Otto von Guericke University Magdeburg

Faculty of Computer Science
The Institute of Technical and Business Information Systems
Data and Knowledge Engineering Group

Research Project

Image registration using Deep Learning

Authors:
Himanshi Bajaj
Mohammad Istiyak Hossain Siddique
Nandish Bandi Subbarayappa
Steve Simon
Suraj Shashidhar

December 13, 2021

Examiner:
Prof. Dr.-Ing. Andreas Nürnberger
Data and Knowledge Engineering Group

Supervisor:
M.Sc. Soumick Chatterjee
Data and Knowledge Engineering Group, and
Department of Biomedical Magnetic Resonance

Himanshi Bajaj , Mohammad Istiyak Hossain Siddique, Nandish Bandi Subbarayappa, Steve Simon, Suraj Shashidhar:
Image registration using Deep Learning
Research Project, Otto von Guericke University
Magdeburg, 2021.

Contents

Abstract

englisch summary of this work

This section is optional! It is basically an motivational cite for this work as it can be found in many books. Example is provided

The validation of clustering structures is

the most difficult and frustrating part of cluster analysis.

Without a strong effort in this direction,
cluster analysis will remain a black art accessible only to those
true believers who have experience and great courage.

Anil K. Jain and Richard C. Dubes

Acknowledgements

This section is optional!

1

Introduction and Motivation

BOVEIRI et al. (2020) defined image registration to be the process whereby we align two given images according to an identical geometrical coordination. This process has its roots in satellite imagery, computer vision, military, etc. fields. One of these fields is medical imaging. Here, image registration refers to the same process of aligning two or more images DE VOS et al. (2018). This has become a cornerstone for many other downstream tasks in medical science and has been researched on for more than a decade now BALAKRISHNAN et al. (2019). There are multiple ways we can perform this registration. Traditional approaches focus on directly optimizing pairs of images based on some criteria or function. More modern approaches include machine learning, deep learning based solutions that try to understand the problem as a whole, rather than focusing on individual pairs of images. Such approaches have far reaching consequences, e.g. they can be very fast and they can have the capability of accommodating a wide array of deformations. Furthermore, some of those solutions do not even require manually delineated training data, meaning that some of those algorithms are unsupervised in nature thus reducing the human effort and possibility of human errors in data.

As mentioned above, traditional approaches to solve medical image registration involves solving some optimization function for each of the image pair. This optimization function is usually made of a similarity metric that allows the non-linear mapping of apparently similarly voxel, while maintaining local smoothness, etc. constraints ZHANG (2018). This is commonly known as non-learning based solution. Because such solutions require optimizing the same criteria from scratch, it requires an extensive amount of time and computational effort to solve such an optimization. To find a middle ground, supervised methods are used. Here, the optimization

does not have to take place for every pair of images, every time. Rather we can utilize the learnt parameters to transform a newly obtained moving image. But such methods require huge amount of manually delineated data which is, firstly difficult to obtain and secondly, and most importantly, more vulnerable to human error.

However, deep learning based solutions try to map the moving images to fixed images by understanding the displacement vector field that caused the change in the two images. We feed both the images to the network and let the network learn on its own what are the parameters to best transform the image from one to another. We have observed plenty of deep learning based solutions. Most of these solutions based their methodology on unsupervised approach BALAKRISHNAN et al. (2019); DE VOS et al. (2018); KIM et al. (2020). In it, they have proposed a variety of approaches to capture the image registration problem, to regularize the generated images, etc. Some of these approaches are based on Generative Adversarial Networks (GAN) LIN et al. (2018); MAHAPATRA (2019). But, owing to the fact that GANs are prone to generate unrealistic imagery and the difficulty around explaining them, led us to not pursue that direction. However, we have also observed some intuitive yet efficient ways of regularizing these networks for producing realistic images, such as cycle consistency KIM et al. (2020); ZHANG (2018), bio-mechanics informed regularizer QIN et al. (2020), etc.

1.1 Motivation

We are motivated to present a solution that is capable of registering images for both inter modal and intra modal cases. As we mentioned in the introduction, we have observed a good number of literature in deep unsupervised approach. From our observations, we found that, most of these approaches either avoided considering or failed to incorporate information about structural connectivity inside brain. As our motivation is to deal with both modalities, we consider it is better to integrate these structural information in our solution. Secondly, we have observed deformations of different sizes in the images of our dataset. This led us to consider a solution which is capable of handling images of different scaling i.e. resolution. Our reasoning for this decision is: if we have a bigger deformation, it can be handled with lower resolution images. In such a case, it is a waste to deal

with images of high resolution. On the other hand, tiny fluctuation or miniscule deformations are hard to see in a lower resolution image, whereas a higher resolution image is much more capable in handling this. We wanted to utilize this concept by developing a pipeline capable of handling multi-scale image. Thirdly, we have observed that there are issues in regularizing the neural network that is being used to learn the deformation field. We underscore the significance of these regularizers in producing a more realistic image, so we decided to utilize the concept of cycle-consistency KIM et al. (2020); ZHANG (2018) whereas a moving image is moved with a regularization that allows the generation of the same image. This cyclic regularizer ensures that no unrealistic image, which cannot be cycled-back to its original, is not produced in the process.

In addition to these approaches, we have also experimented with a non deep learning based. In this approach, we are solely using gradient descent to direcly optimize a pair of images. To our knowledge, such an approach has never been tested. So, we tested its possibility and performance at the same time.

1.2 Aim of this project

As can be evident from the preceding sections, we are researching for a number of items in this project. If we pose these questions as research questions, then we come up with the following:

- RQ 1. How do the state-of-the-art models perform in comparison to each other?
- RQ 2. How well does our method perform in inter-modal and intra-modal scenario?
- RQ 3. What are the contributions of each of the three proposals in the overall performance for either modality?
- RQ 4. Whether direct optimization based solution, solely based on gradient descent, can work to register images? If so, how well do they perform in either modality?

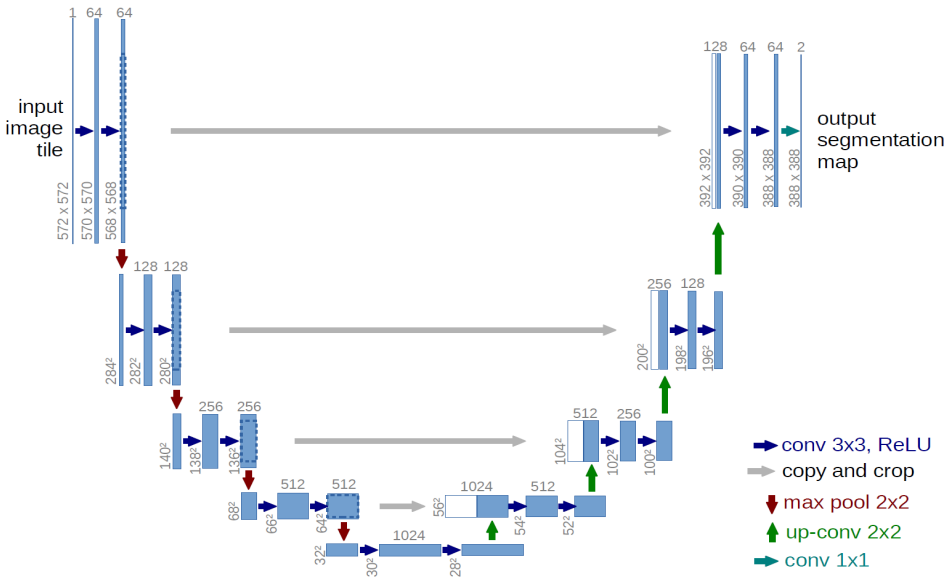
1.3 Structure of this report

This report has been segmented in multiple sections to properly explain the all of our successful and failed experiments. In section 2, we are explaining some of the backgrounds that are the foundation of our work. Following this, we have an extensive literature survey in section 3, which is followed by our methodologies at section 4 and 5. This section will be followed by Experiments and Evaluation section at section 6 where we have detailed our experiments and the evaluation process. And finally, in section 7, we draw our conclusion for this report.

2 Background

2.1 UNet

Semantic Segmentation is a task of assigning each pixel to its class that represents an object.UNet tries to tackle this problem through localization and usage of context. The network consists of a contracting path that generates an image pyramid of features using pooling and convolution operators, whilst the expanding path increases the resolution of the image through upsampling. It uses the contracting paths output to localize the feature and uses multiple channels combined with convolution to learn about semantics RONNEBERGER et al. (2015).



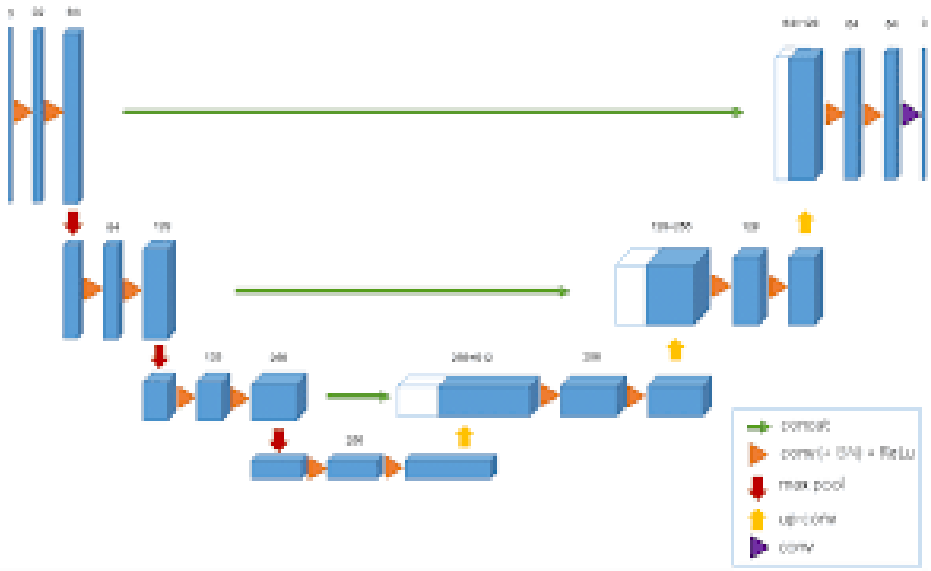
(a)

Figure 2.1: UNet Architecture

Fig 2.1 RONNEBERGER et al. (2015) describes the architecture of UNet. The contracting path consists of back to back convolution blocks where each block performs 3×3 convolutions with unpadded convolutions and applies ReLU activation. The features are downsampled using 2×2 maxpooling operation. In the expanding path, the features are first upsampled through 2×2 transposed convolution which halves the channels. This is followed by concatenation of corresponding feature from contracting path and subsequent application of unpadded 3×3 convolution with ReLU.

2.2 3-D UNet

Vanilla UNet RONNEBERGER et al. (2015) assigned label on each pixel position of a 2-D input image to achieve semantic segmentation. 3-D UNet ÇIÇEK et al. (2016) is similar to 2-D UNet but applied similar principles on volumes. Unlike vanilla UNet, 3-D UNet directly learns on volumes using $3-D$ convolutions. The network was used to segment sparsely annotated data and later trained to perform this task automatically without annotations.



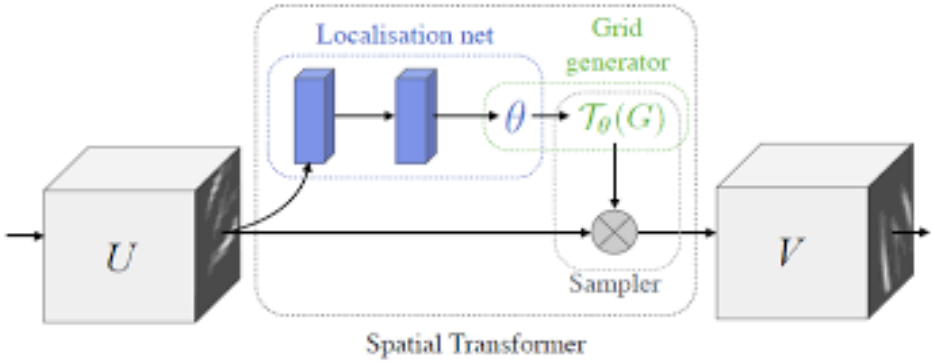
(a)

Figure 2.2: 3D UNet Architecture

Fig 2.2 ÇIÇEK et al. (2016) describes the architecture of 3D UNet. The contracting path consists of convolution blocks where each block performs $3 \times 3 \times 3$ convolution followed by Batch Normalization and ReLU. Batch Normalization helps stabilize training and faster convergence. The contracting path uses $2 \times 2 \times 2$ strided max pooling to reduce spatial dimensions. The expansive path consists of upsampling $2 \times 2 \times 2$ transpose convolution to increase spatial dimensions of the features. This is combined with the corresponding features from contracting path to improve localization. The semantics are further embedded using the $3 \times 3 \times 3$ same padded convolution block with batch normalization and ReLU.

2.3 Spatial Transformers

CNNs can successfully accomplish tasks on spatially equivariant data and are limited in their abilities to be invariant to the input data. Spatial transformers JADERBERG et al. (2016) help bridge this gap by learning the warping parameters that could make the network invariant whilst being trained along with the same CNN network without extra supervision.



(a)

Figure 2.3: Spatial Transformers Architecture

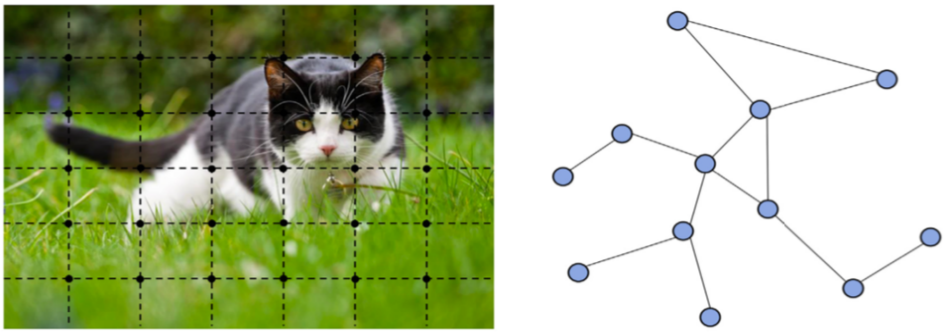
Fig 2.3 JADERBERG et al. (2016) describes the architecture of spatial transformers. The localization network consists of learnable parameters that learn the warping parameters such as affine transformation matrix or deformation field given the input feature maps. The grid generator accepts the parameters from localization network and transforms the input. The transformation is conditional to the input and actual pairs. This means the

spatial transformer localization network needs to learn to adjust the affine or deformable parameters necessary to get closer to actual image.

2.4 Graph Neural Networks

Graph Neural Networks (GNN) BRONSTEIN et al. (2017) are a type of neural network that operate on graph like data structures. A graph typically consists of set of nodes and edges connecting between the nodes. Lot of real world problems could be modelled in terms of graphs such as recommender systems, social media user preference etc.

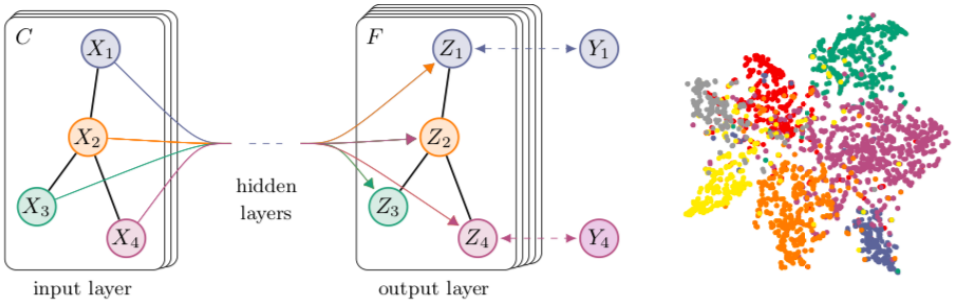
Images could be crudely modelled as graphs where each pixel is represented as a node, each node is connected with its 8 neighbours. Key points extracted from an image could server as nodes and their connections are edges. Text could be modelled as a directed graph where words are nodes and connecting with its next word in the sentence. Molecules can be modelled as a directed graph of different elements with each bond as an edge having specific weightage. Furthermore, Graphs can be used to model problems that occur in non-euclidean domain as shown in Fig 2.4 ZHOU et al. (2020).



(a)

Figure 2.4: Euclidean vs Non-Euclidean Representation

GNNs have a strong inductive bias BATTAGLIA et al. (2018) due to its increased receptive field originating from edges. Moreover, In GNNs the connections between neighbouring nodes are optional, hence its easier to identify semantic structures as shown in Fig 2.5 KIPF und WELLING (2017). GNNs could be used in the following ways



(a)

Figure 2.5: Graph Neural Network Representation

- Graph level task: Predicting the property of the entire graph, Example: predicting the molecule, object detection of an image or sentiment of a sentence.
- Node level task: Predicting the properties of a single node, Example: Semantic segmentation of image where each pixel node class must be predicted, POS of a word in the sentence.
- Edge level task: Predicting the properties of the edge, Example: Supervised link prediction, Next Action prediction.
- Latent space encoding task: The GNN output is not directly used to solve a task rather it is used to encode certain information and to be used by downstream layers.

Our model uses GNN as Latent space encoding to capture semantic information implicitly which would be used to improve the performance in the registration.

2.5 SCG Network

2.5.1 Introduction

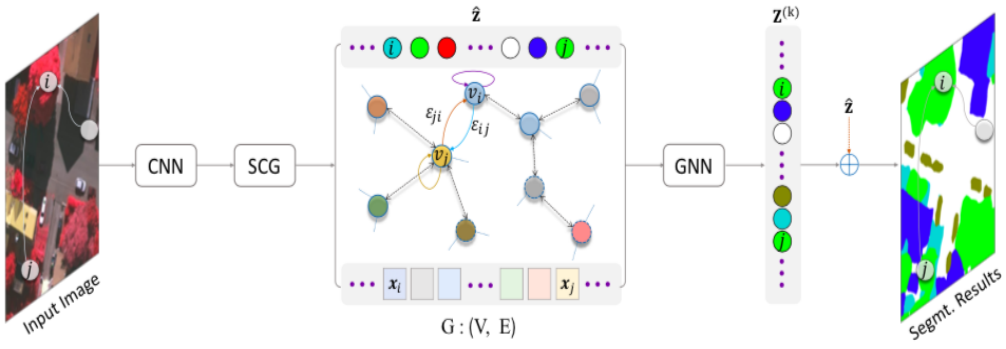
Modelling techniques like CNN, Transformers, Clustering are used to achieve semantic segmentation . This approach requires deep models to improve the receptive field and learn long range dependencies. GNNs

could help increase the receptive field and better learn about long range dependencies between pixels.

Transforming an image into a full graph could be difficult due to number of nodes and edges. The large number of nodes and edges could cause computational issues and would not be a better representation. It would be easier to convert features from an encoder into a graph to reduce the computation and increase receptive field at the same time.

SCGNet LIU et al. (2020) helps to bridge this gap wherein it can self construct a graph from a 2D feature map and map it into vertices and edges in latent space. This is inspired from Variational Auto Encoders VAE KINGMA und WELLING (2019) that finds a mean and variance of the data distribution. Since SCGNet learns from encoders, it can find patches that are similar in latent space but are far away spatially.

The proposed SCG module is extended for the end-to-end semantic segmentation.CNN encoder and decoder is used in the SCGNet framework to extract high-level feature maps, which are then used to build the contextual graph using the SCG. The SCG module generates a global contextual graph, which is then used to train a k-layer GNN to not only learn the latent embedding but also predict the final node-wise labels. Finally, the predicted node labels are projected back onto the original 2D plane. Fig 2.6 LIU et al. (2020) depicts a high-level overview of our method. The network easily switches back and forth between euclidean and graph domains to decrease computation and get best of both worlds.



