Dual-Attention Recurrent Networks for Affine Registration of Neuroimaging Data

Xin Dai* Xiangnan Kong* Xinyue Liu* John Boaz Lee* Constance Moore†

Abstract

Neuroimaging data typically undergoes several preprocessing steps before further analysis and mining can be done. Affine image registration is one of the important tasks during preprocessing. Recently, several image registration methods which are based on Convolutional Neural Networks have been proposed. However, due to the high computational and memory requirements of CNNs, these methods cannot be used in real-time for large neuroimaging data like fMRI. In this paper, we propose a Dual-Attention Recurrent Network (DRN) which uses a hard attention mechanism to allow the model to focus on small, but task-relevant, parts of the input image – thus reducing computational and memory costs. Furthermore, DRN naturally supports inhomogeneity between the raw input image (*e.g.*, functional MRI) and the image we want to align it to (*e.g.*, anatomical MRI) so it can be applied to harder registration tasks such as fMRI coregistration and normalization. Extensive experiments on two different datasets demonstrate that DRN significantly reduces the computational and memory costs compared with other neural network-based methods without sacrificing the quality of image registration.

Keywords: Attention Model; Recurrent Neural Network; Deep Learning; fMRI

1 Introduction

Neuroimaging analysis and mining, which aims to model the functional structure of the brain [35] or extract diagnostic information [29] from a corpus of neuroimaging data, has attracted a lot of interest recently. However, raw neuroimaging data is usually quite noisy and inconsistent across samples [18]. Hence, the data typically undergoes a series of preprocessing steps before it can be further analyzed.

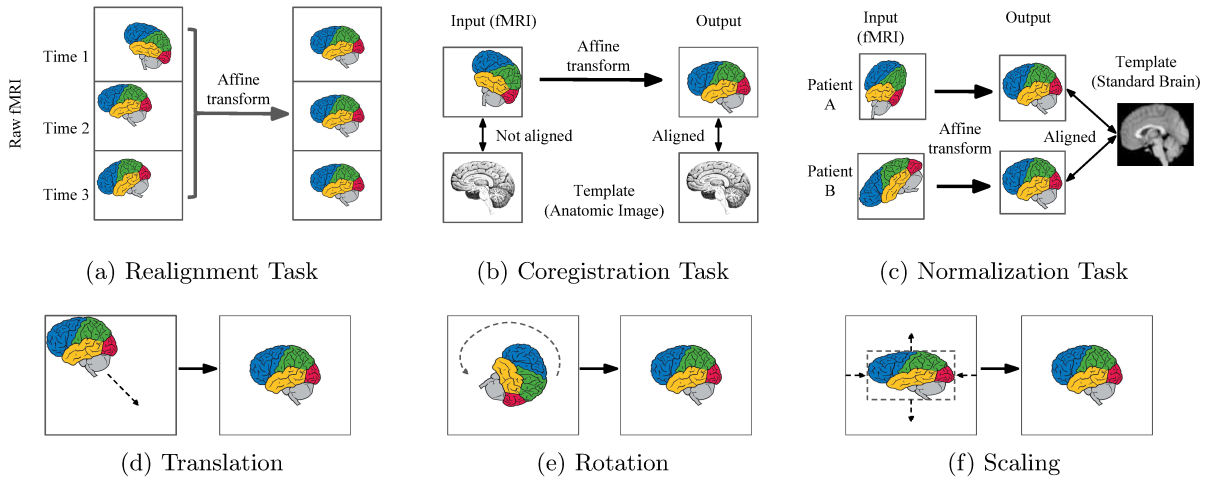
Affine registration is one of the most common tasks performed during preprocessing [2, 7, 14]. The goal of image registration is to spatially transform a raw image to match a given template image. Three types of image registration techniques, *i.e.*, realignment, coregistration, and normalization, are commonly applied on neuroimaging data. All three types of registrations are performed during preprocessing for a variety of brain mining tasks including brain atlas discovery [23], region-of-interest extraction [34], brain network discovery [18], and disease detection [29]. We illustrate the three techniques in Figures 1a-1c.

Automatic image registration has been extensively studied not only in the neuroscience domain [7, 9, 14], but also in other fields like pattern recognition [6, 31] and geoscience [22, 27]. More recently, some studies have used Convolutional Neural Networks (CNN) for medical image registration [19, 24]. Compared to traditional approaches [7, 14], the CNN-based methods can achieve faster processing speeds [24] while avoiding the use of generic matching metrics which have some severe drawbacks [19].

