G53IDS Dissertation

Deep Learning based Brain MRI Registration

Submitted [4th May 2020], in partial fulfilment of

the conditions for the award of the degree BSc Hons Computer Science

and Artificial Intelligence

Ang Ding

16522104

School of Computer Science

University of Nottingham

Supervised by Dr. Xin Chen

I hereby declare that this dissertation is all my own work, except as

indicated in the text:

Data: 04/May/2020

Abstract

In this project, we build a deep neural network for 3D brain MRI affine registration. Unlike conventional registration method, our network directly predicts transformation parameters between a moving image and a fixed image. Three methods are proposed to improve the performance of the network based on ground truth provided by a conventional method. Experiments prove that the performance of our network is close to a conventional method and the speed of affine registration on our network is 10 times faster than a conventional registration method.

Contents

1 Introduction

Image registration is the process of aligning a source image to a target image. More generally, it determines the spatial transformation that maps points in one image to corresponding points in the other image. Image registration has a number of valuable applications in medical image processing. Medical imaging devices can provide images containing accurate anatomical information and functional information such as CT and MRI. Doctors have to make a diagnosis with his spatial imagination and subjective experience by comparing different images. With the correct image registration methods, the information obtained from each image can be accurately integrated into the same image, making it easier and more accurate for doctors to observe lesions and diagnose the disease. In addition, medical images obtained from the same patient at different times may be different due to the development of the disease, the scanning angle or the deformation caused by breathing, etc. By registering images which are collected at different time, the changes of lesions and organs can be quantitatively analyzed, making medical diagnosis and surgical plan more accurate and reliable.

Magnetic resonance imaging (MRI) is a medical imaging technique used in radiology to form pictures of the anatomy and the physiological processes of the body. MRI has wild applications in clinical practice for medical diagnosis. In this project, we apply deep learning to register MR T1 weighted structural image to MNI152 brain template. T1 weighted structural imaging provides information relating to volumes and morphology of brain tissues and structures **Error! Reference source not found.**. MNI152 Template was created in 2001 by the Montreal Neurological Institute (MNI) from 3D brain MRI images of 152 normal subjects [2]. It is a current standard template for brain image registration.

The conventional methods of image registration can be classified into intensity-based method and feature-based method [3]. Intensity-based methods directly calculate the correlation value(grey level or colors) by means of correlation operation to find the best matching position. These methods are preferably applied when the images do not have remarkable feature. Feature-based methods detect salient features of images such as significant regions, lines or points. These methods are typically applied when the features of images are more significant than the image intensities. Many registration tasks require manually alignments in the past, the quality of registration is highly dependent upon the expertise of the user [4]. Although several medical registration tools such as FSL[5] and SimpleITK[6] have been developed with better performance, most of them are time-consuming, especially in the high dimensional deformable registration. The speed of registration cannot meet the requirement of clinical real-time registration.

With the rapid development of deep learning in recent years, significant breakthroughs have been made in many fields through the application of deep learning. Some recent studies have applied deep learning to image registration with remarkable success. Deep learning approaches such as convolutional neural network(CNN) has strong end-to-end learning ability [13]. In this project, we build CNN-based models to directly estimate affine transformation matrix between the input image pair(T1 weighted brain image and MNI152 Template).

The aim in this project is to apply deep learning methods in brain MRI registration, significantly reducing processing time of image registration.

The key objectives of the project are as follows:

1. Build a convolutional neural network to predict affine transformation matrix between a moving image and the fixed image

2. Investigate different approaches to improve the performance of deep learning based affine image registration.

3. Significantly reduce the running time of affine image registration compared with a conventional method.

2 Motivation

Medical imaging is the technology of creating interior tissue images of human body in a non-invasive way for medical treatment or medical research. Medical image registration is an important technique in the field of medical image processing. The aim of medical image registration is to find a spatial transformation to map one image to another image, so that the points corresponding to the same position in the two images can be matched, so as to achieve the information fusion. Image registration is widely applied in various aspects of medical imaging, such as lesion detection, disease diagnosis, surgical planning, surgical navigation, and efficacy evaluation. The first motivation of the project is the importance and wide application of image registration in medical image processing.

The previous conventional methods of image registration have achieved high accuracy. However, the conventional methods are optimization-based, which measures the similarity of two images through iteration. These conventional methods have a slow speed and cannot meet the requirements of real-time registration. Compared with the conventional methods, deep learning based image registration obtains a specific transformation model through training, and registration was completed through the model instead of through iterative algorithm to optimize the cost function from scratch for every new registration task [7]. The second motivation of the project is to overcome the drawbacks of conventional methods.

This project focuses on deep learning based medical image registration. According to

Geert Litjens et al. [8], there are two strategies in this filed: (1) using deep-learning networks to learn a similarity metric for two images and generate estimate transformation parameters by an optimization algorithm, and (2) to directly predict transformation parameters using deep neural networks. The first method still involves time-consuming iterative optimization, so this project adopts the second method for registration. CNNs is the most commonly used deep neural networks in image processing and has strong end-to-end learning ability. The second deep learning based method is a clear end-to-end learning process. Therefore, the third motivation of the project is the advantages and potential of deep learning based image registration.

3 Related Work

Conventional methods in image registration are optimization-based which is accurate but time time-consuming. UK Biobank released 10,000 brain imaging data in 2017, and they developed an automated processing and quality control pipeline using optimization-based image registration to process those brain imaging data. For T1 structural image, they used a library of brain analysis tools called FMRIB Software Library(FSL). The linear registration is applied first using FLIRT(FMRIB's Linear Image Registration Tool), then the non-linear registration is applied using FNIRT(FMRIB's Nonlinear Image Registration Tool). The accuracy of the result is 9821/9995(98.3%) which is high enough [1]. However, in practice the average processing time for registering a 3D brain image using FNIRT is 28.4 minutes [14].

In recent years, deep learning is a popular technology in the research of medical image registration, more and more researchers are studying this method, and many related works have been published. Deep learning based image registration can be divided into three categories: deep iterative registration, supervised image registration, and unsupervised image registration [4].

Deep iterative registration leverages both a metric that quantifies the similarity between two images and an optimization algorithm which generate the transformation matrix between two images. Wu et al. [11] proposed a learning based image registration framework based on deep iterative registration. This method is an improvement compared with conventional methods, but it still cost lots of time when predicting transformation parameters.

The slow registration of iterative methods motivated the development of networks that can directly estimate transformation matrix. Supervised image registration is a popular method to speed up the registration process. In the fully supervised registration, ground truth is used to train deep learning models and estimate transformation parameters. Chee et al. [7] utilized a CNN called affine image registration network (AIRNet) to predict the affine transformation parameters. The performance of their model is 100x speed-up in execution time compared to some conventional methods.

Rohe et al. [12] built a U-net model to estimate deformation field for 3D cardiac MR volumes registration. Cao et al. [9] proposed a CNN based regression model to directly learn the transformation from the input image pair and their corresponding deformation field. Their method outperforms some conventional methods such as SyN and Demons. According to Haskins et al. [4], one of the limitations of fully supervised image registration is the lack of the ideal ground truth for training and the quality of the supervised registration is dependent on the quality of the ground truth. Since it is difficult to obtain transformation of every input image pair, creating a synthetic dataset using the existing medical images can avoid such limitations. However, it is important to ensure that the reliability of the simulated data is equal to that of the clinical data.

Apart from fully supervised registration we mentioned above, dual supervised registration is also a branch of supervised registration. Dual supervised registration uses both ground truth and some image similarity qualification metric to train a model. Fan et al[14] built a dual supervised U-net model to estimate deformation field. They used the Euclidean distance between predicted deformation field and ground truth as well as the Euclidean distance between warped image and the template. The final loss function is the sum of two losses. This method reduces the dependence of registration on ground truth by introducing another loss function.

Although supervised image registration has been a great success, obtaining reliable ground truth remains a big obstacle [15]. Many unsupervised registration methods are motivated to estimate transformations without ground truth. One approach is to evaluate image similarity between a moving image and a fixed image. de Vos et al. [16] proposed a deep learning image registration framework for unsupervised affine and deformable image registration based on NCC and a bending-energy regularization term. Another approach is feature based unsupervised registration. Yoo et al[17] used a spatial transformer network for deformable image registration based on features. Although some studies on unsupervised registration have yield promising results, there are still many limitations of the image similarity metrics especially on multimodal registration.

4 Methodology

4.1 Convolutional Neural Networks(CNNs)

CNN is a kind of deep neural network which involves convolution computation. It is widely applied in the field of image processing. A CNN consists of an input and an output layer, as well as multiple hidden layers. The hidden layers typically consist three common structures: convolutional layer, pooling layer and fully-connected layer. The activation function is commonly a RELU layer which is applied behind convolutional layers. By stacking the convolutional layer, pooling layer and RELU layer, the features of input are transformed into parameters, and finally the output is integrated through the full connection layer. Eventually, backpropagation is applied to update the weights in the network.