(a)

Figure 2.6: SCG Network End to End Model

2.5.2 Model

Encoder module uses adaptive pooling to reduce the dimensionality of the feature maps and transform it into a one dimensional latent space of $n = \overline{(h)} * \overline{(w)} * d$ nodes where $\overline{h}$ and $\overline{w}$ are spatial dimensions of the encoder output. This is further transformed into a distribution represented by mean $\mathcal{M}$ and standard deviation Σ , using a VAE.

$$Mean(\mathcal{M}) = Flatten(Conv_{3\times3}(\overline{FeatureMaps})) \tag{2.1}$$

$$StandardDeviation(\Sigma) = Log(Conv_{1\times1}(\overline{FeatureMaps})) \tag{2.2}$$

The mean and standard deviation are reparameterized using a standard gaussian noise γ~$\mathcal{N}(0,1)$. This helps to centre the distribution during training as shown in equation 2.3 LIU et al. (2020).

$$\mathcal{Z} = \mathcal{M} + \Sigma \cdot \gamma \tag{2.3}$$

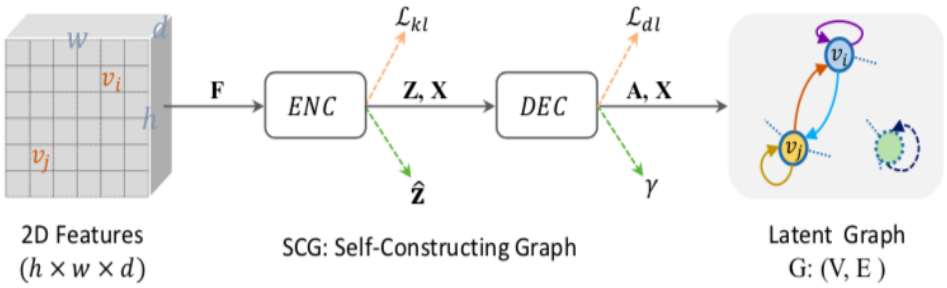
The Kullback-Leibler divergence loss MELBOURNE et al. (2010) is used to minimize the loss along with the residual loss $\hat{Z} = \mathcal{M} \cdot (1 - log\Sigma)$. Logarithm is applied on standard deviation to make the distribution monotonic and stabilize the training regimen as shown in equation 2.4

$$\mathcal{L}_{kl} = \frac{-1}{2nc} \sum_{i=1}^{n} \sum_{j=1}^{c} (1 + log(\Sigma_{ij})^2 \mathcal{M}^2 (\Sigma_{ij})^2) \tag{2.4}$$

Encoder produces node information $\mathcal{Z}$ but not edge details. This is produced by decoder by generating an Adjacency matrix from nodes as shown in equation 2.5. This produces a weighted and undirected graph where similar nodes will have values close to one and dissimilar nodes will have values close to zeros

$$A = ReLU(\mathcal{Z} \cdot \mathcal{Z}^T) \tag{2.5}$$

The node and edge information are further passed to a GNN network which uses $ReLU(Conv_{3\times3})$ to produce a graph of size $\overline{h} * \overline{w} * d$ nodes. The overall model is illustrated by Fig 2.7 LIU et al. (2020).



(a)

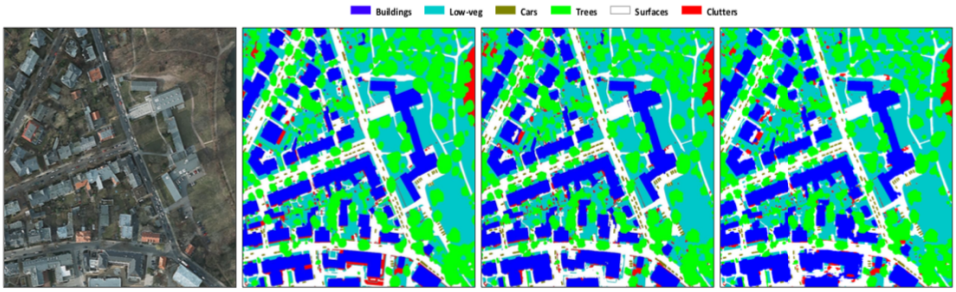
Figure 2.7: SCG Network Core Model

2.5.3 Training Procedure

Since SCGNet was used for segmentation Dice score $\mathcal{L}_{dice}$ DICE (1945) was used as a main loss function along with $\mathcal{L}_{kl}$ and $\mathcal{L}_{dl}$ as regularization. The total loss function was the sum of three losses as shown in equation 2.6

$$\mathcal{L} = \mathcal{L}_{dice} + \mathcal{L}_{dl} + \mathcal{L}_{kl} \tag{2.6}$$

2.5.4 Evaluation



(a)

Figure 2.8: SCG Output Example

The training and evaluation was performed on benchmark datasets like *Postdam* and *Vaihingen* GERKE et al. (2014). Models were trained using Adam optimizer with weight decay 2×10^{-5} and learning rate decay factor of 0.85 every 15 epochs. The SCGNet model is evaluated on the same benchmark dataset. An example output of an original image, ground truth, competitor DDCM-R50 LIU et al. (2019) and SCGNet is shown in the below

Fig 2.8 LIU et al. (2020). SCGNet has competitive F1 score of 0.920 against DDCM 0.928 even though it has significantly less number of trainable parameters than DDCM-R50.

3
Related Work

We begin our discussion by discussing the paper by DE VOS et al. (2018). Authors of this literature, described, pretty much comprehensively, the ideas behind medical image registration system and the ways to evaluate this. Unlike their predecessors, DE VOS et al. (2018) focused on unsupervised learning whereby registration is done by optimizing a loss function. A ConvNet was used here to derive a function like parameter set, which was later used to register the moving image in one-shot.

They started their literature by showing arguments in favour of using deep learning (DL) techniques for medical image registration. They argued about the capacity of DL techniques for automatically extracting the relevant features. They also argued about the qualities of convolutional architecture and deep learning in general like parallelizability and robustness. Following this discussion, they emphasized the necessity of going unsupervised in the field of medical image registration. They argued the drawbacks of supervised methods by mentioning the necessity of manual annotations or artificial synthesization of training examples, which can be difficult to obtain and time consuming to produce. On the other hand, unsupervised techniques do not suffer from these issues.

Bearing these issues in mind, de Vos et al. proposed to calculate a dense displacement vector field (DVF) between the moving and the fixed image. In order to achieve this DVF, they, instead of directly optimizing a set of parameters, used CNNs weights as parameters of the DVF and optimize those to calculate the transformation parameters. Thus, the task of the CNN becomes to detect and utilize the similarity in the fixed and the moving image to understand the parameters of the registration.

VoxelMorph, as proposed by BALAKRISHNAN et al. (2019), is another significant literature that many of the later implementation considered to be a baseline for comparing their implementations. One of the main differences between VoxelMorph and DE VOS et al. (2018) is the way they are inputting the image into the architecture. While de Vos et al. takes two separate inputs, one for moving and another for fixed, and process both of them parallelly, VoxelMorph concatenates them and process them together.

However, authors of VoxelMorph begins their discussion by identifying the shortcomings of traditional registration methods that considers pairwise optimization of moving and fixed images through the alignment of the voxels with similar appearances. But these require higher computational power which makes them difficult to use for real-time test cases. To alleviate this issue, VoxelMorph proposes two alternatives: the first one deals with input volumes and registration field calculated by the CNN based architecture and the second one utilizes anatomical segmentations. The authors focused on 3D volumes of single channel (grayscale). They designed a CNN based network that is used to parameterize the displacement field u between the fixed and the moving image.

The next literature that inspired us is from TANG et al. (2020). They proposed an end-to-end image registration method with three components, namely the affine registration module, the deformable registration module, and the spatial transformer module.

They initiated their discussion by pointing out the drawbacks of traditional optimization based registration processes: they emphasized on the speed of these methods and their inability to handle complex structures. Following this discussion, they stated some of the limitations of the supervised algorithms, e.g the need for labelled ground-truths, etc. These arguments made the ground for the necessity of unsupervised approach to tackle medical image registartion. They acknowledged the contributions of JADERBERG et al. (2016), DE VOS et al. (2018), and BALAKRISHNAN et al. (2019) and drew inspiration from these. However, they mentioned about the necessity of pre-aligned data for training these unsupervised approaches. To mitigate these challenges, TANG et al. (2020) proposed their system as an end-to-end solution for registering an image.

In order to realize such a solution, ADMIR learns, by employing a ConvNet, 12 parameters related to affine transformation. For the deformable transformation, they introduced a ConvNet that is capable of learning the displacement vector field (DVF). They have used Spatial Transformation modules after each of these transformation parameters are learned. They have used a compound loss function which is composed of three individual loss entities: the first loss is calculated between the affinely transformed image and the fixed image, the second one is calculated between the fully warped image and the fixed image, and the last one is calculated as the smoothness loss. One notable distinction between DE VOS et al. (2018) and ADMIR is the concatenation of fixed and moving images, instead of feeding them to separate channel. The same approach was followed by BALAKRISHNAN et al. (2019).

On the other hand, CycleMorph, due to KIM et al. (2020), provided an intuitive solutions to image registration problem. They argued that the existing deep learning based solutions, in most cases, are unable to preserve the topology of the image, resulting in an unwanted degeneracy problem. To mitigate this issue, they used cycle consistency to regularize the diffeomorphism generated by the model. In order to realize their effort, they are generating two diffeomorphic deformation vector field, the first of these takes an image and generates a deformed version of it, this version is then gets propagated to the next network, whereby another network then do the inverse and generates the original image out of this deformed one. Both of the network is updated and optimized using the same loss function. Thus, cycle consistency is used as a constraint that regularizes the deformation vector field, so that it doesn't generate unrealistic image folds which cannot be reverted back to the original image. Furthermore, this paper also proposes a multiscale image registration technique that uses global and local registration methods to optimize the usage of GPU while processing massive dataset.

Loss function that was used here in this paper, has three components. First of them is the registration loss, in which local cross-correlation was used as similarity function for it's less sensitive nature to the contrast variation. In addition to this, an l2 regularizer was also used to make the deformation field smooth. The second part is the cycle loss, in which l1 norm was used to ensure cycle consistency. Apart from these two, a novel loss function

was introduced here that is the identity loss. The rationale for this is: stationary regions of images should not be changed. Identity loss ensures this property. For evaluation, Jacobian matrix was produced to evaluate the diffeomorphic performance of the deformation. Furthermore, Dice coeffiecient and target registration error was used. On the flip side, this approach requires manually delineated anatomical structures and manually indicated landmarks for evaluation of the registration performance, which introduces the risk of human error.

A similar idea was used by ZHANG (2018). He utilized inverse consitency and prepared a pipeline in unsupervised fashion for deformable image registration. In this approach, he took a pair of images and ensured that those are symmetrically deformed toward each other. In order to facilitate the production of realistic image, he introduced an anti-folding constraint and smoothness constraints that restricts the amount of folds in the warped image and ensures the smoothness of the produced image, respectively. In the beginning of the pipeline, the author used two fully connected layers (FCN) to create two non-linear, dense transformation of the input images. This FCN is a U-net style network with varying filter and sampling sizes. These two FCNs share their parameters. Output of these two FCNs are two discrete displacements fields that Zhang ZHANG (2018) labelled as Flow. This flow is then fed to an inverse network and a grid sampler. The grid sampler produces a warped image. On the other hand, to implement the inverse consistent constraint, the author designed an inverse network that takes the discrete displacements fields and the negative of it and adds them together. Thus, by combining two such flow fields from two FCNs, using L2 norm, ICNet calculates the inverse constraint.

However, DALCA et al. (2019) realized that all of these methods considered point estimates of the weights, leaving probabilistic framework aside. They argued that classical registration methods had a rigorous theoretical background, but lacked in speed as they were built to work with pairs of images individually. On the other hand, recent methods, developed in deep learning based models, are much faster, but they lack the theoretical background. DALCA et al. (2019) attempted to bridge the gap by utilizing a probabilistic deep learning method that utilizes registration uncertainty. In addition to this, they ensured the diffeomorphic properties of the registered image which is able to preserve topological information. They modelled

the registration problem as a variational inference on probabilistic generative model. To realize this concept, they utilized a CNN-based model with a combination of upsampling and downsampling, configured to match the GPU setup. The end-to-end system takes two inputs, one moving 3D image and one fixed image, both of which run through the aforementioned CNN network, giving an output of two: the mean and variance of the associated velocity field. These two parameters builds the velocity field that then pass through an integration layer built using squaring and scaling operations. This layer produces a deformation field which is used by a spatial transformer to produce the final moved image. However, because of the probabilistic nature of their algorithm, they had to decide how to solve the intractability issue of the computation of posterior probability for the parameters; they solved it using variational approach and proposed to calculate an approximation of the posterior probability instead of being exact.

KHAWALED und FREIMAN (2020) took a different approach for solving the image registration problem probabilistically. Instead of approximating the posterior probability using some variational method, KHAWALED und FREIMAN (2020) chose to model the true posteriori using Stochastic Gradient Langevin Dynamics (SGLD) method, as proposed by WELLING und TEH. In this method, a small amount of noise is purposefully injected during the training of the mini-batch stochastic gradient descent, to guide the descent to a point that is equivalent to sampling from the true posterior distribution. That way, we don't have to work with an approximation, rather we can have the true distribution parameters.

Khawaled et al. argued in favor of the bayesian treatment of image registration problem because of the small size of the dataset. They argued that this problem alone can cause overfiting if point estimates are used, instead of probability estimates.

On the other hand, HANSEN und HEINRICH (2020) expressed their concern over the sub-standard perfomance of deep learning techniques in case of inter-patient abdominal MRI or CT of lungs during breathe-in and breathe-out phase. They blamed the use of encoder-decoder style architecture, U-net style architecture to be precise, for this. They argued that U-net architectures perform poorly in case of very large deformations, even if multiple levels of encoding-decoding is used. To mitigate this, they proposed to

calculate the displacement vector field directly from the two input images. They utilized a CNN based feature extractor that is able to extract features of interest from the images. But they don't capture all the features in order to generate a vector field, rather they choose a set of key points where they focused their extraction process. They have selected these points based on Foerstner interest operator. Thus, they created a feature map in a low dimensional space that they call displacement map. This map then generates the moved image. The flip side of this method is the usage of fixed points to extract they features. Some of the important features can easily get missed because of this selection process.

GRIGORESCU et al. (2020), on the other hand, argued in favour of combining structural and microstructural information. They argued that, from the point of tracking the microstructural changes, it is even better to combine structural information, obtained from T2-weighted images, and microstructural information, extracted from diffusion tensors. Such an approach makes the process faster to an extent that registration can take place in one pass only. In order to achieve that, they have based their model on VoxelMorph, due to BALAKRISHNAN et al. (2019), and added layers on top of this which are capable of handling diffusion tensor images. Initially, the took a pair of fixed and moving T2 weighted images and combined them together to feed into a CNN based extractor that extracts a velocity field by calculating the displacement of voxels. This then is followed by a topology-preserving scaling and squaring layer that gives the dense displacement field (DDF). This DDF is the used to spatially transform and warp the moving image into a moved image.

QIN et al. (2020) worked with for developing the way we regularize neural networks for image registraion problem. They identified that even if current regularizers are capable of making this image registration problem a well-posed problem, from an ill-posed one, these are unable to incorporate domain specific information. Furthermore, these regularizers often make generic assumption, which might become counter-productive. To alleviate these issues, QIN et al. (2020) proposed a bio-mechanic informed regularizer that incorporates domain specific information which in turn will help immensely in producing much more realistic images as the regularizer itself is more realistic than some generic one like smoothness constraint, etc. To realize this, the authors utilized a variational auto-encoder (VAE)

that learns the bio-mechanically possible deformations. To be more precise, biomechanical simulations, done by partial differential equations, are run to simulate possible deformations. This simulation are then run through a variation auto-encoder which gives the probability of these simulations. This learnt VAE can then used as the regularizer for any registration network.

4
Baseline Methods

4.1 ICNet

4.1.1 Introduction

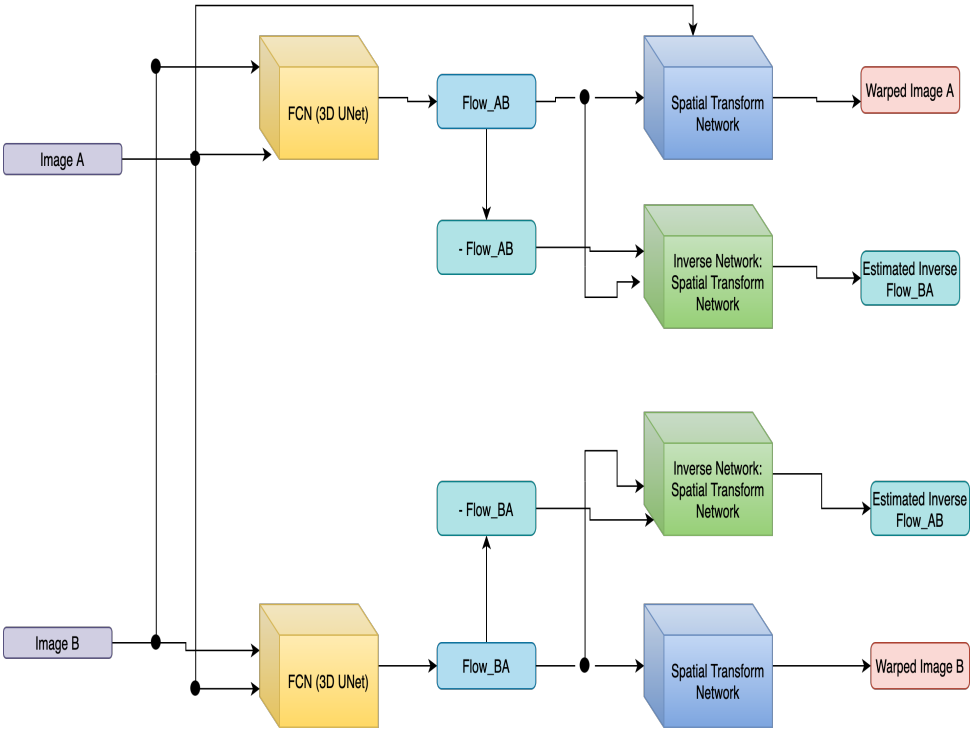
ZHANG (2018) are performing image registration both ways, i.e. Let's assume one image as A and another as B, with ICNet architecture image A is registered to image B and called as warped_A and image B registered to image A is called as warped_B. When image A gets registered to image B, in the process a flow or displacement vector field named as Flow_AB is defined. Similarly, when image B will get registered to image A, flow Flow_BA will be generated. ZHANG (2018) has denoted registered or warped images as warped_A and warped_B.

4.1.2 Architecture

ICNet's ZHANG (2018) architecture consists of a fully convolution network that takes as input image A and B and results in warped_A and warped_B by modeling Flow_AB and Flow_BA. Authors have proposed two modules of fully convolution network for registering, one for image A to B, B being the target image and another one for image B to A, A being the target image. These two FCN modules share parameters and have the same structure. The fully convolution network architecture used in ZHANG (2018) paper is UNet ÇIÇEK et al. (2016) RONNEBERGER et al. (2015). This network comprises of two paths - the contracting path used for an image's undersampling and for oversampling of an image - an expanding path.

The ICNet ZHANG (2018) pipeline consists of a grid sampler that takes as input image A and flow Flow_AB and gives out warped_A. Similarly, image

B and flow Flow_BA as inputs result in warped_B as the output. A grid sampling network is nothing but a spatial transformer network JADERBERG et al. (2016) that is fully differentiable. It consists of a generator that generates a spatial grid and a sampler. This spatial grid gets converted to a sampling grid by using the flow or dvf which we get from the fully convolution network. Finally, the source image is warped by making use of the sampling grid by the sampler. Authors ZHANG (2018) made use of bilinear mode for interpolation. Architecture of ICNet is described in 4.1



(a)

Figure 4.1: ICNet Overview

4.1.3 Training

ICNet ZHANG (2018) proposed three types of constraints, inverse consistent constraint, antifolding constraint, and smoothness constraint. Inverse consistent is a regularisation term used for penalizing the deviation of the transformation fields (flow) from their corresponding opposite or inverse mappings i.e. inverse flow. The second constraint ZHANG (2018) mentioned

is the antifolding constraint that is to prevent folding from happening in the flow generated. The last constraint used is the smoothness constraint. This constraint encourages the local smoothness of transformation fields. For the objective function, ZHANG (2018) used mean squared distance for measuring the similarity between the fixed image and the registered image. So, the final objective function becomes the sum of similarity, inverse consistent, antifolding, and smoothness constraint, where the constraints are multiplied by hyperparameters a,b and c respectively to balance out their values.

4.2 FIRE

4.2.1 Introduction

WANG et al. (2019) talks about unsupervised image registration for inter-modality. This method does multi-modal registration for any dimension of the image, to achieve this, it makes use of cycle consistency loss and inverse consistent property. In this paper, WANG et al. (2019), similar to ICNet ZHANG (2018) both image A and image B gets registered to each other. The model designed gives two fields that is used to register image A to B and image B to A. These transformation fields are achieved by minimizing the objective function.

