /Large}	Small-/ Mid-/ Large Transformer-based network
Diff	Performance difference
RS/RM/RL	Retinal Small-/ Midium- / Large-displacement dataset
BS/BM/BL	Brain Small-/ Midium- / Large-displacement dataset

4.1 Network Architecture Components and Performance

The results of the experiments reported in this section explore the notion of network sizes and components versus image registration performance, using two different architectures, viz. convolutional networks and transformer-based networks. A set of three network sizes, namely large, medium and small, were developed for each of the two architecture types. The construction details have been described in Table 3.1. We hypothesize that both convolutional-based and transformer-based architectures with large sizes are able to improve the generalization ability of the registration network.

4.1.1 Convolutional-based Network: Size vs Performance

In the case of convolutional-based architecture trained to complete retinal image registration, Table 4.2 represents dice scores achieved by different sizes of architectures trained with various loss functions at different scales of transformation fields, and Table 4.3 shows the results of statistical significance test including differences and *p-values*. In this group of controlled experiments, the size of architecture is the independent variable and the dependent variable is dice score performance, whereas there are two extraneous variables that would affect the dependent variable as well: loss functions and scale of transformation fields. Figure 4.1 represents the experimental settings.

Therefore, we controlled these two variables to explore how the size of architectures affects the performance of retinal image registration in different scenarios. In order to have a clear comparison and observation, Figure 4.2 and Figure 4.3 are derived from Table 4.2 to show how average dice scores change with sizes of architectures under two controlled conditions (i.e. scales of deformation fields and loss functions) respectively.

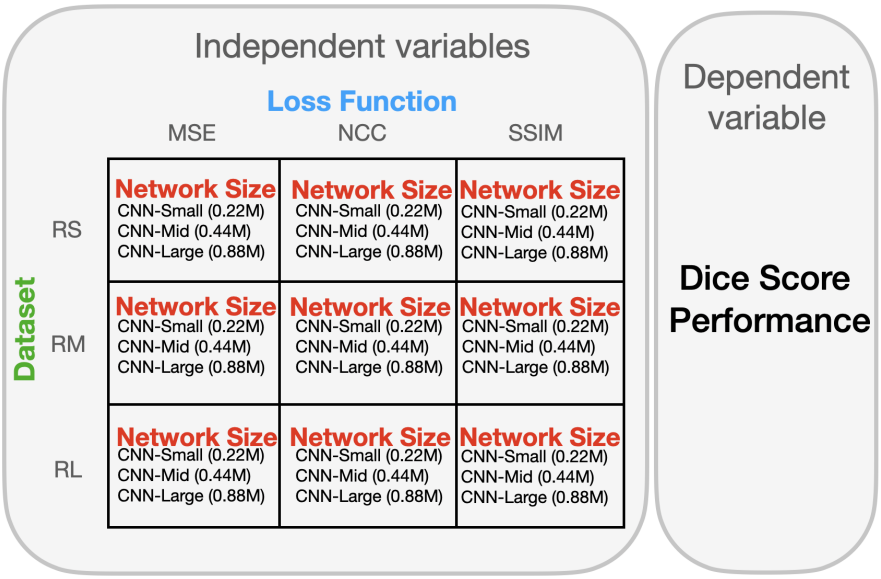


Figure 4.1: Experimental setting of Section 4.1.1 to explore the relationship between the size of convolutional-based network and dice score performance

Table 4.2: Dice scores for convolutional-based architectures at different sizes and different loss functions to complete retinal image registration at various scales of transformation fields

		Small size (0.22M)	**Medium size (0.44M)**	**Large size (0.88M)**
Small-displacement (Before Registration: 0.5048)	MSE	0.7560±0.0065	0.7717±0.0029	0.8063±0.0011
	NCC	0.7807±0.0035	0.6464±0.1013	0.7895±0.0141
	SSIM	0.8182±0.0027	0.8114±0.0163	0.8334±0.0166
Medium-displacement (Before Registration: 0.4516)	MSE	0.7005±0.0030	0.7077±0.0133	0.7526±0.0066
	NCC	0.7899±0.0135	0.5948±0.0281	0.7581±0.0258
	SSIM	0.7782±0.0098	0.7820±0.0067	0.8275±0.0099
Large-displacement (Before Registration: 0.4489)	MSE	0.6609±0.0305	0.6989±0.0045	0.7411±0.0062
	NCC	0.4457±0.0377	0.5698±0.0065	0.7645±0.0084
	SSIM	0.7481±0.0300	0.7776±0.0135	0.8131±0.0161

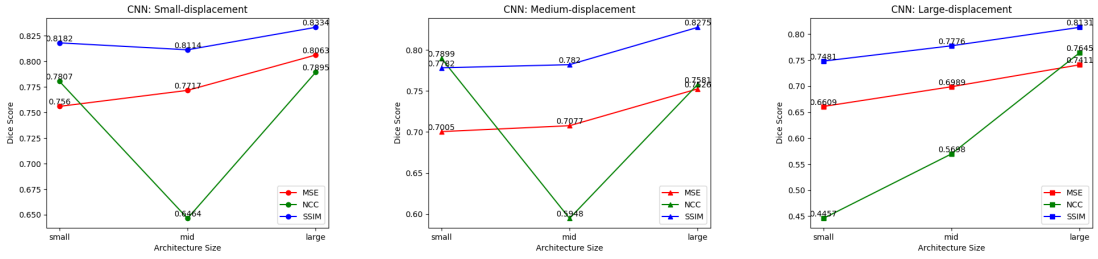
Table 4.3: Dice score differences and p-values for convolutional-based architectures at different sizes to complete retinal image registration at various scales of transformation fields and different loss functions

			Medium-Small	**Large-Medium**	**Large-Small**
Small-displacement	MSE	Diff	0.0157	0.0346	0.0503
		p-value	$< 0.0001^*$	$< 0.0001^*$	$< 0.0001^*$
	NCC	Diff	-0.1343	0.1431	0.0088
		p-value	0.0022^*	0.0014^*	0.1087
	SSIM	Diff	-0.0068	0.0220	0.0152
		p-value	0.2638	0.0181^*	0.0228^*
Medium-displacement	MSE	Diff	0.0072	0.0449	0.0521
		p-value	0.1575	$< 0.0001^*$	$< 0.0001^*$
	NCC	Diff	-0.1951	0.1633	-0.0318
		p-value	$< 0.0001^*$	$< 0.0001^*$	0.0080^*
	SSIM	Diff	0.0038	0.0455	0.0493
		p-value	0.3806	$< 0.0001^*$	$< 0.0001^*$
Large-displacement	MSE	Diff	0.0380	0.0422	0.0802
		p-value	0.0036^*	$< 0.0001^*$	$< 0.0001^*$
	NCC	Diff	0.1241	0.1947	0.3188
		p-value	$< 0.0001^*$	$< 0.0001^*$	$< 0.0001^*$
	SSIM	Diff	0.0295	0.0355	0.0650
		p-value	0.0237^*	0.0003^*	$< 0.0001^*$

In the case of controlling scales of transformation fields, Figure 4.2 represents three line graphs to explore the relationship between the size of convolutional-based architectures and dice scores at three different scales of transformation fields: small, medium and large displacement respectively. In each subgraph, three lines describe the relationship between performance and architecture size for three respective loss functions (MSE, NCC and SSIM). The implications of the three lines are described in conjunction with the relative differences and their statistical significance as shown in Table 4.3.

As shown in Figure 4.2a, for small displacement, the performance of convolutional-based architecture trained with SSIM loss decreased from small- to medium-sized network (0.8182 to 0.8114) and then increased for a large-sized network. The statistical significance test (*p-value* = 0.2638) suggests that the decrease may not be due to the difference in size (at least as defined in our experiment). However, the increase in performance (0.8114 to 0.8334) achieved by the large-sized network is statistically significant and would suggest that the network size influences the performance. Also, the performance of convolutional-based architecture trained with MSE loss steadily increased from small-sized to medium-sized and then large-sized networks. The increases are also statistically significant. Additionally, the results for convolutional-based networks trained with NCC loss decreased sharply from small- to medium-sized networks (0.7807 to 0.6464), the decrease in performance (*p-value* = 0.0022) is statistically significant. The performance then increased to 0.7895 in the large-sized network. As for statistical significance tests, the *p-value* (0.0014) suggests that this improvement is influenced by the size of the network. However, compared to the small-sized network, the *p-value* (0.2087) indicates that the mild increase in performance from small- to large-sized networks (0.7807 to 0.7895) is not statistically significant. Similarly to Figure 4.2a, Figure 4.2b depicts the results of the convolutional-based architectures trained with three loss functions for medium-displacement retinal image registration. The performance of SSIM remained stable from small- to medium-sized networks (0.7782 and 0.7820) and then went up to 0.8275 achieved by the large-sized network. The result of the statistical significance test achieved a lower *p-value* than 0.0001 and would suggest that the growth in performance

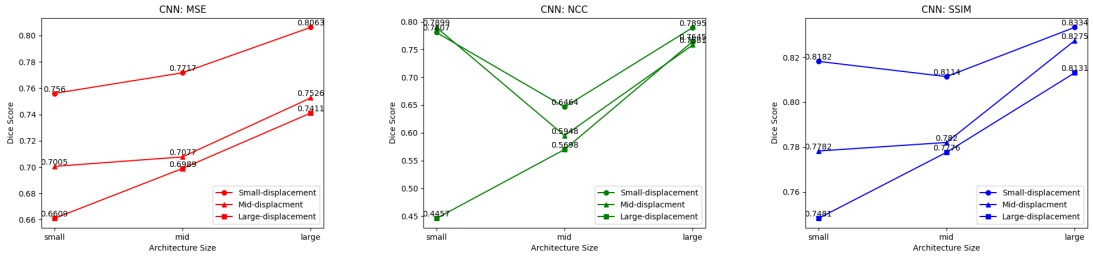
is caused by the increased size of the network. Likewise, the performances of networks trained with MSE loss rose slightly from small- to medium-sized networks (0.7005 to 0.7077), then the performance rose to 0.7526 achieved by the large-sized network. In this case, the increase is not statistical significant (*p-value* = 0.1575) between small- and medium-sized networks while the performance improvement from medium- to large-sized networks is statistically significant (*p-value* <0.0001). In terms of the results of convolutional-based networks trained with NCC loss function, the medium-sized network degraded the performance significantly compared to the small-sized network (0.7899 to 0.5948). Even though the performance achieved an improvement to 0.7581 by the large-sized network, the large-sized network did not behave as well as the small-sized network. Compared to the small-sized network, these performance decreases in medium- and large-sized networks are statistically significant. Lastly, for large displacement, the performances of convolutional-based networks at different sizes are represented in Figure 4.2c. Both the performances of convolutional-based networks trained with SSIM and MSE loss functions showed gradual increases from small- to medium- and then to large-sized networks. Also, training with NCC grew the performance rapidly from 0.4457 to 0.5698 and to 0.7645; achieved by small-, medium- and large-sized networks respectively. Importantly, these improved performances acquired from all cases in large-displacement retinal image registration are statistically significant.



(a) Small-displacement **(b)** Medium-displacement **(c)** Large-displacement

Figure 4.2: Average dice scores of convolutional-based architectures at different sizes interacted with different loss functions to complete retinal image registration

Under the controlled condition of loss functions, Figure 4.3 shows how average dice scores change with different sizes of CNN-based architectures to complete retinal image registration trained with a set of loss functions: MSE, NCC and SSIM. Each line in the



(a) MSE **(b)** NCC **(c)** SSIM

Figure 4.3: Average dice scores of convolutional-based architectures at different sizes interacted with scales of transformation fields to complete retinal image registration

subgraph represents the change of performance at small-, mid- and large-displacement transformation fields respectively. First, the performances of convolutional-based networks trained with MSE loss are shown in Figure 4.3a. Overall, all three lines show an upward trend. Both the performances of convolutional-based networks in small- and large-displacement transformation fields increased gradually from small- to medium- and to large-sized networks; all increases are statistically significant. In contrast, for mid-displacement image registration, the performance increased mildly from small- to medium-sized networks (0.7005 to 0.7077) and then it increased sharply from medium- to large-sized networks (0.7077 to 0.7526). The mild increase is not statistically significant (*p-value* = 0.1575), which suggests that the larger size is not helpful to improve performance in this case. However, with the increasing size of networks, these improved performances are statistically significant with a lower *p-value* than 0.0001. Second, Figure 4.3b shows the change in convolutional-based networks performance trained with NCC loss. The performance plunged to hit the lowest points at medium-sized networks both in small- and mid-displacement transformation fields, whereas the performance of large-sized networks went up to a comparable dice score as small-sized networks. However, the performance showed rapid growth from small- to medium- then to large-sized networks in the large-displacement transformation field. As for the results of statistical significance tests, all performance differences are successful to reject the null hypothesis except the difference (*p-value* = 0.1087) between the small- and large-sized networks in medium-displacement. Lastly, the performances of convolutional-based networks trained with SSIM are represented in Figure 4.3c. For small-displacement retinal image registration,

the performance decreased gently from small- to medium-sized networks while the performance increased from medium- to large-sized networks. The performance achieved 0.8182, 0.8114, and 0.8334 at small-, medium- and large-sized networks respectively. In comparison, the performance increased slightly from small- to medium-sized networks (0.7782 to 0.7820) and then surged to 0.8275, achieved by the large-sized network with the mid-displacement transformation field. Similarly, in large-displacement registration, the performance showed a steady upward trend from small- to medium- and to large-sized networks. These increases in the performance of the adjacent size networks (0.7481 to 0.7776 to 0.8131) are statistically significant.

4.1.2 Transformer-based Network: Size vs Performance

Similarly to the previous subsection, a set of transformer-based networks constructed with various sizes are trained to complete retinal image registration. Figure 4.4 represents experimental settings based on independent and dependent variables.