4.1.1 Convolutional Operation

Convolution

Convolution is a mathematical operator that generates a third function from two functions f and g. The aim of convolution in CNN is to down-sample the input and extract features from the input.

$$(f * g)(n) = \int_{-\infty}^{\infty} f(\tau)g(n-\tau)d\tau \tag{1}$$

the first argument (function f) to the convolution is the input, and the second (function g) is the kernel[18]. The kernel values are weight for each pixel value. As shown in Figure1, A kernel is placed in the top-left corner of the image. The pixel values are multiplied with the corresponding kernel values and these products are summed up and the sum is placed in the next layer at the point corresponding to the center of the kernel. The kernel carries on the convolution operation from left to right and then top to bottom on the input image according to the stride. The output of a convolutional layer is its feature map and the input of next layer.

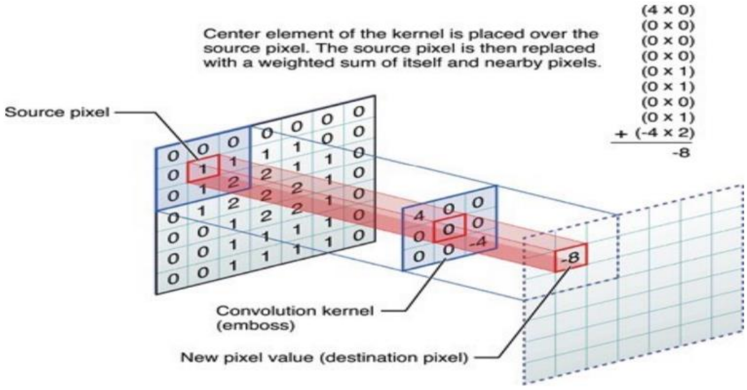

Center element of the kernel is placed over the source pixel. The source pixel is then replaced with a weighted sum of itself and nearby pixels.
(4 x 0)
(0 x 0)
(0 x 0)
(0 x 0)
(0 x 1)
(0 x 1)
(0 x 0)
(0 x 1)
+ (-4 x 2)
-8
Source pixel
Convolution kernel (emboss)
New pixel value (destination pixel)

Figure 1: Example of convolutions

4.1.2 Rectified Linear Units layer(RELU layer)

f(x) = max(0,x)   where x is the input to a neuron.

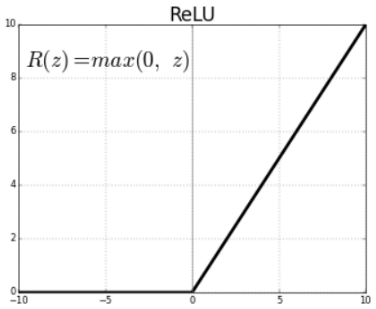
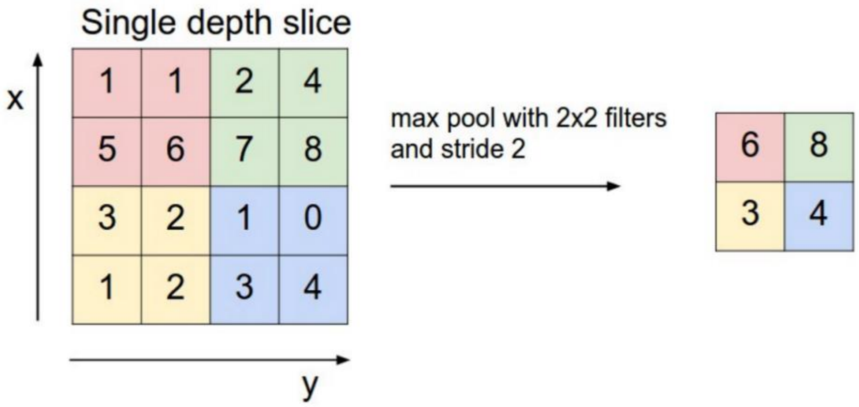


Figure 2: RELU

The rectified linear activation function(Figure 2) is a common activation function in artificial neural network. The function will output itself if it is positive, otherwise, it will output zero. The aim of RELU is to increase the nonlinearity of neural network model.

4.1.3 Max Pooling

Pooling layers reduce the spatial size of the input to decrease the number of parameters and computation. The aim of pooling is downsampling. Max pooling is a common method of pooling which uses the maximum value from each patch of the feature map. As shown in Fig.4, the input is a 4*4 matrix and a 2*2 filter with a stride of two pixels is applied to it. For each region covered by filters, take the maximum value of the region. The output is the result of max pooling.



1	1	2	4
5	6	7	8
3	2	1	0
1	2	3	4

6	8
3	4

Figure 3: Example of max pooling

4.1.4 Backpropagation

Backpropagation is a common method used to train artificial neural networks. When we train a neural network, if the output does not achieve the expectation, the backpropagation recalculates the gradient of the loss function in the network layer by layer. This gradient is fed back to the gradient descent algorithm to update the weights to minimize the loss function.

4.2 Transformation models

Spatial transformation models play an important role in image registration. It maps the source image to the target image mathematically. There are three types of transformation model including rigid transformation, affine transformation and non-rigid transformation.

4.2.1 Rigid Transformation

A rigid transformation (also called an isometry) is a transformation of the plane that preserves length. The distance between any two points before and after the transformation remains the same. The rigid transformations include rotations, translations, reflections and their combination.

4.2.2 Affine Transformation

Affine transformation is a linear transformation of a vector space with a translation into another vector space. Sets of parallel lines remain parallel and the ratios between sets of lines remain the same after an affine transformation. Affine transformations include translation, rotation, scaling, reflection, shear and compositions of them in any combination and sequence.

An affine transformation contains two functions: a translation b and an affine map A which can be represented as:

$$\vec{\boldsymbol{y}} = \boldsymbol{A}\vec{\boldsymbol{x}} + \vec{\boldsymbol{b}}$$

It is equivalent to the matrix in figure 4.

$$\begin{bmatrix} \vec{y} \\ 1 \end{bmatrix} = \begin{bmatrix} A & \vec{b} \\ 0, \ldots, 0 & 1 \end{bmatrix} \begin{bmatrix} \vec{x} \\ 1 \end{bmatrix}$$

Figure 4: Form of affine transformation

The last row vector 0…0,1 is a technique for augmented matrix. This technique maps the space to a subset of a space with an additional dimension. The affine transformation can be applied at the higher-dimensional space by means of linear transformation.

In this project, we apply 3D affine transformation to warp 3D brain MR images. A 3D affine transformation is defined by 3 translations, 3 shear, 3 scale and 3 rotations along x, y, z axes as shown in figure 5.

$$\underset{\text{3 translations}}{\begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}} \times \underset{\text{3 shear}}{\begin{bmatrix} 1 & h_{xy} & h_{xz} & 0 \\ h_{yx} & 1 & h_{yz} & 0 \\ h_{zx} & h_{zy} & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}} \times \underset{\text{3 scale}}{\begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}} \times \underset{\text{3 rotations}}{\begin{bmatrix} r_{11} & r_{12} & r_{13} & 0 \\ r_{21} & r_{22} & r_{23} & 0 \\ r_{31} & r_{32} & r_{33} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}$$

Figure 5: Different transformation types in 3D affine transformation

An affine transformation matrix is obtained by applying dot product of above transformation matrices. There are twelve parameters in a 3D affine transformation matrix, nine for shear/scale/rotation and three for translation(figure 6).

$$\begin{bmatrix} X' \\ Y' \\ Z' \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & t_x \\ a_{21} & a_{22} & a_{23} & t_y \\ a_{31} & a_{32} & a_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Figure 6: Example of 3D affine transformation matrix

4.2.3 Deformable Transformation

Deformable transformation is a non-linear transformation which can locally warp the source image to align with the target image[19]. In linear transformation, the transformation of all pixels in the image are described by transformation parameters. However, the deformable translation allows non-uniform mapping between images. Each of pixel in the images can be warped based on deformation field between two images.

4.3 Image Warping and Interpolation

This section introduces how to compute a warped image based on a given transformation. Section 4.3.1 outlines the theory of image warping. Section 4.3.2 provides interpolation method used in this project.

4.3.1 Image Warping

Image warping is a procedure to generate warped images from source image based on transformation parameters. In image registration, we apply image warping to align a moving image to the fixed image. There are two warping method in the image warping.

- **Forward Warping**

Forward warping maps original image pixels onto warped image. As shown in figure 7, the coordinates of transformed pixels are not integer in the warped image. In this case, there are two problems in forward warping. First, multiple pixels in the original image could hit the same pixel in the warped image. Second, the warped image may have holes as some pixels are not hit.