However CNNs may not be an ideal solution for *real-time* registration tasks on functional Magnetic Resonance Imaging (fMRI), due to their high memory and computational costs. As we illustrate in Figure 3, applying CNNs on high-dimensional fMRI data may result in extremely large feature maps. For example, the 3D fMRI data from a *single* timepoint can have a size of $96 \times 96 \times 96$. Suppose we use only 10 convolutional filters at the first layer, the dimension of the resulting feature map will be $96 \times 96 \times 96 \times 10 = 8,847,360$. Furthermore, CNNs also suffer from heavy computational costs. The number of multiplications in a 2D convolutional layer is $O(H \times W \times N \times x \times y \times c)$, where H, W, N are the corresponding height, width, and number of filters, while x, y, c are the height, width, and channels of the inputs. Such high costs in memory and computation make CNNs inefficient for real-time registration of neuroimaging data.

CNNs have a high associated cost because they have to scan the entire input image to calculate global features. However, the nature of the image registration task lends itself well to solutions that merely consider very limited partial information from the image. As shown in the examples in Figure 2, we only need to look at the nose (small region) to align a rotated human face,

*Worcester Polytechnic Institute
†University of Massachusetts Medical School



(a) Realignment Task (b) Coregistration Task (c) Normalization Task

(d) Translation (e) Rotation (f) Scaling

Figure 1: (a), (b), (c) show three different tasks to preprocess neuroimaging data. All three tasks can be considered as special cases of affine registration. (d), (e), (f) are three atomic types of affine trasformation

or we only need the position of the face (coarse global information) while ignoring a lot of other details to align a translated face. These properties of our task inspired us to consider the hard attention mechanism to solve this highly complex problem.

In 2014, Mnih et al. proposed a reinforcement learning approach for visual tasks called the Recurrent Attention Model (RAM) [25]. RAM controls an n-step agent as it performs actions by moving its sensor over the input image (environment). At each step, the sensor of RAM takes a "glimpse" of the image. Because the size of each glimpse is typically much smaller than that of the input image, and the number of glimpses is usually a tiny constant, RAM's computational and memory costs can be much lower than that of a CNN.

In this paper, we discuss how to build a recurrent attention-based model for 3D image registration on high-dimensional fMRI image. We address several unique challenges for applying a recurrent attention model on our task. Firstly, the range and histogram of voxel values between the raw image and the template image can differ significantly. Furthermore, the size of the objects in the two images can differ quite significantly as well. This inhomogeneity between the raw image and the template image can harm the performance of RAM significantly, since it only has a single attention mechanism. To solve these problems we propose a model with dual-attention.

Our idea and contribution: In this paper, we formulate image registration as a regression problem. We study how to design a neural network based model which can handle different kinds of registration. To deal with the problems of high computational complexity and image heterogeneity, we proposed a Dual-Attention Recurrent Network (DRN) and compared it against multiple state-of-the-art approaches, including CNN, RAM [25], and DSL [19], on four different image registration tasks. The experiment results clearly show that DRN outperforms all the baselines on all the tasks, indicating that it is a promising approach for real-time and universal image registration.

2 Problem Formulation

In this section, we introduce some related concepts and then define the problem.

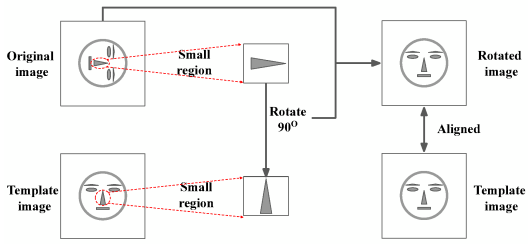
2.1 3D affine registration on neuroimaging data

An affine transformation is a function that maps an object from an affine space to another while preserving distance ratios but changing the position, orientation, or size of the object. Three atomic operations of affine transformation: translation, rotation, and scaling, are shown in Figures 1d-1f.

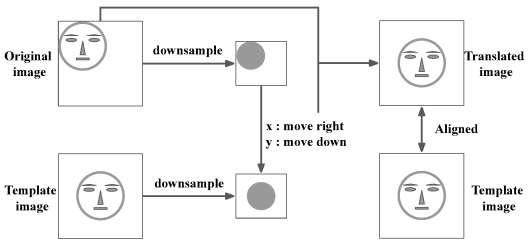
In neuroimage preprocessing, we often encounter three kinds of 3D image registration tasks involving affine transformation: Realignment, Coregistration and Normalization (Figures 1a-1c).