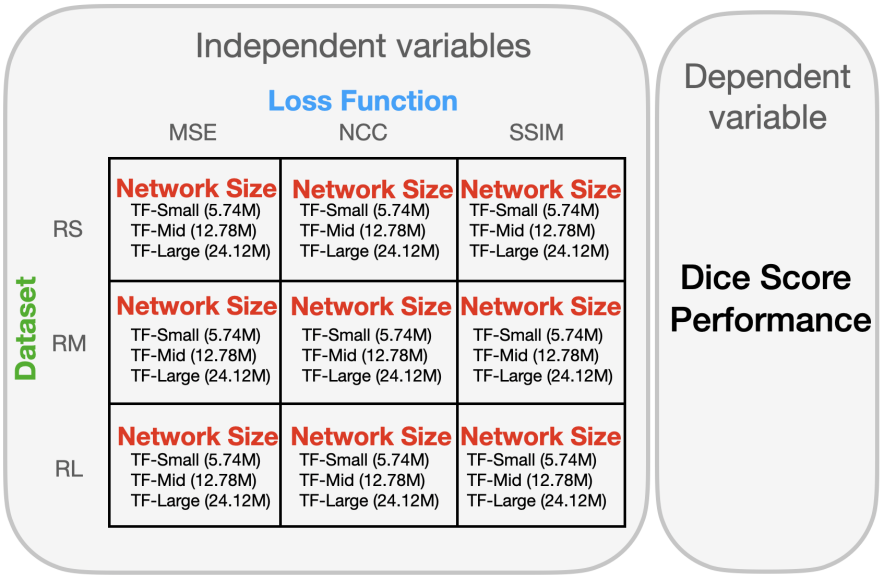


Figure 4.4: Experimental setting of Section 4.1.2 to explore the relationship between the size of transformer-based network and dice score performance

Table 4.4 and Table 4.5 represent the registration performance in dice score and the statistical significance test respectively. Additionally, how the performance changes with the size of transformer-based architectures are shown in Figure 4.5 and Figure 4.6 which are derived from Table 4.4 according to different controlled conditions.

Specifically, Figure 4.5 containing three subgraphs depicts the relationship between performance and the size of architecture trained with various loss functions under a controlled scale of transformation field (small-, mid- and large-displacement). As shown in Figure 4.5a, at small-displacement transformation field, the performances of transformer-based networks trained with SSIM gradually grew from small- to medium- and to large-sized networks (0.8141 to 0.8275 and to 0.8506 in dice score). In this case, all increases are statistically significant. However, the medium-sized network trained with MSE degraded slightly compared to the small-sized network (0.7588 to 0.7560), the performance difference is not statistically significant (*p-value* = 0.7274) and would suggest that the mild decrease is not due to the size of networks. Then, the performance of the large-sized network rose to 0.8066 and the growth is statistically significant (*p-value* $<$0.0001). Conversely, the performances of transformer-based networks trained with NCC rose steadily over the set of different sizes. The performance increased from 0.6224 to 0.6999 and to 0.7928 in dice scores; achieved by small-, medium- and large-sized transformer-based networks respectively. Importantly, these increases are recognised as statistically significant, which suggests that the performance is affected by the size of the network in this case. Similarly to Figure 4.5a, Figure 4.5b shows the performances of transformer-based networks at medium-displacement transformation field. The performances of networks trained with SSIM and MSE showed a continual upward trend from small- to medium- and large-sized networks. As for SSIM, the performances increased from 0.7562 to 0.7855 and to 0.8280 in dice scores. Similarly, the performances of MSE rose from 0.6972 to 0.7132 and to 0.7579. All the observed increases in these cases are statistically significant. Also, the performances of NCC increased sharply from small- to medium-sized networks (0.5421 to 0.7924), and then increased smoothly to 0.8175 in the large-sized network. The results of statistical significance tests for all these increases acquired lower *p-values* than 0.0001. Likewise, the performances of transformer-based networks in the large-displacement transformation field are shown in Figure 4.5c. As for training with SSIM, the performances remained stable (0.7617 and 0.7619) in small- and medium-sized networks, this trivial difference (*p-value* = 0.9803) is not statistically significant. Com-

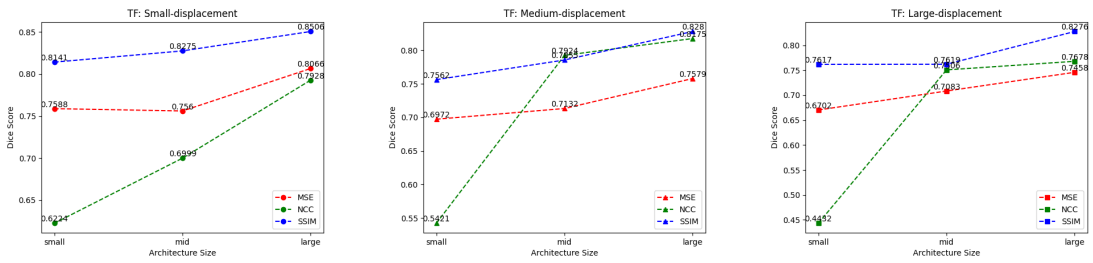
pared to the medium-sized network, the performance of the large-sized network gained 0.0657 to 0.8280 in dice scores. This marked increase is statistically significant (*p-values* <0.0001). Correspondingly, the performances of MSE showed a gradual upward trend from 0.6702 to 0.7083 and to 0.7458; achieved by small- to medium- and large-sized networks respectively. Also, all these rises (*p-values* <0.0001) are statistically significant. In terms of training with NCC, there was a substantial increase in performance of small- and medium-sized networks (0.4432 to 0.7506), *p-value* of this difference is lower than 0.0001 and it suggests that the performance is due to the increased size of the networks. Then the performance rose marginally from 0.7506 to 0.7678 achieved by medium- and large-sized networks respectively, while this mild increase (*p-value* = 0.0057) is statistically significant.

Meanwhile, Figure 4.6 represents how the performance changes with the size of transformer-based networks under the controlled condition of three loss functions: MSE, NCC and SSIM. As shown in Figure 4.6a, there are three lines to depict the performances of transformer-based networks trained with MSE loss at different scales of transformation fields. Since we have mentioned specific values for the comparison of performance when describing Figure 4.5, we pay attention to describing the trend of change at different scales of the transformation fields. In small-displacement transformation field, the slight decrease from small- to medium-sized networks is not statistically significant (*p-value* = 0.7274), but the increase from medium- to large-sized networks is statistically significant. Also, in the medium-displacement transformation field, the growth was gentle from small- to medium-sized networks, then became sharper from medium- to large-sized networks. In the case of the large-displacement transformation field, the performance increased steadily between the adjacent-sized network. Furthermore, all increases in medium- and large-displacement transformation fields are statistically significant. Correspondingly, Figure 4.6b represents the performances of transformer-based networks trained with NCC loss. How the performance changes with the size of networks in the small-displacement transformation field showed different behaviours compared to the performance of transformer-based networks in mid- and large-displacement transformation

fields. To be precise, the performances of transformer-based networks rose steadily with the increasing size of networks in the small-displacement transformation field, and these rises are statistically significant. Conversely, in medium- and large-displacement transformation fields, the performance surged from small- to medium-sized and grew gently to large-sized networks. In these cases, all performance differences are recognized as statistically significant. Also, the performances of transformer-based networks trained with SSIM loss are shown in Figure 4.6c. In small- and mid- displacement transformation fields, the performance of transformer-based networks increased steadily among the adjacent-sized network. In the large-displacement transformation field, the performance remained stable from small- to medium-sized networks, and then the performance rose abruptly from medium- to large-sized networks. Except in the case of the large-displacement transformation field where the slight performance difference between the small- and medium-sized network is not statistically significant ($p\text{-}value$ = 0.9893), all increases are deemed statistically significant ($p\text{-}value$ <0.0001).

Table 4.4: Dice scores for transformer-based architectures at different sizes and different loss functions to complete retinal image registration at various scales of transformation fields

		Small size (5.74M)	**Medium size (12.78M)**	**Large size (24.12M)**
Small-displacement	MSE	0.7588±0.0082	0.7560±0.0207	0.8066±0.0038
(Before Registration:	NCC	0.6224±0.0695	0.6999±0.0034	0.7928±0.0067
0.5048)	SSIM	0.8141±0.0072	0.8275±0.0020	0.8506±0.0089
Medium-displacement	MSE	0.6972±0.0150	0.7132±0.0037	0.7579±0.0039
(Before Registration:	NCC	0.5421±0.1174	0.7924±0.0038	0.8175±0.0065
0.4516)	SSIM	0.7562±0.0183	0.7855±0.0176	0.8280±0.0202
Large-displacement	MSE	0.6702±0.0187	0.7083±0.0069	0.7458±0.0112
(Before Registration:	NCC	0.4432±0.1677	0.7506±0.0106	0.7678±0.0105
0.4489)	SSIM	0.7617±0.0159	0.7619±0.0160	0.8276±0.0050

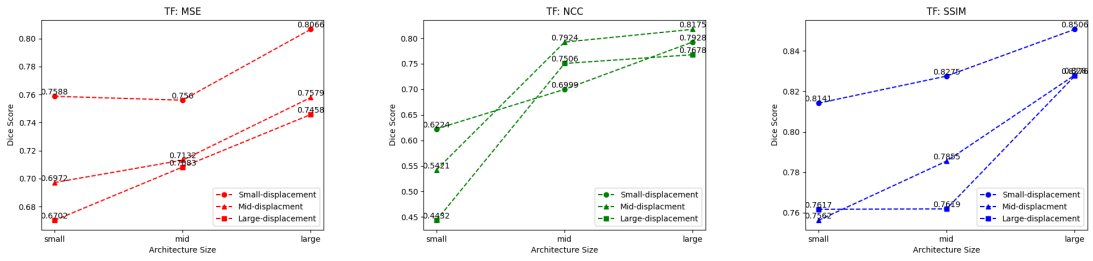


(a) Small-displacement **(b)** Medium-displacement **(c)** Large-displacement

Figure 4.5: Average dice scores of transformer-based architectures at different sizes interacted with different loss functions to complete retinal image registration

Table 4.5: Dice score differences and p-values for transformer-based architectures at different sizes to complete retinal image registration at various scales of transformation fields and different loss functions

			Medium-Small	**Large-Medium**	**Large-Small**
Small-displacement	MSE	Diff	-0.0028	0.0506	0.0478
		p-value	0.7274	$< 0.0001^*$	$< 0.0001^*$
	NCC	Diff	0.0775	0.0929	0.1704
		p-value	0.0071^*	$< 0.0001^*$	$< 0.0001^*$
	SSIM	Diff	0.0134	0.0231	0.0365
		p-value	0.0002^*	$< 0.0001^*$	$< 0.0001^*$
Medium-displacement	MSE	Diff	0.0160	0.0447	0.0607
		p-value	0.0110^*	$< 0.0001^*$	$< 0.0001^*$
	NCC	Diff	0.2503	0.0251	0.2754
		p-value	$< 0.0001^*$	$< 0.0001^*$	$< 0.0001^*$
	SSIM	Diff	0.0293	0.0425	0.0718
		p-value	0.0057^*	0.0005^*	$< 0.0001^*$
Large-displacement	MSE	Diff	0.0381	0.0375	0.0756
		p-value	$< 0.0001^*$	$< 0.0001^*$	$< 0.0001^*$
	NCC	Diff	0.3074	0.0172	0.3246
		p-value	$< 0.0001^*$	0.0057^*	$< 0.0001^*$
	SSIM	Diff	0.0002	0.0657	0.0659
		p-value	0.9803	$< 0.0001^*$	$< 0.0001^*$



(a) MSE **(b)** NCC **(c)** SSIM

Figure 4.6: Average dice scores of transformer-based architectures at different sizes interacted with scales of transformation fields to complete retinal image registration

4.1.3 Transformer-based Network: Multi-head vs Performance

The above experiments were conducted to explore the effect of different sizes from the general viewpoint of architectures, viz. convolutional-based networks and transformer-based networks. Compared to a convolutional-based network, the only additional component in a transformer-based network is the multi-head self-attention mechanism. However, it is not clear how the multi-head self-attention component affects the performance of transformer-based networks in a variety of network sizes (small, medium and large). Therefore, we further explored transformer-based networks by conducting additional controlled experiments, where the independent variable is the number of heads in self-attention components and dice score performance is the dependent variable. As shown in Figure 4.1.3, a set of three multi-head self-attention components, namely 6-head, 9-head and 12-head, were incorporated to construct transformer-based networks at different sizes. In order to maintain a low level of variability, we trained transformer-based architectures with SSIM loss to complete small-displacement retinal image registration. Besides, the

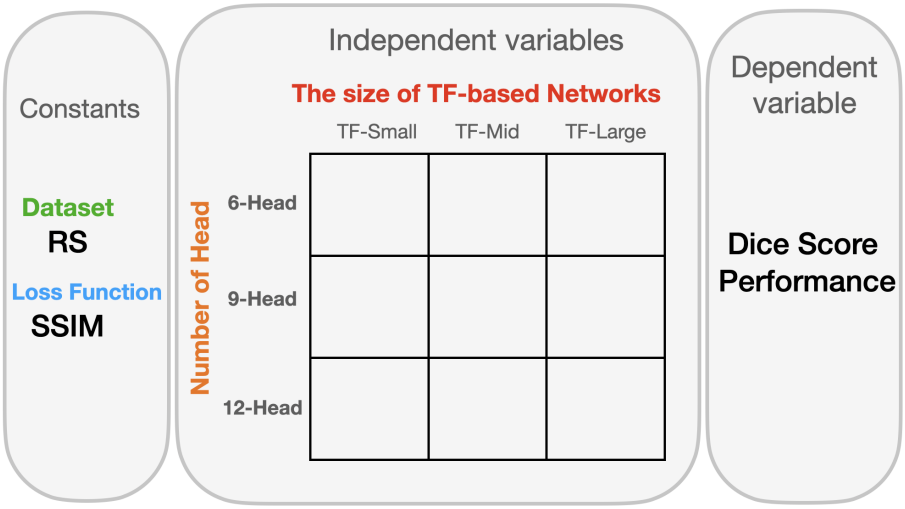


Figure 4.7: Experimental setting of Section 4.1.3 to explore how the number of heads in self-attention components affects the performance of transformer-based networks in retinal image registration

performances are compared to the set of convolutional-based networks at corresponding sizes. Consequently, Table 4.6 represents the results of transformer-based networks with various multi-head self-attention components at different network sizes and the results of the corresponding convolutional-based networks. Meanwhile, Table 4.7 shows differ-

ences and the results of the statistical significance tests (*p-values*) to ascertain whether the performance is affected by multi-head self-attention components. Also, we extracted information from Table 4.6 to draw a line graph for the sake of clearly observing and comparing how performance changes with multi-head self-attention components at different sizes of networks.