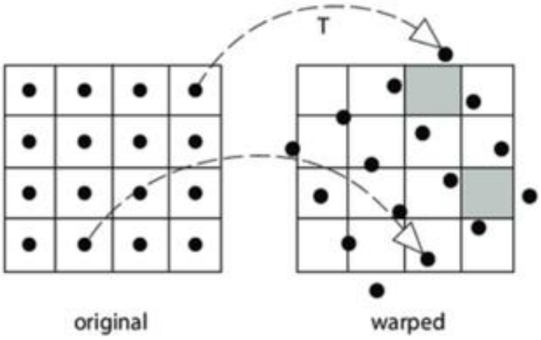


Figure 7: A visualization of forward warping

- **Backward Warping**

Backward warping maps pixels of the warped image back onto the original image. As shown in figure 8, all coordinates in the warped image are transformed back to the original image to avoid holes and overlaps. However, we need to use interpolation to estimate the pixel intensity of the original image. Backward warping is commonly used in image processing.

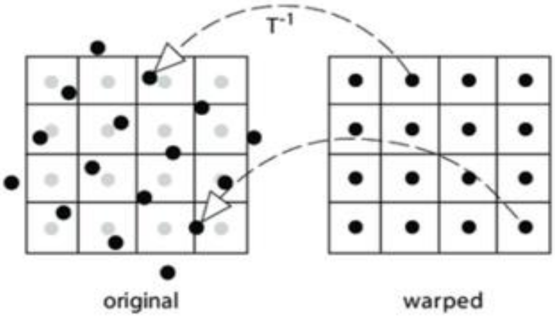


Figure 8: A visualization of backward warping

4.3.2 Interpolation

Image Interpolation is widely used in image generation or resampling [20]. The simplest interpolation is nearest neighbor interpolation. This method takes the value of the closest pixel as the intensity of the pixel in the warped image. The interpolation method applied in this project is tri-linear interpolation. Tri-linear interpolation is a method to interpolate a pixel (x, y, z) in a 3 dimensional space. It calculates a weighted average of the eight neighbor pixels as the estimated intensity of a pixel in the warped image.

5 Design

This section introduces the design of the project. Section 5.1 outlines the data preprocessing pipeline. FMRIB Software Library(FSL) is applied in this section to generate ground truth from raw images. Section 5.2 introduces the architecture of the AIRNet. Section 5.3 shows the loss functions and validation method used in this project. Section 5.4 provides an overview of affine image registration pipeline. Three frameworks are used to investigate a better affine registration performance.

5.1 Data Preprocessing

The raw data of this project is 335 brain MR images and an MNI-152 brain template. These raw image data cannot be directly used as the input of CNN because they have different size and intensity. To solve this problem, raw images require to be preprocessed before inputting into a network. All images are resized to same image size and the range of pixel intensity values of each image is normalized, making the range 0 to 1. A detailed description of data preprocessing is in the section 6.1.

In addition, ground truth is crucial for supervised image registration. A ground truth generation pipeline is provided in this Figure 9. FLIRT is a robust and accurate tool for affine brain image registration. The input of FLIRT is a brain MRI image and the MNI-152 brain template. The output is the ground truth of this project which has two parts. The warped brain is the registered image of the input brain and the affine transformation matrix contains twelve parameters which can map the input brain to the registered brain.

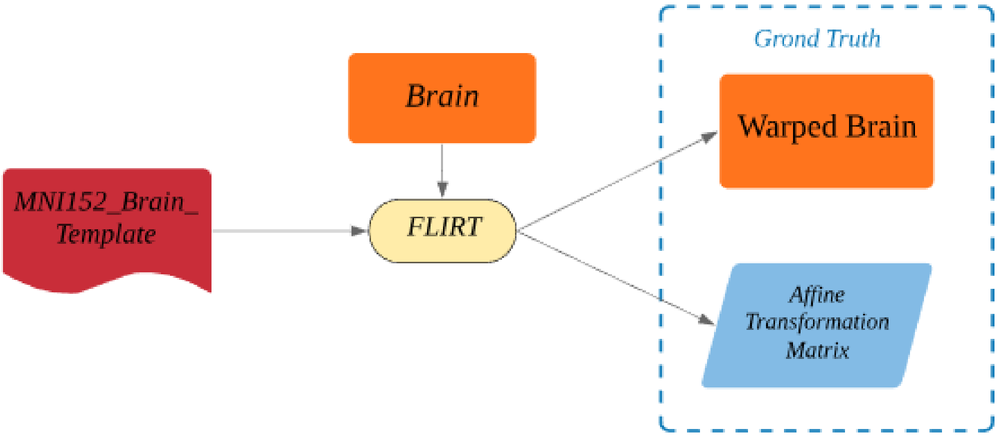


Figure 9: A visualization of ground truth generation

5.2 Architecture of AIRNet

The AIRNet was proposed by Chee et al. [7] in 2018. We imitate the architecture of AIRNet and build a network for this project to estimate the affine transformation matrix between two input images. The architecture of AIRNet is shown in figure 10. There are two parts in the AIRNet: the feature extraction part which extracts features from input images and regression part which integrates features to affine transformation matrix.

The feature extraction part contains two parallel pathways, one for the fixed image and another for the moving image. The benefit of this design is to extract features of each input image separately and avoid features being mixed at the beginning of training. In the feature extraction part, a 3D filter is used in convolutional layer to extract feature maps of the input image. Each convolutional layer is followed by a pooling layer which are added to downsize the feature maps. Two pathways share same parameters and weights in this part.

In regression part, the feature maps of two images are concatenated as the input of fully connected layers. There are three fully connected layers in the regression part. The regression will output an affine transformation matrix with twelve parameters which can register the moving image to the fixed image.

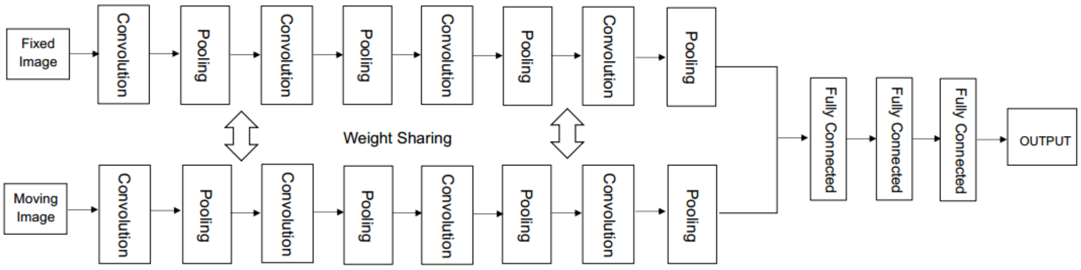


Figure 10: The architecture of AIRNet. Convolution: convolutional layer. Pooling: Max pooling layer. Fully Connected: Fully connected layer. Output: twelve affine transformation parameters.

5.3 Loss Function and Validation

Robust metrics and validation method are important for the accuracy and reliability of a project. This section provides loss functions and validation method we used. In section 6.3.1, two loss functions are introduced to evaluate the performance of models. In section 6.3.2, we validate our models are robust by applying cross validation.

5.3.1 Loss Function

A loss function shows differences between estimated values and true values. Backpropagation algorithm are applied to the network and update weights of the models based on the result of loss functions. In this project, two loss function are used to evaluate the performance of models.

Mean Square Error(MSE)

$$\boldsymbol{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2 \quad (2)$$

Where n is the number of data points, Y_i is the vector of actual values, $\hat{Y}_i$ is vector of the predicted values.

The mean square error(MSE) measures the average squared difference between the predicted data and the real data. In this project, there are two different loss functions use MSE: MSE_transformation and MSE_consistency. MSE_transformation is the MSE between the predicted affine transformation matrix and the ground truth affine transformation matrix. MSE_consistency is the MSE of transformation parameters that estimated using the original image and an augmented version of the original image.

Normalized Cross Correlation(NCC)

$$\boldsymbol{NCC} = \frac{1}{n}\sum_{x,y}\frac{1}{\sigma_f\sigma_t}f(x,y)t(x,y) \quad (3)$$

Where f(x, y) and t(x, y) are two images, σ_f and σ_t are standard deviation of f(x, y) and t(x, y), n is the number of pixels of image.

The normalized cross correlation(NCC) measures the similarity between two images. The range of NCC is -1 to 1. If NCC is close to 1, two images have a high correlation. If NCC is close to -1, two images have a negative correlation. If NCC is equal to 0, two images have no correlation. In this project, the loss function NCC_similarity is the NCC between a warped image and the template.

5.3.2 Cross Validation

Cross validation is a common approach to test the performance, prevent overfitting and evaluate the generalization capability of the model. We apply k-fold cross validation for each experiment to prove that the model is stable on different datasets.

K-fold cross validation splits a dataset into k equally sized subsets. One of the subsets is selected as the validation set and the other k-1 subsets are used as training sets. This process is repeated k times until all of the subsets are used as the validation set. K models are obtained after training on each training set. Then we test K models on a testing set and the average of k results is the estimated performance of the model.

In this project, we have 335 brain MR images as original dataset. We take 65 images for the testing set and 270 images for cross validation. Then we split 270 images to 3 folds and apply a 3-fold cross validation. For each experiment, we obtain 3 models as the results of cross validation.