Realignment: Also called motion correction. An fMRI record of a patient can be viewed as a time sequence containing multiple 3D fMRI images. Patients may move their heads slightly while undergoing an MRI scan. This causes the voxels in the same position to correspond to different anatomical locations at different time points. As its name suggests, the goal of the task is to realign the images in an fMRI record.

Coregistration: It is an alignment between two brain images which are usually derived using different techniques, *e.g.*, anatomical MRI and functional MRI scans. Coregistration is usually used to line up a functional image and a structural image, so the voxels



(a) Example: efficient registration for rotation



(b) Example: efficient registration for translation

Figure 2: The computational and memory costs of image registration can be reduced by using only local details or coarse global information. (a) For registrations which involve only rotation, the local details in a small region are sufficient to solve the problem (*e.g.*, we can rotate the face to align it with the template by simply focusing on the nose). (b) For registrations which involve a translation, the coarse global information in a small downsampled image is sufficient to solve the problem.

from the same position in each image now correspond to the same anatomical location.

Normalization: Normalization is like coregistration, but with different inputs. To overcome variability in the shapes and sizes of scans from different individuals, normalization aligns the neuroimaging scans of multiple individuals to a single brain template. In this paper, we focus on normalization with linear transformation.

2.2 Problem formulation Figures 1a-1c show that realignment, coregistration, and normalization can be viewed as matching a raw fMRI image to a template image via affine transformation. The only differences between them are the particular templates and transformations that are used.

General affine transformation can be denoted as a vector $\mathbf{a} = (t_x, t_y, t_z, r_x, r_y, r_z, s_x, s_y, s_z)$, in which t_x, t_y, t_z indicate translations along three axes, r_x, r_y, r_z indicate rotations around three axes, and s_x, s_y, s_z indicate scaling. For realignment and coregistration, the transformations are rigid (*i.e.*, no scaling is involved). Based on the above, we can formulate realignment, coregistration, and normalization as regression problems.

Problem: We are given a set of samples $\mathcal{D}=\{(\mathbf{R}_i, \mathbf{T}_i)\}$, where $\mathbf{R}_i \in \mathbb{R}^{x\times y\times z}$ denotes a raw 3D image while $\mathbf{T}_i \in \mathbb{R}^{x\times y\times z}$ denotes the 3D template image. For each pair $(\mathbf{R}_i, \mathbf{T}_i)$, there exists a vector $\mathbf{a}_i \in \mathbb{R}^9$ which denotes the ideal affine transformation from $\mathbf{R}_i$ to $\mathbf{T}_i$ (similarly, $\mathbf{a}_i \in \mathbb{R}^6$ if the task is realignment or coregistration). Given a parametric function $f_\theta : \mathcal{D} \mapsto \mathbb{R}^9$, the goal is learn parameters θ such that:

$$\arg\underset{\theta}{\text{minimize}} \sum_{(\mathbf{R}_i,\mathbf{T}_i)\in\mathcal{D}} \|f_\theta(\mathbf{R}_i, \mathbf{T}_i) - \mathbf{a}_i\|^2 \tag{2.1}$$

Now the problem is a typical regression problem.

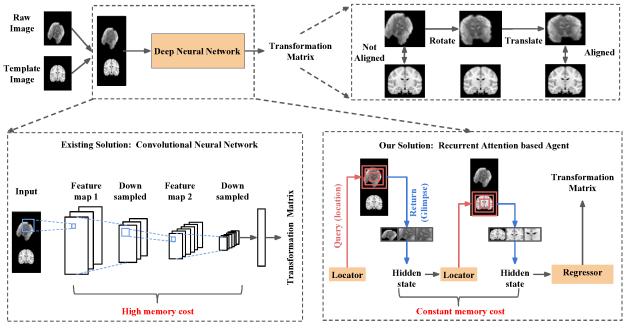


Figure 3: Existing CNN-based approaches [19, 24] have high computational memory cost when the input image is very large. On the other hand, a recurrent attention model's memory cost is not related to input size.

Note that 2D image registration can be defined in a similar way where the transformation is in the 2D space.

3 The Proposed Method

We now describe our proposed Dual-Attention Recurrent Model which is inspired by the RAM model [25].

3.1 3D glimpse sensor and dual-attention glimpse network At each time step, the DRN utilizes its 3D glimpse sensor as well as the dual-attention glimpse network to construct a glimpse representation $\mathbf{g}$. The representation $\mathbf{g}$ encodes information from small parts of the input images (*i.e.*, the raw image and the template image).