As shown in Figure 4.8, the performance of convolutional-based networks (baseline) decreased slightly from small- to medium-sized networks (0.8182 to 0.8114), and then the performance gained a rapid growth from medium- to large-sized networks which arrived at the best performance of 0.8334 in dice score. Conversely, the performance of transformer-based networks with various numbers of heads in self-attention components showed different behaviours. To be precise, 6-head transformer-based network gained a marked improvement to reach a peak from small- (0.8141) to medium-sized networks (0.8275), while the performance fell sharply to 0.8205 at the large-sized network. On the contrary, the performance of a 9-head transformer-based network declined considerably from small- to medium-sized networks (0.8260 to 0.8007), then the performance surged to the highest point (0.8502) at the large-sized network. Besides, the performance of a 12-head transformer-based networks rose gently from small- to medium-sized networks (0.7897 to 0.7928) and then grew rapidly to 0.8506 achieved by the large-sized network.

Compared to the baseline (i.e. the performance of the corresponding sizes of convolutional-based networks), in the case of small-sized networks, the 9-head transformer-based network achieved the best performance, which was 0.0078 higher than the small-sized convolutional-based network (0.8182). Importantly, the result of the statistical significance test for the sharp difference achieved a lower *p-value* than 0.0001, and it would suggest that the 9-head self-attention component contributes to improving performance significantly. However, the performance of the 6-head transformer-based network is comparable to the baseline (0.8141 and 0.8182), the slight difference is not deemed statistically significant (*p-value* = 0.1538). Additionally, the performance of the 12-head transformer-based network hit the lowest performance of 0.7897, the dramatic decline compared to the baseline (0.8182) is statistically significant (*p-value* = 0.0068). When it comes to a medium-

sized network, incorporating a 6-head self-attention component is beneficial to gain an increase, and the improvement is statistically significant (*p-value* = 0.0150). Except that 6-head transformer-based network outperformed the medium-sized convolutional-based network, 9-head and 12-head transformer-based networks underperformed relative to the baselines, but these marginal decreases are not statistically significant, for their *p-values* are 0.1735 and 0.1409 respectively. Lastly, as for large-sized networks, the performance of the 6-head transformer-based network (0.8205) was slightly under baseline, i.e. the large-sized convolutional-based performance (0.8334), whereas the slight decrease is not statistically significant (*p-value* = 0.3017). With the increasing number of heads in transformer-based networks, 9-head and 12-head transformer-based networks outperformed the baseline considerably, achieving 0.8502 and 0.8506 respectively. These sharp growths acquired *p-values* around 0.02 in the statistical significance test. Therefore, these improved performances are likely owed to incorporating multi-heads self-attention components in these cases.

Table 4.6: Dice scores for multi-head self-attention components to complete small-displacement retinal image registration with SSIM loss function

	CNN-based (baseline)	Transformer-based		
		6-Head	9-Head	12-Head
Small-size	0.8182 ± 0.0027	0.8141 ± 0.0072	0.8260 ± 0.0004	0.7897 ± 0.0253
Medium-size	0.8114 ± 0.0163	0.8275 ± 0.0020	0.8007 ± 0.0134	0.7928 ± 0.0295
Large-size	0.8334 ± 0.0166	0.8205 ± 0.0297	0.8502 ± 0.0089	0.8506 ± 0.0089

Table 4.7: p-values for multi-head self-attention components to complete small-displacement retinal image registration with SSIM loss function

		6-Head	9-Head	12-Head
Small-size	Diff(TF-CNN)	-0.0041	0.0078	-0.0285
	p-value	0.1538	$< 0.0001^{*}$	0.0068^{*}
Medium-size	Diff(TF-CNN)	0.0161	-0.0107	-0.0186
	p-value	0.0150^{*}	0.1735	0.1409
Large-size	Diff(TF-CNN)	-0.0129	0.0168	0.0172
	p-value	0.3017	0.0244^{*}	0.0217^{*}

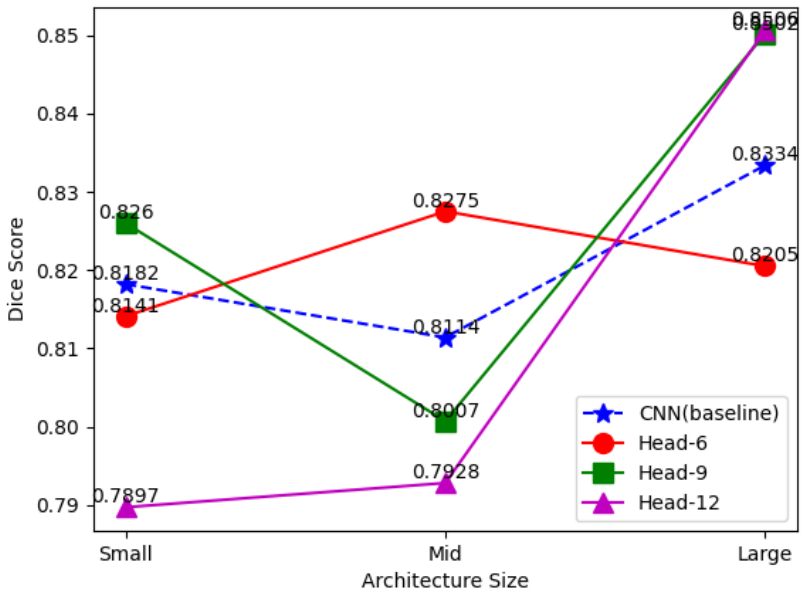


Figure 4.8: Average dice scores of transformer-based architectures changed with the size and number of multi-head self-attention components in retinal image registration

4.2 Further Exploration of Transformer-based Architecture

Generally, most transformer-based networks outperformed convolutional-based networks in previous experiments, and the performance of transformer-based architecture has been explained by the ability to find relationships between distant features in the image. Thus, the question arises as to the scale of the deformation field at which transformer-based architecture becomes superior in performance relative to convolution-based architecture. Our expectation is that the answer is not straightforward, since there are other factors to be taken into consideration, including datasets and loss functions. In this section, we explore how the transformer-based network addresses the limitation of the convolutional-based network in medical image registration by conducting a series of controlled experiments. Instead of comparing the specific dice score of convolutional-based and transformer-based networks respectively, we describe how the difference in average dice scores between these networks changes with scales of transformation fields to provide a more intuitive comparison. To keep other external variables the same, namely the size of networks and the number of heads in self-attention components, we trained the large-sized

convolutional-based network and 12-head large-sized transformer-based network under different controlled conditions such as datasets including retinal images and brain images, and a set of three loss functions: MSE, NCC and SSIM.

4.2.1 Retinal Image Registration: Scale of Transformation Field vs Performance Differences

In the context of retinal image registration, Figure 4.9 represents the experimental settings. As shown in Table 4.8, we compared the performance of large-sized convolutional-

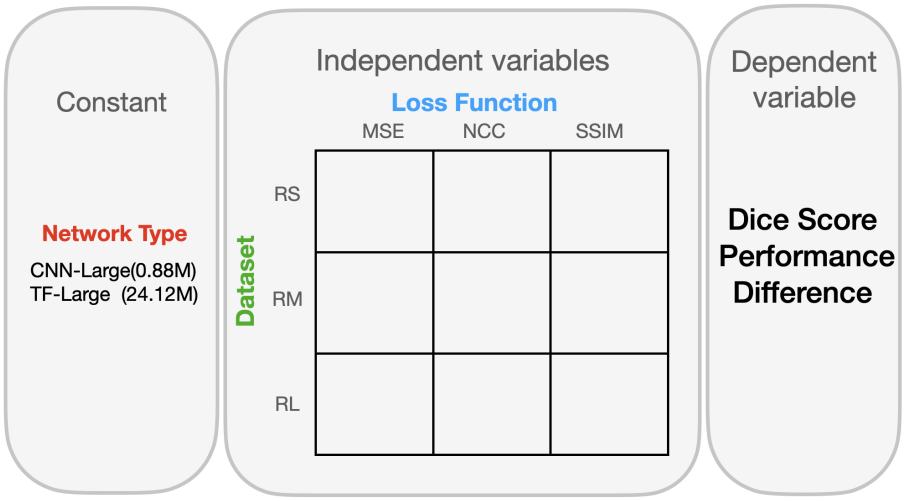


Figure 4.9: Experimental setting of Section 4.2.1 to explore how transformer-based networks address limitation of convolutional-based networks in retinal image registration

based network (Table 4.2) and large-sized 12-head transformer-based network (Table 4.4) in retinal image registration. The results of the statistical significance tests are provided as well. In addition, Figure 4.10a represents the differences in average dice scores between convolutional-based and transformer-based networks at three different scales of transformation fields. Generally, the performance of transformer-based networks exceeded convolutional-based networks in all cases, but most of these improved performances are trivial, and from Table 4.8, the results of statistical significance tests acquired higher *p-values* than 0.05, which suggests these difference performances are not statistically significant. Under the controlled condition of loss functions, the difference between transformer-based and convolutional-based networks trained with NCC showed the most prominent change with scales of transformation fields. The improvement of

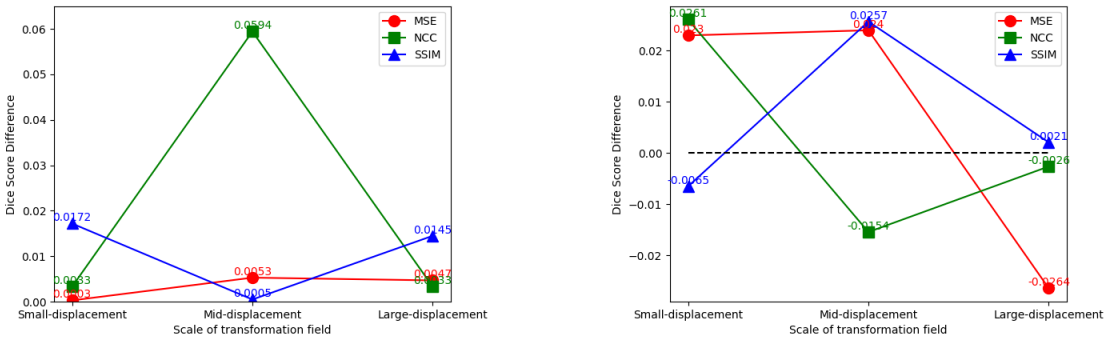
the transformer-based network trained with NCC compared with the convolutional-based network went up to reach the peak from small- to mid-displacement transformation fields (0.0033 to 0.0594). After this considerable increase, the improvement fell back at large-displacement transformation field (0.0033) as the same as the difference at small-displacement transformation field. In this case, these mild improvements at small- and large-displacement transformation fields are not statistically significant, which acquired *p-value* 0.5595 and 0.4990 respectively. Nevertheless, the improvement at mid-displacement transformation field is statistically significant (*p-value* $<$0.0001). Therefore, apart from the improved performance at the medium-displacement transformation field, transformer-based networks were not able to achieve superior performance than convolutional-based networks at small- and large-displacement transformation fields. In contrast, the improvement in the case of SSIM loss decreased from small- to mid-displacement transformation fields (0.0172 to 0.0005) while it went up to 0.0145 at the large-displacement transformation field. As for the statistical significance test, the differences at small- and large-displacement transformation fields are statistically significant; their *p-values* are 0.0217 and 0.0290 respectively. Besides, the slight improvement at the mid-displacement transformation field is not statistically significant (*p-value* = 0.9508). In the case of MSE loss, the improved performance slightly increased from 0.0003 to 0.0053 and decreased gently to 0.0047 at small-, mid- and large-displacement transformation fields. As a result, none of these mild improvements is statistically significant.

Furthermore, we observed how these improvements vary with loss functions under the controlled condition of the scale of transformation fields. At the small-displacement transformation field, the improved performance achieved the highest point when networks are trained with SSIM loss (0.0172). Also, the result of the statistical significance test (*p-value* = 0.0217) indicated that the improvement is statistically significant. As for NCC and MSE losses, the performances of convolutional-based networks were closely near to transformer-based networks, and these differences (0.0033 and 0.0003) are not statistically significant. When it comes to the mid-displacement transformation field, NCC loss was the most helpful loss function to enlarge the gap between the performance of

transformer-based and convolutional-based networks, and the increase of 0.0594 is recognized as statistically significant. Conversely, in the case of MSE and CNN loss functions, these mild improvements (0.0053 and 0.0005) are not statistically significant. Similarly to the case of the small-displacement transformation field, at the large-displacement transformation field, the improved performance is statistically significant only when networks are trained with SSIM loss (*p-value* = 0.0290). Apart from SSIM loss, the differences between convolutional-based and transformer-based networks trained with MSE and NCC losses were trivial (0.0047 and 0.0033), these tiny differences are not statistically significant.