4.2.2 Architecture

FIRE's WANG et al. (2019) architecture consists of total 5 components. First is an encoder which is used for extracting features that are not modality dependent, then comes two decoders, which are used for mapping the attributes extracted by the encoder with the warped images. And lastly, two transformation networks are used for generating transformation fields. During the registration process, the encoder's output is also warped to the transformation field.

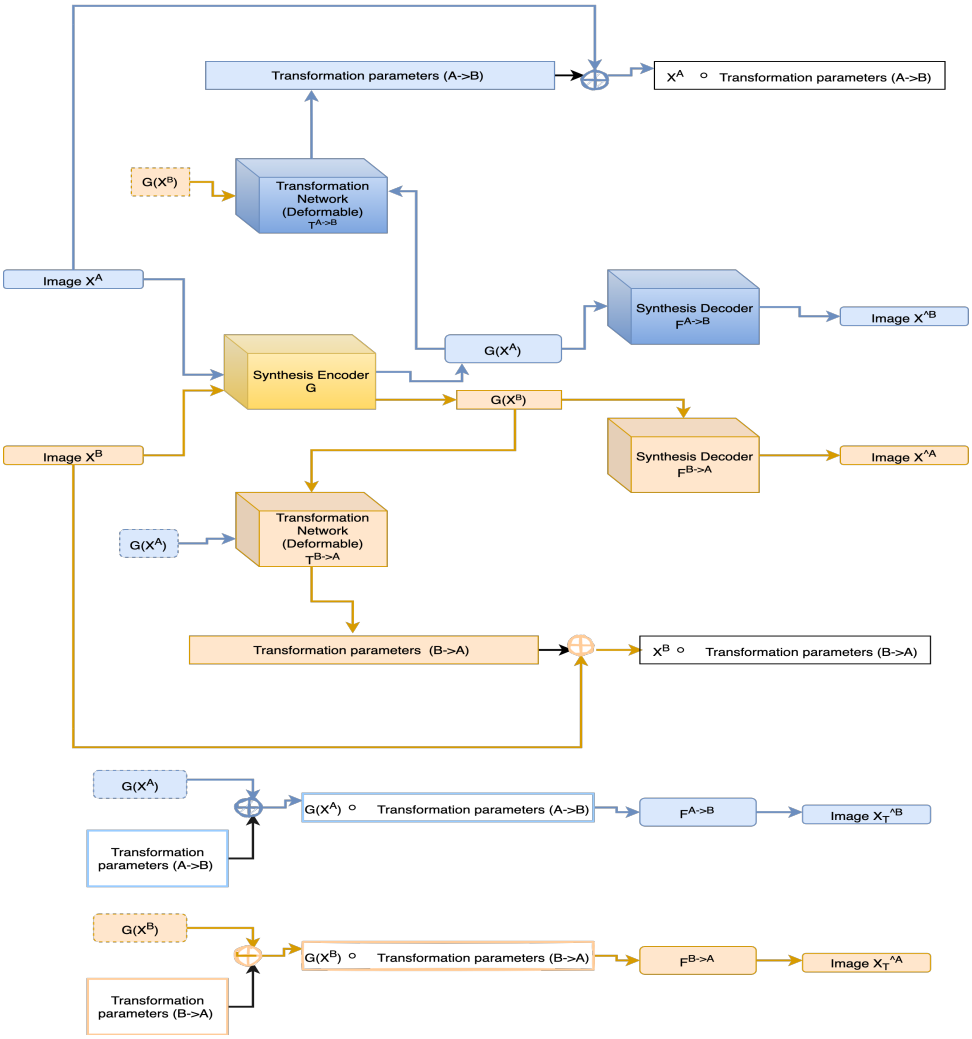
We have first defined a residual block or a resnet block according to HARA et al. (2018) and used this resnet block in encoder and decoder networks. Encoder's architecture consists of 1 convolution layer followed by two layers of downsampling and then there are 4 residual blocks. Whereas, in the

decoder network, we first have 4 of the residual blocks, then comes two oversampling convolutional layers where we have have used 3D transposed convolution operator and then final convolutional layer along with a 1D convolutional layer. Kernel size for the convolutional layers is 3 and we have used instance normalization as defined in paper WANG et al. (2019) and leaky relu as the activation function.

In the fire paper, WANG et al. (2019) for generating transformation field, authors have made use of both affine as well as deformable transformation networks. But we have used already affine registered images (i.e T1 affinely registered to T2 and T2 affinely registered to T1), hence we are exploring only deformable network in this case. For deformable transformation network, there are two convolutional layers that intakes encoded image x and image y respectively and then activation function leaky relu is used. These two layers are then concatenated and passed to a residual block followed instance normalization, leakyrelu, convolutional layer and again instance normalization and finally tanh. All of these extracted attributes result in a displacement vector field. This field is then passed to a grid sampler or a spatial transformation network JADERBERG et al. (2016). Architecture of FIRE is described in 4.2

4.2.3 Training

The objective function for this network has three components, the first is synthesis loss, the second one is registration loss and the last is a regularization term. Synthesis loss WANG et al. (2019) again has four parts, the first is accuracy loss, the second one is for the features extracted, next is the loss for cycle consistency for the intermodality synthesis and lastly is the alignment loss. Registration loss consists of two losses, the first is accuracy loss for the registration and then a loss function for inverse consistency. All of these components for synthesis and registration loss makes use of the Root Mean Square error function. The regularization term in the original paper has three terms, we are not considering the first two terms - synthesis regularization and registration regularization as both of them make use of affine network output and we are already feeding the network affinely registered images. The third component of regularization is a smoothness constraint which is calculated by non-rigid transformation fields to the



(a)

Figure 4.2: FIRE Overview

smooth loss function and summing and multiplying it with a hyperparameter. Value of this hyperparameter or scaling parameter WANG et al. (2019) (let's say a) is calculated as -

$$a = (2^{2m})/(10M) \tag{4.1}$$

where m stands for the dimension of the images (3 in our case) and M represents the number of pixels in the image. (128 or 64 in our case).

4.3 ADMIR

4.3.1 Introduction

ADMIR TANG et al. (2020) (Affine and Deformable Medical Image Registration) is an affine and deformable image registration that works end to end. This uses the convolution neural network (ConvNet) to perform unsupervised image registration. This method doesn't require the images to be pre-aligned, which in turn helps to do image registration quickly with good accuracy.

An affine registration network, a deformable registration network, and a spatial transformer are the three fundamental components of ADMIR. Both Affine and Deformable networks were trained at the same time in the original publication. The traditional image registration techniques improve the registration by iteratively optimizing the similarity function. However, these are time-consuming and incapable of dealing with complex shape changes. ADMIR made an effort to decrease time while simultaneously improving the quality of registration results. We know that getting labelled data is always not easy. The ADMIR tries to solve this problem as it is an unsupervised learning algorithm, which means our data need not be labelled with information like landmarks, ground-truths, or pre-known transformations.

4.3.2 Model

ADMIR TANG et al. (2020) has two sub-networks, one focuses on affine registration and the other one focuses on deformable registration. The fixed and moving pictures are concatenated and sent into the Affine ConvNet, which predicts 12 affine transformation parameters (rotation, translation, scaling, and shearing) that are used to calculate the DVF u_a, which is then used by the spatial transform to coarsely warp the moving image. The coarsely warped and fixed images are then combined and fed into the Deformable ConvNet, which calculates the DVF u_d. To get the final registration result, the final DVF u_f is calculated by aggregating DVF u_a and DVF u_d with the help of a spatial transformer, which warps the moving image to fully register the moving and fixed images.

4.3.3 Training

We tried to do the end-to-end affine registration with all the modules as mentioned in the ADMIR TANG et al. (2020). However, we did not get good results as the affine registration module did not produce the desired output. So, we reverted to the usual preprocessing pipeline that we are using in this paper with the help of ANTS and Freesurfer, and then we used the deformable module of ADMIR to perform registration. This also allows us to perform aggregation of DVFs. We have changed the loss function as we were not using the affine module.

Firstly, to obtain the local difference between the two images, the fixed and moving images are first concatenated and then input into the deformable network. The deformable network contains encoder and decoder stages. The encoder stage contains a stack of convolutional and strided convolutional layers to lower the spatial dimension step by step, allowing the network to learn local similarity from features with varied resolutions. Similarly, the Decoder Stage contains successive deconvolutional and convolutional layers, allowing the network to restore the low-resolution feature to the same resolution as the input image.

As mentioned in ADMIR TANG et al. (2020), We do element-wise addition of two features between two same-size cubes to make maximum use of the information in low- and high-resolution features. This in turn connects to a final fully convolutional layer which determines the final DVF. The final DVF is used by the spatial transformer to fully register the moving image to the fixed image. After each convolution operation, LeakyRelu and Batch Normalization are utilized, except for the final convolutional layer.

Each convolution block is made up of a strided convolutional layer with a kernel size of $3 \times 3 \times 3$ and a stride of 2 and 1 for the decoder and encoder, respectively. A Batch Normalization layer and a LeakyRelu activation layer are also added after each layer. We changed the loss function a little because we aren't utilizing the affine module. As indicated in the paper, we employed gradient loss for smoothness in addition to normalized cross-correlation (NCC) as a similarity loss.

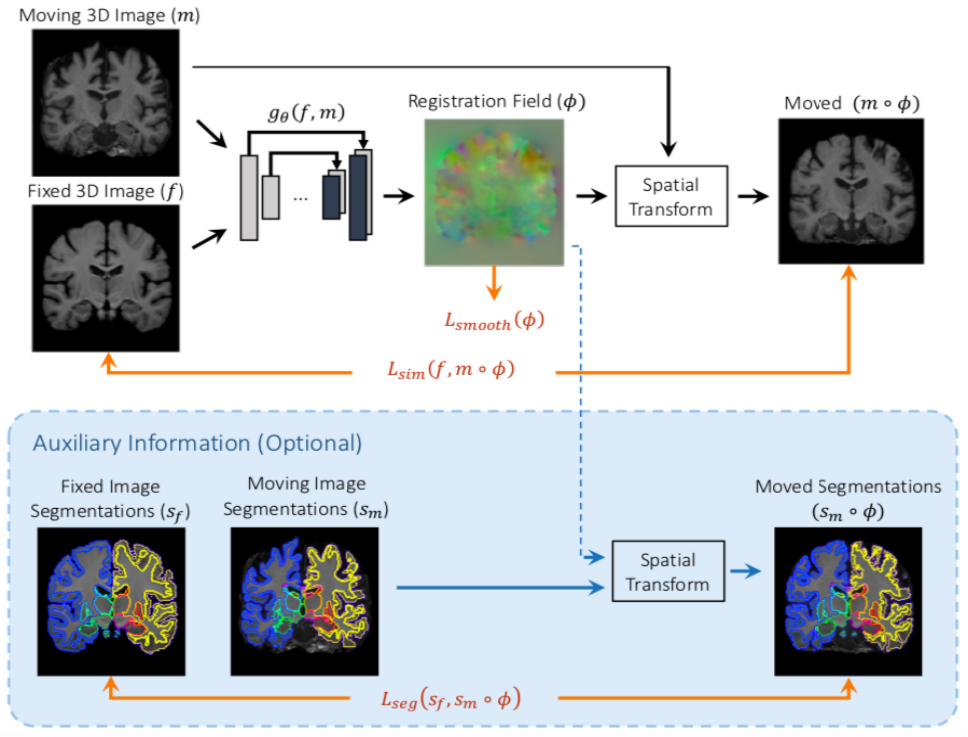
4.4 Voxelmorph

4.4.1 Introduction

Voxelmorph BALAKRISHNAN et al. (2019) is a deep learning based, unsupervised, pairwise, deformable registration framework. The framework aims to use CNN for parameterizing the function to calculate deformation field. The model is unsupervised and learns based on image similarity metric, Hence it does not require costly supervision and generation of deformation field for training. The framework also incorporates additional supervised segmentation training regimen that could help further improve the performance on the dataset. Traditional iterative approaches have the disadvantage of optimizing deformation field for each pair of images independently of others and takes minutes to compute on GPU. Voxelmorph takes in two n-D volumes and compute deformation field in few seconds. This reduces compute time whilst maintaining performance. An overview has been shown in below Fig 4.3 BALAKRISHNAN et al. (2019)

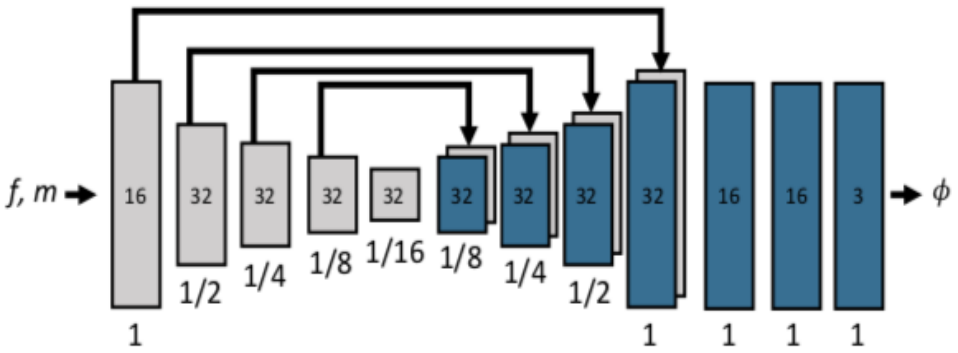
4.4.2 Model

Voxelmorph is based on UNet ÇIÇEK et al. (2016) and has a similar encoder, decoder based blocks with skip connections. Voxelmorph takes a single input of concatenated moving and fixed image. Each encoder block halves the spatial dimensions using strides of 2 and applies non linear activation LeakyReLU with parameter 0.2. As we reduce the feature dimensions subsequent encoders learn more high level information about the images. The decoder block consists of an upsampling block that upscales the feature dimensions by a factor of 2 and convolution layer that concatenates the corresponding encoder features to upsampled features and applies convolution on them. This is finally fed to few convolution layers that maintains the spatial dimensions of the deformation field and captures of finer changes in anatomy. Fig 4.4 BALAKRISHNAN et al. (2019) shows the architecture of Voxelmorph.



(a)

Figure 4.3: Voxelmorph Overview



(a)

Figure 4.4: Voxelmorph Architecture

4.4.3 Training Procedure

The dataset comprises of T1 3731 brain volumes from publicly available dataset, it consists of both healthy and diseased person of varied age dis-

tribution. The volumes are preprocessed using Freesurfer FISCHL (2012). Scans were resampled to standard $256 \times 256 \times 256$ with 1mm isotropic size. The scans are then removed of bias fields, skull stripped and affine registration applied. After visual inspection few of the bad preprocessed scans are rejected. The affinely registered volumes are fed into the network and the resulting deformation field is used to transform moving image using Spatial Transformer JADERBERG et al. (2015). The warped image is then compared against the fixed image using a similarity metric such as MSE and NCC. The deformation field is regularized using a smoothness loss function that penalizes unrealistic sharp deformations. Equation 4.2 shows the overall loss function where f is fixed image, m is moving image, ϕ is the deformation field and $\mathcal{L}_{sim}$ is the similarity loss, $\mathcal{L}_{smooth}$ is the deformation field smoothness loss, λ acts as hyperparameter for regularization term.

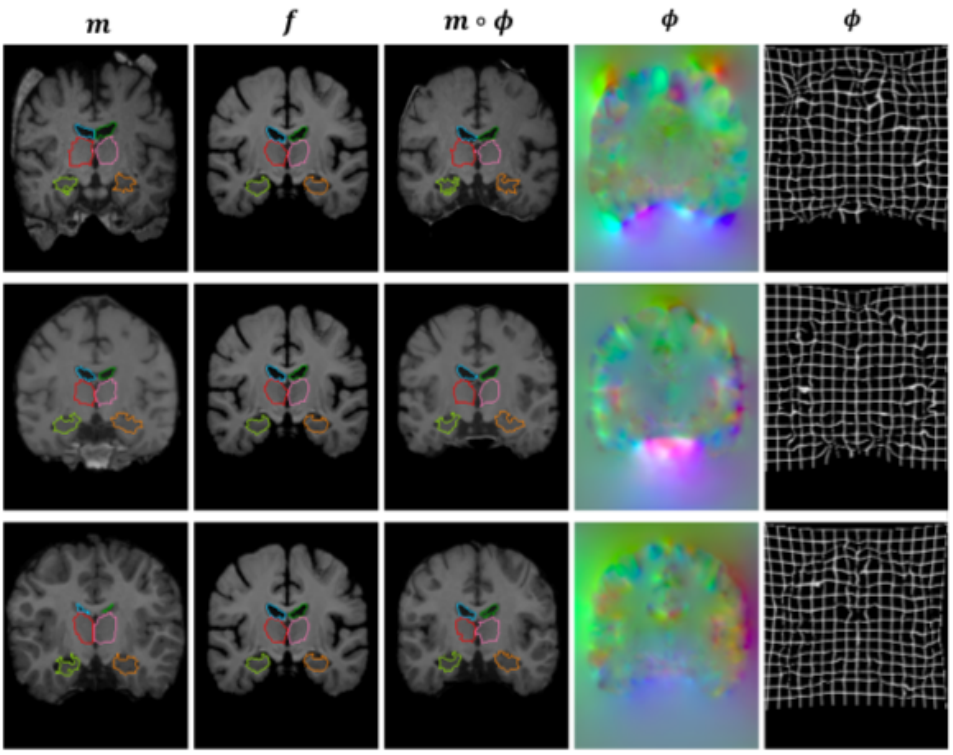
$$\mathcal{L}_{us}(f, m, \phi) = \mathcal{L}_{sim}(f, m \circ \phi) + \lambda \mathcal{L}_{smooth}(\phi) \tag{4.2}$$

If we consider additional segmentation for training then the loss function gets modified as below equation 4.3 where $\mathcal{L}_{seg}$ is the segmentation loss

$$\mathcal{L}_{a}(f, m, s_f, s_m, \phi) = \mathcal{L}_{us}(f, m, \phi) + \gamma \mathcal{L}_{seg}(s_f, s_m \circ \phi) \tag{4.3}$$

4.4.4 Evaluation

The model is evaluated using DICE score against the segmentation from the fixed image. Voxelmorph outperforms traditional techniques such as ANTS-Syn method. It obtains an average dice of 0.786 to that of 0.776. Below Fig 4.5 BALAKRISHNAN et al. (2019) shows the registration of sample volumes that highlights registered landmarks and the deformation field.



(a)

Figure 4.5: Voxelmorph Example Output

5
Proposed Methods

5.1 Direct Optimization based Method

5.1.1 Introduction

In the previous sections we have seen parameterizing a function using large deep learning models to calculate deformation fields. We also have discussed its advantages against traditional iterative algorithms in terms of performance and speed. In current method, we ditch the usage of deep learning model and optimize the deformation field directly using gradient descent without any model parameters. The rationale behind direct optimization is to experiment on registration performance without any deep learning model or complex mathematical optimization procedure. FUSE et al. (2000) describes various techniques based on gradients to estimate optical flow, SHERINA et al. (2020) discusses using multi-scale, iterative, numerical approaches and using additional speckle information to solve registration problem.

5.1.2 Training Procedure

Our focus in this experiments is to directly optimize a deformation field of size $3 \times 128 \times 128 \times 128$ for a pair of images using image similarity metric such as NCC or NMI and smoothness loss. Equation 5.1 explains the loss function used to optimize the deformation field. f and m represents fixed and moving image, $l_{sim_{m->f}}$ represents similarity loss and $l_{sm_{m->f}}$ represents smoothness loss to regularize deformation field. α and β We experiment using various optimizers with default settings in Pytorch PASZKE

et al. (2019), 1500 epochs and report the result for the same evaluation dataset as used for other deep learning based methods.

$$l_{total} = \alpha(l_{sim_{m->f}}) + \beta(l_{sm_{m->f}}) \tag{5.1}$$

5.2 MSCGUNet

5.2.1 Introduction

Deep Learning based techniques have been applied successfully to tackle various complex medical image processing problems. Over the years, several image registration techniques have been proposed using deep learning. Deformable image registration techniques such as Voxelmorph BALAKRISHNAN et al. (2019) have been successful in capturing finer changes and providing smoother deformations. However, Voxelmorph, as well as ICNet ZHANG (2018) and FIRE WANG et al. (2019) do not explicitly encode global dependencies (i.e. overall anatomical view of the supplied image) and track large deformations. This research improves upon Voxelmorph by employing self-constructing graph network SCGNet LIU et al. (2020) to encode semantics which can improve the learning process of the model and help the model to generalise better, multi-scale supervision to be able to work well in case of small as well as large deformations, and cycle consistency for making the deformations consistent.