5.4 Affine Image Registration Pipeline

The aims of this project are to investigate different methods to improve the performance of deep learning based affine image registration and reduce its running time. To achieve these objectives, five experiments are designed to find the method with best performance. In this section, three registration models are provided. The first model is a supervised single step registration model designed for experiment 1 and 2. The second model is a dual supervised registration model for experiment 3 and 4. The third model is a hybrid method model proposed for experiment 5.

As this project is based on deep learning, we build an affine image registration network(AIRNet) to estimate affine transformation matrix between two input images. The fixed image is the MNI-152 brain template and the moving image is a brain MR image each time. The output of the AIRNet is twelve affine transformation parameters which estimate affine registration of two input images. The details of AIRNet are provided in section 6.2.

5.4.1 Supervised Single Step Registration Model

The first model we build is a supervised single step registration model which was proposed by Chee et al. [7]. In this model, the input are two 3D images of the same image size. The fixed image is an MNI-152 brain template and the moving image is a brain MR image to be registered. The output of the AIRNet is twelve affine transformation parameters which predict the affine transformation between two input images. The ground truth in this model is an affine transformation matrix obtained from FSL. We use MSE_transformation as the loss function in this model. After the model converges, an image resampler can register a moving image to the fixed image based on the predicted affine transformation matrix. A visualization of supervised single step registration model is given in figure 11.

This supervised single step registration model is designed for two experiments, experiment 1 and experiment 2. The difference between two experiments is their ground truth. The ground truth in experiment 1 is the affine transformation matrix directly obtained from FSL. And for the ground truth in experiment 2, three translation parameters in twelve affine transformation parameters are normalized to the same range of the other nine shear/scale/rotation parameters. In this case, twelve parameters will have a similar effect in the loss function.

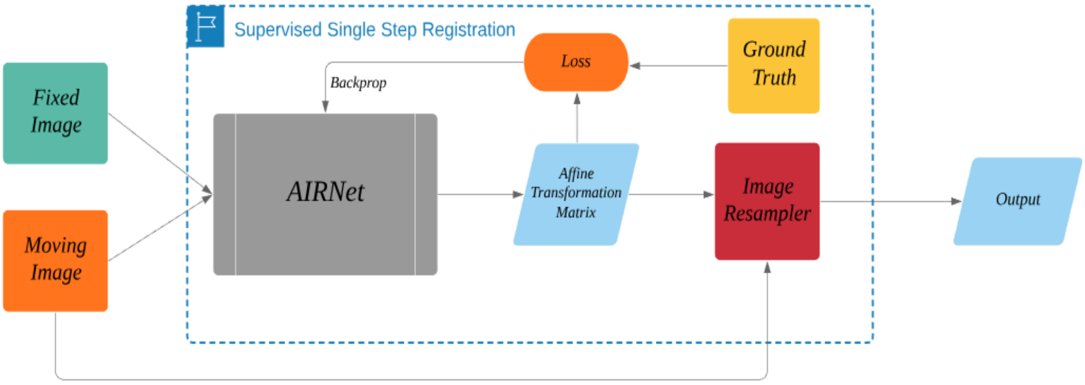


Figure 11: A visualization of Supervised single step registration

5.4.2 Dual Supervised Registration Model

The second model we build is a dual supervised registration model. In this model, the architecture is similar with the supervised single step registration model. The input is two images to be registered, and the output is twelve predicted affine transformation parameters. As we discussed earlier, dual supervised registration uses both ground truth and a metric that qualified image intensity to train the model [4]. A visualization of dual supervised registration model is given in figure 12. The difference between supervised single step registration model and dual supervised registration model is that there are two loss functions in this model. Besides the MSE_transformation between the predicted affine transformation matrix and ground truth, NCC_similarity is also adopted in this model to evaluate the similarity between a warped image and the fixed image. The final loss function is the sum of MSE_transformation and NCC_similarity which combines the advantages of both methods.

Experiment 3 and 4 are designed for the dual supervised registration model. The idea of experiment 3 is based on a paper that use dual supervision. Fan et al.[14] used dual supervising to predict the deformation field for 3D brain MR image registration. One of the loss functions they chose quantify image similarity between the fixed image and the warped image. The NCC_similarity in experiment 3 is the NCC between the fixed image and the warped image currently estimated by the network. However, the difference of image similarity between two images includes local difference which cannot be registered by affine registration. In order to eliminate the effect of local difference of two images, we propose the experiment 4 to use the NCC between ground truth image and the warped image as the NCC_similarity.

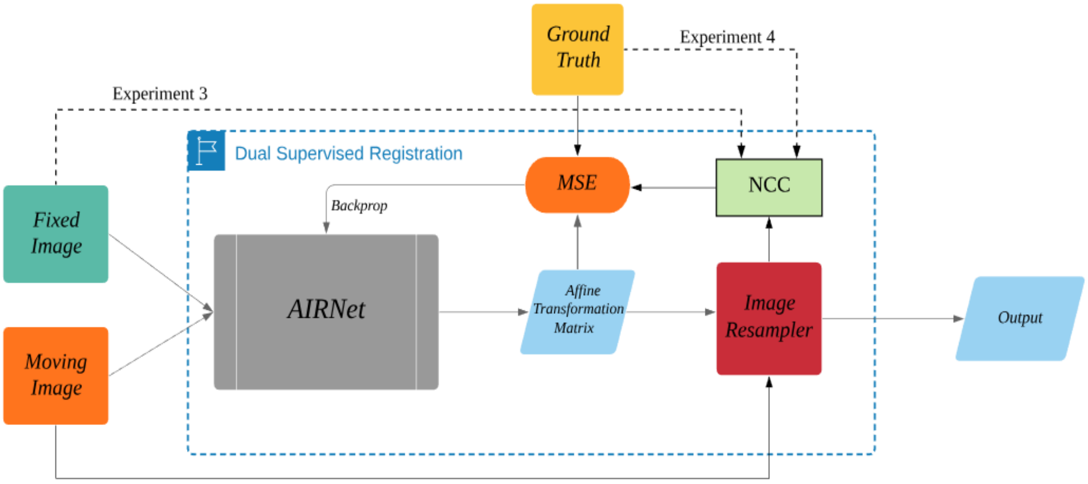


Figure 12: A visualization of dual supervised registration

5.4.3 Novel Dual Supervised Registration Model

The third model we build is a novel dual supervised registration model. There are two pairs of input image in this model. The first pair is the same with previous models, one is the fixed image and another one is the moving images. The second pair is the fixed image and an augmented moving image. The ground truth in this model is an affine transformation matrix obtained from FSL. The augmented moving image is transformed from the moving image in the first pair by a randomly generated augmentation matrix T which contains translation, scale and rotation. Then, two pairs of images are input into the network model successively, and the output is their corresponding predicted affine transformation matrices. The MSE_transformation is the MSE between the ground truth and predicted affine transformation matrix to register the moving images. The MSE_consistency is the MSE between the predicted affine transformation matrix of the moving image and the product of predicted augmented moving image's affine transformation matrix and the augmentation matrix T. The final loss function in the experiment 5 is the sum of MSE_transformation and MSE_consistency. A visualization of this novel dual supervised registration model is given in figure 13.

This model is designed to enhance the ability of the network. In the theory, the registered image of moving image and augmented moving image is identical. The reason is that the augmented image is the same brain as the moving image, even if it is warped by a random augmentation matrix T. For the moving image, its registered image is transformed by its affine transformation matrix predicted by the network. And for the augmented moving image, it is transformed from the moving image by an augmentation matrix T first, and then registered to the fixed image based on its predicted affine transformation matrix. This process is equivalent to the moving image being transformed by the dot product of augmentation matrix T and the predicted affine transformation matrix of the augmented image. Therefore, mathematically, the predicted affine transformation matrix of moving image is equal to the dot product just introduced. However, in reality, there is an error between above two matrices. We hope to minimize this error to improve the robustness of the model in producing consistent prediction.