Table 4.8: Dice score differences and p-values between convolutional-based and transformer-based architectures at different scales of transformation fields and loss functions in retinal image registration

		Small-displacement	Medium-displacement	Large-displacement
MSE	Diff(TF-CNN)	0.0003	0.0053	0.0047
	p-value	0.8333	0.0708	0.3167
NCC	Diff(TF-CNN)	0.0033	0.0594	0.0033
	p-value	0.5595	$< 0.0001^*$	0.4990
SSIM	Diff(TF-CNN)	0.0172	0.0005	0.0145
	p-value	0.0217^*	0.9508	0.0290^*



(a) Retinal image registration

(b) Brain image registration

Figure 4.10: The differences of average dice scores between transformer-based and convolutional-based networks at different scales of transformation fields and loss functions

4.2.2 Brain Image Registration: Scale of Transformation Field vs Performance Differences

In order to investigate whether the transformer-based architecture is able to addresses the limitation of the convolutional-based architecture on other datasets, we conducted

the same controlled experiments on brain slice datasets. The experimental settings are shown in Figure 4.11. Table 4.9 and Table 4.10 represent the performances of large-sized

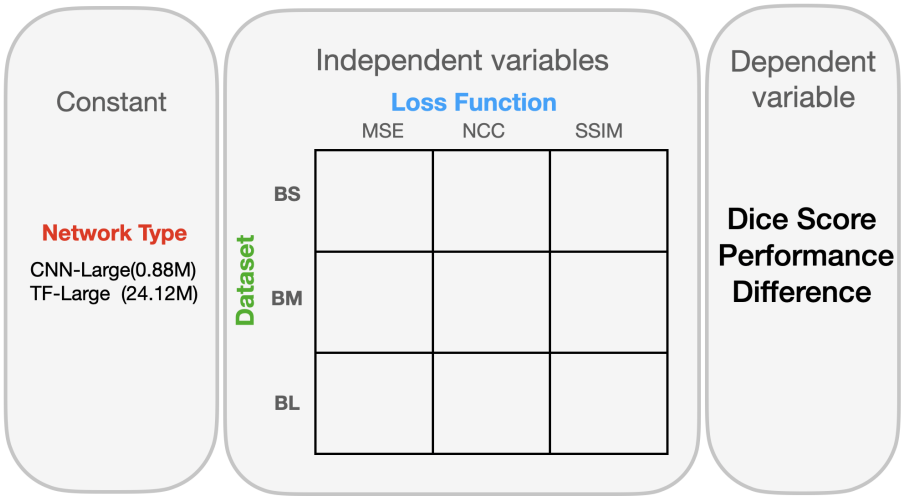


Figure 4.11: Experimental setting of Section 4.2.2 to explore how transformer-based networks address the limitation of convolutional-based networks in brain image registration

convolutional-based network and 12-head large-sized transformer-based network respectively. Further, Table 4.11 shows the comparison between the performance of transformer-based and convolutional-based networks in a variety of cases, and it shows the results of the statistical significance test. How the performance difference changes with scales of transformation fields in terms of various loss functions can be seen clearly in Figure 4.10b. Since all *p-values* are lower than 0.0001, we omit to describe the results of statistical significance tests for these performance differences in the later description of Figure 4.10b for brevity.

As shown in Figure 4.10b, in the case of MSE loss function, the performance of transformer-based networks were 0.023 and 0.024 higher than convolutional-based networks at small- and medium-displacement transformation fields respectively, while convolutional-based overtook transformer-based network by 0.0264 at large-displacement transformation field. In contrast, the difference between the performance of transformer-based and convolutional-based networks changed differently under the guidance of NCC loss. Precisely, the transformer-based network trained with NCC surpassed the convolutional-based network by 0.023 at the small-displacement transformation field. However, transformer-

based networks deteriorated the performance compared to convolutional-based networks at mid- and large-displacement transformation fields by 0.0154 and 0.0026 respectively. In terms of SSIM loss, the transformer-based network degraded in performance by 0.0065 compared with the convolutional-based network at the small-displacement transformation field. With the increasing displacement in the transformation field, transformer-based networks achieved 0.0257 and 0.0021 higher performance than convolutional-based networks at mid- and large-displacement transformation fields.

Moreover, the performance difference between convolutional-based and transformer-based networks varied with loss functions at the same scale of transformation field. At the small-displacement transformation field, transformer-based networks transcended convolutional-based networks under the guidance of NCC and MSE losses, but the transformer-based network trained with SSIM loss failed to show its advantage to achieve superior performance. Still, at the mid-displacement transformation field, SSIM and MSE losses were able to guide transformer-based networks to exhibit superior performance, while the transformer-based network was outperformed by the convolutional-based network under the guidance of NCC loss. At the large-displacement transformation field, except that SSIM loss, NCC and MSE loss were not supportive for transformer-based networks to outperform convolutional-based networks.

Table 4.9: Dice scores for the convolutional-based architecture at different scales of transformation fields and loss functions in brain image registration

	Small-displacement	Medium-displacement	Large-displacement
Before	0.5953	0.4939	0.4517
MSE	0.9045±0.0073	0.7721±0.0040	0.7063±0.0046
NCC	0.5456±0.0019	0.4507±0.0044	0.4563±0.0003
SSIM	0.8213±0.0007	0.7150±0.0020	0.7102±0.0002

Table 4.10: Dice scores for the transformer-based architecture at different scales of transformation fields and loss functions in brain image registration

	Small-displacement	Medium-displacment	Large-displacement
Before	0.5953	0.4939	0.4517
MSE	0.9275±0.0003	0.7961±0.0037	0.6799±0.0040
NCC	0.5717±0.0006	0.4353±0.0017	0.4537±0
SSIM	0.8148±0.0002	0.7407±0.0003	0.7123±0.0003

Table 4.11: Differences and p-values between convolutional-based and transformer-based architectures at different scales of transformation fields and loss functions in brain image registration

		Small-displacement	Medium-displacement	Large-displacement
MSE	Diff(TF-CNN)	0.0230	0.0240	-0.0264
	p-value	$< 0.0001^*$	$< 0.0001^*$	$< 0.0001^*$
NCC	Diff(TF-CNN)	0.0261	-0.0154	-0.0026
	p-value	$< 0.0001^*$	$< 0.0001^*$	$< 0.0001^*$
SSIM	Diff(TF-CNN)	-0.0065	0.0257	0.0021
	p-value	$< 0.0001^*$	$< 0.0001^*$	$< 0.0001^*$

4.3 Loss Functions and Architectures

Since we noticed that loss functions affected architectures in different ways in previous experiments, we conducted a series of experiments in this section to explore how a set of loss functions, namely MSE, NCC and SSIM loss, interact with different architectures viz. convolutional-based network and transformer-based network to affect registration performance. Thus, the experiment design is a 3×2 factorial design including two factors (i.e. loss functions with 3 levels and architectures with 2 levels) under the controlled conditions of datasets and scales of transformation fields. Figure 4.12 and Figure 4.14 represent experimental settings in retinal and brain image registration respectively. In the evaluation, we compared the performance among the set of loss functions at different scales of transformation fields in terms of the same large-sized architectures.

4.3.1 Retinal Image Registration: Loss functions vs Performance

In retinal image registration, based on the performance of large-sized convolutional-based networks trained with MSE, NCC and SSIM loss functions as shown in Table 4.2, Table 4.12 denotes these differences among various loss functions and corresponding *p-values*. In a similar manner, Table 4.13 shows these performance differences in terms of transformer-based networks, which were derived from Table 4.4. In order to have a clear comparison and observation, Figure 4.13 contains three subgraphs to show how average dice scores change with two independent variables/factors (i.e. loss functions and architectures) at various scales of transformation fields: small-, mid- and large-displacement transformation fields respectively.

As shown in Figure 4.13a, for the small-displacement transformation field, the per-

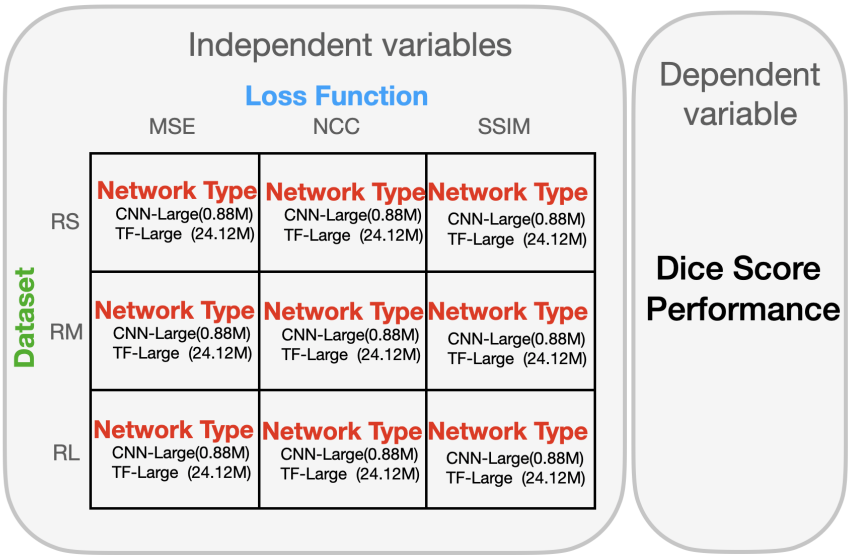
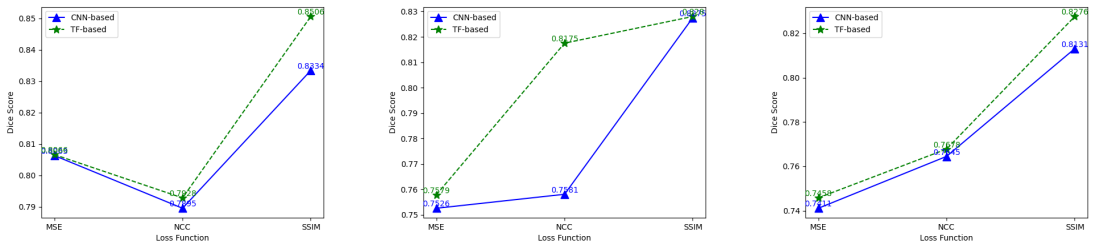


Figure 4.12: Experimental setting of Section 4.3.1 to explore how loss functions interacting with architectures affect the performance of retinal image registration

formance of convolutional-based networks (depicted in blue solid line) decreased from 0.8063 to 0.7895 and then increased to 0.8334; achieved by the guidance of MSE, NCC and SSIM loss function respectively. The statistical significance test (*p-value* = 0.0083) suggests that the decrease in performance may be due to training the convolutional-based network with NCC instead of MSE loss function. Also, the increase in performance from NCC to SSIM loss function and the increase from MSE to SSIM are statistically significant; their *p-values* are 0.0004 and less than 0.0001 respectively. Similarly to the performance of convolutional-based networks, the performance of transformer-based networks (depicted in green dashed line) decreased from MSE to NCC (0.8066 to 0.7928) and increased when the network is trained with SSIM loss function (0.8506). As for the statistical significance tests, these lower *p-values* suggest that all differences among performances are caused by the guidance of different loss functions. Correspondingly, Figure 4.13b represents the performance of convolutional-based and transformer-based networks trained with various loss functions at the mid-displacement transformation field. In terms of convolutional-based network, the performance increased slightly from MSE to NCC (0.7526 to 0.7581), the slight difference is not statistically significant (*p-value* = 0.5684). The performance then increased sharply to 0.8275 in the case of training the convolutional-based network with SSIM loss function. As for the statistical signifi-

cance test, the *p-value* is less than 0.0001, which would suggest that the improvement may be influenced by the choice of loss functions. Also, the improved performance between MSE and SSIM loss function (0.7526 to 0.8275) is statistically significant (*p-value* <0.0001). Conversely, the performance of transformer-based networks increased sharply from MSE to NCC (0.7579 to 0.8175), the result of the statistical significance test gained a lower *p-value* less than 0.0001. Then the performance increased slightly to SSIM (0.8280), the slight performance difference between NCC and SSIM loss function is not statistical significant (*p-value* = 0.1834). However, compared to the performance of MSE loss function, the statistical significance test (*p-value* <0.0001) would suggest that SSIM loss function is better than MSE to guide the transformer-based network to complete mid-displacement retinal image registration. Lastly, as shown in Figure 4.13c, for the large-displacement transformation field, the performance of convolutional-based and transformer-based networks trained with different loss functions showed a continual upward trend from MSE to NCC and to SSIM loss function. As for convolutional-based networks, the performance increased from 0.7411 to 0.7645 and to 0.8131 in dice score. Similarly, the performance of transformer-based networks rose from 0.7458 to 0.7678 and to 0.8276. All the observed increases in these cases are statistically significant. Therefore, these improved performances are likely owed to the choice of the loss function in these cases.



(a) Small-displacement **(b)** Mid-displacement **(c)** Large-displacement

Figure 4.13: Average dice scores of different architectures trained with different loss functions at various scales of transformation fields in retinal image registration

Table 4.12: p-values for the convolutional-based architecture at different scales of transformation fields and loss functions in retinal image registration

		Small-displacement	Medium-displacement	Large-displacement
NCC-MSE	Diff	-0.0252	0.0055	0.0234
	p-value	0.0083*	0.5684	< 0.0001*
SSIM-MSE	Diff	0.0271	0.0749	0.0720
	p-value	0.0004*	< 0.0001*	< 0.0001*
SSIM-NCC	Diff	0.0523	0.0694	0.0486
	p-value	< 0.0001*	< 0.0001*	< 0.0001*

Table 4.13: p-values for the transformer-based architecture at different scales of transformation fields and loss functions in retinal image registration

		Small-displacement	Medium-displacement	Large-displacement
NCC-MSE	Diff	-0.0138	0.0596	0.0220
	p-value	0.0002*	< 0.0001*	0.0012*
SSIM-MSE	Diff	0.0440	0.0701	0.0818
	p-value	< 0.0001*	< 0.0001*	< 0.0001*
SSIM-NCC	Diff	0.0578	0.0105	0.0598
	p-value	< 0.0001*	0.1834	< 0.0001*

4.3.2 Brain Image Registration: Loss functions vs Performance

Similarly to retinal image registration, we conducted controlled experiments to investigate the effect of loss functions interacting with architectures in brain image registration. Based on the performance of convolutional-based and transformer-based network as shown in Table 4.9 and Table 4.10 respectively, Table 4.14 and Table 4.15 represent differences along with p-values among the performance of networks trained with various loss functions in terms of convolutional-based and transformer-based networks respectively. Since all *p-values* are lower than 0.0001 in the case of brain image registration, we omit to describe the results of the statistical significance test for these performance differences in a later description for brevity. Likewise, Figure 4.15 clearly depicts how these performances are influenced by loss functions related to convolutional-based and transformer-based networks at different scales of transformation fields.