5.2.2 Hypothesis

The MSCGUNet tries to capture global dependencies using SCGNet LIU et al. (2020), wherein encoded CNN features are used to find out semantics. We hypothesize that these dependencies could help the model to learn relationships even between two distantly located structures in the fixed and moving image or voxels of ventricles filled with CSF and helps generate better deformations. As mentioned earlier, SCGNet was able to find clusters of similar information in the image using latent space features. If the model knows about the landmarks then it would be helpful for improving the deformation field.

CHATTERJEE et al. (2020) uses multi-scale supervised UNet to perform segmentation of vessels. We apply the multi-scale supervision to handle different amount of deformations, where deformations are predicted and losses are calculated for both original image and a downsampled image. We hypothesise that deformations on downsampled image would cover for larger deformations while the deformations on original image would produce finer deformations. The multi-scale supervision also allows for faster convergence, brings stability in the training as there would be multiple channels through which gradients can flow.

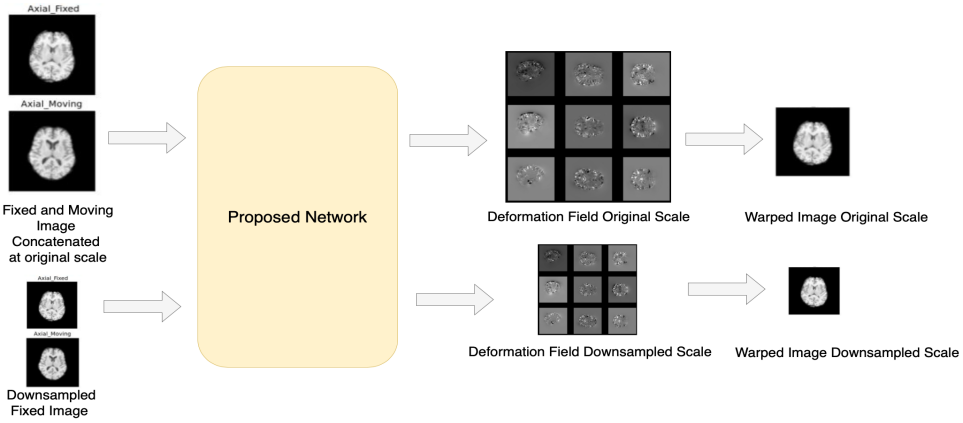
An adopted version of the cycle consistency loss, inspired by ZHOU et al. (2016), was employed to maintain consistent deformations. Cycle consistency helps to maintain flow correspondence between fixed and moving image thereby drastically reducing the number of possible deformation fields to achieve the same deformation .Regularisation of sharp deformation was also used to obtain smooth deformation fields and remove unrealistic transformation.

5.2.3 Method

Let f, m be fixed and moving n-D images where they belong to domain $\Omega \subset \mathbb{R}^n$. We use 3-D single channel grayscale scans from IXI dataset BY BRAIN-DEVELOPMENT.ORG for training but the framework is applicable also for other dimensions. We apply preprocessing treatment such as bias field correction, skull stripping, affine registration and intensity normalization on both the images. Both the images are then downscaled by a factor of two to be fed as secondary input. The complete CNN SCGNet based model parameterizes the function G_θ that calculates the deformation field ϕ given a pair of fixed f and moving m images. The deformation field will have n-channels as it has to represent displacement in each of the n-axes. If its a 3-D image then deformation field will have 3 channels where each channel represent displacement in x, y and z axes respectively.

MSCGUNet takes in fixed f and moving m image along with its downsampled counterparts f_d and m_d and computes deformation field ϕ and ϕ_d. Both the pairs are passed through the network and corresponding deformation fields are then applied to $m \circ \phi$ and $m_d \circ \phi_d$. Spatial transformers are used to warp the moving image and a similarity metric is used to compare f

with m and f_d with m_d. Gradient descent is used to minimize the loss and find optimal parameter $\hat{\theta}$. Fig 5.1 describes overview of the MSCGUNet.



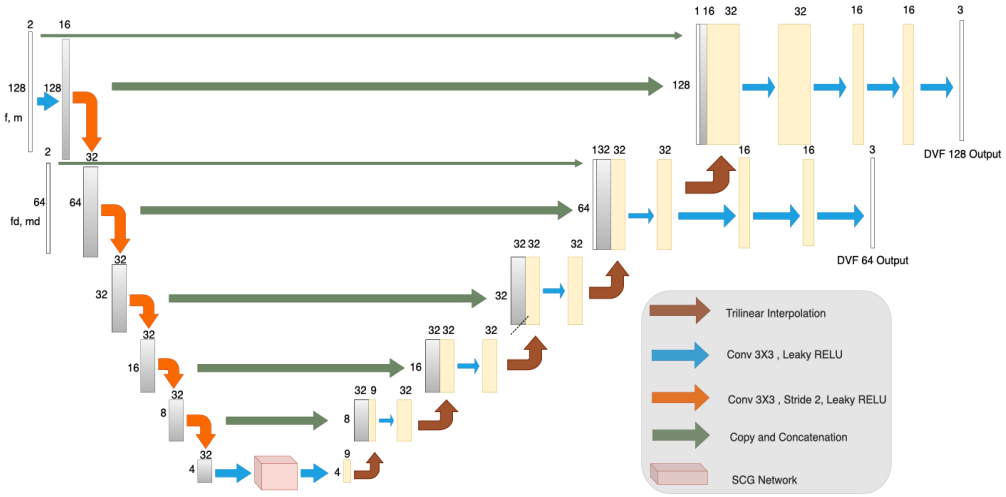
(a)

Figure 5.1: Multi Scale UNet Overview

5.2.4 Architecture

The fixed f and moving m images are concatenated to the dimensions of $2 \times 128 \times 128 \times 128$ similarly downsampled images are concatenated with combined dimension of $2 \times 64 \times 64 \times 64$. The model uses UNet like structure to encode and decode the image information. For encoding, it uses convolution with stride of 2 and activation LeakyReLU to downsample the image and learn relevant features. Encoder near the latent space learn more high level features, whilst encoder at the beginning learns low level features that acts as an image pyramid. The encoded information is fed into the latent space of the SCGNet, which first parameterizes the encodings to latent embeddings with mean and standard deviation and derives the undirected graph through an inner product of latent embeddings. The output is then fed into a fully-connected graph convolution network (GCN) which attempts to learn semantics of the brain anatomy. The GCN output is upsampled through trilinear interpolation and supplied to a CNN decoder which takes in also input from the respective encoder as skip connections. Each decoder block scales up the spatial dimensions of the image and reduces the channels by a factor of 2. Skip connections help improve gradient back propagation and training convergence. It also supplies the

corresponding encoder information that provides location information that would be missing from decoder. Fixed image f and f_d are concatenated to the UNet output to maintain contrast of the fixed image DEWEY et al. (2019). Finally, deformation fields are smoothened by convolution layers which helps learn deformation at finer scales. Fig 5.2 illustrates the model architecture.



(a)

Figure 5.2: Multi Scale UNet with SCGNet Architecture

5.2.5 Training Procedure

We use publicly available IXI dataset BY BRAIN-DEVELOPMENT.ORG to demonstrate the registration. 1000 images are used for training. The images are preprocessed using ANTs BRIAN B. AVANTS. During preprocessing we apply skull stripping, bias field correction, intensity normalization and affine registration on the images. Fixed f and moving m images were downsampled, the fixed f and moving m original images were concatenated and sent as an input to the network. Network processed the images and provides deformation fields at both original and downsampled scale. This is applied on both original, downsampled moving image and compared with both the fixed images. This process is repeated for the same image pair by swapping fixed and moving images. Finally both pair of losses equation 5.2 are added up and back propagated through the network. Similarity measure Normalized Cross Correlation NCC and Normalized

Mutual Information NMIl_{sim}were used for intramodal and intermodal registration for training. Smoothness loss l_{sm} HORN und SCHUNCK (1981) acts as regularizer that penalizes sharp deformations and scg loss l_{scg} helps to learn similar features in the brain. α, β and λ are hyper parameters. We experiment using various hyper parameters and make ablations to the network to measure the impact of each component on the registration performance. We run the training process on 200 images for 1000 epochs with learning rate of $3e-4$. The hyper parameters α and α_d are set to -1.2 and -0.6, β and β_d are set to 0.5 and 0.25 and λ is 5. We achieve the best performance using these hyper parameters for the combination of dataset, preprocessing and the model.

$$l_{total} = \alpha(l_{sim_{f->m}} + l_{sim_{m->f}}) + \alpha_d(l_{sim_{f_d->m_d}} + l_{sim_{m_d->f_d}}) +$$
$$\beta(l_{sm_{f->m}} + l_{sm_{m->f}}) + \beta_d(l_{sm_{f_d->m_d}} + l_{sm_{m_d->f_d}}) +$$
$$\lambda(l_{scg_{f->m}} + l_{scg_{m->f}})$$
(5.2)

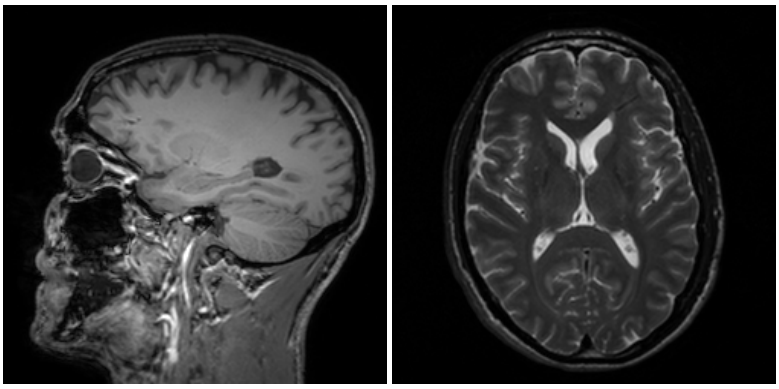
6

Experiments and Evaluation

6.1 Dataset and Preprocessing

6.1.1 IXI Dataset

The IXI Dataset project consists of nearly 600 Magnetic resonance(MR) images from healthy subjects. For each subject the dataset consist of three MR image acquisition protocols, namely, T1, T2 and PD-weighted images, MRA images and Diffusion weighted images. These images are collected at three different hospitals in London: Hammersmith Hospital using Philips 3T system, Guy's Hospital using Philips 1.5T system and Institute of Psychiatry using GE 1.5T system and have been downloaded from BY BRAIN-DEVELOPMENT.ORG.



(a) (b)

Figure 6.1: (a) T1-w image (b) T2-w image

Our experiments makes use of the T1 and T2 weighted images to perform both intermodal and intramodal registrations. The T1 weighted im-

ages with voxel ordering AIL (A-P within I-S within L-R) consist of 150 slices each of size 256 x 256 pixels having a resolution of 0.9375 x 0.9375 x 1.2000m^3/voxel. The T2 weighted images with RPI (R-L within P-A within I-S) voxel ordering consist of 116 to 130 slices each of size 256 x 256 pixels having a resolution identical to that of the T1 weighted images.

We discuss the Data preprocessing pipelines we have experimented in detail in section 6.1.2 followed by experiments and different approaches in section 6.2. The section 6.3.1 describes the evaluation metrics that we have adopted and the section 6.3.2 describes the baseline which we compared with followed by results in section 6.4

6.1.2 Data Preprocessing

Data pre-processing has always been an integral part of Machine Learning applications going hand in hand with the (Garbage in Garbage out) GIGO concept. The information that the user perceives to be useful might need different preprocessing steps and the quality of the data directly affects the learning ability of our models. These techniques may involve removal or reduction of noise and artefacts, accomodate for the intensity differences of images based on scanners using techniques like image filtering, resampling, intensity normalization. Pre-processing techniques employed by our pipeline encompasses the analyzes, processing and visualization of 3D MRI data.

Experiments were conducted with different preprocessing pipelines and the we describe each of these in detail in the following section. The pipelines were in some cases directly implemented exactly based on requirements set forth by the research papers which we tried to replicate and in some cases were adapted to give better registration performance.

Pipeline 1: FreeSurfer Preprocessed Dataset

FreeSurfer facilitates analysis and visualization of structural and functional neuroimaging data and the software package provides a full preprocessing stream which we have employed in our pipeline. The full preprocessing stream of FreeSurfer cortical reconstruction process named *recon-all*, consists of distinct 31 preprocessing stages. These 31 stages are further divided

into sub directives and the user can decide as to perform the whole cortical reconstruction or any sub particular directive. The FreeSurfer *recon-all* can be executed using a Command Line Interface (CLI) with the following command which takes the subject name and input volumes and the recon command as input.

```
recon-all -subject subjectname -i invol1 <-i invol2> -all
```

The recon command specifies if the complete cortical reconstruction process has to be executed or some of the sub directives. BALAKRISHNAN et al. (2019) uses the preprocessing pipeline employing Fressurfer and we use a similar preprocessing pipeline in our experiments. The preprocessing pipeline for our experiment uses the first sub directive, *autorecon1* which consists of stages as shown in Figure 6.2. The following section describes in-

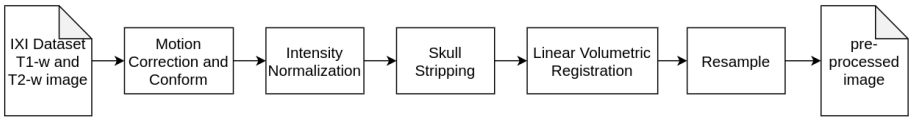


Figure 6.2: Pre-processing Pipeline

dividual preprocessing steps and the time taken for each of these steps are approximated for an AMD Opteron 64bit 2.5GHz processor as mentioned in the FreeSurfer documentation and described in DALE et al. (1999).

Motion Correction and Conform: The motion correction step takes multiple source volumes as input and correct for small motions between these volumes and averages them together. Processing time for the Motion Correction step is estimated to be less than 5 minutes.

NU Intensity Correction: Variations in scanners or parameters during MR image acquisition, often results in intensity non-uniformity in MR data. Hence the Non-paramteric Non-Uniform Normalization (N3) step is incorporated to account for this intensity differences. This step makes relatively few assumptions about the data (Non Parametric) and four iterations of normalization is performed. NU Intensity correction takes approximately 3 minutes for processing.

Talairach: Talairach is a FreeSurfer script which computes affine transform from the input volume to the MNI305 atlas. The transform computation internally uses the MINC program *mritotal* and these coordinates are used

as seed points in subsequent stages of cortical reconstruction process. Talairach computations is performed approximately within a minute.

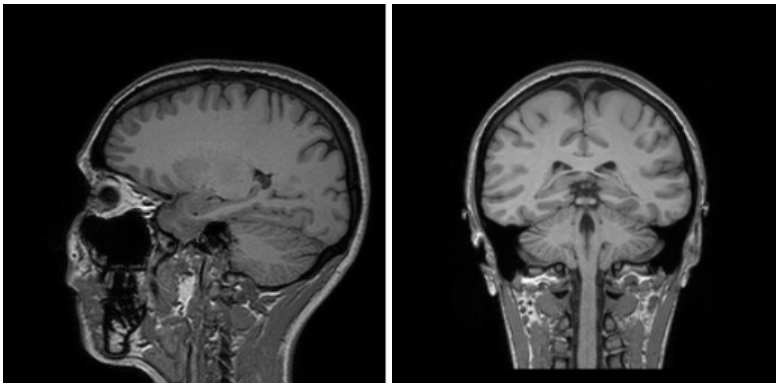
Normalization: The Freesurfer *autorecon* directive performs further intensity normalization of the volume to correct for fluctuations in intensity that would otherwise make segmentation difficult. Normalization step scales intensities of all voxels in such a way that the mean intensity of white matter becomes 110. Further control points can be added by the user, but in our experiments we have kept the default settings and this configurations takes approximately 3 minutes for a volume.

Skull Strip: Eliminating extra-cranial and non brain tissues might be an important step to ensure better segmentation of brain regions for many clinical applications and analysis. The skull strip step removes the skull from normalized input volume and generates a brain mask. *mri_watershed* program is then executed with or without seed points specified by user to get the skull stripped volume. The whole procedure takes under a minute to complete. The *autorecon1* directive of FreeSurfer cortical reconstruction process end with this stage.

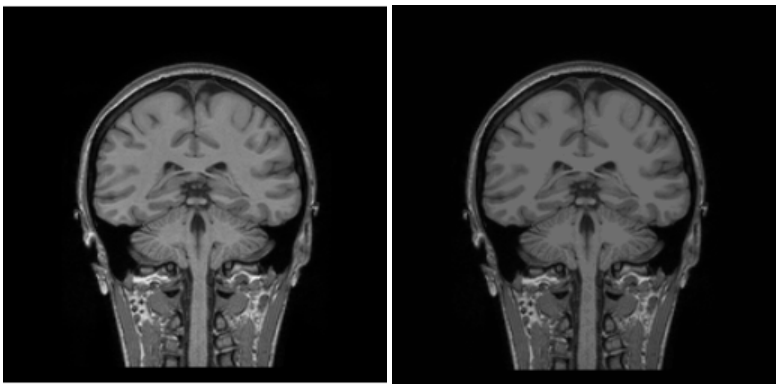
We further perform affine registration on the FreeSurfer preprocessed volumes to form image pairs. Affine registration between the image pairs is performed using Advanced Normalization Tools(ANTS) toolkit by setting the *type_of_transform* to 'Affine'. Experiments without affine transformations were also conducted and we discuss the details in evaluation section. The affine registered images were further resampled to 128 x 128 x 128 grid with 1mm isotropic voxels. The preprocessed dataset was split into 200 image pairs for train and 50 image pairs for test. The preprocessing stages with a sample output for each stage is depicted in Figure 6.3

Pipeline 2

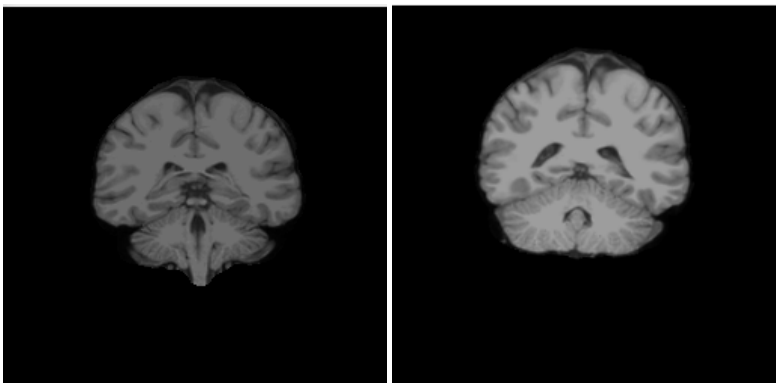
A different preprocessing pipeline was used to create a validation set to check the generalization performance of our models and details of this pipeline will be described in the following section. The validation set uses the same datasource namely IXI Dataset, however uses different stages and tools. The pipeline 2 follows steps similar to the preprocessing pipeline as described in ZHANG (2018). We experimented with some of the commonly used software packages such as Advanced Normalization Tools(ANTS),



(a) (b)



(c) (d)



(e) (f)

Figure 6.3: (a) Input Volume (256 x 256 x 150) 0.9375 x 0.9375 x 1.2000m^3/voxel
(b)Motion Correct and Conform (256 x 256 x 256) 1mm isotropic voxels
(c) Intensity Correction for better segmentation
(d) Intensity Normalized (e) Skull Stripped volume (f) Affine registered volume

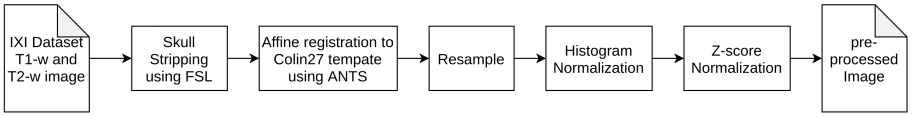


Figure 6.4: Pre-processing Pipeline 2

FMRIB Software Library(FSL), ITK-SNAP for the preprocessing steps. Individual stages of the pipeline is depicted in Figure 6.4

Skull Strip using FSL: Brain Extraction Tool(BET) is used to extract the brain removing non brain tissues from the whole head. The tool also gives provision to create brain mask and let user chose the intensity threshold to estimate brain outline. We have used the default settings in our experiments for skull stripping.