3D glimpse sensor: As shown in Fig 4b, given a 3D image $\mathbf{I}$, a location $\mathbf{l} = (i, j, k)$, and a glimpse scale s, the sensor extracts a set of s cropped images $\{\mathbf{C}_1, \cdots, \mathbf{C}_s\}$ from image $\mathbf{I}$. The cropped images are centered at (i, j, k). The length of a cropped image $\mathbf{C}_{m+1}$'s sides are always twice that of $\mathbf{C}_m$. For instance, if $\mathbf{C}_1$ has shape $2 \times 2 \times 2$, then $\mathbf{C}_2$ should have shape $4 \times 4 \times 4$. Finally, all cropped images $\mathbf{C}_1, \cdots, \mathbf{C}_s$ are

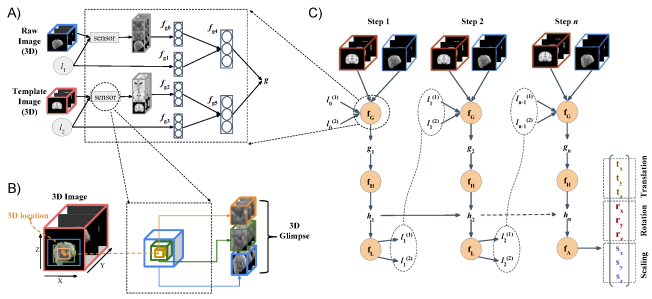


Figure 4: **A) Dual-attention glimpse network** f_G: Given a raw image, a template image, and two locations $\mathbf{l_1}$ and $\mathbf{l_2}$ (one for each image, respectively), f_G uses two 3D sensors to extract a 3D glimpse from each of the two images at locations $\mathbf{l_1}$ and $\mathbf{l_2}$, respectively. The glimpse and location for the raw image are mapped by fully connected layers f_{g0} and f_{g1}, respectively, and their outputs are concatenated and encoded by layer f_{g4}. Similarly, we use layers f_{g2}, f_{g3}, f_{g5} to encode the glimpse and location for the template image. Finally, the outputs of f_{g4} and f_{g5} are concatenated as a single glimpse representation $\mathbf{g}$. **B) 3D glimpse sensor**: Given a 3D image and a location $\mathbf{l}$, the sensor extracts 3 cropped images centered at $\mathbf{l}$ with varying scales forming a 3D retina-like glimpse [25]. **C) Overall structure of DRN**: The DRN is an RNN. At the i-th step, the core network f_H takes $\mathbf{g}_i$ generated by f_G and internal state $\mathbf{h}_i$ from the previous step and generates a new internal state $\mathbf{h}_{i+1}$. The location network f_L uses $\mathbf{h}_{i+1}$ to stochastically generate the next locations $\mathbf{l}_{i+1}^{(1)}$ and $\mathbf{l}_{i+1}^{(2)}$ (for the raw image and the template image). At the last step, the action network f_A uses the final internal state $\mathbf{h}_{n+1}$ to produce an affine transformation, which matches the raw image to the template image.

resized to the same size as $\mathbf{C}_1$, concatenated together and then flattened into a glimpse vector $\mathbf{x}$.

DRN uses two sensors to extract two glimpse vectors. The first vector $\mathbf{x}_r$ is extracted from the raw image $\mathbf{I}_r$ at location $\mathbf{l}_r$ while the second glimpse $\mathbf{x}_t$ is taken from the template image $\mathbf{I}_t$ at location $\mathbf{l}_t$. We then use a dual-attention glimpse network to encode $\mathbf{x}_r$, $\mathbf{l}_r$, $\mathbf{x}_t$, $\mathbf{l}_t$ into a *single* glimpse representation $\mathbf{g}$.

Dual-attention glimpse network f_G: As shown in Figure 4a, the glimpse network is composed by 6 fully connected layers. Let $f_{gi}(\mathbf{x}) = \sigma(\mathbf{W}_i^\top \mathbf{x} + \mathbf{b}_i)$ denote a fully-connected layer, parameterized by weight matrix $\mathbf{W}_i$ and bias vector $\mathbf{b}_i$, with ReLU activation σ and input $\mathbf{x}$. The glimpse $\mathbf{x}_r$ and the location $\mathbf{l}_r$ from the raw image are encoded as $\mathbf{x}'_r = f_{g0}(\mathbf{x}_r)$ and $\mathbf{l}'_r = f_{g1}(\mathbf{l}_r)$, respectively. Similarly, $\mathbf{x}_t$ and $\mathbf{l}_t$ from the template image are encoded as $\mathbf{x}'_t = f_{g2}(\mathbf{x}_t)$ and $\mathbf{l}'_t = f_{g3}(\mathbf{l}_t)$. Finally, we construct the glimpse vector for the raw image $\mathbf{g}'_r = f_{g4}(\mathbf{x}'_r \,||\, \mathbf{l}'_r)$ where $||$ represents concatenation. Similarly, $\mathbf{g}'_t = f_{g5}(\mathbf{x}'_t \,||\, \mathbf{l}'_t)$. Finally, we concatenate to produce the glimpse representation $\mathbf{g} = \mathbf{g}'_r \,||\, \mathbf{g}'_t$.