In the manner of quick summary, the relationship between the performance and loss functions exhibits a similar trend in respect of convolutional-based and transformer-based networks at all three scales of transformation fields. Generally, both the performances of convolutional-based and transformer-based networks decreased rapidly from MSE to NCC and then increased significantly with SSIM loss function. To be more precise, as shown in Figure 4.15a, the performance of convolutional-based networks reduced from

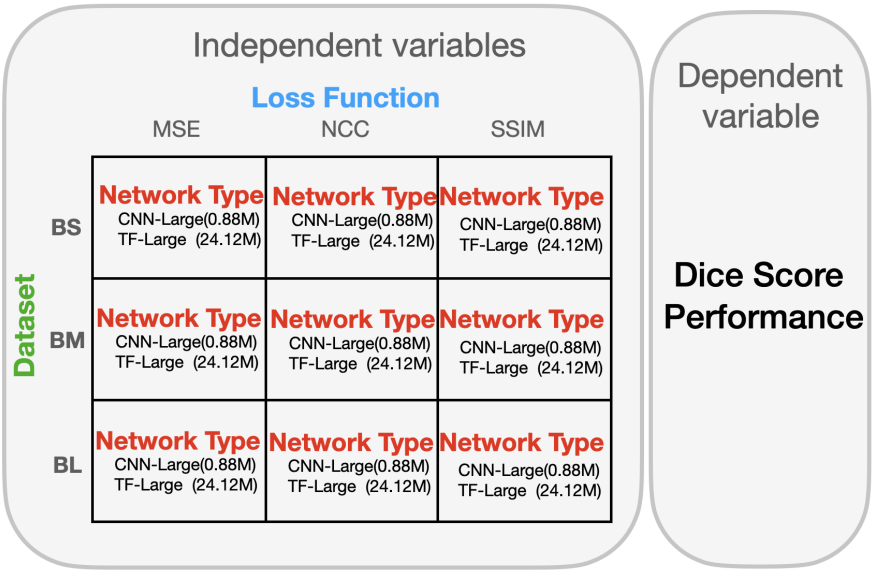
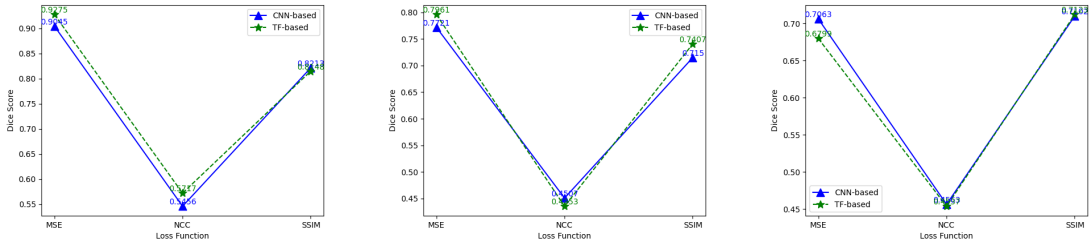


Figure 4.14: Experimental setting of Section 4.3.2 to explore how loss functions interacting with architectures affect the performance of brain image registration

0.9045 to 0.5456 and increased to 0.8213; achieved by the guidance of MSE, NCC and SSIM loss functions respectively. Similarly, the performance of transformer-based networks decreased from 0.9275 to 0.5717 and grew to 0.8148 in dice score. Figure 4.15b represents performances achieved at mid-displacement transformation field. As for convolutional-based networks, there was a fall from 0.7721 to 0.4507 between the MSE and NCC loss functions, while the performance increased dramatically to 0.7150 in the case of training the convolutional-based network with SSIM loss function. In terms of transformer-based networks, the performance declined from 0.7961 to 0.4353 and then rose to 0.7407; achieved with MSE, NCC and SSIM loss functions. Similarly, as shown in Figure 4.15c, at the large-displacement transformation field, the performance of convolutional-based networks dipped from MSE (0.7063) to NCC (0.4563) but peaked at 0.7102 under the guidance of SSIM loss function. Likewise, the performance of transformer-based networks decreased from 0.6799 to 0.4537; achieved by the guidance of MSE and NCC loss functions respectively. However, the performance of the transformer-based network trained with SSIM hit the highest point (0.7123) in the case of the large-displacement transformation field. Overall, all performance differences are statistically significant, which would suggest the decrease or improvement in performances are likely due to the different guidance of loss functions in the case of brain image registration.



(a) Small-displacement **(b)** Mid-displacement **(c)** Large-displacement

Figure 4.15: Average dice scores of different architectures trained with different loss functions at various scales of transformation fields in brain image registration

Table 4.14: p-values for the convolutional-based architecture at different scales of transformation fields and loss functions in brain image registration

		Small-displacement	Medium-displacement	Large-displacement
NCC-MSE	Diff	-0.3589	-0.2793	-0.2500
	p-value	$< 0.0001^*$	$< 0.0001^*$	$< 0.0001^*$
SSIM-MSE	Diff	-0.0832	-0.0571	0.0039
	p-value	$< 0.0001^*$	0.0311^*	$< 0.0001^*$
SSIM-NCC	Diff	0.2757	0.2222	0.2539
	p-value	$< 0.0001^*$	$< 0.0001^*$	$< 0.0001^*$

Table 4.15: p-values for the transformer-based architecture at different scales of transformation fields and loss functions in brain image registration

		Small-displacement	Medium-displacement	Large-displacement
NCC-MSE	Diff	-0.3558	-0.3608	-0.2262
	p-value	$< 0.0001^*$	$< 0.0001^*$	$< 0.0001^*$
SSIM-MSE	Diff	-0.1127	-0.0554	0.0324
	p-value	$< 0.0001^*$	$< 0.0001^*$	$< 0.0001^*$
SSIM-NCC	Diff	0.2431	0.3054	0.2586
	p-value	$< 0.0001^*$	$< 0.0001^*$	$< 0.0001^*$

4.4 Loss Functions and Dataset Characteristics

Based on previous experiments results, we summarize the order of dice score for retinal and brain image registrations at different scales of transformation fields as shown in Table 4.16. In the case of retinal image registration, the order of performance from the best to the worst is SSIM, MSE and NCC at the small-displacement transformation field, while the order is changed as SSIM, NCC and MSE at medium- and large-displacement transformation fields. Nevertheless, loss functions perform differently in brain image registration. At small- and medium-displacement transformation fields, the best performance is achieved by training networks with MSE. Then SSIM outperforms NCC to guide networks to complete brain image registration. At the large-displacement transformation field, SSIM outperformed MSE to achieve the best generalization ability, and the performance of NCC was the worst in this case.

It is clear from Table 4.16 that loss functions show different behaviours in the task of retinal and brain image registration. Also, in the same dataset, the performance of the same loss function is different at various scales of transformation fields. Therefore, we make a further discussion about how these different loss functions in conjunction with datasets affect registration performance at various scales of transformation fields in respect of convolutional-based and transformer-based networks. In order to have a clear comparison and observation, Figure 4.17 and Figure 4.19 depict how performances change with datasets at various scales of transformation fields under the guidance of different loss functions viz. MSE, NCC and SSIM loss in terms of convolutional-based and transformer-based networks respectively.

Table 4.16: The order of dice score for retinal and brain image registrations at different scales of transformation fields

	Small-displacement	Medium-displacement	Large-displacement
Retina	SSIM > MSE > NCC	SSIM > NCC > MSE	SSIM > NCC > MSE
Brain	MSE > SSIM > NCC	MSE > SSIM > NCC	SSIM > MSE > NCC

4.4.1 Convolutional-based Network: Dataset Vs Performance

In the case of convolutional-based network, Figure 4.16 shows the experimental settings. Figure 4.17 are derived from Table 4.2 and Table 4.9. Specifically, Figure 4.17 containing

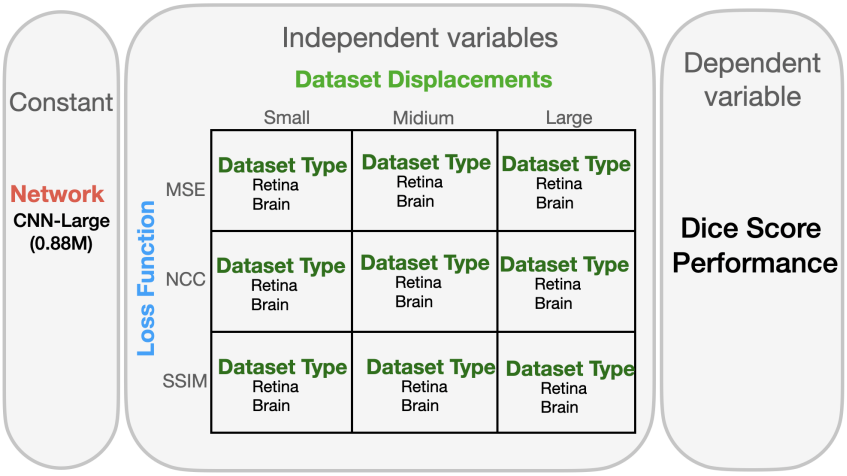
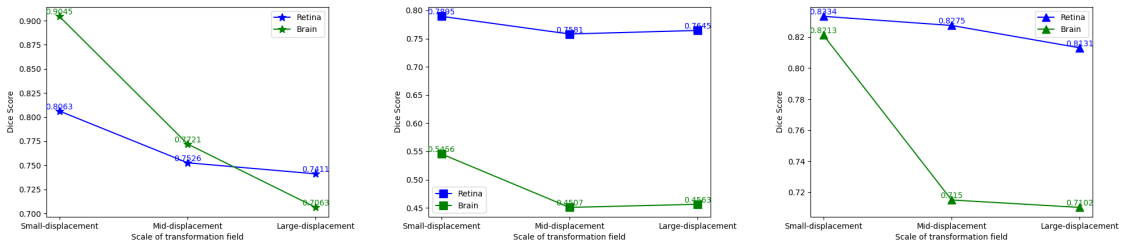


Figure 4.16: Experimental setting of Section 4.4.1 to explore how loss functions interacting with dataset characters affect the performance of convolutional-based networks.

three subgraphs depicts the relationship between the performance of convolutional-based networks and scales of transformation fields in terms of retinal and brain image datasets under controlled loss functions (MSE, NCC and SSIM). As shown in Figure 4.17a, when the convolutional-based network is trained with MSE loss function, the performance of retinal image registration decreased slightly from small- to mid-displacement transformation fields (0.8063 to 0.7526), and then remained stable at the large-displacement transformation field (0.7411). In this case, the performance of brain image registration decreased continuously and sharply from small- to mid- and to large-displacement transformation fields; achieved 0.9045, 0.7721 and 0.7063 in dice scores respectively. Similarly to Figure 4.17a, Figure 4.17b represents the comparison information in the case of training networks with NCC loss function. For retinal image registration, the convolutional-based network showed some fluctuations in dice scores from small- (0.7895) to mid- (0.7581) and to large-displacement transformation fields (0.7645). Instead, the performance of brain image registration dipped sharply from small- to mid-displacement transformation fields (0.5456 to 0.4507), then the performance remained static (0.4563) at the large-displacement transformation field. Correspondingly, as shown in Figure 4.17c, under

the guidance of SSIM loss function, the performance of retinal image registration decreased gradually from 0.8334 to 0.8275 and to 0.8131; achieved at small-, mid- and large-displacement transformation fields respectively. Conversely, in the case of brain image registration, the performance fell dramatically from small- to mid-displacement transformation fields (0.8213 to 0.7150), and then the performance decreased marginally to 0.7123 at the large-displacement transformation field.



(a) MSE **(b)** NCC **(c)** SSIM

Figure 4.17: Average dice scores of convolutional-based networks trained with different loss functions at different scales of transformation fields in retinal and brain image registration

4.4.2 Transformer-based Network: Dataset Vs Performance

Similarly, as shown in Figure 4.19, we extracted information from Table 4.4 and Table 4.10 to draw line graphs for the sake of clearly observing and comparing how the performance of transformer-based networks changes with scales of transformation fields differently in retinal and brain image datasets. The experimental settings are shown in Figure 4.18. As shown in Figure 4.19a, in the case of training transformer-based network with MSE loss function, the performance of retinal image registration decreased gradually from small- to mid- and to large-displacement transformation fields (0.8066 to 0.7569 to 0.7458), while the performance of brain image registration showed a sharp downward trend from small- to mid- and to large-displacement transformation fields; achieved 0.9075, 0.7961 and 0.6799 respectively. Likewise, as shown in Figure 4.19b, when the transformer-based network is trained with NCC loss function, the performance of retinal image registration increased slightly from small- to mid-displacement transformation fields (0.7028 and 0.8175), while the performance decreased to 0.7678 at the

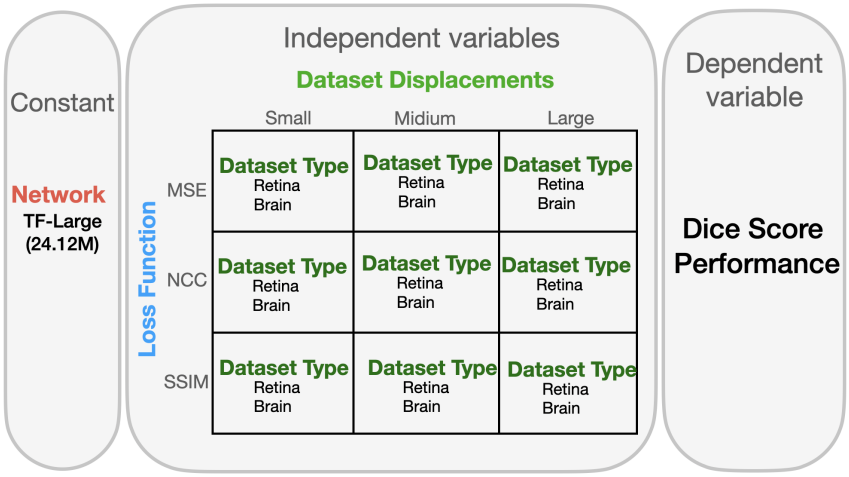
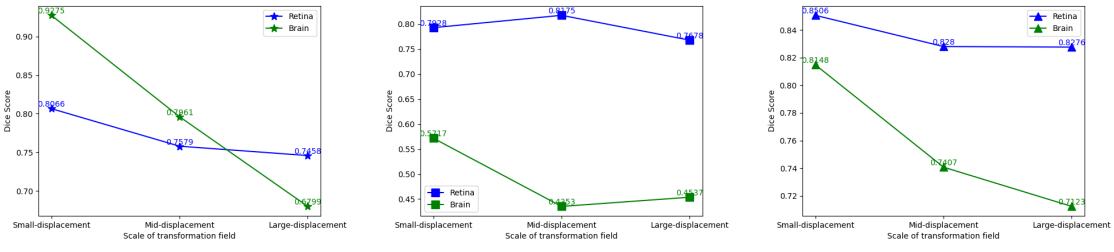


Figure 4.18: Experimental setting of Section 4.4.2 to explore how loss functions interacting with dataset characters affect the performance of transformer-based networks.

large-displacement transformation field. Conversely, in the case of brain image registration, the performance decreased considerably from small- to mid-displacement transformation fields (0.5717 and 0.4353), and then the performance increased marginally at the large-displacement transformation field (0.4537). Lastly, how the performance of transformer-based network trained with SSIM loss function changes with scales of transformation fields is shown in Figure 4.19c. In the case of retinal image registration, the performance decreased slightly and then levelled off from small- to mid- and to large-displacement transformation fields; achieved 0.8506, 0.8280 and 0.8276 in dice scores respectively. For brain image registration, the performance decreased significantly from 0.8148 to 0.7407 at small- and mid-displacement transformation fields respectively. Also, the performance decreased less sharply from mid- to large-displacement transformation fields (0.7407 and 0.7123).