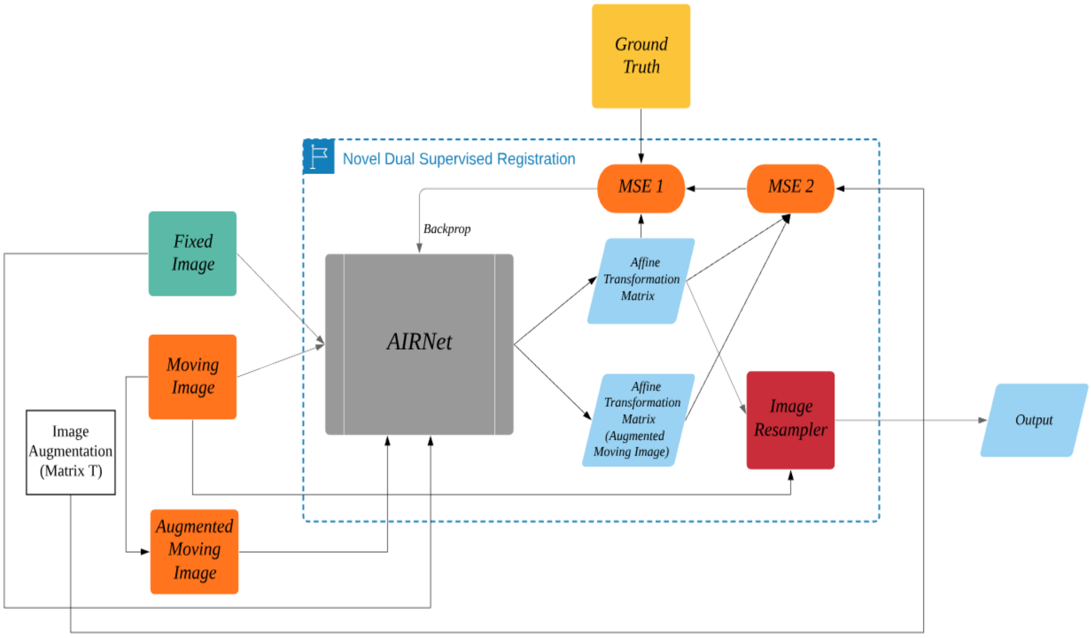

Ground Truth
Novel Dual Supervised Registration
MSE 1
MSE 2
Backprop
Fixed Image
Moving Image
AIRNet
Affine Transformation Matrix
Affine Transformation Matrix (Augmented Moving Image)
Image Resampler
Output
Image Augmentation (Matrix T)
Augmented Moving Image

Figure 13: A visualization of a novel dual supervised affine registration

6 Implementation

This section provides implementation details of this project. Section 6.1 introduces the process of data preprocessing. Section 6.2 shows the architecture of AIRNet. Section 6.3 provides loss functions applied in this project and validation steps used. Section 6.4 provides implementation details and training procedure of each experiment.

6.1 Data Preprocessing

Data preprocessing in this project includes two parts. Details of Image resizing is provided in section 6.1.1. The method of ground truth generation is introduced in section 6.1.2.

6.1.1 Image Resizing

In this project, the raw imaging data is 335 3D brain MR images and an MNI-152 brain template. These images have different image size and need to be resized before training deep learning models. For the given images, the length of each dimension is in the range of 120 to 240 pixels. In order to make the image size same, we pad each image to the size of 256*256*256. However, my GPU is unable to process 256*256*256 images. Therefore, we resize all the images to 64*64*64. Now, we have 335 3D MR images and an MNI-152 brain template in size of 64*64*64 pixels.

6.1.2 Ground Truth Generation

After obtaining resized images, we need to generate the ground truth of them. FSL is a comprehensive library of analysis tools for FMRI, MRI and DTI brain imaging data which runs in Linux. Here, we apply a robust and accurate tool in FSL called FLIRT for affine brain image registration.

We use FLIRT to register brain MR images to MNI-152 template as:

```
flirt -in Brain -ref MNI152_Template -out warped_Brain
-omat transformation_matrix.mat
```

The input for FLIRT each time is a 3D brain MR image and the MNI-152 template. The output is the ground truth of the input brain image which includes an affine transformation matrix and a registered brain image.

6.2 Implementation of AIRNet

This section provides the parameter settings of the AIRNet. For each pathway in the feature extraction part, the input image is firstly convoluted by a 3*3*3 convolutional filter and then activated by a RELU layer. After that, the output feature maps will be downsized by a 2*2*2 max pooling layer. The input image undergoes four convolutional layers and four max pooling layers alternatively. The feature maps are doubled by each convolutional layer and downsized to half by each max pooling layer. The output of each pathway is sixteen 4*4*4 feature maps.

In regression part, the feature maps of two images are concatenated as the input of fully connected layers. There are three fully connected layers in this part. 2048 parameters are regressed to twelve affine transformation parameters after passing through three fully connected layers. In the first layer, a dropout layer is added to prevent the network from overfitting [21]. Twenty percent parameters are dropped out to ensure the neural network will not completely match the training samples, which will help alleviate the overfitting problem. The regression will output an affine transformation matrix with twelve parameters which registers the moving image to the fixed image.

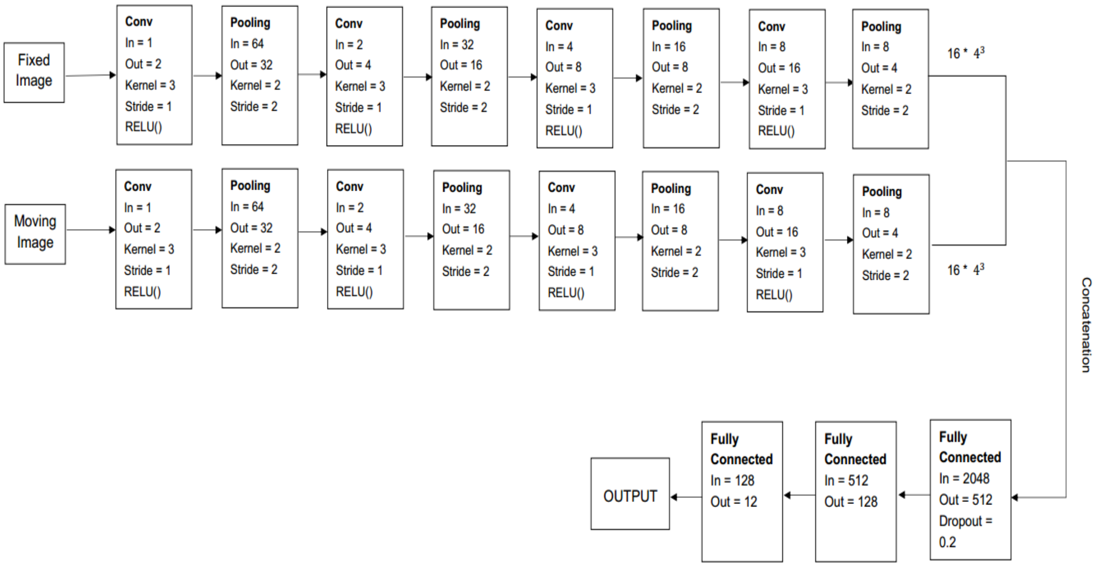

Fixed Image
Conv In = 1 Out = 2 Kernel = 3 Stride = 1 RELU()
Pooling In = 64 Out = 32 Kernel = 2 Stride = 2
Conv In = 2 Out = 4 Kernel = 3 Stride = 1 RELU()
Pooling In = 32 Out = 16 Kernel = 2 Stride = 2
Conv In = 4 Out = 8 Kernel = 3 Stride = 1 RELU()
Pooling In = 16 Out = 8 Kernel = 2 Stride = 2
Conv In = 8 Out = 16 Kernel = 3 Stride = 1 RELU()
Pooling In = 8 Out = 4 Kernel = 2 Stride = 2
16 * 4^3
Moving Image
Conv In = 1 Out = 2 Kernel = 3 Stride = 1 RELU()
Pooling In = 64 Out = 32 Kernel = 2 Stride = 2
Conv In = 2 Out = 4 Kernel = 3 Stride = 1 RELU()
Pooling In = 32 Out = 16 Kernel = 2 Stride = 2
Conv In = 4 Out = 8 Kernel = 3 Stride = 1 RELU()
Pooling In = 16 Out = 8 Kernel = 2 Stride = 2
Conv In = 8 Out = 16 Kernel = 3 Stride = 1 RELU()
Pooling In = 8 Out = 4 Kernel = 2 Stride = 2
16 * 4^3
Concatenation
Fully Connected In = 2048 Out = 512 Dropout = 0.2
Fully Connected In = 512 Out = 128
Fully Connected In = 128 Out = 12
OUTPUT

Figure 14: Parameter settings of AIRNet. In: The number of input channel. Out: The number of output channel. Kernel: The kernel size of the 3D filter. Stride: The stride of the 3D filter.

6.3 Implementation of Experiments

The details of each experiment are provided in this section. As we discussed earlier, there are three models in this project. In section 6.4.1, experiment 1 and 2 evaluate the supervised single step affine registration model. In section 6.4.2, experiment 3 and 4 evaluate the dual supervised affine registration model. In section 6.4.3, experiment 5 assesses the novel dual supervised affine registration model. The experiments in this project are based on PyTorch on a single GTX 950M GPU.

6.3.1 Experiment 1 and 2

- **Experiment 1**

The experiment 1 is implemented on the supervised single step registration model. First, a dataset for the experiment is built before training. The dataset includes MNI-152 template, 3D brain MR images and ground truth affine transformation matrices which are processed in the section 6.1. In this step, we normalize MNI-152 template and 3D brain MR images to the range of 0 to 1 as the intensity of given images are not in the same range. In addition, the data types of above data are transformed to tensor float 32.