3.2 Dual-attention recurrent network

Similar to RAM, the DRN model is an RNN. Fig 4c shows the whole structure of DRN. We have described the 3D glimpse sensor and the glimpse network f_G above, here we describe the remaining components of DRN.

Core network f_H: Given the glimpse representation $\mathbf{g}_i$ and the hidden internal state $\mathbf{h}_i$ at step i, we calculate the new internal state $\mathbf{h}_{i+1} = f_H(\mathbf{g}_i, \mathbf{h}_i)$ using the core network, so the history of all the glimpses we have seen up to step i is encoded in $\mathbf{h}_{i+1}$. Here we use basic LSTM cells to form f_H.

Location network f_L: The locations of the two sensors are denoted as a vector $\mathbf{l}$=$(l_r^x, l_r^y, l_r^z, l_t^x, l_t^y, l_t^z)$. The sub-vector $\mathbf{l}^{(1)}$=(l_r^x, l_r^y, l_r^z) denotes the location on the raw image, and $\mathbf{l}^{(2)}$=(l_t^x, l_t^y, l_t^z) denotes the location on the template image. For step i, location $\mathbf{l}_i$ is chosen from a 6D Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\sigma})$. Here $\boldsymbol{\mu}_i = f_L(\mathbf{h}_i)$, where f_L is a single fully connected layer with *tanh* activation. On the other hand, $\boldsymbol{\sigma}$ is the user-specified standard deviation.

Action network f_A: Taking the internal state $\mathbf{h}_n$ at the last recurrent step n as the input, the action is to predict the transformation parameter vector $\mathbf{a}_p$ which maps the raw image to the template. The action network $f_A(\mathbf{h}_n) = \mathbf{a}_p$ is a three layer fully connected network. The activation functions are ReLU's except for the last layer which has no activation. The reward for the action is formulated as:

$$r = 1 - \frac{\|\mathbf{a}_p - \mathbf{a}_t\|^2}{\text{length}(\mathbf{a}_p)} \tag{3.2}$$

where $\mathbf{a}_t$ is the ground truth transformation that aligns the two images perfectly. Prediction is only performed at the last step and each of the agent's movements (*i.e.*, generated locations) is assigned the same reward (Eq. 3.2).

The n-step agent's interaction on the input image can be denoted by a sequence $\mathbf{S}_{1:n} = (\mathbf{x}_1, \mathbf{l}_1, \mathbf{x}_2, \mathbf{l}_2, \cdots, \mathbf{x}_n, \mathbf{a}_p)$. It can be viewed as a case of Partially Observable Markov Decision Process [25]. The true state of the environment is static but unknown. Here we use θ to denote the parameters of the above RNN. The agent needs to learn a policy $\pi(\mathbf{l}_i|\mathbf{S}_{1:i-1}; \theta)$ to maximize the expectation of reward:

$$J(\theta) = \mathbb{E}_{p(\mathbf{S}_{1:n};\theta)}\left[\sum_{i=1}^{n} r_{\mathbf{l}_i|\mathbf{S}_{1:i-1}}\right] \tag{3.3}$$

The $r_{\mathbf{l}_i|\mathbf{S}_{1:i-1}}$ is the reward for the location at the i-th

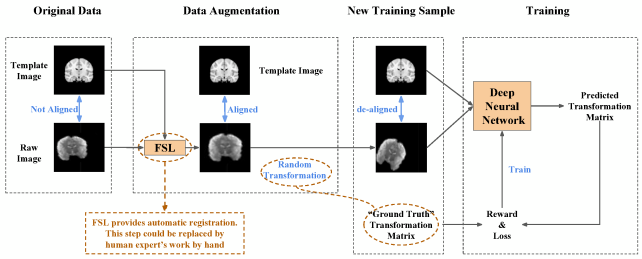


Figure 5: Data augmentation for coregistration and normalization for fMRI data

step. In this work, all the rewards $r_{\mathbf{l}_i|\mathbf{S}_{1:i}}$ are equal and computed by Eq. 3.2.