(a) MSE **(b)** NCC **(c)** SSIM

Figure 4.19: Average dice scores of transformer-based networks trained with different loss functions at different scales of transformation fields in retinal and brain image registration

4.5 Chapter Summary

This chapter reported the results of a series of controlled experiments and ablation studies. It provided experimental settings based on independent variables and dependent variables. There are a variety of groups of experiments to explore the relationship between registration performance and architectures, datasets and loss functions, respectively. However, these factors do not influence registration performance separately. Thus, we introduced factorial designs to explore how different factors interact to affect performance.

In terms of the relationship between network architecture components and performance, results presented in Section 4.1 concluded that the larger-size networks are likely to achieve better performance. In terms of multi-head self-attention components, it is helpful to improve performance compared to pure convolutional-based networks in most scenarios. However, one cannot simply summarize that more multi-head self-attention components provide better performance. Additionally, Section 4.2 reported the results of the performance differences in the comparison of convolutional-based networks and transformer-based networks in the case of retinal and brain image registration respectively. The results demonstrated that transformer-based networks are able to address the limitation of convolutional-based networks, especially to register medium- to large-displacement image pairs. Moreover, how loss functions interacting with architectures and datasets to affect performance are represented in Section 4.3 and Section 4.4 respectively. By observing the trends of the performance of convolutional-based networks and transformer-based networks changing with loss functions respectively, loss functions behave differently on different architectures. In terms of the relationship between loss functions and datasets, in the case of retinal image registration, SSIM is most helpful to guide networks to achieve the best performance (0.8506). In the case of brain image registration, the performance of training networks with MSE is the best (0.9045). Therefore, a loss function tailored to a salient feature of the image will most likely lead to better registration. More analysis and discussion of these results are provided in the next chapter.

Chapter 5

Discussion

This research aimed to empirically study the effects of medical image registration performance on a series of factors: architectures, scales of transformation fields, loss functions and datasets. Based on experimental results represented in Chapter 4, we found that the answer to how registration performance changed with these factors is not straightforward, these factors were interacting to affect registration performance. In this chapter, we interpret experimental results to briefly answer research questions listed in section 1.3 respectively.

5.1 Network Architecture Components and Performance

In terms of exploring the notion of network sizes and components versus image registration performance, the results represented in section 4.1 indicate that both convolutional-based and transformer-based architectures with larger sizes are able to improve the generalization ability. The dice scores of the large-size convolutional-based and transformer-based networks are 0.8334and 0.8506 respectively in retinal image registration, achieving the best performance compared to smaller sizes of corresponding architectures. As we mentioned in Chapter 1, all possible functions mapping input to output constitute a hypothesis space $\mathscr{H}_m$, where m is the number of model parameters. The approximate function (f_m) is selected by a minimizer to obtain a small generalization error; the minimizer

can be defined as

$$f_m = \arg\min_{f \in \mathscr{H}_m} \mathscr{R}(f), \tag{5.1}$$

where $\mathscr{R}(f)$ is an empirical risk calculated as shown in Equation 2.5. Ma et al. (2020) proposed a direct approximation theorem:

$$inf_{f \in \mathscr{H}_m} \mathscr{R}_f \lesssim \frac{\|f^*\|_*^2}{m}, \tag{5.2}$$

where f^* is the target function in natural function space, and the target function is unknown. The number of parameters is denoted as m. From this equation, we can tell that the generalization error is smaller when the model contains more parameters, thus the larger-sized model can achieve a better generalization ability.

Generally, there are two main hyperparameters to decide the number of parameters of a neural network: the number of layers (depth) and the number of neurons in each layer (width). In Ma et al. (2020)'s work, a multi-layer neural network is defined as:

$$f(x) = \sum_{i_L=1}^{m_L} a_{i_L}^L \sigma \left(\sum_{i_{L-1}=1}^{m_{L-1}} a_{i_L i_{L-1}}^{L-1} \sigma \left(... \sigma \left(\sum_{i_1=1}^{m_1} a_{i_2 i_1}^1 \sigma \left(\sum_{i_0=1}^{d+1} a_{i_1 i_0}^0 x_{i_0} \right) \right) \right) \right), \tag{5.3}$$

where the depth of layer is denoted as L; the width of layer is denoted as m_l; and the weights of layer is denoted as $a_{i_{l+1} i_l}^l$, including weight matrix (W_l) and bias (b_l), $W_l \in \mathbb{R}^{m_{l+1} \times m_l}$ and $b_l \in \mathbb{R}^{m_l}$. ReLU activation function, input and the dimension of input are denoted as σ, x and d respectively.

Since the original decoder stage in convolutional-based and transformer-based networks have already been constructed by 12 layers, and Zagoruyko and Komodakis (2016) suggested that widening layers instead of deepening architectures is more effective to improve performance, thus we enlarged the size of models by fixing the depth of neural networks and roughly enlarging the width of each layer. Generally, the number of neurons in each layer is the number of filters, each filter is looking for a specific type of template or concept in the input volume. Therefore, widening layers is to extract more feature maps. After training a network, each filter generates a specific feature map. In a neural network, these multiple layers are stacked to generate a hierarchy of filters. At the

earlier layers, the output usually represents low-level features such as edges. And then at the mid-level layers, more complex features are found such as corners and blobs and so on. At high-level layers, filters are used to generate outputs that resemble concepts more than blobs. In our experiments, we constructed larger-sized models by enlarging the width of each layer. Thus, larger-size models are able to generate more features in each hierarchical layer, which might be helpful to realize the alignment of anatomical structures. Precisely, more simple features are propagated and combined to form complex features, which might be helpful to capture (or encode) more details. The complex features generated help the network find corresponding anatomical structures in moving and fixed images, and it is beneficial to calculate specific displacements for these matched anatomical structures.

Additionally, the size of the model has a close relationship with the size of the training dataset (Nguyen et al., 2020). Therefore, one cannot simply summarize that more parameters alone contributed to better generalization ability. When the amount of model parameters is larger than the size of the training dataset, the model is overparameterized. Nguyen et al. (2020) visualized feature maps to show that overparameterized models are more likely to generate repeated features. Therefore, increasing the size of a model may not always yield improved performance. Also, overparameterized models normally meet the problem of overfitting. This is why some large-sized networks in our experiments are only able to achieve comparable performance to small-sized networks instead of improving performance. For instance, in the case of convolutional-based networks trained with NCC loss function, the number of parameters of small- and large-sized convolutional-based networks are 0.22M and 0.88M respectively, while the mild increase in dice score performance from small- to large-sized networks (0.7807 to 0.7895) is not statistically significant ($p\text{-}value = 0.2087$).

With regard to transformer-based networks, we explore how the number of multi-head self-attention components affects registration performance as well. We noticed that more heads in self-attention components do not provide better performance in the case of medical image registration. For example, a 12-head transformer-based network decreased

the performance on dice score compared to 9-head transformer-based network (0.7897 and 0.8260). Ji et al. (2019) defined the output of the *i-th* head (H_i) in the multi-head self-attention components as:

$$H_i = W_v^{(i)} V \times softmax\left((W_k^{(i)} K)^\top (W_q^{(i)} Q)\right) \in \mathbb{R}^{q_i \times n}, \tag{5.4}$$

where the learnable matrix weights of value (V), query (Q) and key (K) matrices are denoted as W_v, W_q and W_k respectively. Here, $((W_k^{(i)} K)^\top (W_q^{(i)} Q))$ are used to obtain the similarity scores between query and key by a matrix multiplication. Softmax function normalizes (i.e. sum to one) these similarities scores. The multiplicands of the the matrix multiplication, normalized similarity scores and value, generate the corresponding output. In the case of self-attention mechanism, key, value and query are the same as input. Essentially, applying different learned weights matrices $W_k^{(i)}$, $W_q^{(i)}$ and $W_v^{(i)}$ on key, value and query is actually applying linear transformation on different parts of input into different representation subspaces, and the final output is obtained by concatenation of the *i-th* output (H_i) as:

$$O = W_o \begin{bmatrix} H_1 \\ \vdots \\ H_M \end{bmatrix} \in \mathbb{R}^{q \times n}, \tag{5.5}$$

where W_o is the matrix to map the dimension of multiple heads into the desired dimension. Therefore, the final output is the combination of the similarities results in different representation subspaces. The number of representation subspaces equals the number of heads in multi-head self-attention components.

However, a self-attention with more heads in a multi-head configuration could possibly produce redundant similarities between query and key in different representation subspaces. These extra multi-head self-attention components might learn overlapping features in some representation subspaces, or these extra heads might break a complex feature such as a part of the anatomical structure into several smaller unrealistic features which are meaningless to the characteristics of images to be registered. Therefore, additional heads do not always play a positive role to improve performance. Sometimes, the additional heads even degrade the performance when these images or feature maps are not

sufficiently complex to contain many prominent features. This explains why the performance decreased with incorporating more heads in terms of medium-sized transformer-based networks, where incorporating 6-head self-attention components outperformed 9-head and 12-head transformer-based networks (the dice scores are 0.8275, 0.8007 and 0.7928 respectively).

5.2 Further Exploration of Transformer-based Architecture

In order to answer the research question of how transformer-based networks address the limitation of convolutional-based neural networks in medical image registration, we conducted a series of controlled experiments in Section 4.2. We aimed to determine the scale of the deformation field at which transformer-based architecture becomes superior in performance relative to convolution-based architecture. Comparing the change of performance differences between the convolutional-based and transformer-based networks, the results shown in Table 4.8 and Table 4.11 demonstrated that transformer-based architecture is able to improve the performance of convolutional-based architecture in most scenarios, especially when registering medium to large-displacement image pairs.

Even though the transformer network was designed for natural language processing tasks, we believed that a transformer-based network is able to realize the task of medical image registration. Our experimental results were in line with this hypothesis. As we described in Section 2.4, a transformer-based network is able to replace a convolutional-based network to realize two major functions: feature aggregation and feature transformation (Zhao et al., 2020). Also, Cordonnier et al. (2019) demonstrated from a theoretical perspective that the multi-head self-attention layer is able to be re-parametrized as any convolutional layer.

Moreover, in the larger-displacement transformation field, we applied a larger extent of transformations including translation, rotation and elastic transformations on images to generate image pairs; more details are described in Section 3.4.3. Thus, the distances between similar features in moving and fixed images are larger than in small-displacement

image pairs, which might be beyond the ability of convolutional-based architecture. The ability of an architecture to capture information can be expressed as an effective receptive field (ERF), which is the area of original input that a neuron can observe. Le and Borji (2017) proposed an equation to calculate effective receptive field as follows.

$$R_k = R_{k-1} + (f_k - 1) \prod_{i=1}^{k-1} s_i, \tag{5.6}$$

where the kernel size of layer k is denoted as f_k; stride is denoted as s_i; and R_{k-1} is the effective receptive field of previous layer $k-1$. Therefore, the effective receptive field is related to the size of filters, stride and the depth of networks. Increasing the depth of networks is helpful to enlarge the effective receptive field, but it is still limited to a local range. Based on Equation 5.6, the effective receptive field of the end of the decoder in our convolutional-based network is 62×62, while our original input size is 192×224. For each layer, the receptive field is the number of pixels of the previous layer that are used to output a pixel in the next layer. The size of the receptive field depends on the size of the kernel and the dimension of the input. In our experiments, the receptive field on each convolutional layer is 3×3, if we ignore the dimension of input. However, according to the definition of the self-attention layer (Equation 2.18), self-attention is able to simultaneously calculate the similarities in the global range, thus the receptive field equals the size of input of this layer. This may theoretically account for the superior performance of an attention-based network over a convolutional-based counterpart.

Essentially, image registration is a pixel-level computer vision task, where it regresses displacement for each pixel from a moving image to a fixed image. Importantly, permutation equivariance is the most prominent characteristic in pixel-level prediction tasks. Permutation equivariance which refers to the fact that if reordering input or applying operations on input, the output will change in the same way can be expressed as:

$$\mathscr{T}_\pi(A(X)) = A(\mathscr{T}_\pi(X)), \tag{5.7}$$

where any spatial permutation is denoted as $\mathscr{T}_\pi$, and any operations and input are denoted as A and X respectively. Therefore, the output value is independent of the order of input.

In the case of medical image registration, the order of pixels might be different in moving and fixed images at a larger-displacement transformation field, because there exists a large extent of rotation and elastic transformations. However, the convolutional operation considers the neighbour pixels and the fixed order of these neighbour pixels to output a weighted sum. Thus, it does not satisfy permutation equivariance when the kernel size is larger than 1. The same anatomical structure might be represented in different representation spaces. Hence, the convolutional layer might detect them as two different structures and miss aligning them together.