Affine registration to Colin27 template: The input volume is affine registered to Colin27 atlas which is a 181 x 217 x 181 grid with isotropic voxels of 1mm. The template image is skull stripped as well prior to registration to ensure the volumes are not distorted. Colin27 atlas with high SNR and structure definition is an average of 27 T1 weighted MRI scans of the same individual. The *type_of_transform* parameter for ANTS registration was set to 'Affine' to perform this registration and the registration is approximated to complete in less than a minute.

Resample: After affine registration with Colin27 atlas the image volumes are of dimensions 181 x 218 x 181. To ensure consistency with the pipeline 1 dataset, we further resample the images to 128 x 128 x 128 grid having istropic voxels of 1mm. Resampling is performed with the help of nibabel open source library.

Histogram and Z-score Normalization: Intensity normalization is performed by matching intensity histogram of each brain MRI to Colin27 template using Histogram matching algorithm. We also perform Z-score normalization to ensure the mean intensity of each image is zero and ensure the standard deviation is one.

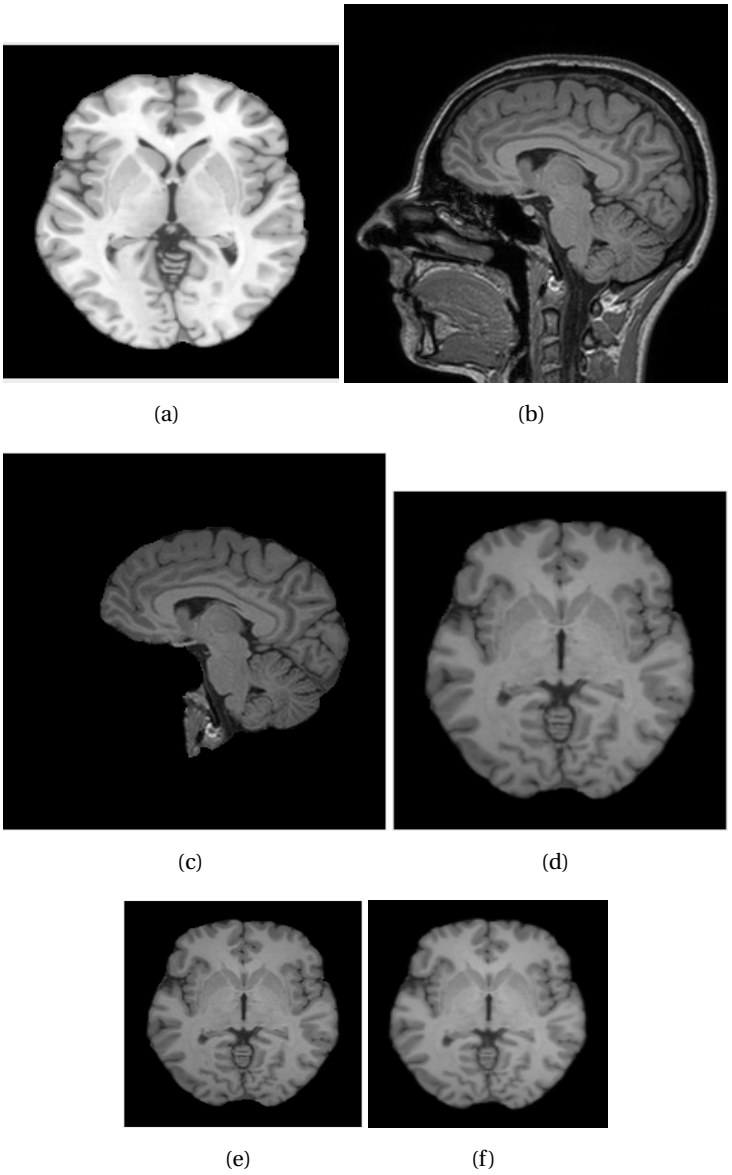
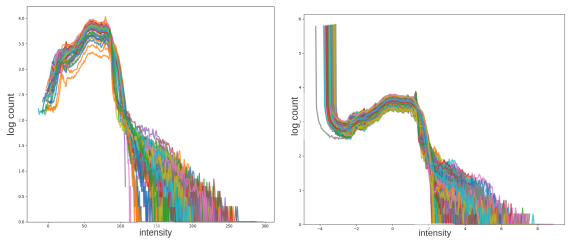


Figure 6.5: (a) T1 weighted Colin27 Template (b) input volume (256 x 256 x 150) 0.9375 x 0.9375 x 1.2000m^3/voxel (c) Skull stripped using FSL (d) affine registered with Colin27 template (181 x 217 x 181) (e) resampled and histogram normalized (128 x 128 x 128) volume (f) z-score normalized volume



(a) (b)

Figure 6.6: (a) Histogram Normalized Images (b) Z-score normalized images

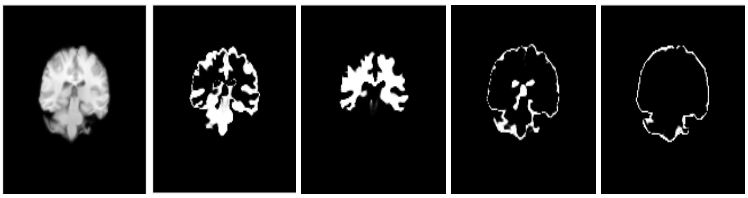
6.2 Preliminary Experiments

6.2.1 Segmentation

A key step in most clinical applications involves image segmentation, one of the major tasks in medical image analysis. As discussed by DESPOTOVIĆ et al. (2015) image segmentation is extensively used to analyze changes in brain and also visualize brain's anatomical structures. Although several segmentation techniques have evolved with time, we found that the segmentation techniques does not always gives the expected results. One of the evaluation metric used for measuring image registration quality in our report is using Dice score. We compute Dice score for individual segments of the brain namely White matter(WM), Gray matter(GM) and Cerebrospinal fluid(CSF) and is averaged to get the final score per input volume. The experiments were conducted with popular MRI toolboxes such as Advanced Normalization Tools(ANTS), FMRIB Software Library(FSL), FreeSurfer, Trainable Weka Segmentation (ImageJ). The following section gives a brief comparison between the segmentation results of these toolboxes.

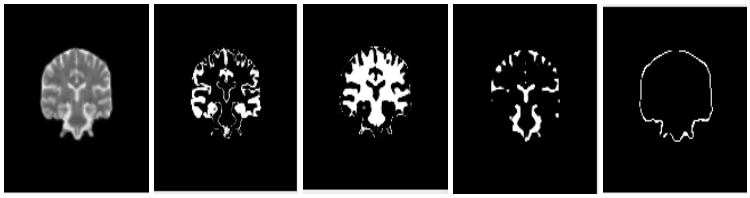
Advanced Normalization Tools(ANTS): The ANTS toolbox uses a finite mixture model(FMM) segmentation algorithm, called the atropos and is described in BRIAN B. AVANTS. Atropos provides the provision to specify prior constraints such as specification of prior label image, MRF prior which ensures label are spatially smoothened. The function takes the image to segmented along with the image mask and also the number of classes so as to initialize K-means clustering. We also specify mrf parameters such as smoothingFactor and radius which determines the amount of smoothing

and the radius respectively. Convergence parameters includes number of iterations and the convergence threshold. We performed experiments with values of k for the K-means algorithm set to 3 and 4 to segment the brain volume into CSF, WM and GM for the first case and also to segment the brain volume into CSF, WM, GM and the background voxels for the latter. We observed that the T1 weighted volume segmentation often results in parts of GM wrongly segmented into the CSF and to the background labels as observed in 6.7. For T2 volumes ANTS atropos segmentation does not work very well and we observe most of the Gray matter is wrongly segmented into White matter and CSF segments as shown in figure 6.8



(a) (b) (c) (d) (e)

Figure 6.7: Ants atropos Segmentation(a) T1 weighted input volume (b) GM (c) WM (d) CSF (e) Background



(a) (b) (c) (d) (e)

Figure 6.8: Ants atropos Segmentation(a) T2 weighted input volume (b) GM (c) WM (d) CSF (e) Background

FAST: FSL is another commonly used toolbox for medical image analysis and FAST(FMRIB's Automated Segmentation Tool) is the module used to perform segmentation in FSL. Unlike ANTS where segmentation is performed using scripts, we can perform segmentation task in FAST using a GUI as well as the command line. FAST uses hidden markov random field model and as associated Expectation-Maximization(EM) algorithm to perform the segmentation task. The FAST algorithm takes as input the

image to be segmented, the number of image channels, number of classes, mrf parameters similar to ANTS atropos toolbox. However the interesting parameter is the *type of image* parameter which allows us to specify the modality of the image which was not provided in ANTS. This parameter can take T1, T2 and Proton Density modalities as parameters. As observed from figures 6.9 and 6.10 FSL segmentation maps the CSF regions into GM and attributes most of the brain volume into WM. Irrespective of specifying the modality using *type of image* parameter we still observe lower segmentation performance compared to ANTS atropos.

(a) (b) (c) (d)

Figure 6.9: FSL Segmentation(a) T2 weighted input volume (b) GM (c) WM (d) CSF

(a) (b) (c) (d)

Figure 6.10: FSL Segmentation(a) T2 weighted input volume (b) GM (c) WM (d) CSF

FreeSurfer: The freesurfer toolbox also allows us to segment a brain volume and attempts to separate the White Matter from everything else. The *mri_segment* performs the segmentation task and the volume must be normalized such that white matter voxels are 110-valued. As we needed the Gray matter and Cerebrospinal fluid segments as well in our evaluation we did not use the freesurfer segmentation in our evaluation pipeline.

Trainable Weka Segmentation: Trainable Weka Segmentation is a Fiji plugin which can be effectively used to combine machine learning algo-

rithms with manual annotation to produce pixel-based segmentations and is based on works of ARGANDA-CARRERAS et al. (2017). The GUI allows user to load any volume and with MRI experts annotate areas of the brain belonging to different classes. The GUI with manual annotations is shown in Fig 6.11 Once the regions are manually annotated with the help of an

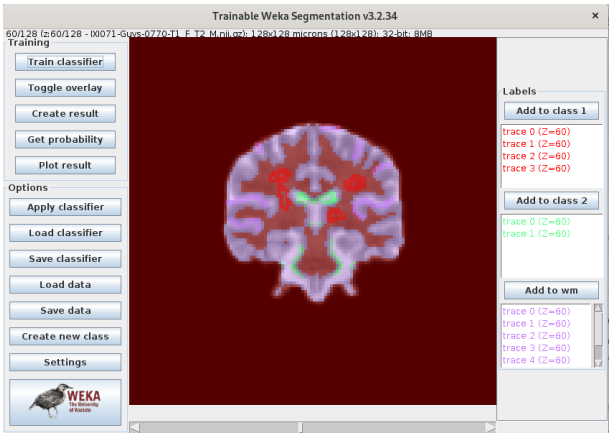


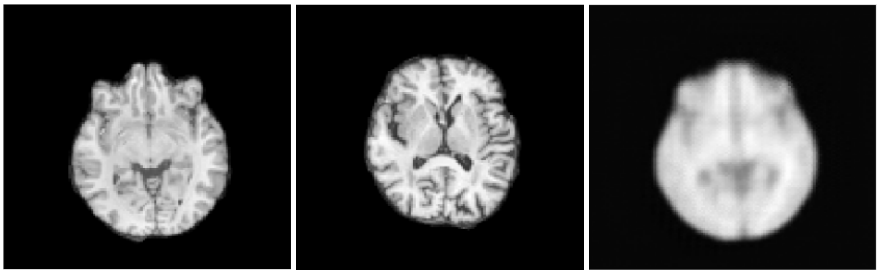
Figure 6.11: Fiji Weka Segmentation GUI with manual annotations

expert we can train a classifier. The default classifier is FastRandomForest which is multi-threaded version of random forest and the user can choose to use any available classifier in the weka. Once the model is trained we can further save the model and load in future and simply use the apply classifier option to segment any image or stack of images that the user wants to segment. However annotation would require expertise in the medical imaging field and also needs to performed on multiple images to ensure maximum accuracy. As manual annotation and experimenting on different set of results would be time consuming and out of the scope of this project, we decided to not include Trainable weka segmenation in our pipeline.

We observed that segmentation performed using various toolboxes performs for T1 weighted images when compared to T2 weighed images. Most of the deep learning models that we explored also used T1 weighted images for training and not for T2 weighted images.As we observed better generalization performance and reasonable segmentation performance over the dataset we explored ANTS atropos segmentation algorithm was used in the evaluation pipeline.

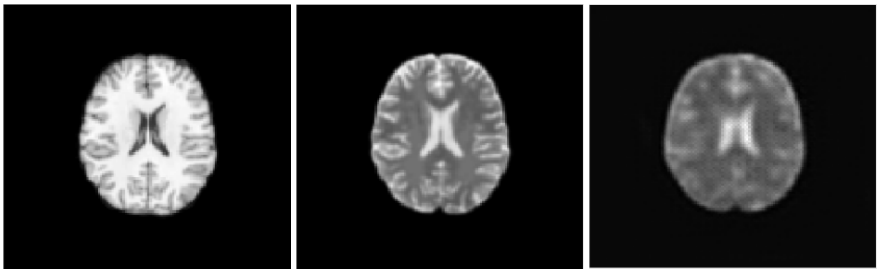
6.2.2 FIRE

For Fire since code wasn't available we have replicated the architecture as defined in the paper WANG et al. (2019). As mentioned in section 4.2.2 we are passing affine registered images to the network and hence we have made use of encoder, decoder and deformable networks. Code is implemented in Pytorch PASZKE et al. (2019). We have used two Adam optimizer KINGMA und BA (2014) to optimize the objective function described in section 3.2. The learning rate considered for deformable network is 0.00005 and for encoder/decoders is 0.0001. We have experimented with Fire code only on the pipeline 1 dataset. We have conducted 5 experiments for Fire WANG et al. (2019). Since the model is large, we were unable to train



(a) (b) (c)

Figure 6.12: Fire intramodal registration(a) Fixed Volume (b) Moving Volume (c) Registered Volume



(a) (b) (c)

Figure 6.13: Fire intermodal registration(a) Fixed Volume (b) Moving Volume (c) Registered Volume

128x128x128 images on 32GB GPU memory. To overcome this, we experimented with 2 options - The first option is downsampling the images to

64x64x64 and passing it to the entire network, and checking the results. The second option is to take a 128x128x128 image and before passing it to the decoder network, interpolate the transformation field to 64x64x64 using spatial transformation network and final synthesized images are 128x128x128. The second option was better as compared with the first one. So we went ahead and ran the rest of the experiments using the same method as option two. Two experiments are for intramodal registration for coregistered T1 and T2 images, and two experiments were conducted for intermodal registration with interpolation and one was for intermodal registration with downsampled images. The batch size is kept as 2 for downsampled case and 3 for interpolated case, iterations are set to 200 and epochs are set as 1000 for all the experiments.

Algorithm	IntraModal			
	SSIM	PCC	Dice Score	MSE
FIRE	0.6132 ±0.0102	0.9724 ±0.0048	0.5732 ±0.0243	0.0027 ±0.0004

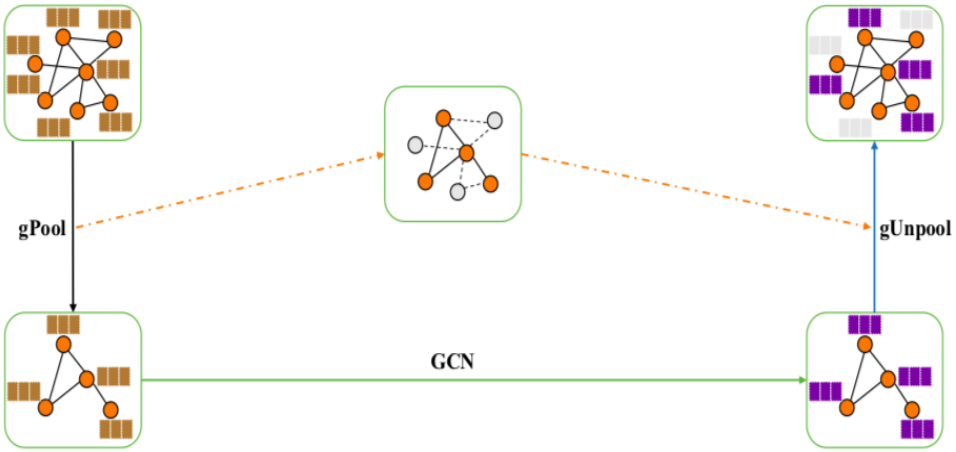
Table 6.1: Intramodal Evaluation for FIRE on dataset preprocessed with pipeline 1

Algorithm	InterModal	
	Pearson Correlation	Dice Score
FIRE	0.8593 ±0.0211	0.5174 ±0.0128

Table 6.2: Intermodal Evaluation for FIRE on dataset preprocessed with pipeline 1

6.2.3 Graph UNet

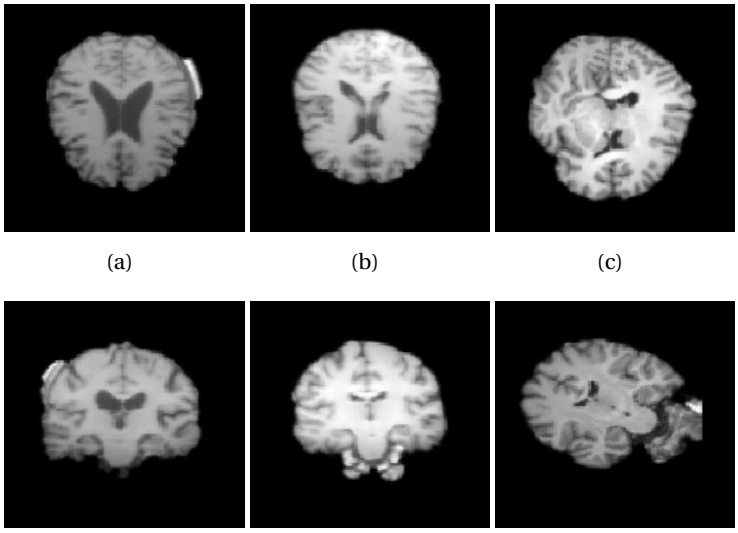
GAO und JI (2019) proposes an end to end graph based UNet where tasks on data such as texts, images can be learned in its natural form without resorting to manual creation of node and edge information. The model also comes up with novel graph pool and graph unpool operations that are analogous to downsample and upsample operations for typical images which can work on non-euclidean domain. Fig 6.14 GAO und JI (2019) provides an overview of Graph UNet. On images for instance, It can easily group pixels related to same object into one cluster to achieve semantic segmentation. It can also establish relationships between those objects in



(a)

Figure 6.14: Graph UNet Overview

the image. This property of graph UNets could be used to find the deformation field given a pair of images. We hypothesize that the Graph UNet



(a) (b) (c)

(d) (e) (f)

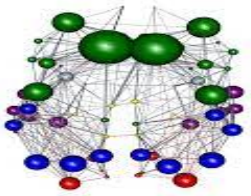
Figure 6.15: Graph UNet Example Output (a) Fixed volume Axial slice (b) Moving volume Axial slice (c) Registered volume Axial slice (d) Fixed volume coronal slice (e) Moving volume coronal slice (f) Registered volume coronal slice

would be able to find such relationship between landmarks of both the images with which it can easily find out the displacement field required to

warp the moving image. We trained the network to perform Intramodal registration using NCC and Smoothness loss for optimization. We trained using the identical dataset with same preprocessing steps for 100 epochs with hyperparameters for NCC being -1.0 and smoothness loss of 0.5. We provide an example visual of the registration below Fig 6.15 where first row corresponds to fixed image , second row corresponds to moving image and finally we have warped image. It shows that that the warping has not happened properly and the warped images are not even corresponds to the same axes. We evaluated the model on the same dataset and obtained low SSIM of 0.7834 ± 0.0093. We believe that the deformation field is not getting optimized correctly due to unbalanced loss function with insufficient constraints, but this hypothesis needs to be tested further to confirm this.