3.3 Training Similar to [25], we use the REINFORCE algorithm as defined by [32] to solve the problem above. The gradient of J can be approximately computed by:

$$\nabla_\theta J = \frac{1}{m}\sum_{j=1}^{m}\sum_{i=1}^{n}\nabla_\theta \log\left(\pi\left(\mathbf{l}_i^j|\mathbf{S}_{1:i-1}^j;\theta\right)\right)r^j \tag{3.4}$$

where m denotes the number of episodes. Recall we assume that $\mathbf{l}_i$ is generated from a Gaussian distribution, of which the covariance matrix is fixed and the mean is the output of location network at step i, hence Eq. 3.4 is equivalent to:

$$\nabla_\theta J = \frac{1}{C}\sum_{j=1}^{m}\sum_{i=1}^{n}\nabla_\theta\left(\mathbf{l}_i^j - f_\theta\left(\mathbf{x}_{i-1}^j, \mathbf{l}_{i-1}^j, \mathbf{h}_{i-1}^j\right)\right)r^j \tag{3.5}$$

where $f_\theta(\mathbf{x}_i^j, \mathbf{l}_i^j, \mathbf{h}_i^j)$ is the network at the i-th step for the j-th episode, taking the outputs from previous steps and outputting the mean value $\boldsymbol{\mu}_i^j$ of the Gaussian distribution generating $\mathbf{l}_i^j$. C is a constant and $\nabla_\theta f_\theta$ can be computed by standard backpropagation.

Using hybrid supervised loss: The above can only be used to train the glimpse, core, and location networks but not the action network. Because the final action $\mathbf{a}_p$ is only used to compute the reward, which is not differentiable (Eq. 3.2). So we also combine the mean squared loss between $\mathbf{a}_p$ and $\mathbf{a}_t$ to train the action network, glimpse network and core network. The location network is trained via REINFORCE only.

4 Experiments

4.1 Data collection In order to evaluate the performance on different registration tasks, we test our methods on 2 datasets.

• **MNIST dataset:** The original MNIST dataset can be obtained using the TensorFlow API[1]. It contains 60,000 handwritten digits in the training set and 10,000 in the test set. We first use MNIST to study 2D image registration tasks (since our model can also handle the 2D case), in which we randomly translate, rotate, and rescale the digits in each image.

• **Bipolar disorder dataset (BD):** We also test 3D image realignment, coregistration, and normalization on a Bipolar Disorder Dataset which we obtained from the University of Massachusetts Medical School. It contains neuroimaging data for 39 subjects. For each subject, we have a corresponding fMRI image with dimension $96\times96\times50\times156$ (3D spatial + 1D temporal dimensions) and a T1-Weighted (anatomical brain structure) image with dimension $170 \times 256 \times 256$.

4.1.1 Data augmentation Data augmentation is a necessity in medical image registration research [19]. We use data augmentation to generate more samples and labels. We now explain the two ways we apply data augmentation for the different registration tasks.

We use the first approach to generate data for 3D realignment. Similar to the approach used in [19], we first extract a 3D brain image (a time slice) from 4D fMRI data and center it in a $96 \times 96 \times 96$ black background. This image is used as the template. We then de-align the template via a random 3D rigid transformation $\mathbf{t}$ and treat the de-aligned image as the raw image. The ground truth is simply the inverse transformation of $\mathbf{t}$. We also used the same data augmentation on the 2D registration task, but the random transformations on MNIST are 2D affine transformation with scaling.

For 3D coregistration and normalization, we need to start with an fMRI image that is correctly aligned with a given template. To achieve this, we aligned each fMRI image $\mathbf{R}$ to a template $\mathbf{T}$ by using the well-known neuroimaging toolbox FSL[2] to get an aligned image $\mathbf{R}'$. A human expert can also accomplish this using manual alignment. We then randomly de-align via a random transformation of which the inverse is the ground truth and use the de-aligned image as the raw image. For the task of 3D coregistration, the template for each subject is the subject's brain anatomic scan and the random transformation is rigid. For the task of 3D normalization, all the subjects take the same MNI152 standard image as the template, and the random transformation is affine including scaling. We illustrate this process in Fig 5. We summarize the random transformations for each task in Table 1.

[1] https://www.tensorflow.org/
[2] https://fsl.fmrib.ox.ac.uk/fsl/

4.2 Compared methods In order to validate the effectiveness of DRN, we compared the following:

• **Fully Connected Neural Network (FC):** We compare with a fully connected neural network with two hidden layers. For 2D registration, the first hidden layer consists of 100 neurons while the second has 50. For 3D registration, both layers have 50 neurons due to GPU memory limit.