As for transformer-based networks, Ji et al. (2019) proved that self-attention operator (A_s) is permutation equivariant. In their work, self-attention operator (A_s) is denoted as:

$$A_s(X) = W_v X \cdot softmax\left((W_k X)^\top \cdot W_q X\right), \tag{5.8}$$

where in self-attention operator, query, key and value matrices are the same as input (X). Their corresponding weight matrices are denoted as W_q, W_k, and W_v. When applying a spatial permutation ($\mathscr{T}_\pi$) to the input X, we have

$$\mathscr{T}_\pi(X) = XP_\pi, \tag{5.9}$$

where P_π is an orthogonal matrix, thus $P_\pi^\top P_\pi = I$.

Moreover, the proof is shown as (Ji et al., 2019):

$$\begin{aligned}
A_s(\mathscr{T}_\pi(X)) &= W_v \mathscr{T}_\pi(X) \cdot softmax\left((W_k \mathscr{T}_\pi(X))^\top \cdot W_q \mathscr{T}_\pi(X)\right) \\
&= W_v XP_\pi \cdot softmax\left((W_k XP_\pi)^\top \cdot W_q XP_\pi\right) \\
&= W_v XP_\pi \cdot softmax\left(P_\pi^\top (W_k X) \cdot W_q XP_\pi\right) \\
&= W_v X(P_\pi P_\pi^\top) \cdot softmax\left((W_k X)^\top \cdot W_q X\right) P_\pi \\
&= A_s(X)P_\pi \\
&= \mathscr{T}_\pi(A_s(X)).
\end{aligned}$$

According to the definition of spatial permutation equivariance as shown in Equation 5.7, self-attention operator satisfies spatial permutation equivariance (easily shown

by demonstrating that $A_s(\mathscr{T}_\pi(X)) = \mathscr{T}_\pi(A_s(X))$. Therefore, compared to convolutional operation, a self-attention operator is able to not only find the longer-range information, but also to predict displacement for each pixel even when the location and deformation of anatomical structures change to a large degree.

5.3 Loss Functions and Architectures

For the third research question, we explored how loss functions interacting with architectures affect generalization performance. The results represented in Section 4.3 indicate that the loss function interacting with different architectures behaves differently. Roughly speaking, transformer-based networks outperformed convolutional-based networks under the guidance of the same loss function in most scenarios. Besides, the relationship between the different loss functions and generalization performance exhibit a similar trend in respect of convolutional-based and transformer-based networks. Theoretically, the loss function is designed to calculate the gap between prediction (denoted as $f(x)$) and expectation (y). Hence, loss function is denoted as $\mathscr{L}(f(x),y)$, where the desired function approximated by network is denoted as f. Also, this function (f) is parameterized by multi-dimensional trainable weights. Therefore, the loss function is closely related to the parameters of architecture, and there exist a lot of parameters in architecture, thus the loss function is usually a high-dimensional function. From the perspective of loss landscape, Ma et al. (2020) suggested that the smoother landscape is able to realize better generalization ability. Also, they proved that with the increasing parameters of architectures, the corresponding loss landscape becomes smoother, because the over-parametrization architecture is beneficial to avoiding bad local minima. In addition, Huang et al. (2020) suggested that higher-dimensionality architecture is easier to arrive at good minima. In our case, there are 0.88M and 24M parameters in large-sized convolutional-based and transformer-based networks respectively, thus the more parameters contained in transformer-based networks might be helpful to smoothen the loss landscape, then the smoother loss landscape helps transformer-based networks to achieve better generalization ability. Furthermore, the additional parameters (around 23M) of the transformer-based network are used

to construct multi-head self-attention components. Liu et al. (2020) proved that even if the optimizer is stuck in local minima, the multi-head attention components are still helpful to improve the generalization ability.

5.4 Loss Functions and Datasets

As for the last research question, we explored how different loss functions (viz. MSE, NCC and SSIM) in conjunction with datasets (viz. retinal fundus images and brain slice images) affect registration performances; results are represented in Section 4.4. The study demonstrates a correlation between loss functions and datasets. The results suggest that the changes in performance with loss function for a given network vary according to the image datasets used in the task of image registration. Since the network learns the prominent features of specific images through iterative training under the guidance of loss functions, what prominent features found by networks would change if the loss function changes in a given deep network. In other words, not every feature of an image makes the same contribution to the learning process of network (Rajput, 2021). Also, the prominent features vary with the tissue structures of a given medical image. Therefore, the results are in line with our hypothesis that a loss function tailored to the salient feature of the image will most likely lead to better registration.

As we described in Section 2.5, the loss function is to compute the dissimilarities between a pair of images. We trained networks with a set of loss functions including MSE, NCC and SSIM loss to complete retinal and brain image registration. These loss functions are designed to calculate dissimilarities based on different considerations. Specifically, as shown in Equation 2.19, the MSE loss function essentially calculates the average of the square of pixel-to-pixel dissimilarities between the moved and fixed images. It calculates the difference from the global perspective without considering the influence of neighbour pixels. However, NCC and SSIM loss functions address the limitation of MSE. Both NCC and SSIM calculate dissimilarities locally with the consideration of neighbour pixels within a 9×9 window in our implementation. Precisely, from the definition of NCC and SSIM as shown in Equation 2.20 and Equation 2.21 respectively, NCC and SSIM

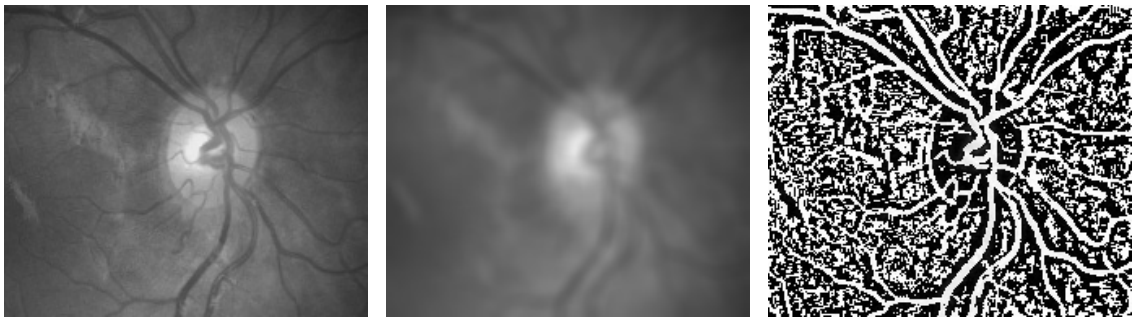
are able to capture structural information directly by subtracting the corresponding local mean intensity (denoted as $I - \bar{I}$), where mean intensity is denoted as $\bar{I}$. Figure 5.1 shows that structural information can be captured directly. Then NCC and SSIM compared the dissimilarities by calculating the cosine of the angle between structural information captured from moved and fixed images (Z. Wang et al., 2004). The major difference between NCC and SSIM is their normalization approach. In the work of Z. Wang et al. (2004), they recognized that the luminance and contrast are related to the mean intensity and standard deviation of intensity respectively. Thus, the normalization of NCC is only for luminance, which helps NCC to be less sensitive to illumination changes (Rao et al., 2014). Nevertheless, the design of SSIM closely mimics the human visual system (HSV), because it is normalized for luminance and contrast at the same time. According to the definition of SSIM, as shown in Equation 2.21, SSIM compared the dissimilarities based on structural information after removing the influence of luminance and contrast (Z. Wang et al., 2004); a consideration that is not found in the NCC loss function.

In the case of retinal image registration, the network trained with SSIM achieved the best performance (0.8506 on dice score). Retinal images are characterized by the line structure of the vascular network constructed by the blood vessels (Nirmala et al., 2011). Therefore, SSIM and NCC which are based on structural information outperform the MSE loss function. As we mentioned before, SSIM is less sensitive to the change of luminance as well as contrast, while NCC is only less sensitive to the change of luminance. Therefore, in retinal image registration, SSIM guides networks to achieve the best performance.

Correspondingly, in the case of brain images (shown in Figure 5.2), the cortical feature is the most salient visible feature of the human brain (Thompson et al., 2000). In most cases, networks trained with the MSE loss function achieve the best performance (0.9275 on dice score) in brain image registration. Because MSE calculates the dissimilarities globally, it kept the prominent feature information of brain image viz. cortical feature to a large degree. However, when networks were trained with NCC and SSIM, the information on cortical features would have been removed during the process of extracting structural

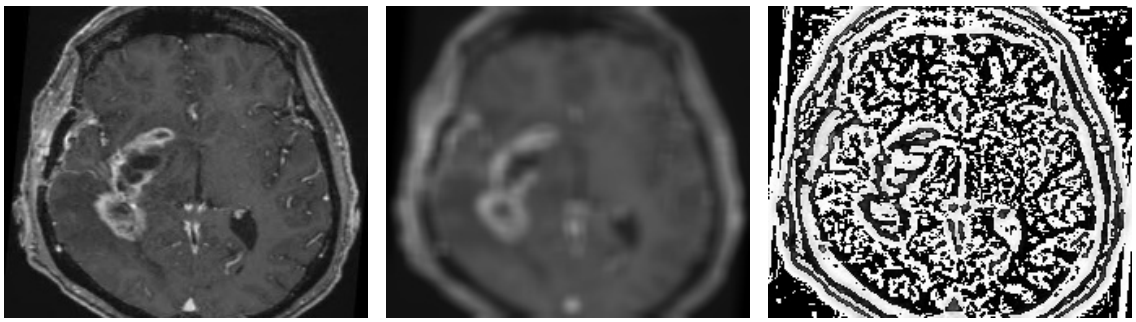
information (as shown in Figure 5.2c). Cortical is a wrinkled appearance unlike vessels as a noticeable structure, the pixel intensities of the cortical feature are very close, thus the mean intensity is close to each pixel intensity within a patch. After subtracting mean intensity, the value of cortical features might be near zero. Besides, it is clearly observed in Figure 5.2c that some noises were introduced into the captured structural information, these noises would affect the accurate measurement of dissimilarities as well. In addition, we observed that when the networks are trained with the NCC loss function, the performances are lower than the dice score calculated from the image pairs before registration. The reason might be that NCC cannot calculate the accuracy similarities between a pair of images if there is not enough structural information in this image pair (Rao et al., 2014).

Further, we noticed that when scales of the transformation field change sufficiently enough, the prominent feature to be learned by networks would vary (Rajput, 2021). For example, when there exists a larger transformation field between brain image pairs, the most prominent feature might change to the edge structure such as sulcal features (Thompson et al., 2000). Therefore, in the case of large-displacement brain image registration, SSIM which is tailored to calculate difference based on structure performed better than MSE in similar networks. As an illustration, the dice scores of large-sized transformer-based networks trained with SSIM and MSE are 0.7123 and 0.6799 respectively.



(a) Original image **(b)** Mean image **(c)** Original-Mean

Figure 5.1: An example of a retinal image to capture structural information by subtracting mean intensity



(a) Original image **(b)** Mean image **(c)** Original-Mean

Figure 5.2: An example of a brain image to capture structural information by subtracting mean intensity

5.5 Chapter Summary

This chapter theoretically analyzed the results represented in Chapter 4. In terms of the relationship between performance and architecture, larger-sized networks are helpful to improve performance, because more neurons in each hierarchical layer can generate more features from simple and complex, then these generated features provided rich information to align anatomical structures. In terms of architecture components, multi-head self-attention components are incorporated to predict the similarities results in different representation subspaces, which is useful to improve the registration performance. However, it is important to choose the number of heads in self-attention components, because additional heads might degrade the performance by learning overlapping features or breaking a complex feature into several smaller unrealistic and meaningless features.

Further, we explained why transformer-based networks are able to improve the registration performance of convolutional-based networks. From the perspective of the effective receptive fields, self-attention components are able to calculate the similarities in the global range, while convolutional operations are limited to a local range depending on the size of the kernel and the dimension of input. Essentially, image registration is a pixel-level task. Self-attention components satisfied spatial permutation equivariance, which is the most prominent characteristic in a pixel-level task. Satisfying this characteristic makes self-attention components play a positive role in predicting displacement for each pixel even when registering large-displacement image pairs.

Lastly, loss functions are related to the parameters of architectures and these two factors

interact to affect the registration performance. Basically, more parameters are helpful to smoothen the loss landscape to improve the generalization abilities. Besides, the choice of loss function should consider the prominent features in datasets. To be precise, SSIM compared the dissimilarities based on structural information, thus it is the best loss function to calculate the dissimilarities of retinal images containing multiple line structures of the vascular network. In addition, unlike vessels which are the noticeable structures in retinal images, cortical features are the salient visible structures in brain images. Thus, MSE is the most suitable loss function to guide networks to complete brain image registrations.

Chapter 6

Conclusion

This chapter briefly summarises our key findings and contributions, addresses our limitations and recommends further study.

6.1 Summary

This thesis aimed to empirically study deep neural networks for non-rigid medical image registration. To the best of our knowledge, this is the first study that comparatively analyzes how registration performance is influenced by a variety of factors: components of architectures, the scale of transformation fields, loss functions and datasets in the task of medical image registration. Based on a series of controlled experiments and ablation studies, it can be concluded that these four factors are interacting to affect registration performance. To be precise, from the viewpoint of architecture, the results indicated that larger-sized architectures in respect of convolutional-based and transformer-based networks are able to improve the generalization ability in most scenarios. Also, a transformer-based network with a suitable number of heads in multi-head self-attention components could provide better performance compared to a convolutional-based counterpart. Otherwise, additional heads could possibly degrade the performance due to producing redundant similarities. As for the relationship between registration performance and scale of transformation field, the results indicated that the performance of transformer-based architecture becomes superior to the performance of convolutional-based architecture, especially in medium to large-displacement transformation fields. Again, from the viewpoint of loss

functions, we noticed that the loss function interacting with different architectures behaves differently. The change in registration performance with loss function exhibits a similar trend with regard to different architectures viz. convolutional-based and transformer-based networks. Lastly, from the viewpoint of the dataset, the underlying probability distributions of retinal and brain image datasets are different; the salient features representative of the images (retina and brain in our example) are different. The results indicated that loss function tailored to salient features of the image will most likely lead to better registration. In sum, the registration performance is not only influenced by one factor alone.