6.2.4 Self Attention Based UNet

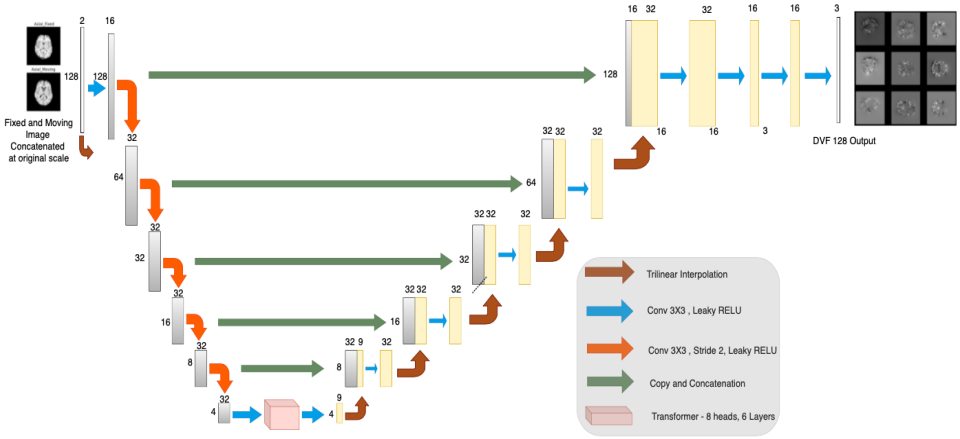
VASWANI et al. (2017) applies self attention mechanism to find out multiple types of relationships in a sequence without recurrence and convolution and successfully applied on various textual translation tasks. We hypothesize that such self attention mechanism based transformer model acts as a fully connected graph which could define various types of relationships between pixels of two different images. The relationships could be to used to find similar anatomical parts of the brain between two scans which could further aid the registration as explained for SCGNet. Another relationship could be finding structural connectives within same scan as shown by Fig 6.16 DELIGIANNI et al. (2019).



(a)

Figure 6.16: Structural Connectivity in Brain

We model the function as a form of UNet that has Transformer Encoder with 8 heads and 6 layers and without Decoder in the latent space as shown in the Fig 6.17. The pair of fixed f and moving m images are concatenated along the gray-scale axis and passed through the network. The encoder distills the information into featuremaps using convolution with LeakyReLU and stride of 2. At the latent space the feature maps are flattened and fed to transformer as a sequence. The output of the transformer is then reshaped back to 3d feature maps which are concatenated with the corresponding encoder to get the precise location information. This is then upsampled to the same spatial dimensions as original image using trilinear upsampling operation.



(a)

Figure 6.17: Self Attention based UNet Architecture

We trained using the identical dataset with same preprocessing steps for 300 epochs, batch size of 2 due to computational constraints presented by transformer, hyperparameters for NCC being -1.0 and smoothness loss of 0.5. We provide an example visual of the registration below Fig 6.18 where first row corresponds to fixed image , second row corresponds to moving image and finally we have warped image. It shows that that the warping has not happened properly and the warped images are not even corresponds to the same axes. We observe better registration performance as compared to Graph UNet as the training regimen was more stable and the overall loss function seems to have been partially balanced. We evaluated the model and obtained SSIM of 0.9619 ± 0.0097. We still observe that the overall shape

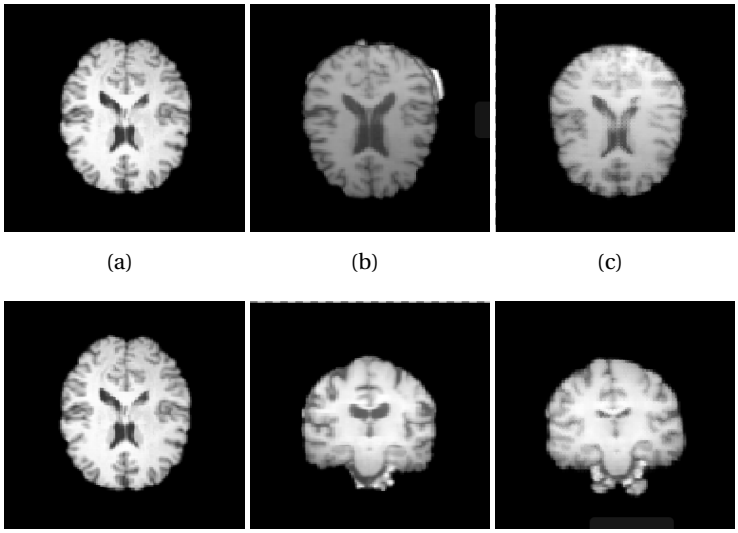


Figure 6.18: Self Attention based UNet Example Output (a) Fixed volume Axial slice (b) Moving volume Axial slice (c) Registered volume Axial slice (d) Fixed volume coronal slice (e) Moving volume coronal slice (f) Registered volume coronal slice

of the CSF, Brain outline in warped image is not fully deformed, but there has been a significant improvement as compared to other methods. We also observe severe pixellations in the warped image due to sharp deformation fields. Even though we regularized the deformation field to reduce such unrealistic deformations, it seems to have not fully smoothened resulting in overall less SSIM scores. We believe that the deformation field must be smoothened out by adding more constraints and reducing the possible deformations available through cycle consistency and anti folding loss, but this hypothesis needs to be tested further to confirm this. There is also an issue of computational efficiency due to large transformer encoders whose heads and layers could be reduced and experimented.

6.3 Evaluation Metrics and Baselines

Ensuring consistent evaluation strategy across our experiments is vital to give us results that are comparable. Evaluation strategy for image registration is a challenging problem as image registration tasks deals with differ-

ent image types, modalities, optimization criteria etc. Similarity measures based on image intensity has always been a popular approach for evaluating image registration and such measures can be broadly be classified into three groups as described by RAZLIGHI et al. (2013): information theoretic measures, measures which considers spatial dependency of neighbouring voxels and statistical measures.

6.3.1 Evaluation Metrics

Structural Similarity Index Measure: The structural similarity index measure (SSIM) is effectively used to evaluate the perceived quality of digital images and the variants of SSIM are extensively compared in RENIEBLAS et al. (2017). The visual image quality is evaluated by measuring the structural similarities between two images where one image is the reference image. Given images X and Y to be compared, and x and y are pairs of square windows of same size of X and Y the SSIM is calculated as

$$SSIM(x,y) = (2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)/(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2) \quad (6.1)$$

where μ_x and μ_y are average pixel values with pixel value standard deviations σ_x and σ_y and covariance σ_{xy}. The SSIM(x,y) value takes a value between 0 and 1 indicating completely different patches and identical patches respectively.

Pearson Correlation Coefficient: Pearson Correlation Coefficient (PCC) is a statistical measure used to understand the departure of two random variables from independence. The p values calculated in this approach assumes that the dataset is normally distributed. The pearson correlation is calculated as

$$PCC = 1/N - 1\sum_{i=1}^{N}((x_i - \mu_x)/\sigma_x)((y_i - \mu_y)/\sigma_y) \quad (6.2)$$

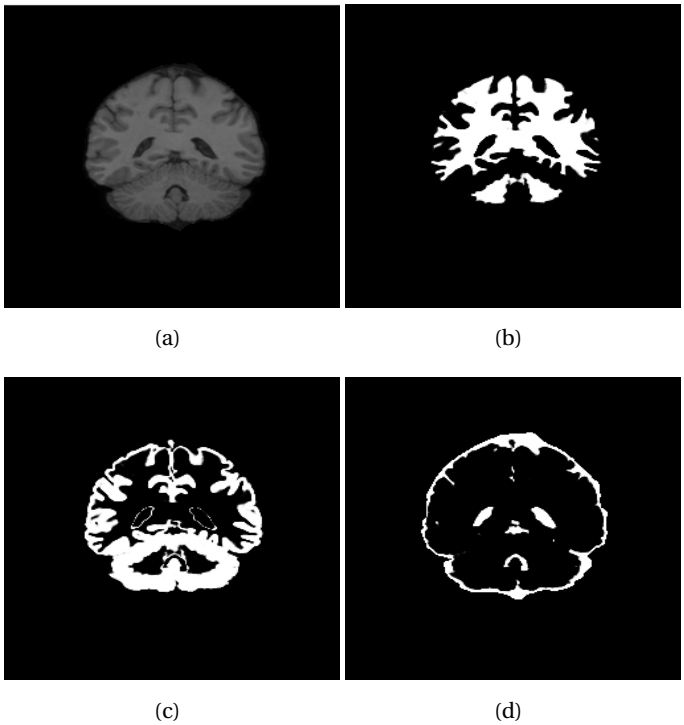
where $_i$ and $_i$ are realizations of random variables X and Y and μ_x and μ_y are the means of X and Y respectively, σ_x and σ_y are standard deviations of X and Y respectively and N is the number of sample pairs.

Dice Score: Dice coefficient (DICE) is a overlap based metric extensively used in evaluating image segmentation tasks and TAHA und HANBURY (2015) discusses multiple metrics to compare segmentation performance.

To evaluate the Dice Score, we first segment the template image and the registered image into four segments namely Cerebrospinal fluid(CSF), White Matter(WM), Gray Matter(GM) and the background. We then compute the dice score for individual segments and average the scores to get averaged dice score for the whole volume. Dice score between template image X and registered image Y is computed as

$$DICE = 2|X \cap Y|/|X| + |Y| \tag{6.3}$$

Segmentation accuracy is a major factor affecting the dice scores as the evaluation relies completely on the segmented results. Popular MR image toolboxes such as ANTS, FSL, Freesurfer, WEKA were used to perform the segmentation tasks. Segmentation quality was further evaluated with different modalities and different parameters and we observed ANTS atropos segmentation to be significantly better than other toolboxes.



(a) (b)

(c) (d)

Figure 6.19: Segmentation result using ANTS atropos (a) Input volume (b) White matter (c) Gray matter (e) Cerebrospinal fluid segmentation

Kullback-Leibler Distance: Kullback-Leibler Distance (KLD) is a evaluation metric used for multimodal brain image registration belonging to

information theory evaluation measures. KLD gives a statistical distance measure between two joint intensity distributions . Given observed and expected intensity distributions as $_o(x,y)$ and $_s(x,y)$ the KLD is computed as

$$KLD = \sum_{x\in\chi}\sum_{y\in\chi} p_o(x,y)log(p_o(x,y)/p_e(x,y)) \tag{6.4}$$

Hausdorff Distance: Spatial distance based metrics are another set of evaluation measures used mainly for image segmentation tasks. The distances are calculated in voxel and the spatial position of voxels are taken into consideration. The Hausdorff distance between two volumes having finite point sets A and B is calculated as

$$HD(A,B) = max(h(A,B),h(B,A)) \tag{6.5}$$

where (A,B) is the directed hausdorff distance given by

$$h(A,B) = max_{a\in A}min_{b\in B}||a-b|| \tag{6.6}$$

where $||a-b||$ can be L1 or L2 norm. The Hausdorff distance is however sensitive to outliers and is not recommended to be used directly. Alternative would be to use Average Hausdorff distance is the Hausdorff distance averaged over all points.

Apart from these quantitative evaluation measures, images were also compared visually and by overlaying template and registered image and highlighting areas which does not match. For intramodal registration we evaluated SSIM, Dice score and the PCC whereas for intermodal we evaluated only the Dice Score and the pearson score as SSIM is an intensity based technique and would not work for evaluation with different modalities.

6.3.2 Evaluation Baselines

In the following section we will describe in detail the different baselines and their configurations used for evaluation.

ANTS: Advanced Normalization Tools(ANTS) is regarded as the state-of-the-art medical image registration and segmentation toolkit and is considered to be a gold standard in comparative studies. ANTS which is non deep learning framework offers several types of linear and non linear transforms and

different choices of optimization metrics. The common optimization metrics includes mutual information, cross-correlation, mattes and demons. For intramodal registrations we used Symmetric Normalization (SyNCC) transform with cross-correlation optmization metric and for intermodal we used the same transform with mutual information as optimization metric.

Voxelmorph: The Voxelmorph implementation was already provided by the author and the hyperparameters were as specified by BALAKRISHNAN et al. (2019). We trained the network with λ value set to 1. The batch size was set to 5 and was trained for a total of 1000 epochs. The learning rate was set to 10^{-4} and ADAM optimizer was used. For intramodal we used Normalized-Cross-correlation(NCC) loss function and for intermodal we used the Normalized mutual information((NMI) as implemented by the authors.

ICNet:

We conducted 6 different experiments for ICNet. 2 out of which were conducted on the pipeline 2 dataset. And the remaining four on pipeline 1 datasets. Experiments conducted on pipeline 2 were - Intramodal registration of 200 T1 images and Intermodal registration of both T1 to T2 and T2 to T1 for the set of 200 images. For pipeline 2, we tried out intramodal registration for both coregistered T1 image pair and T2 image pair, for intermodal, we experimented with T1 affinely registered to T2 and vice versa and recorded the results.

For ICNet we have taken the code that was provided by the author. The code is implemented in Pytorch PASZKE et al. (2019). Adam optimizer KINGMA und BA (2014) is used to optimize the objective function described in section 3.1. Gradients of the loss function are calculated by the backpropagation algorithm. We have taken the same values of hyperparameters as set by the author. For instance, the learning rate value is 0.0005. For pipeline 1 intramodal registration, we have set iterations as 50, batch size as 2, and the total number of epochs as 2000. For pipeline 2 intramodal registration, we have set iterations as 50, total epochs as 1000, and batch size as 2. For intermodal registration with pipeline 1, for both T1 as fixed and T2 as moving, T2 as fixed and T1 as moving, batch size as 2, iterations as 100, and epochs equal to 1000 are considered. Hyperparameter for the inverse consistent constraint is by default set to 0.05, for antifolding as

100000 and smoothness constraint as 0.5. These hyperparameter values are taken from the original paper code.

ADMIR:

As specified in the ADMIR TANG et al. (2020), the number of epochs for the entire registration network was set to 1000. We used Adam with a learning rate of 1e 4 as the optimizer in the network. For every 10 epochs, all network parameters are saved. Every image in the training set is successively selected as the moving image and concatenated with the fixed image as a whole to feed into the deformable convNet in each epoch. The fully warped image generated by the spatial transformer with the help of the final DVF is used to find the total loss.

Even though the ADMIR TANG et al. (2020) claims that the proposed ADMIR model can perform end-to-end registration better than existing techniques, we were unable to replicate the results with the same architecture. We had to make some changes in the design by removing the affine module and altering the loss function.

6.4 Results

The evaluation metrics discussed in section 6.3.1 were effectively used to quantitatively and qualitatively measure the registration performance of the proposed system with respect to non deep learning and deep learning based frameworks. The evaluations were performed with a standardized dataset having the preprocessing pipelines as discussed in section 6.1.2. This ensures that the model performances are comparable and if they are reproducible.

6.4.1 Direct Optimization

We observe from visuals that the registration happens quite well using gradient descent and use the SSIM WANG et al. (2004), Pearson correlation coefficient metric FREEDMAN et al. for quantitative evaluation. We observe that ADAM KINGMA und BA (2014) optimizer has the highest metric at 1500 epoch with 0.97622 ± 0.0081 when compared against ADAMW, RMSPROP and SGD RUDER (2016).

Optimizer	IntraModal			
	SSIM	Pearson Correlation	Dice Score	MSE
ADAM	0.9762 ± 0.0081	0.9886 ± 0.0041	0.7824 ± 0.0346	0.0010 ±0.0004
ADAMW	0.9753 ± 0.0082	0.9868 ± 0.0051	0.7836 ±0.0355	0.0010 ±0.0004
RMSPROP	0.9729±0.0076	0.9912 ± 0.0017	0.7926 ±0.0195	0.0007 ±0.0001
SGD	0.9315 ± 0.0088	0.9584 ± 0.0073	0.6044 ±0.0229	0.0036 ±0.0005

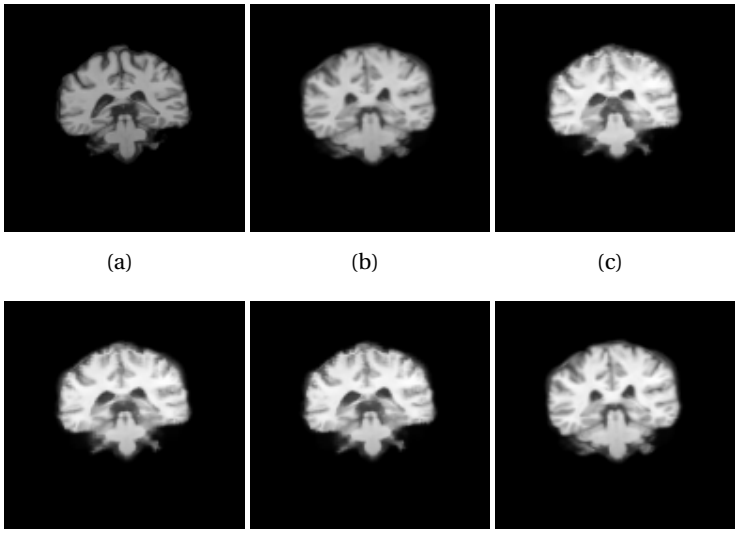
Table 6.3: Intramodal Evaluation for Direct Optimization

Epochs	IntraModal - RMSPROP			
	SSIM	Pearson Correlation	Dice Score	MSE
1000	0.9713 ± 0.0042	0.9831 ± 0.0023	0.7901 ± 0.0293	0.0010 ± 0.0002
1500	0.9729 ± 0.0076	0.9912 ± 0.0017	0.7926 ± 0.0195	0.0007 ± 0.0001
2000	0.9725 ± 0.0063	0.9910 ± 0.0025	0.7919 ± 0.0271	0.0008 ± 0.0002
2500	0.9722 ± 0.0069	0.9871 ± 0.0022	0.7881 ± 0.0313	0.0009 ± 0.0002

Table 6.4: Intramodal Direct Optimization Evaluation by Epochs

The optimization takes 3.47 minutes for deformable registration on 12 GB GPU. As we run for more epochs, the deformation field could become too smoothened or unrealistic, Hence we observe better registration for epochs 1000 and 1500 6.4. Alternatively, It could also mean that more constraints should be added to better update the deformation field during training that has to be explored in the future work. We applied the direct optimization on the same images used for evaluation for MSCGUNet and Voxelmorph. Intramodal and Intermodal results have been reported in the Table 6.3 and Table 6.5.

The direct optimization technique seems to perform better than ANTS SyN for the given dataset and other parameters for Intramodal registration. As observed in fig 6.20 we see registration happening for intramodal T1 weighted volumes using ADAM, ADAMW, RMSPROP whereas SGD optimizer fails to perform any registration. Intermodal registration the registration performance is poorer than ANTS SyN when comparing with the Person Correlation metric and the KL Distance. However direct optimization techniques performs comparable to ANTS SyN and deep learning based techniques when evaluated with respect to the Dice score. RMSPROP is observed to perform better than ADAM and ADAMW optimizers. We hypothesize that the performance depends on the construction of the loss function which could affect the loss surface and thereby optimizer per-



(a) (b) (c)

(d) (e) (f)

Figure 6.20: Intramodal T1 weighted registration using direct optimization, Sagittal view (a) Fixed volume (b) Moving Volume (c) Adam Optimizer (d) AdamW Optimizer (e) RMSPROP Optimizer (f) SGD Optimizer

Optimizer	InterModal		
	Pearson Correlation	Dice Score	KL Distance
ADAM	0.7695 ±0.0055	0.5908 ±0.0498	0.8780 ±0.4452
ADAMW	0.8234 ±0.0344	0.6108 ±0.0524	0.8577 ±0.2838
RMSPROP	0.8222 ±0.0345	0.6114 ±0.0533	0.8556 ±0.4435
SGD	0.8165 ±0.0039	0.6081 ±0.0722	0.8656 ±0.4435

Table 6.5: Intermodal Evaluation for Direct Optimization

formance. Intermodal registration may require more than (Normalized Mutual Information) NMI loss to be optimized better. Using KL Divergence or harmonized images may help improve the performance of the optimization.

6.4.2 Deep Learning Networks

The registration performance is evaluated for the baselines as shown in table 6.6 for the dataset preprocessed using pipeline 1. We compare deep learning based image registration methods against the gold standard ANTS

SyN registration algorithm. The deep learning methods with the exception

Algorithm	SSIM	PCC	IntraModal Dice Score	MSE	KL Distance
Ants(SyN)	0.9611 ±0.0065	0.9792 ±0.0042	0.7166 ±0.0245	0.0017 ±0.0003	0.0734 ±0.0825
ADMIR	0.9775 ±0.0064	0.9894 ±0.0031	0.7860 ±0.0310	0.0009 ±0.0002	0.0551 ±0.0250
ICNet	0.9271 ±0.0129	0.9542 ±0.0110	0.5996 ±0.0350	0.0040 ±0.0010	0.1294 ±0.2857
VoxelMorph	0.9711 ±0.0052	0.9878 ±0.0029	0.7747 ±0.0260	0.0010 ±0.0002	0.0535 ±0.0246
MSCGUNet	0.9808 ±0.0050	0.9916 ±0.0020	0.8013 ±0.0243	0.0007 ±0.0001	0.0506 ±0.0222

Table 6.6: Intramodal Evaluation on dataset preprocessed with pipeline 1

of ICNet consistently seems to outperform ANTS SyN registration algorithm when evaluated on all the metrics. Although improvement in SSIM scores for the deep learning based algorithms over ANTS SyN was only marginal, Dice score showed significant difference of 0.0694 when compared with ADMIR.