• **Convolutional Neural Network (CNN):** The CNN has a convolutional layer, a pooling layer, and two fully connected layers similar as above. The convolutional layer has 128 filters with filter size 5×5 followed by 2×2 max-pooling.

• **Recurrent Attention Model (RAM):** We compared against an RAM [25] model whose parameters are almost comparable with our proposed model. For fair comparison, this model is still able to see both the de-aligned and the template images.

• **Deep Supervised Learning agent (DSL):** DSL is a state-of-the-art artificial agent for 3D rigid registration [19]. The agent is instructed to mimic a greedy registration path, which minimizes the distance between two images step by step. The overall structure looks like a DQN, however the ground truth Q value can be calculated explicitly. To predict the Q value, we trained a CNN with five convolution layers with 8, 32, 32, 128, and 128 filters. We didn't adopt the hierarchical strategy designed for extremely high resolution images in the original paper since our dataset has lower resolution.

• **Double-attention Recurrent Network (DRN):** This is our proposed model. In the glimpse network, each of the four layers encoding image or location is composed by 128 neurons, and each of the last two layers encoding glimpse representation is composed by 256 neurons. The core network has 128 LSTM cells. The action network is the same as the FC above.

4.3 Performances evaluation

4.3.1 Assessment metric We use an assessment metric similar to [7]. Since we treated image registration as a regression problem, the assessment metrics are the average errors of translation, rotation and scaling. Error in translation is defined by Manhattan distance between the predicted translation and ground truth translation. The errors for rotation and scaling can be defined similarly.

4.3.2 Performances on 2D hand-writing digits image registration We first study the effectiveness of the proposed method on 2D registration. For all methods, the max training iterations is $700K$, initial learning rate is 0.1 while decay rate is 0.1, and batch size is 128. For our method, the number of glimpses is 8. The sensors used in RAM and DRN are 2D sensors. The crop size of the sensor is 8×8 and the glimpse is composed by three different resolutions (8×8, 16×16, and 32×32). Samples are generated using the process described in Section 4.1.1 and images are embedded into a 100×100 black background.

Table 1: Range of affine transformation on each task.

Task	Transformation		
	Translation (pixels)	Rotate (degree)	Scale (times)
2D registration	$\pm$ 20	$\pm$ 120	$1 \sim 3$
3D realignment	$\pm$ 20	$\pm$ 45	-
3D coregistration	$\pm$ 20	$\pm$ 45	-
3D normalization	$\pm$ 20	$\pm$ 45	$0.8 \sim 1.3$

Table 2 shows the average error for all methods on the three tasks. We also show each methods relative rank for each task. The results show that the proposed method clearly outperforms all the baselines. Compared to FC and CNN, DRN uses much less neurons, but achieved obviously better results. In particular, RAM has the lowest performance, which supports our assumption that adding a second attention mechanism can significantly reduce regression error. DSL achieves slightly better results for rotation, but its translation error is very high. There is no scaling result for DSL since it is designed for rigid transformations.

4.3.3 Performances on 3D brain image registration We then study the effectiveness of the proposed method on 3D realignment, coregistration, and normalization. Due to the large size of brain images, the batch size is reduced to 16. For RAM and DRN, we increase the number of glimpses to 16, and the crop size of the sensor to $20 \times 20 \times 20$, because compared with MNIST, the object-to-background ratio is much higher for brain images. For each subject (patient) we extract a time slice from their fMRI and generate synthetic samples using the method in Section 4.1.1.

Table 3 shows the performances of the compared methods on BD including their relative performance. Again, our method significantly outperforms all the baselines. Again, DRN always outperforms RAM, especially on the task of normalization. The low performance of RAM on normalization probably indicates that the use of a single sensor is very sensitive to scaling. By contrast, our dual-sensor architecture is able to greatly reduce the average error on all three kinds of transformation. We also tested DSL on realignment, showing average errors which are very high. One possible reason may be that DSL is originally designed for CT images which have higher resolution but are less

Table 2: Results on MNIST for 2D registration. The results are reported as "average performance (rank)".

Method	Average Error		
	Translation	Rotation	Scaling
FC	0.11 (2)	0.46 (4)	0.40 (3)
CNN	0.12 (3)	0.40 (3)	0.35 (2)
RAM	0.35 (4)	0.49 (5)	0.65 (4)
DSL	0.58 (5)	0.19 (1)	-
DRN	0.09 (1)	0.20 (2)	0.26 (1)

Table 3: Results on BD for 3D registrations. The results are reported as "average performance (rank)".