Unlike that, a variety of survey papers that only grouped registration works based on various taxonomies such as typical structures of architectures, modality of input image pairs and learning algorithms, our thesis provides a comprehensive analysis of performance comparisons along with statistical significance test results. Furthermore, our work was not directed at improving the performance of transformer-based networks, but rather to provide empirical insights into the relationship between performance and a series of factors viz. architectures, scale of transformation fields, loss functions and datasets respectively. Also, we provided insights into how these factors interacting with others affect performance. Further, we investigated the machinery of transformer-based networks in medical image registration, we interpreted how and why transformer-based networks are able to address the limitation of convolutional-based networks. In addition, we explored the intuitive theory of loss functions commonly used in medical image registration, exploring the correlation between loss functions and datasets. In order to enable our work, we generated image registration datasets by applying realistic transformation with random groups of parameters on image segmentation datasets, which address the problem of the shortage of publicly available image registration datasets. Overall, our work provides insightful knowledge about the relationship among architectures, the scale of transformation fields, loss functions and the characteristics of datasets to consider when we design a deep neural network for registration tasks.

6.2 Further Work

The research clearly illustrates how loss function interacting with architecture and datasets affects performance, but it also raises the question of how loss function interacting with different optimization methods achieves the global minima. Thus, determining the relationship between the registration performance and the setting of hyperparameters such as learning rate, dropout rate and batch size is significant.

Additionally, although transformer-based networks outperformed convolutional-based networks in most cases, the amount of training dataset might not be large enough to make transformer-based networks achieve the best generalization ability. In our work, even the amount of parameters of small-sized architecture is larger than the amount of training dataset. Therefore, further research could investigate the relationship between the amount of dataset and the size of architecture in the case of medical image registration.

Also, we designed an adaptive combination loss function to train registration networks. The combination loss function consisted of three different loss functions (viz. MSE, NCC and SSIM) with corresponding trainable weights. However, the results were not as expected and we suspect that placing the loss layer at the end of the network may not have been a good strategy. We expected that the characteristics of the training dataset would have guided the appropriate weighting of the respective loss functions. Fruitful further work could be directed towards placing the separate loss functions at different levels in the network.

Furthermore, based on our insight into how self-attention components work in medical image registration, proposing a pure and novel transformer network to improve the registration performance within the consideration of loss functions, the scale of transformation fields and datasets is recommended for further study.

Bibliography

Adal, K. M., van Etten, P. G., Martinez, J. P., van Vliet, L. J., & Vermeer, K. A. (2015). Accuracy assessment of intra-and intervisit fundus image registration for diabetic retinopathy screening. *Investigative ophthalmology & visual science*, *56*(3), 1805–1812.

Avants, B. B., Epstein, C. L., Grossman, M., & Gee, J. C. (2008). Symmetric diffeomorphic image registration with cross-correlation: Evaluating automated labeling of elderly and neurodegenerative brain. *Medical image analysis*, *12*(1), 26–41.

Balakrishnan, G., Zhao, A., Sabuncu, M. R., Guttag, J., & Dalca, A. V. (2019). Voxelmorph: A learning framework for deformable medical image registration. *IEEE transactions on medical imaging*, *38*(8), 1788–1800.

Balakrishnan, G., Zhao, M. R., Amyand Sabuncu, Guttag, J., & Dalca, A. V. (2018). An unsupervised learning model for deformable medical image registration. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 9252–9260.

Bhuvaji, S., Kadam, A., Bhumkar, P., Dedge, S., & Kanchan, S. (2020). Brain tumor classification (MRI). https://doi.org/10.34740/KAGGLE/DSV/1183165

Boveiri, H. R., Khayami, R., Javidan, R., & Mehdizadeh, A. (2020). Medical image registration using deep neural networks: A comprehensive review. *Computers & Electrical Engineering*, *87*, 106767.

Budai, A., Bock, R., Maier, A., Hornegger, J., & Michelson, G. (2013). Robust vessel segmentation in fundus images. *International journal of biomedical imaging*, *2013*.

Chen, J., He, Y., Frey, E. C., Li, Y., & Du, Y. (2021). Vit-v-net: Vision transformer for unsupervised volumetric medical image registration. *arXiv preprint arXiv:2104.06468*.

Chen, X., Diaz-Pinto, A., Ravikumar, N., & Frangi, A. (2020). Deep learning in medical image registration. *Progress in Biomedical Engineering*.

Chollet, F. (2018). *Deep learning mit python und keras: Das praxis-handbuch vom entwickler der keras-bibliothek*. MITP-Verlags GmbH & Co. KG.

Cordonnier, J.-B., Loukas, A., & Jaggi, M. (2019). On the relationship between self-attention and convolutional layers. *arXiv preprint arXiv:1911.03584*.

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, *2*(4), 303–314.

Dalca, A. V., Balakrishnan, G., Guttag, J., & Sabuncu, M. R. (2018). Unsupervised learning for fast probabilistic diffeomorphic registration. *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 729–738.

De Silva, T., Chew, E. Y., Hotaling, N., & Cukras, C. A. (2021). Deep-learning based multi-modal retinal image registration for the longitudinal analysis of patients with age-related macular degeneration. *Biomedical Optics Express*, *12*(1), 619–636.

De Vos, B. D., Berendsen, F. F., Viergever, M. A., Sokooti, H., Staring, M., & Išgum, I. (2019). A deep learning framework for unsupervised affine and deformable image registration. *Medical image analysis*, *52*, 128–143.

Decencière, E., Cazuguel, G., Zhang, X., Thibault, G., Klein, J.-C., Meyer, F., Marcotegui, B., Quellec, G., Lamard, M., Danno, R., et al. (2013). Teleophta: Machine learning and image processing methods for teleophthalmology. *Irbm*, *34*(2), 196–203
e-ophtha dataset http://www.adcis.net/en/third-party/e-ophtha/.

Decencière, E., Zhang, X., Cazuguel, G., Lay, B., Cochener, B., Trone, C., Gain, P., Ordonez, R., Massin, P., Erginay, A., et al. (2014). Feedback on a publicly distributed image database: The messidor database. *Image Analysis & Stereology*, *33*(3), 231–234.

Deng, L. (2012). The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, *29*(6), 141–142.

Devore, J. L. (2008). Probability and statistics for engineering and the sciences.

Dinh, L., Pascanu, R., Bengio, S., & Bengio, Y. (2017). Sharp minima can generalize for deep nets. *arXiv preprint arXiv:1703.04933*.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Fraz, M. M., Remagnino, P., Hoppe, A., Uyyanonvara, B., Rudnicka, A. R., Owen, C. G., & Barman, S. A. (2012). An ensemble classification-based approach applied to retinal blood vessel segmentation. *IEEE Transactions on Biomedical Engineering*, *59*(9), 2538–2548. https://doi.org/10.1109/TBME.2012.2205687

Fu, Y., Lei, Y., Wang, T., Curran, W. J., Liu, T., & Yang, X. (2020). Deep learning in medical image registration: A review. *Physics in Medicine & Biology*.

Géron, A. (2019). *Hands-on machine learning with scikit-learn, keras, and tensorflow: Concepts, tools, and techniques to build intelligent systems*. " O'Reilly Media, Inc.".

Goodfellow, I., Bengio, Y., Courville, A., & Bengio, Y. (2016). *Deep learning* (Vol. 1). MIT press Cambridge.

Haskins, G., Kruger, U., & Yan, P. (2020). Deep learning in medical image registration: A survey. *Machine Vision and Applications*, *31*(1), 1–18.

Hennig, C., & Kutlukaya, M. (2007). Some thoughts about the design of loss functions. *REVSTAT–Statistical Journal*, *5*(1), 19–39.

Hernandez-Matas, C., Zabulis, X., Triantafyllou, A., Anyfanti, P., Douma, S., & Argyros, A. A. (2017). Fire: Fundus image registration dataset. *Modeling and Artificial Intelligence in Ophthalmology*, *1*(4), 16–28.

Hu, J., Shen, L., & Sun, G. (2018). Squeeze-and-excitation networks. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 7132–7141.

Huang, W. R., Emam, Z., Goldblum, M., Fowl, L., Terry, J. K., Huang, F., & Goldstein, T. (2020). Understanding generalization through visualizations.

Jaderberg, M., Simonyan, K., Zisserman, A., et al. (2015). Spatial transformer networks. *Advances in neural information processing systems*, 2017–2025.

Ji, S., Xie, Y., & Gao, H. (2019). A mathematical view of attention models in deep learning. *Texas A&M University*.

Kauppi, T., Kalesnykiene, V., Kamarainen, J.-K., Lensu, L., Sorri, I., Raninen, A., Voutilainen, R., Uusitalo, H., Kälviäinen, H., & Pietilä, J. (2007). The diaretdb1 diabetic retinopathy database and evaluation protocol. *BMVC*, *1*, 1–10.

Keskar, N. S., Mudigere, D., Nocedal, J., Smelyanskiy, M., & Tang, P. T. P. (2016). On large-batch training for deep learning: Generalization gap and sharp minima. *arXiv preprint arXiv:1609.04836*.

Krebs, J., Mansi, T., Mailhé, B., Ayache, N., & Delingette, H. (2018). Unsupervised probabilistic deformation modeling for robust diffeomorphic registration. *Deep learning in medical image analysis and multimodal learning for clinical decision support* (pp. 101–109). Springer.

Kuckertz, S., Papenberg, N., Honegger, J., Morgas, T., Haas, B., & Heldmann, S. (2020). Learning deformable image registration with structure guidance constraints for adaptive radiotherapy. *International Workshop on Biomedical Image Registration*, 44–53.

Le, H., & Borji, A. (2017). What are the receptive, effective receptive, and projective fields of neurons in convolutional neural networks? *arXiv preprint arXiv:1705.07049*.

Li, H., Xu, Z., Taylor, G., Studer, C., & Goldstein, T. (2018). Visualizing the loss landscape of neural nets. *Advances in Neural Information Processing Systems*, 6389–6399.

Liu, B., Balaji, Y., Xue, L., & Min, M. R. (2020). Analyzing attention mechanisms through lens of sample complexity and loss landscape.

Ma, C., Wojtowytsch, S., Wu, L., et al. (2020). Towards a mathematical understanding of neural network-based machine learning: What we know and what we don't. *arXiv preprint arXiv:2009.10713*.

Mahapatra, D., Sedai, S., & Garnavi, R. (2018). Elastic registration of medical images with GANs. *arXiv preprint arXiv:1805.02369*.

Marstal, K., Berendsen, F., Staring, M., & Klein, S. (2016). Simpleelastix: A user-friendly, multi-lingual library for medical image registration. *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 134–142.

Mohri, M., Rostamizadeh, A., & Talwalkar, A. (2018). *Foundations of machine learning*. MIT press.

Mok, T. C., & Chung, A. (2020). Fast symmetric diffeomorphic image registration with convolutional neural networks. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4644–4653.

Nguyen, T., Raghu, M., & Kornblith, S. (2020). Do wide and deep networks learn the same things? uncovering how neural network representations vary with width and depth. *arXiv preprint arXiv:2010.15327*.

Nirmala, S., Nath, M. K., & Dandapat, S. (2011). Retinal image analysis: A review. *International Journal of Computer & Communication Technology (IJCCT)*, *2*(6), 11–15.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. (2019). Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, *32*, 8026–8037.

Rahane, A. A., & Subramanian, A. (2020). Measures of complexity for large scale image datasets. *2020 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, 282–287.

Rajput, V. (2021). Robustness of different loss functions and their impact on networks learning capability. *arXiv preprint arXiv:2110.08322*.

Rao, Y. R., Prathapani, N., & Nagabhooshanam, E. (2014). Application of normalized cross correlation to image registration. *International Journal of Research in Engineering and Technology*, *3*(5), 12–16.

Rosasco, L., Vito, E. D., Caponnetto, A., Piana, M., & Verri, A. (2004). Are loss functions all the same? *Neural Computation*, *16*(5), 1063–1076.

Staal, J., Abràmoff, M. D., Niemeijer, M., Viergever, M. A., & Van Ginneken, B. (2004). Ridge-based vessel segmentation in color images of the retina. *IEEE transactions on medical imaging*, *23*(4), 501–509.

Stergios, C., Mihir, S., Maria, V., Guillaume, C., Marie-Pierre, R., Stavroula, M., & Nikos, P. (2018). Linear and deformable image registration with 3D convolutional neural networks. *Image analysis for moving organ, breast, and thoracic images* (pp. 13–22). Springer.

Swinscow, T. D. V., Campbell, M. J. et al. (2002). *Statistics at square one*. Bmj London.

Swirszcz, G., Czarnecki, W. M., & Pascanu, R. (2016). Local minima in training of neural networks. *arXiv preprint arXiv:1611.06310*.

Thiese, M. S., Ronna, B., & Ott, U. (2016). P value interpretations and considerations. *Journal of thoracic disease*, *8*(9), E928.

Thompson, P. M., Mega, M. S., Narr, K. L., Sowell, E. R., Blanton, R. E., & Toga, A. W. (2000). Brain image analysis and atlas construction. *Handbook of Medical Image Proc. and Analysis*.

Tuli, S., Dasgupta, I., Grant, E., & Griffiths, T. L. (2021). Are convolutional neural networks or transformers more like human vision? *arXiv preprint arXiv:2105.07197*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, *30*.

Wang, J., & Zhang, M. (2020). Deepflash: An efficient network for learning-based medical image registration. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4444–4452.

Wang, Z., Bovik, A. C., Sheikh, H. R., & Simoncelli, E. P. (2004). Image quality assessment: From error visibility to structural similarity. *IEEE transactions on image processing*, *13*(4), 600–612.

Wang, Z., & Delingette, H. (2021). Attention for image registration (AiR): A transformer approach.

Zagoruyko, S., & Komodakis, N. (2016). Wide residual networks. *arXiv preprint arXiv:1605.07146*.

Zhao, H., Jia, J., & Koltun, V. (2020). Exploring self-attention for image recognition. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10076–10085.

Zitova, B., & Flusser, J. (2003). Image registration methods: A survey. *Image and vision computing*, *21*(11), 977–1000.