During the training, the model loads the MNI-152 template, a 3D brain image and its corresponding ground truth affine transformation matrix each time. Then the template and brain image are input to the AIRNet and the AIRNet outputs a predicted affine transformation matrix. The MSE_transformation is calculated and used for backpropagation algorithm to update parameters of the AIRNet. This procedure will iterate the number of epoch times for all images in the training set. Parameter settings of experiment 1 are specified in Table 1.

- **Experiment 2**

The model of experiment 2 is the same with experiment 1. The change in experiment 2 is we normalized three translation parameters of ground truth affine transformation matrix. Originally, the range of nine shear/scale/rotation parameters is -0.3 to 1.3 and the range of 3 translation parameters is -15 to 15. To normalize the three translation parameters to the same range of the other nine parameters, we normalize the translation parameters with a certain number which is called norm in this project. In this case, the predicted translation parameters are also in a normalized form in the training. Therefore, to obtain a correct transformation matrix, we multiply the norm back with the translation parameters in the testing. This method is also applied in the experiment 3,4 and 5. Parameter settings of experiment 2 are shown in Table 2.

Parameters	Value
Learning rate	0.0001
epoch	500
optimizer	Adam
batch size	5

Table 1: Parameter Settings of experiment 1

Parameters	Value
Learning rate	0.0001
epoch	500
optimizer	Adam
batch size	5
norm	32

Table 2: Parameter Settings of experiment 2

6.3.2 Experiment 3 and 4

- **Experiment 3**

The Experiment 3 is implemented on the dual supervised affine registration model. The dataset of this experiment is the same with the experiment 2. The network takes a brain image and the template as input and outputs a predicted affine transformation matrix each iteration. In experiment 3, the final loss function is Loss = MSE_transformation + NCC_similarity. First, we calculate the MSE_transformation between ground truth affine transformation matrix and the predicted matrix. Then, we warp the moving image based on the predicted matrix and calculate the NCC_similarity between the template and the warped image. A build-in affine transformation function in SciPy are used to warp the moving image. The sum of MSE_transformation and NCC_similarity is used for backpropagation algorithm to update parameters of AIRNet. Parameter settings of experiment 3 are specified in Table 3.

- **Experiment 4**

The model implemented on experiment 4 is the same with experiment 3. Compared with experiment 3, the experiment 4 uses ground truth warped image as template when calculating NCC_similarity of two images. Besides the MNI-template, brain images and ground truth transformation matrix, ground truth warped images generated by FSL are also normalized and added to the dataset before training. During the raining, the MSE_transformation is calculated in the same way with experiment 3. However, as we find that the image intensity range between the template and warped images is still different after normalization, we calculate the NCC_similarity between the predicted warped image and its corresponding ground truth warped image to minimize the influence. Parameter settings of experiment 4 are shown in Table 4.

Parameters	**Value**
Learning rate	0.0001
epoch	800
optimizer	Adam
batch size	5
norm	32

Table 3: Parameter Settings of experiment 3

Parameters	**Value**
Learning rate	0.0001
epoch	800
optimizer	Adam
batch size	5
norm	32

Table 4: Parameter Settings of experiment 4

6.3.3 Experiment 5

The experiment 5 is implemented on a novel dual supervised affine registration model we proposed. The first step of this experiment is to generate augmented moving images. For every moving image, a random transformation matrix T are generated for augmentation. In order to ensure the authenticity of the augmented images, we constraints the range of each transformation types in matrix T: The range of translation is -5 to 5 along each axis; scaling ranges from 0.9 to 1.1 for each axis; the range of rotation is -5 to 5 degree along each axis. We generate one augmented moving image for each moving image based on a random transformation matrix T. The dataset in this experiment includes the MNI-template, moving image, ground truth affine transformation matrix, augmented moving image and transformation matrix T.

In the training, we input the moving image and template into the AIRNet first and obtain a predicted affine transformation matrix Y. Then we input the augmented moving image and template into the AIRNet and obtain a predicted transformation matrix Z. There are two MSE in this experiment. MSE_transformation is the MSE between the ground truth affine transformation matrix and predicted matrix Y. As we explained in section 5.2.4, the dot product of matrix T and Z is equal to the matrix Y in mathematics. Because the output of AIRNet is twelve transformation parameters, we need to extend the matrix Z to its homogenous coordinate. After that we apply dot product to matrix Z and T and obtain a matrix A. MSE_consistency is the MSE between the twelve transformation parameters in matrix Y and matrix A. The final loss function in experiment 5 is Loss = MSE_transformation + MSE_consistency. The sum of two MSE are used for backpropagation to update the parameters of AIRNet. Parameter settings of experiment 5 are specified in Table 5.

Parameters	**Value**
Learning rate	0.0001
epoch	800
optimizer	Adam
batch size	5
norm	32

Table 5: Parameter Settings of experiment 5

7 Evaluation Results

This section will provide the performance of each experiment and compare the running time with a conventional image registration tool FSL. In addition, section 7.3 analysis the parameter settings we tried in the project.

7.1 Performance

This section evaluates performance of experiments by the difference between the template and registered images. The registered images are transformed by its corresponding affine transformation matrix predicted by AIRNet. The following two image similarity metrics are used through all experiments: MSE and NCC(see section 6.3.1). MSE measures the error of each pixel between two images. NCC measures the similarity of two images.

The image affine registration performance of each experiment is shown in Table 6, Table 7, Figure 15 and Figure 16. The MSE between moving images and the template is 0.0188 and the NCC between two images is 0.8343. The ground truth of this project is generated by FLIRT and its MSE with the template is 0.111 and its NCC with the template is 0.9550. The images registered by FLIRT are well aligned with the template.

The experiment 1 and 2 are conducted on the supervised single step model. The experiment 1 directly use the affine transformation matrix predicted by FLIRT as ground truth. And the experiment 2 normalizes three translation parameters of the matrix before training. In experiment 1, average MSE between registered images and the template is 0.0151 and average NCC is 0.8973. In experiment 2, average MSE is 0.0134 and average NCC is 0.9213. The result shows that the experiment 2 outperforms the experiment 1. Normalizing three translation parameters of the ground truth matrix improves the performance of the AIRNet. During the training, we find there is a trade-off between experiment 1 and 2. In experiment 1, The square errors in nine shear/scale/rotation parameters are larger than those error in the experiment 2. However, the situation of three translation parameters is reversed. It means the accuracy of nine shear/scale/rotation parameters has more impact than that of three translation parameters in affine registration.

The experiment 3 and 4 use the dual supervised registration model. Experiment 3 calculates the NCC_similarity between a predicted registered image and the template. Experiment 4 computes the NCC_similarity between a predicted registered image and the ground truth registered image generated by FLIRT. As shown in Table 6, the performance of experiment 3 does not improve compared with the result of experiment 2. The reason probably is the NCC_similarity calculated during the training is influenced by local differences between two images which require deformable transformation to register. In addition, different image intensity range of two images may also impact the performance. The performance of experiment 4 is better than experiment 2 and 3. Both MSE and NCC between two images outperform the results in experiment 2. As a valid metric NCC_similarity is added, the dual supervised image registration model predicts a more accurate affine transformation matrix than the supervised single step registration model.

Experiment 5 is conducted on a novel dual supervised registration proposed by us. The result of this experiment is worse than experiment 2 which means the MSE_consistency in experiment 2 has a negative impact on the AIRNet. In addition, the value of MSE_consistency fluctuates within a certain range during the training.

In conclusion, the performance of the dual supervised registration model is better than other models in this project. The performance of the model in experiment 4 is close to the performance of the conventional approach. An illustration of a registered images is shown in Figure 17(a). However, for some images shown in Figure 17(b), the network performs a failed registration. The reason might be there is a significant error on one or two predicted affine transformation parameters which leads to a mismatch between the warped image and the template.