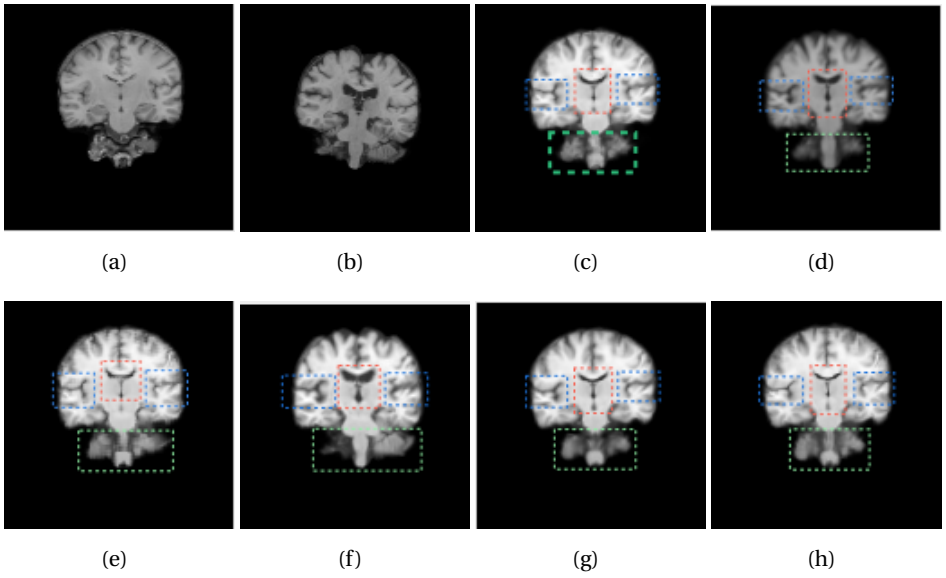


Figure 6.21: Intramodal T1 weighted registration Coronal view (a) Fixed volume (b) Moving Volume (c) ADMIR registered volume (d) ANTS SyN registered volume (e) Direct Optimization registered volume (f) ICNet registered volume (g) Voxelmoprh registered (h) MSCGUNet registered volume

From our comparative study spanning across deep learning registration networks including ADMIR, ICNet, FIRE and Voxelmorph we found ADMIR to be the best performing network for intramodal registration and was

considered to be the deep learning image registration benchmark. The MSCGUNet was able to perform significantly better this benchmark when evaluated with 2 tailed T-test with significance level of 0.05. The t-value was found to be -4.18385 and the p-value was found to be 0.000063 when evaluated for SSIM. We observed similar results when evaluated for dice score with the t-value being -3.82488 and the p-value as .00023 with significant result at $p < .05$. When performing statistical test for pearson correlation coefficent we observed the t-value -3.99969 and the p-value 0.000123 and the result is significant at $p < 0.05$. The statstical test performed over mean sqaured error metric also gave also us statistically significant results with the t-value 3.46466 and the p-value 0.000789 at $p < 0.05$. The figures 6.21 and 6.22 provides a visual comparison with annotations to the areas of the brain where the MSCGUNet is able to better map to the template image.

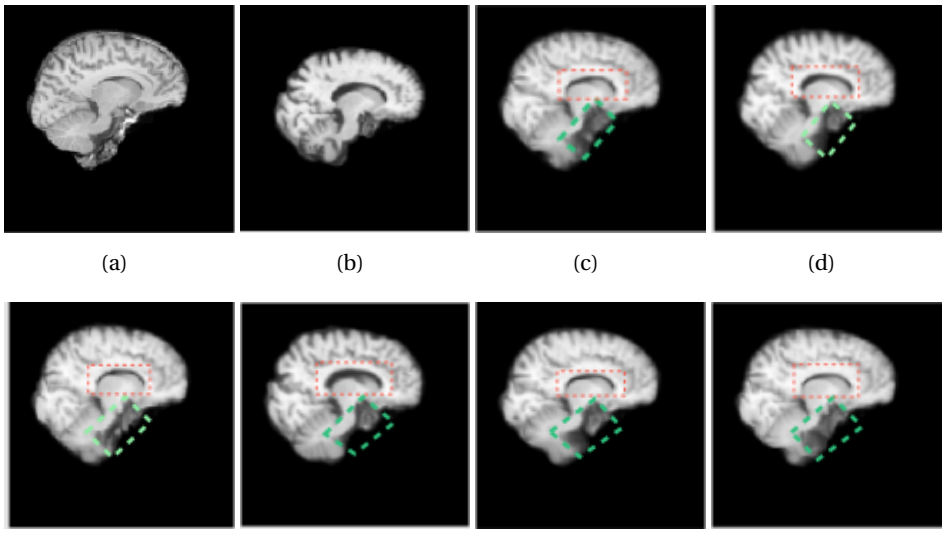


Figure 6.22: Intramodal T1 weighted registration Sagittal view (a) Fixed volume (b) Moving Volume (c) ADMIR registered volume (d) ANTS SyN registered volume (e) Direct Optimization registered volume (f) ICNet registered volume (g) Voxelmoprh registered (h) MSCGUNet registered volume

We futher evaluated the performance of these networks on data preprocessed with pipeline 2 to compare the generalization performance of these networks on data which is completely unseen and is not used for training. As observed from table 6.7 the MSCGUNet showed significant improve-

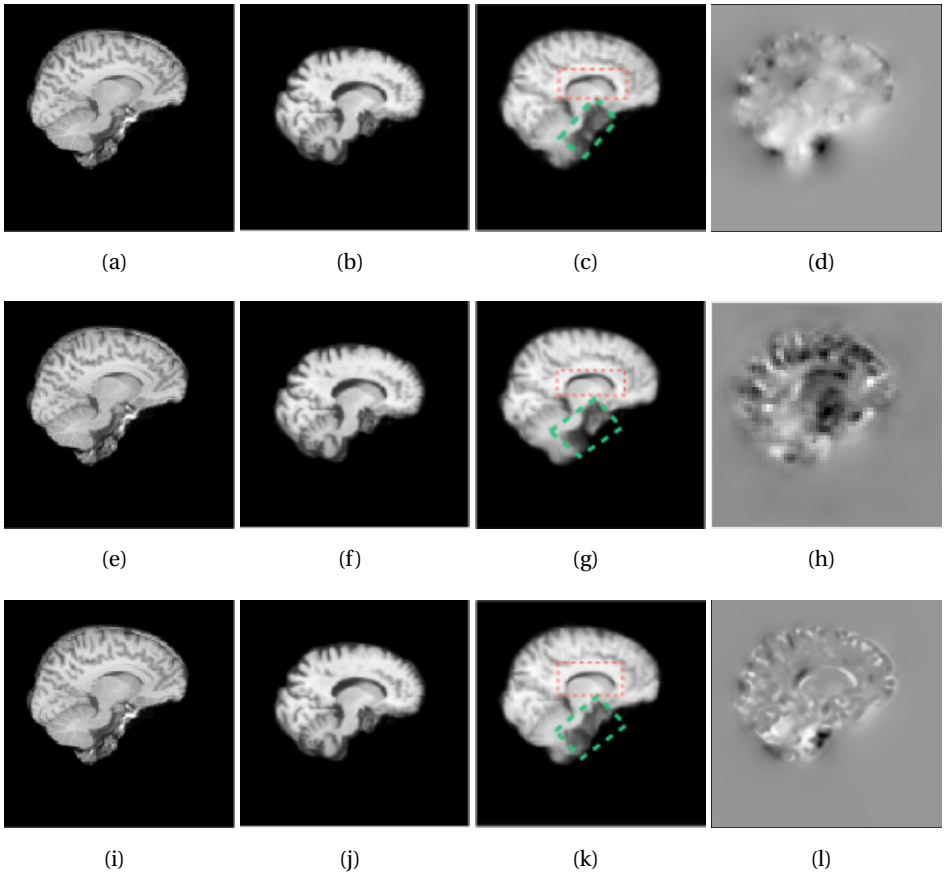


Figure 6.23: Intramodal T1 weighted registration Sagittal view (a) Fixed volume (b) Moving Volume (c) ADMIR registered volume (d) ADMIR DVF (e) Fixed Volume (f) Moving Volume (g) Voxelmorph registered (h) Voxelmorph DVF (i) Fixed Volume (j) Moving Volume (k) MSCGUNet registered (l) MSCGUNet DVF

Algorithm	IntraModal			
	SSIM	PCC	Dice Score	MSE
Ants(SyN)	0.8734 ±0.0326	0.9702 ±0.0189	0.7103 ±0.0616	0.0051 ±0.0037
ICNet	0.7879 ±0.0576	0.9371 ±0.0243	0.5598 ±0.1181	0.0098 ±0.0062
VoxelMorph	0.8851 ±0.0260	0.9724 ±0.0130	0.6999 ±0.0580	0.0048 ±0.0030
MSCGUNet	0.9025 ±0.0236	0.9761 ±0.0116	0.7445 ±0.0447	0.0041 ±0.0037

Table 6.7: Intramodal Evaluation on dataset preprocessed with pipeline 2

ment in SSIM and Dice scores when compared to gold standards ANTS SyN and deep learning networks such ass ICNet and Voxelmorph. Voxelmorph was found to be under-performing with respect to the Dice score when

compared with ANTS and the mean squared error was found to be the lowest for the MSCGUNet with significant improvements over ANTS and Voxelmorph. These results strongly suggests that the MSCGUNet is also able to generalize much better than the other deep learning networks and provides better performance even with respect to the ANTS SyN registration algorithm.

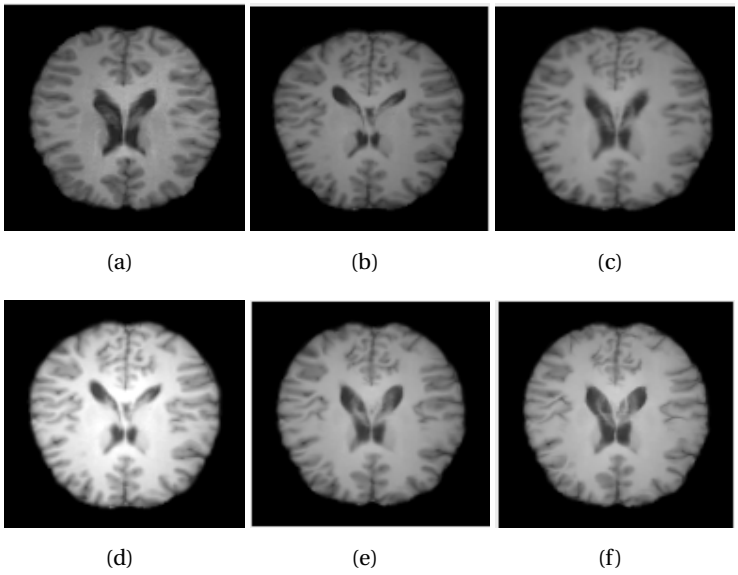


Figure 6.24: Intramodal T1 weighted registration with data preprocessed using pipeline 2, Axial view (a) Fixed volume (b) Moving Volume (c) ANTS SyN registered volume (d) ICNet registered volume (e) Voxelmoprh registered (f) MSCGUNet registered volume

When evaluated with dataset preprocessed using pipeline 2, we observe comparable performance for ANTS SyN and Voxelmorph with better SSIM, PCC and MSE scores for Voxelmorph and ANTS outperforming Voxelmorph with respect to the DICE score metric. ICNet is consistently underperforming when compared to non deep learning and deep learning techniques and this reflects in all the metrics. As Voxelmorph has marginally better performance over ANTS SyN over most of the metrics we use Voxelmorph in the statistical tests against MSCGUNet. Although the SSIM scores for dataset preprocessed with pipeline 2 was found not to be statistically significant we observe that the PCC has a t-value of -2.02001 and p-value of 0.047212. The result is significant at $p < 0.05$ when evaluated for a 2 tailed Test with significance level of 0.05.

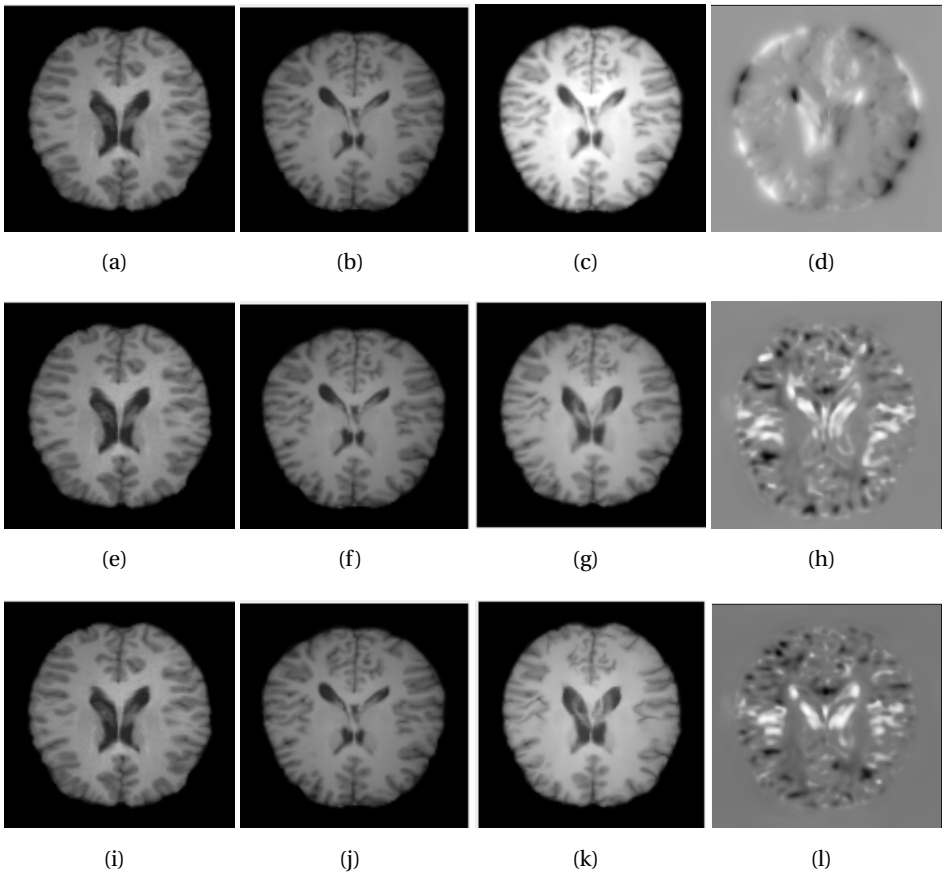


Figure 6.25: Intramodal T1 weighted registration on pipeline 2 Axial view (a) Fixed volume (b) Moving Volume (c) ICNet registered volume (d) ICNet DVF (e) Fixed Volume (f) Moving Volume (g) Voxelmorph registered (h) Voxelmorph DVF (i) Fixed Volume (j) Moving Volume (k) MSCGUNet registered (l) MSCGUNet DVF

Algorithm	InterModal		
	Pearson Correlation	Dice Score	KL Distance
Ants(SyN)	0.8555 ±0.0263	0.5243 ±0.1318	0.8690 ±0.4904
ICNet	0.7490 ±0.0640	0.5921 ±0.0528	0.9398 ±0.4124
VoxelMorph	0.8525 ±0.0245	0.6071 ±0.0510	0.8290 ±0.4575
MSCGUNet	0.8539 ±0.0245	0.6211 ±0.0309	0.8338 ±0.4587

Table 6.8: Intermodal Evaluation

Experiments were conducted registering volumes of different modality namely T1 weighed and T2 weighted volumes. The intermodal registra-

tion was performed keeping T1 weighted volumes as fixed volumes and T2 weighted volumes as the moving volumes. SSIM, Mean Squared error (MSE) evaluation metrics were not used for evaluating intermodal registrations as SSIM and MSE are intensity based metrics and will fail to provide satisfactory results for our evaluation. As described by RAZLIGHI et al. (2013) we have used Kullback-Leibler Distance (KLD) which belongs to the class of information theoretic measures and Pearson Correlation coefficient which is a statistical evaluation measure as described in detail in section 6.3.1.

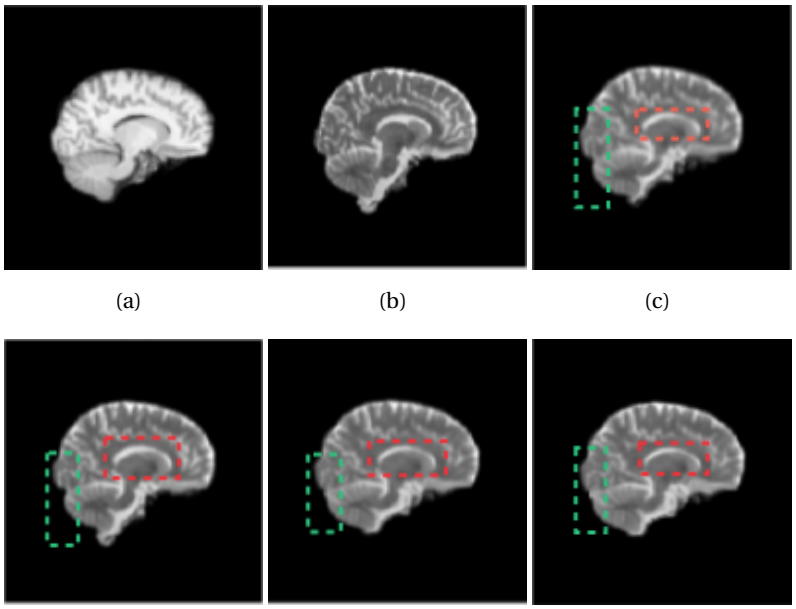


Figure 6.26: Intermodal registration with data preprocessed using pipeline 2, Axial view (a) Fixed volume (b) Moving Volume (c) ANTS SyN registered volume (d) ICNet registered volume (e) Voxelmoprh registered (f) MSCGUNet registered volume

As observed from table 6.8 the MSCGUNet performs better than ANTS SyN, ICNet and Voxelmorph considering the Dice score. However due to inaccuracies in the segmentation algorithm the Dice score evaluation can only be taken as secondary evaluation metric. The experiments conducted comparing various toolboxes for segmentation was already described in detail in section 6.2.1. Although we observed that the MSCGUNet performs better than the other baselines, the improvement was not statistically significant.

We compare the volumes visually as shown in figure 6.26. As observed in the fig 6.26 the CSF segments are mapped slightly better in the MSCGUNet compared to other deep learning networks.AS the volumes are affine registered, and represent just a different modality of the same brain volume, the differences are subtle. However as annotated with the green box, we can observe that the MSCGUNet is able to map the overall brain structure much better than ANTS SyN and ICNet registration network. Evaluation metrics which can precisely quantify the registration performance for intermodal registration is an issue that needs to be addressed and can be a part of future research.

6.4.3 Ablation Study of MSCGUNet

Algorithms			Evaluation Metric	
SCG	MSS	IC	SSIM	DICE
x	x	x	0.9713 ±0.0048	0.7738 ±0.0243
x	x	✓	0.9767 ±0.0047	0.7866 ±0.0228
x	✓	x	0.9782 ±0.0053	0.7902 ±0.0211
✓	x	x	0.9757 ±0.0051	0.7760 ±0.0232
x	✓	✓	0.9802 ±0.0047	0.7980 ±0.0233
✓	✓	✓	0.9822 ±0.0047	0.8070 ±0.0234

Table 6.9: Intramodal registration performance evaluation for Test set preprocessed with pipeline 2 T1 weighted volumes for different network configurations

As described in detail in section 5.2, MSCGUNet extends the UNET architecture by adding the concepts of encoding global dependencies employing self-constructing graph network(SCG), multi-scale supervision for faster convergence(MSS) and cycle consistency(IC) which encourages pair of images are symmetrically deformed towards one another. We observe better registration performances which are evaluated in depth in chapter 6 and in the following section we try to quantify the improvements and attribute the contributions of individual concepts. For future references we name the network without any of the extended concepts as the base network and the experiments following combines this base network with one or more of the concepts such as SCG, MSS and IC. To this end, we run the training and testing loop for these combinations and perform evaluations on a fixed test set. The experiments were conducted on T1 weighted volumes and for

intramodal registration and future work can extend these experiments for intermodal registration as well.