Experiment	MSE	NCC
No registration	0.0188	0.8343
FLIRT	0.0111	0.9550
Experiment 1	0.0152	0.8973
Experiment 2	0.0134	0.9213
Experiment 3	0.0138	0.9240
Experiment 4	0.0125	0.9438
Experiment 5	0.0162	0.9003

Table 6: Performance of FLIRT and each experiment

Experiment	Cross Validation	MSE	std	NCC	std
No registration	-	0.0188	0.0047	0.8343	0.0407
FLIRT	-	0.0111	0.0036	0.9550	0.0071
Experiment 1	1	0.0152	0.0052	0.8938	0.0457
	2	0.0147	0.0053	0.9001	0.0377
	3	0.0158	0.0059	0.8981	0.0475
Experiment 2	1	0.0135	0.0050	0.9230	0.0347
	2	0.0134	0.0051	0.9217	0.0280
	3	0.0134	0.0053	0.9192	0.0424
Experiment 3	1	0.0137	0.0040	0.9227	0.0285
	2	0.0139	0.0053	0.9237	0.0285
	3	0.0136	0.0058	0.9255	0.0366
Experiment 4	1	0.0126	0.0044	0.9433	0.0269
	2	0.0123	0.0041	0.9445	0.0289
	3	0.0127	0.0051	0.9436	0.0416
Experiment 5	1	0.0171	0.0053	0.8953	0.0470
	2	0.0163	0.0051	0.9012	0.0469
	3	0.0153	0.0055	0.9045	0.0410

Table 7: Cross validation result of each experiment

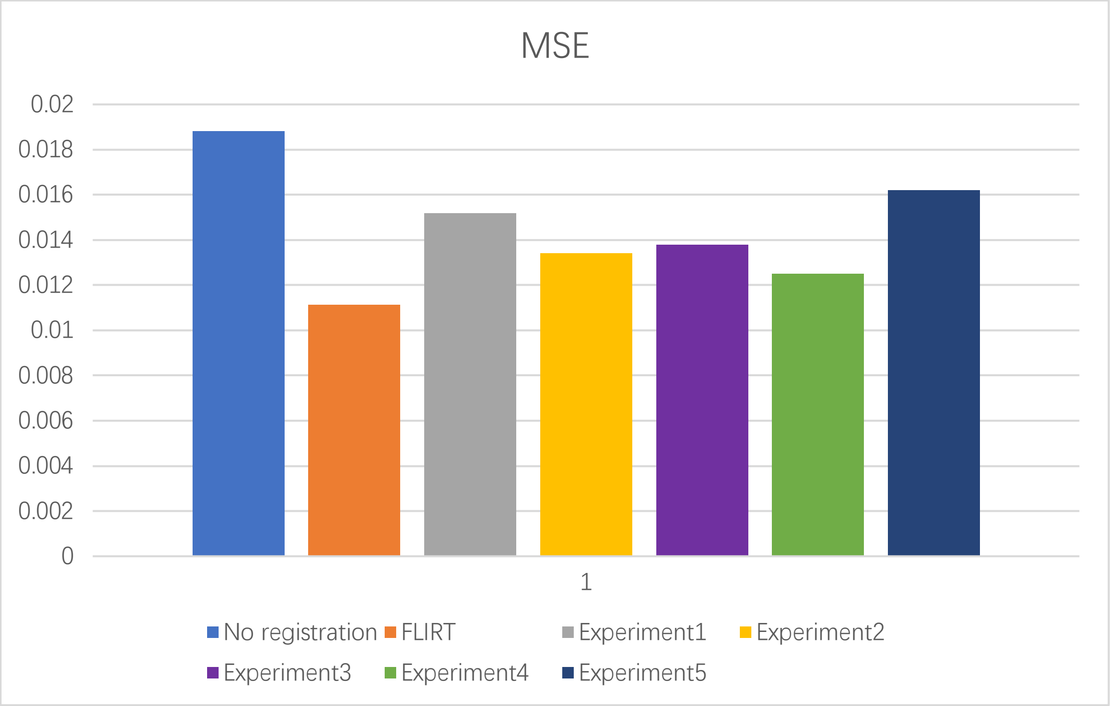


Figure 15: MSE of experiments(Lower is better)

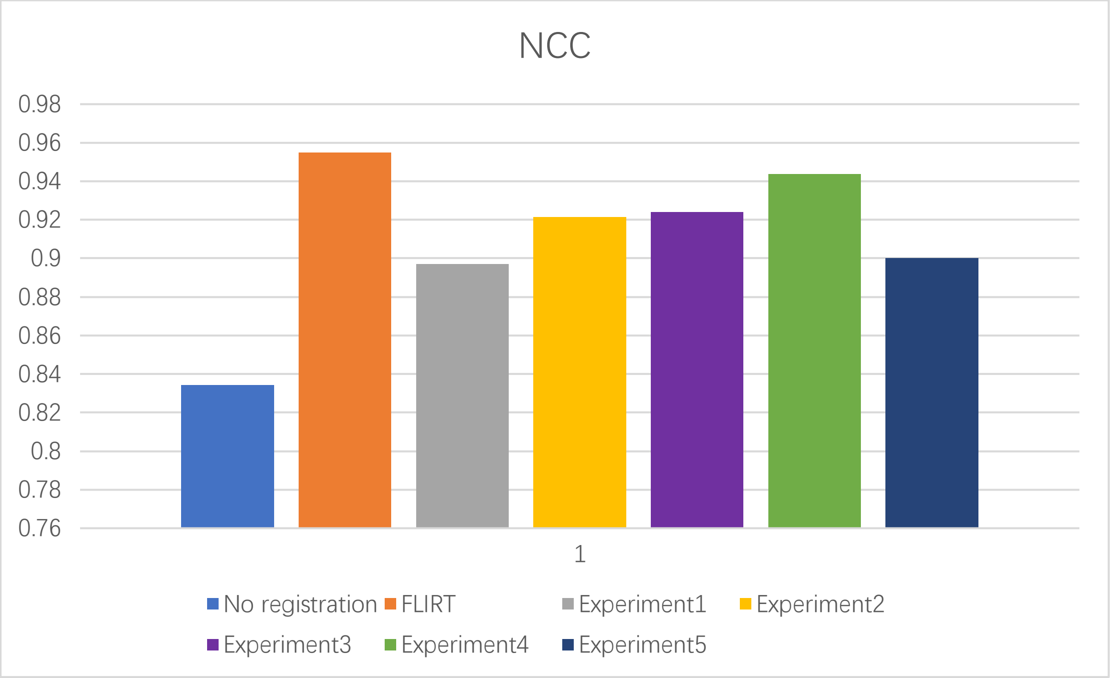


Figure 16: NCC of each experiment(Higher is better)

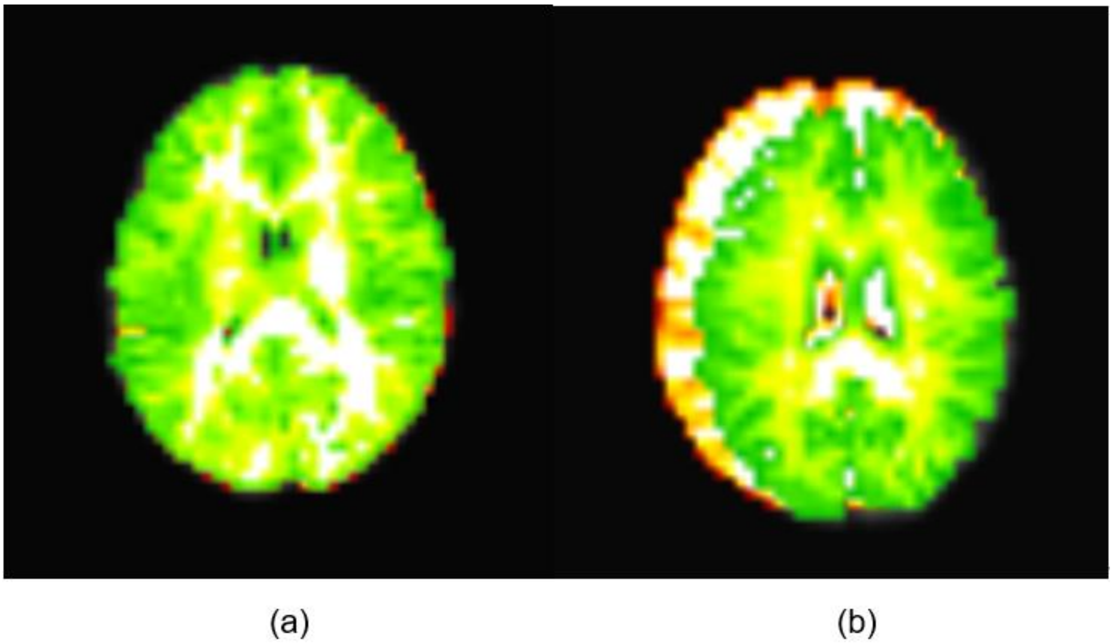


Figure 17: Illustration of a registered image and the template. The orange brain is the MNI152 template. The green brain is the registered image. (a) registered image aligns with the template. (b) An example of failed registration.

7.2 Running Time

This section evaluates the running time of conventional image registration and deep learning based registration. As shown in table 8, the running time of affine image registration using FLIRT is 5.77±0.53 seconds per image. The speed of AIRNet is 0.48±0.05 seconds per image. Deeping learning based affine image registration is 10 times faster the conventional affine image registration.

Method	Running Time(s)
FLIRT	5.77±0.53
AIRNet	0.48±0.05

Table 8: Running time of FLIRT and AIRNet per image

8 Summary and Reflections

8.1 Project Management

This project was a huge challenge to me as I have not learned any methodology used in this project at the beginning. To meet objectives of the project, I have done lots of work and effectively managed time and resources in this academic year. Although we made changes on objectives in the course of the project because of some factors, the project was still a success as several experiments are designed and implemented to improve the performance of affine image registration.

8.1.1 Time Management

This project is deep learning based image registration. It was a new topic to me. Therefore, I spent 2 months to do literature review and learn methodologies. During this period, I figured out the theory of image registration and deep neural network. Meanwhile, I also taught myself Python to build deep learning models. I preprocessed image data and generate ground truth in this time.

In December and January, I start to build network for affine transformation. Due to the influence of the mid-term exam and other factors, the project progressed slowly during this period. At the beginning of February, I realized that the project had fallen behind. I spent 6 hours on the project per day, and finished experiment 1 and 2 in several week. However, the performance of experiment 1 and 2 is does not meet expectations. In this case, instead of going ahead for deformable registration, my supervisor suggested me to change the objectives and investigate more methods to improve the performance of affine image registration.

In March, I designed experiment 3 and 4 to evaluate the performance of dual supervised learning. In addition, a novel affine image registration model was proposed. Because experiment 3, 4 and 5 resample images during the training, training time for each experiment soared to several hours. This is unexpected in my time management. The project was going well until the Covid-19 outbroke in the UK. As the university were closed, I had to work in my accommodation. The efficiency of studying in the accommodation was very low. I cost some time to adopt the life in lockdown and maintain good mental state to complete the project. Also, weekly face to face meeting were changed to online meeting which caused some misunderstanding as I cannot show my project to the supervisor. To overcome those problem, I read numerous papers to learn different methods proposed for image registration. I learned many practical ideas from papers.

Overall, I am not satisfied with my time management of this project. As I have no experience on deep learning before, many plans exceeded the deadline set for the project. In the future, extra time should be considered when managing time in case of delay caused by the failure of the experiment.

8.1.2 Resource Management

In this project, there are 335 3D brain MR images and an MNI-152 template as raw data. In addition, data preprocessing will generate a large number of images. Therefore, the naming conventions must be strictly followed to avoid contaminating data because of label confusion.

Besides brain images, ground truth generated by FSL also need to be carefully categorized. As FSL runs on Linux, we mounted the image data folder under Linux system. In this way, both Windows and Linux systems can share imaging data in the folder.

In addition, after training a model, the model should be stored with proper naming convention to avoid overlapping previous models.

8.2 Contributions and Reflections

8.2.1 Contributions

In this project, we implement five experiments on three different models. Some methods are proved to improve the performance of affine image registration.

On the supervised single step image registration model, we use ground truth affine transformation matrix to train the model first. Then we normalize three translation parameters of the ground truth in the experiment 2. We prove that using a normalized ground truth in the training can reduce the impact of translation when calculating MSE between a predicted transformation matrix and the ground truth. In this way, the ability of predicting nine shear/scale/rotation parameters are improved. There are a number of papers used this model, but no one explicitly pointed out normalizing translation parameters before training can improve the performance of the model.

Another achievement is we improve the performance of affine transformation by implementing a dual supervised model. Some papers implemented this model using the MNI-152 template as the template of NCC as the method in experiment 3. However, we find that using ground truth registered image as the template of NCC outperform the former method. We believe the method in experiment 4 is more accurate because The NCC between two images are not influenced by local difference in the MNI152 template.

In addition, we propose a novel dual supervised registration method based on mathematical theories. Although this method fails to improve the performance of affine registration, it is still an interesting topic to investigate why there are differences between matrices that should theoretically be the same.

8.2.2 Project Appraisal

Review the project throughout the year, I have learned a lot from it, both in terms of knowledge and project management. However, there are many difficulties in the course of the project.

First, literature review was not done well. I underestimated the complexity of image registration, which is a technology widely used in different fields. When I did literature review at the beginning, I was interfered by many image registrations examples in other fields, which led to my misunderstanding of brain image registration. In the initial literature survey, I did not make full effort on it. As a result, I could not understand the supervisor’s explanation in the weekly meetings. I processed the data and wrote the code before I understood the concepts and methods of brain image registration, which caused me many troubles.

Second, the limitation of image size may reduce the accuracy of image registration. Originally, to ensure the input images have the same image size, the given images are padded to 256*256*256. However, a 256*256*256 image is too large to train in my GPU. I have to resize the images from 256*256*256 to 64*64*64. The resolution of 64*64*64 images is much lower than original images. In this case, when I train the network with 64*64*64 images, the network is probably unable to extract the details of images. In the future, we can implement the experiments on Google Cloud with original images and compare the result of them.

Third, in the last three experiment, the training time increased from fifteen minutes to five hours. This problem has seriously delayed the progress of my project. The reason for the increase in training time is low GPU utilization. In the last three experiments, I need to process images using NumPy on CPU during the training. In this case, the thread frequently switches between CPU and GPU. I tried different method to solve this problem but did not succeed.

Although various difficulties are encountered throughout the project, I achieved the objectives I set for this project. Multiple approaches were used to improve the performance of deep neural network as well as a novel hybrid method was proposed and evaluated. The affine registration process speeds up 10 times compared with a conventional method.

Bibliography

[1] Alfaro-Almagro, F., Jenkinson, M., Bangerter, N. K., Andersson, J. L., Griffanti, L., Douaud, G., ... & Vidaurre, D. (2018). Image processing and Quality Control for the first 10,000 brain imaging datasets from UK Biobank. *Neuroimage*, *166*, 400-424.

[2] Mandal, P. K., Mahajan, R., & Dinov, I. D. (2012). Structural brain atlases: design, rationale, and applications in normal and pathological cohorts. Journal of Alzheimer's disease : JAD, 31 Suppl 3(0 3), S169–S188. doi:10.3233/JAD-2012-120412

[3] Barbara Zitová and Jan Flusser, (2003). Image registration methods: a survey. Image and Vision Computing 21(11), pp.977-1000.

[4] Haskins, G., Kruger, U., & Yan, P. (2019). Deep Learning in Medical Image Registration: A Survey. arXiv preprint arXiv:1903.02026.

[5] Woolrich, M. W., Jbabdi, S., Patenaude, B., Chappell, M., Makni, S., Behrens, T., ... & Smith, S. M. (2009). Bayesian analysis of neuroimaging data in FSL. Neuroimage, 45(1), S173-S186.

[6] Lowekamp, B. C., Chen, D. T., Ibáñez, L., & Blezek, D. (2013). The design of SimpleITK. Frontiers in neuroinformatics, 7, 45.

[7] Chee, E., & Wu, J. (2018). Airnet: Self-supervised affine registration for 3d medical images using neural networks. *arXiv preprint arXiv:1810.02583*.

[8] Geert Litjens and Thijs Kooi el al., (2017). A survey on deep learning in medical image analysis. Medical Image Analysis 42, pp.60-88.

[9] M. Jenkinson and S.M. Smith. (2001). A global optimisation method for robust affine registration of brain images. Medical Image Analysis, 5(2):143-156, 2001.

[10] Andersson, J.L., Jenkinson, M., Smith, S., (2007). Non-linear registration, aka Spatial normalization FMRIB technical report TR07JA2. FMRIB Analysis Group of the University of Oxford 2.

[11] Wu, G., Kim, M., Wang, Q., Munsell, B. C., & Shen, D. (2015). Scalable high-performance image registration framework by unsupervised deep feature representations learning. IEEE Transactions on Biomedical Engineering, 63(7), 1505-1516.

[12] Rohé, M. M., Datar, M., Heimann, T., Sermesant, M., & Pennec, X. (2017, September). SVF-Net: Learning deformable image registration using shape matching. In International Conference on Medical Image Computing and Computer-Assisted Intervention (pp. 266-274). Springer, Cham.

[13] Cao, X., Yang, J., Zhang, J., Nie, D., Kim, M. J., Wang, Q., & Shen, D. (2017). Deformable Image Registration based on Similarity-Steered CNN Regression. *Medical image computing and computer-assisted intervention : MICCAI ... International Conference on Medical Image Computing and Computer-Assisted Intervention*, *10433*, 300–308. doi:10.1007/978-3-319-66182-7_35

[14] Fan, J., Cao, X., Yap, P. T., & Shen, D. (2019). BIRNet: Brain image registration using dual-supervised fully convolutional networks. *Medical image analysis*, *54*, 193-206.

[15] Uzunova, H., Wilms, M., Handels, H., & Ehrhardt, J. (2017, September). Training CNNs for image registration from few samples with model-based data augmentation. In International Conference on Medical Image Computing and Computer-Assisted Intervention (pp. 223-231). Springer, Cham.

[16] 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.

[17] Yoo, I., Hildebrand, D. G., Tobin, W. F., Lee, W. C. A., & Jeong, W. K. (2017). ssemnet: Serial-section electron microscopy image registration using a spatial transformer network with learned features. In Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support (pp. 249-257). Springer, Cham.

[18] Goodfellow, I., Bengio, Y. and Courville, A. (2016). Deep learning. MIT Press, p.327.

[19] Sotiras, A., Davatzikos, C., & Paragios, N. (2013). Deformable medical image registration: A survey. IEEE transactions on medical imaging, 32(7), 1153-1190.

[20] Lehmann, T. M., Gonner, C., & Spitzer, K. (1999). Survey: Interpolation methods in medical image processing. IEEE transactions on medical imaging, 18(11), 1049-1075.

[21] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. The journal of machine learning research, 15(1), 1929-1958.