Algorithms			Evaluation Metric	
SCG	MSS	IC	SSIM	DICE
x	x	x	0.8794 ±0.0495	0.6768 ±0.1305
✓	x	x	0.8875 ±0.0504	0.6905 ±0.1335

Table 6.10: Intramodal registration performance evaluation for Test set preprocessed with pipeline 2 T1 weighted volumes with and without SCG in latent

As observed from the SSIM and DICE scores evaluated over 50 T1 weighted volumes, we infer that the best performance is observed with all the extensions on the base network. The SCG component seems to only marginally improve the scores when compared to the base network, however improves the generalization capability when evaluated on pipeline 2 dataset as observed in table 6.10. Configurations involving MSS and IC significantly improves the registration performance as observed in table 6.9 and combining all the three components we get the best scores of 0.9822±0.0047 for SSIM and 0.8070±0.0234 for Dice score. Hence MSCGUNet implements all the three components and their corresponding loss functions in the training loop.

7

Conclusions and Future Work

7.1 Discussion and Conclusion

The proposed method with Multi-scale SCG UNet performs significantly better than state-of-the-art ANTS SyN registration algorithm and marginally better than deep learning networks such as ADMIR and Voxelmorph. The proposed network tries to explicitly encode global dependencies and semantics, i.e structure and overall view of the anatomy in the supplied image, by incorporating self-constructing graph network in the latent space of a UNet model. The multi-scale architecture helps in tracking larger deformations and the inverse consistency constraint ensures the deformations are consistent. We hypothesise that the generalization ability of our network is significantly better than other deep learning networks such as ADMIR, ICNet and Voxelmorph and quantitatively substantiate this observation as shown in table 6.7 and this is attributed to the self-constructing graph network which semantically encodes the overall anatomical view of supplied volume. The multi-scale architecture with different receptive field and richer feature information was found to improve the overall registration performance and make the network more robust quantified using SSIM and DICE.

As visually compared in 6.22 the proposed network better maps the CSF segment of the brain when compared to all other baselines. The grey matter segments are mapped more or less in similar manner for all baselines we evaluated. The proposed network, ADMIR and Voxelmorph was able to replicate the shape of the brain volume in a much precise manner as compared to ANTS SyN and ICNet. The proposed network was also able to replicate the lobes significantly better as annotated by green rectangle in 6.22 when compared to all the other baselines. We observed that the

proposed network was able to generate state-of-the-art registration scores with small dataset of 200 MR volumes and also on unseen data with a completely different preprocessing pipeline.

Experiments were conducted evaluating major aspects of the network as observed in table as 6.9. The network showed significantly improved performance for intramodal registration. Evaluations were also performed with different MR image modalities. However we observed comparable performance to that of state-of-the-art ANTS SyN registration algorithm and Voxelmorph deep learning network. We believe the evaluation metrics Pearson Correlation and Dice Score is not completely accurate for intermodal registration due to the segmentation inaccuracies as discussed in section 6.2.1.

The direct optimization technique where the deformation field is directly optimized using gradient descent gave surprisingly significant results. We observed that these techniques without any training and model parameters was able to perform comparable to ANTS SyN and other deep learning baselines. We also observe that the RMSPROP optimizer gave better results for intramodal and intermodal registrations and improvements in loss function implementation could further improve the results.

7.2 Future Work

The current research was focused on both deep learning and direct optimization. We list some ideas that could be tested in detail in the future.

1. Improve the self attention based deep learning model by designing loss function that targets transformer loss just like SCG loss. Try using only transformer encoder layers to increase computational efficiency.

2. Training and Evaluating intermodal images are tricky due to failure of segmentation of warped image. Having a better metric to measure intermodal image similarity would make training and evaluation much easier.

3. Intermodal registration could be converted to Intramodal registration by converting the modality of the moving image to that of fixed

image ZUO et al. (2021) and then registering it using the proposed Intramodal network and trained end to end.

4. Proposed model could be improved further by integrating an affine network on top of deformable network and training them end to end. The deformation field could be optimized better through addition of other losses such as anti-folding loss.

5. Direct Optimization gave surprisingly good results. The performance of direct optimization depends mostly on the loss function and optimizer used. The loss surface would impact the optimizer which would have significant effects on registration. Hence a better loss function could be constructed to serve intermodal and intramodal scenarios.

6. Speed of the direct optimization could be improved by better application of GPU programming techniques.

A

Contribution Sheet

Topic	Execution Task	Person
Literature Research	Systematic Literature Review	Istiyak
Literature Research	Comparative Study of DLIR	Istiyak
Report Writing	Introduction, Motivation	Istiyak
Report Writing	Structure, Related Work	Istiyak
Preprocessing Pipeline	FSL, Freesurfer	Nandish
Evaluation	FSL, ADMIR	Nandish
Literature Research	FSL, ADMIR	Nandish
Report Writing	FSL, ADMIR	Nandish
Model Development and Training	ADMIR	Nandish
Literature Research	ICNet, FIRE, ADMIR	Himanshi
Model Development and Training	ICNet, FIRE	Himanshi
Model Development and Training	ADMIR(Deformable network)	Himanshi
Report Writing	ICNet, FIRE	Himanshi
Literature Research	Voxelmorph	Suraj
Literature Research	Graph UNet, Self Attention	Suraj
Literature Research	SCGNet, Direct Optimization	Suraj
Model Development and Training	Voxelmorph	Suraj
Model Development and Training	Graph UNet, Self Attention	Suraj
Model Development and Training	Direct Optimization	Suraj
Model Development	ADMIR	Suraj
Model Development	Proposed Network (SCGNet, Multiscale)	Suraj
Report Writing	Background, Voxelmorph	Suraj
Report Writing	Proposed Method, Direct Optimization	Suraj
Literature Research	ANTS, GNN, Freesurfer	Steve
Literature Research	SCGNet, Evaluation	Steve
Preprocessing Pipeline	ANTS, Freesurfer	Steve
Model Development	Proposed Network (Cycle Consistency)	Steve
Model Development	Proposed Network, Ablation Study	Steve
Model Training	Proposed Network, Ablation Study	Steve
Evaluation Pipeline Development	SSIM, DICE, PCC, KLD, MSE	Steve
Evaluation	ANTS, FIRE, Proposed Network, Voxelmorph, ICNet	Steve
Report Writing	Evaluation	Steve
Report Writing	Results, Conclusion	Steve

B

Abbreviations and Notations

Dataset, Algorithms and Metrics acronyms

Acronym	Meaning
DL	deep learning
GAN	Generative Adversarial Networks
RQ	Research Question
ReLU	Rectified Linear Unit
CNN	convolutional neural network
GNN	Graph Neural Networks
SCG	Self Constructing Graph
DVF	displacement vector field
FCN	fully connected layers
GPU	Graphics processing unit
SGLD	Stochastic Gradient Langevin Dynamics
MRI	Magnetic resonance imaging
VAE	Variational auto-encoder
NCC	Normalized cross-correlation
NMI	Normalized Mutual Information
ANTS	Advanced Normalization Tools
FSL	FMRIB Software Library
SNR	Signal-to-noise ratio
WM	White matter
GM	Gray matter
CSF	Cerebrospinal fluid
FMM	finite mixture model
EM	Expectation-Maximization
SSIM	Structural Similarity Index Measure
PCC	Pearson Correlation Coefficient
KLD	Kullback-Leibler Distance

List of Figures

D

List of Tables

E

Bibliography

[ARGANDA-CARRERAS et al. 2017] I. Arganda-Carreras, V. Kaynig, C. Rueden, K. W. Eliceiri, J. Schindelin, A. Cardona und H. S. Seung. **Trainable Weka Segmentation: A machine learning tool for microscopy pixel classification**. Bioinformatics, Vol. 33:2424–2426, 2017.

[BALAKRISHNAN et al. 2019] G. Balakrishnan, A. Zhao, M. R. Sabuncu, J. Guttag und A. V. Dalca. **VoxelMorph: A Learning Framework for Deformable Medical Image Registration**. IEEE Transactions on Medical Imaging, Vol. 38(8):1788–1800, 2019. ArXiv: 1809.05231.

[BATTAGLIA et al. 2018] P. W. Battaglia, J. B. Hamrick, V. Bapst, A. Sanchez-Gonzalez, V. F. Zambaldi, M. Malinowski, A. Tacchetti, D. Raposo, A. Santoro, R. Faulkner, Çaglar Gülçehre, H. F. Song, A. J. Ballard, J. Gilmer, G. E. Dahl, A. Vaswani, K. R. Allen, C. Nash, V. Langston, C. Dyer, N. M. O. Heess, D. Wierstra, P. Kohli, M. M. Botvinick, O. Vinyals, Y. Li und R. Pascanu. **Relational inductive biases, deep learning, and graph networks**. ArXiv, Vol. abs/1806.01261, 2018.

[BOVEIRI et al. 2020] H. R. Boveiri, R. Khayami, R. Javidan und A. Mehdizadeh. **Medical image registration using deep neural networks: A comprehensive review**. Computers & Electrical Engineering, Vol. 87:106767, 2020.

[BRIAN B. AVANTS] G. S. a. J. C. G. Brian B. Avants, Nicholas J. Tustison. **ANTS: Open-Source Tools for Normalization And Neuroanatomy**. ????

[BRONSTEIN et al. 2017] M. M. Bronstein, J. Bruna, Y. LeCun, A. D. Szlam und P. Vandergheynst. **Geometric Deep Learning: Going beyond Euclidean data**. IEEE Signal Processing Magazine, Vol. 34:18–42, 2017.

[BY BRAIN-DEVELOPMENT.ORG]

[CHATTERJEE et al. 2020] S. Chatterjee, K. Prabhu, M. V. Pattadkal, G. Bortsova, F. Dubost, H. Mattern, M. de Bruijne, O. Speck und A. Nürnberger. **DS6: Deformation-aware learning for small vessel segmentation with small, imperfectly labeled dataset**. ArXiv, Vol. abs/2006.10802, 2020.

[DALCA et al. 2019] A. V. Dalca, G. Balakrishnan, J. Guttag und M. R. Sabuncu. **Unsupervised Learning of Probabilistic Diffeomorphic Registration for Images and Surfaces**. Medical Image Analysis, Vol. 57:226–236, 2019. ArXiv: 1903.03545.

[DALE et al.] **Cortical Surface-Based Analysis I. Segmentation and Surface Reconstruction**.

[DE VOS et al. 2018] B. D. de Vos, F. F. Berendsen, M. A. Viergever, H. Sokooti, M. Staring und I. Isgum. **A Deep Learning Framework for Unsupervised Affine and Deformable Image Registration**. arXiv:1809.06130 [cs], 2018. ArXiv: 1809.06130.

[DELIGIANNI et al. 2019] F. Deligianni, J. D. Clayden und G.-Z. Yang. **Comparison of Brain Networks Based on Predictive Models of Connectivity**. 2019 IEEE 19th International Conference on Bioinformatics and Bioengineering (BIBE), pp. 115–121, 2019.

[DESPOTOVIĆ et al. 2015] I. Despotović, B. Goossens und W. Philips. **MRI segmentation of the human brain: Challenges, methods, and applications**. Computational and Mathematical Methods in Medicine, Vol. 2015, 2015.

[DEWEY et al. 2019] B. E. Dewey, C. Zhao, J. C. Reinhold, A. Carass, K. C. Fitzgerald, E. S. Sotirchos, S. Saidha, J. Oh, D. L. Pham, P. A. Calabresi, P. C. M. van Zijl und J. Prince. **DeepHarmony: A deep learning approach to contrast harmonization across scanner changes.** Magnetic resonance imaging, 2019.

[DICE 1945] L. R. Dice. **Measures of the Amount of Ecologic Association Between Species**. Ecology, Vol. 26:297–302, 1945.

[FISCHL 2012] B. R. Fischl. **FreeSurfer**. NeuroImage, Vol. 62:774–781, 2012.

[FREEDMAN et al.] D. Freedman, R. Pisani und R. Purves. **Statistics (international student edition)**. ????

[FUSE et al. 2000] T. Fuse, E. Shimizu und M. Tsutsumi. 2000, **A Comparative Study on Gradient-Based Approaches for Optical Flow Estimation**.

[GAO und JI 2019] H. Gao und S. Ji. **Graph U-Nets**. IEEE transactions on pattern analysis and machine intelligence, Vol. PP, 2019.

[GERKE et al. 2014] M. Gerke, F. Rottensteiner, J. Wegner und G. Sohn. 2014, **ISPRS Semantic Labeling Contest**.

[GRIGORESCU et al. 2020] I. Grigorescu, A. Uus, D. Christiaens, L. Cordero-Grande, J. Hutter, A. D. Edwards, J. V. Hajnal, M. Modat und M. Deprez. **Diffusion tensor driven image registration: a deep learning approach**. arXiv:2005.06926 [eess], 2020. ArXiv: 2005.06926.

[HANSEN und HEINRICH 2020] L. Hansen und M. P. Heinrich. **Tackling the Problem of Large Deformations in Deep Learning Based Medical Image Registration Using Displacement Embeddings**. arXiv:2005.13338 [cs, eess], 2020. ArXiv: 2005.13338.

[HARA et al. 2018] K. Hara, H. Kataoka und Y. Satoh. **Can Spatiotemporal 3D CNNs Retrace the History of 2D CNNs and ImageNet?** In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 6546–6555.

[HORN und SCHUNCK 1981] B. K. P. Horn und B. G. Schunck. **Determining Optical Flow**. Artif. Intell., Vol. 17:185–203, 1981.

[JADERBERG et al. 2015] M. Jaderberg, K. Simonyan, A. Zisserman und K. Kavukcuoglu. **Spatial Transformer Networks**. 2015, In: NIPS.

[JADERBERG et al. 2016] M. Jaderberg, K. Simonyan, A. Zisserman und K. Kavukcuoglu. **Spatial Transformer Networks**. arXiv:1506.02025 [cs], 2016. ArXiv: 1506.02025.

[KHAWALED und FREIMAN 2020] S. Khawaled und M. Freiman. **Unsupervised Deep-Learning Based Deformable Image Registration: A Bayesian Framework**. arXiv:2008.03949 [cs], 2020. ArXiv: 2008.03949.

[KIM et al. 2020] B. Kim, D. H. Kim, S. H. Park, J. Kim, J.-G. Lee und J. C. Ye. **CycleMorph: Cycle Consistent Unsupervised Deformable Image Registration**. arXiv:2008.05772 [cs, eess, stat], 2020. ArXiv: 2008.05772.

[KINGMA und BA 2014] D. P. Kingma und J. Ba. **Adam: A Method for Stochastic Optimization**. 2014.

[KINGMA und WELLING 2019] D. P. Kingma und M. Welling. **An Introduction to Variational Autoencoders**. Found. Trends Mach. Learn., Vol. 12:307–392, 2019.

[KIPF und WELLING 2017] T. Kipf und M. Welling. **Semi-Supervised Classification with Graph Convolutional Networks**. ArXiv, Vol. abs/1609.02907, 2017.

[LIN et al. 2018] C.-H. Lin, E. Yumer, O. Wang, E. Shechtman und S. Lucey. **ST-GAN: Spatial Transformer Generative Adversarial Networks for Image Compositing**. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9455–9464. 2018, IEEE, Salt Lake City, UT.

[LIU et al. 2019] Q. Liu, M. C. Kampffmeyer, R. Jenssen und A.-B. Salberg. **Dense Dilated Convolutions Merging Network for Semantic Mapping of Remote Sensing Images**. 2019 Joint Urban Remote Sensing Event (JURSE), pp. 1–4, 2019.

[LIU et al. 2020] Q. Liu, M. C. Kampffmeyer, R. Jenssen und A.-B. Salberg. **Self-Constructing Graph Convolutional Networks for Semantic Labeling**. IGARSS 2020 - 2020 IEEE International Geoscience and Remote Sensing Symposium, pp. 1801–1804, 2020.

[MAHAPATRA 2019] D. Mahapatra. **GAN Based Medical Image Registration**. arXiv:1805.02369 [cs], 2019. ArXiv: 1805.02369.

[MELBOURNE et al. 2010] A. Melbourne, G. R. Ridgway und D. J. Hawkes. **Image similarity metrics in image registration**. 2010, In: Medical Imaging.

[PASZKE et al. 2019] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai und S. Chintala. **PyTorch: An Imperative Style, High-Performance Deep Learning Library**. In: H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox und R. Garnett, Eds., Advances in Neural Information Processing Systems 32, pp. 8024–8035. 2019. Curran Associates, Inc.

[QIN et al. 2020] C. Qin, S. Wang, C. Chen, H. Qiu, W. Bai und D. Rueckert. **Biomechanics-informed Neural Networks for Myocardial Motion Tracking in MRI**. arXiv:2006.04725 [cs, eess], 2020. ArXiv: 2006.04725.

[RAZLIGHI et al. 2013] Q. R. Razlighi, N. Kehtarnavaz und S. Yousefi. **Evaluating similarity measures for brain image registration**. Journal of Visual Communication and Image Representation, Vol. 24:977–987, 2013.

[RENIEBLAS et al. 2017] G. P. Renieblas, A. T. Nogués, A. M. González, N. Gómez-Leon und E. G. del Castillo. **Structural similarity index family for image quality assessment in radiological images**. Journal of Medical Imaging, Vol. 4:035501, 2017.

[RONNEBERGER et al. 2015] O. Ronneberger, P. Fischer und T. Brox. **U-Net: Convolutional Networks for Biomedical Image Segmentation**. 2015.

[RUDER 2016] S. Ruder. **An overview of gradient descent optimization algorithms**. arXiv preprint arXiv:1609.04747, 2016.

[SHERINA et al. 2020] E. Sherina, L. Krainz, S. Hubmer, W. Drexler und O. Scherzer. **Displacement field estimation from OCT images utilizing speckle information with applications in quantitative elastography**. ArXiv, Vol. abs/2008.07373, 2020.

[TAHA und HANBURY 2015] A. A. Taha und A. Hanbury. **Metrics for evaluating 3D medical image segmentation: Analysis, selection, and tool**. BMC Medical Imaging, Vol. 15, 2015.

[TANG et al. 2020] K. Tang, Z. Li, L. Tian, L. Wang und Y. Zhu. **ADMIR–Affine and Deformable Medical Image Registration for Drug-Addicted Brain Images**. IEEE Access, Vol. 8:70960–70968, 2020.

[VASWANI et al. 2017] A. Vaswani, N. M. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser und I. Polosukhin. **Attention is All you Need**. ArXiv, Vol. abs/1706.03762, 2017.

[WANG et al. 2019] C. Wang, G. Papanastasiou, A. Chartsias, G. Jacenkow, S. A. Tsaftaris und H. Zhang. **FIRE: Unsupervised bi-directional inter-modality registration using deep networks**. 2019.

[WANG et al. 2004] Z. Wang, A. C. Bovik, H. R. Sheikh und E. P. Simoncelli. **Image quality assessment: from error visibility to structural similarity**. IEEE Transactions on Image Processing, Vol. 13:600–612, 2004.

[WELLING und TEH] M. Welling und Y. W. Teh. **Bayesian Learning via Stochastic Gradient Langevin Dynamics**. p. 8, ????

[ZHANG 2018] J. Zhang. **Inverse-Consistent Deep Networks for Unsupervised Deformable Image Registration**. arXiv:1809.03443 [cs], 2018. ArXiv: 1809.03443.

[ZHOU et al. 2020] J. Zhou, G. Cui, Z. Zhang, C. Yang, Z. Liu und M. Sun. **Graph Neural Networks: A Review of Methods and Applications**. ArXiv, Vol. abs/1812.08434, 2020.

[ZHOU et al. 2016] T. Zhou, P. Krähenbühl, M. Aubry, Q. Huang und A. A. Efros. **Learning Dense Correspondence via 3D-Guided Cycle Consistency**. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 117–126, 2016.

[ZUO et al. 2021] L. Zuo, B. E. Dewey, A. Carass, Y. Liu, Y. He, P. A. Calabresi und J. L. Prince. **Information-based Disentangled Representation Learning for Unsupervised MR Harmonization**. 2021, In: IPMI.

[ÇIÇEK et al. 2016] Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox und O. Ronneberger. **3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation**. ArXiv, Vol. abs/1606.06650, 2016.

Declaration of Academic Integrity

We hereby declare that we have written the present work by ourselves and did not use any sources or tools other than the ones indicated.

Datum: .....

(Signature)