Task	Method	Average Error		
		Translation	Rotation	Scaling
Realignment	FC	0.39 (4)	0.68 (4)	-
	CNN	0.28 (3)	0.41 (3)	-
	RAM	0.24 (2)	0.35 (2)	-
	DSL	1.21 (5)	0.80 (5)	-
	DRN	0.20 (1)	0.20 (1)	-
Coregistration	FC	0.38 (4)	0.61 (4)	-
	CNN	0.27 (3)	0.48 (3)	-
	RAM	0.22 (1)	0.39 (2)	-
	DRN	0.22 (1)	0.32 (1)	-
Normalization	FC	0.31 (3)	0.55 (3)	0.91 (3)
	CNN	0.25 (2)	0.40 (2)	0.89 (2)
	RAM	0.44 (4)	1.38 (4)	1.37 (4)
	DRN	0.19 (1)	0.20 (1)	0.77 (1)

noisy compared to fMRI. Fig. 6 visualizes examples of the results of all the 3D registrations achieved by DRN.

4.4 Computational and memory complexity analysis The most important motivation of this paper is to find an alternative deep network architecture with lower computational and memory costs as an alternative for CNN-based methods for image registration. In Tables 4 and 5, we report the costs of all baselines and our method on 2D and 3D tasks.

Memory cost can be divided into two parts: the number of neurons and weights. We don't consider the biases since the number of weights is dominant. Similarly, for computational cost we only consider the number of multiplications instead of additions.

From Tables 4 and 5, we can see that CNN always has very high computational cost as well as number of neurons and parameters. The reason for the larger parameter size is because the CNN uses a single convolutional layer, so the input feature to its fully connected layer is large. DSL is also based on CNNs but its parameter size is smaller, because it has five convolutional layers and fewer filters at early layers. However, the computational cost of DSL is much higher than other methods, because DSL needs to call its CNN many times. The fully-connected network has the smallest number of neurons, but its parameter size is also large. RAM has the lowest cost for both computation and memory, however its accuracy can be very low in some cases as we have pointed out. The proposed DRN is significantly more efficient in computation and space than most of the baselines. Compared with RAM, our proposed DRN has slightly higher computational and memory costs, because DRN has an additional attention mechanism for two sensors while RAM only controls one sensor. Even so, the costs for both DRN and RAM are actually at the same order of magnitude.

Table 4: Computational and memory costs on MNIST Dataset. Results are reported as "cost (rank)".

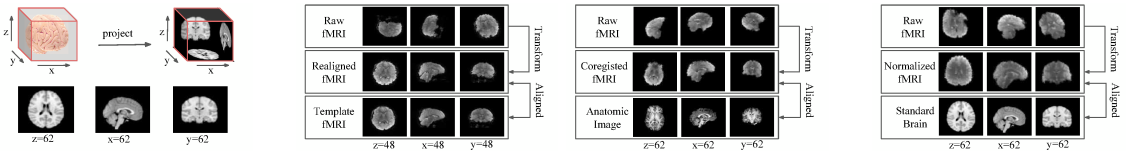
Method	Cost		
	Computational cost(flop)	Number of neurons	Number of weights
FC	1.28×10^6 (3)	156 (1)	10^6 (4)
CNN	3.20×10^7 (4)	1.28×10^6 (5)	3.20×10^7 (5)
RAM	6.07×10^5 (1)	796 (2)	1.01×10^5 (1)
DSL	3.56×10^9 (5)	2.07×10^5 (4)	6.27×10^5 (3)
DRN	6.84×10^5 (2)	1052 (3)	1.75×10^5 (2)

Table 5: Computational and memory costs on Bipolar Disorder Dataset. Results are reported as "cost (rank)".

Method	Cost		
	Computational cost(flop)	Number of neurons	Number of weights
FC	2.50×10^8 (3)	156 (1)	1.95×10^8 (4)
CNN	6.29×10^9 (4)	2.50×10^8 (5)	3.20×10^9 (5)
RAM	5.02×10^7 (1)	796 (2)	3.16×10^6 (1)
DSL	8.64×10^{11} (5)	9.76×10^6 (4)	6.53×10^6 (3)
DRN	9.93×10^7 (2)	1052 (3)	6.30×10^6 (2)

4.5 Influence of parameters In this section, we study the influence of glimpse scale. Recall that a glimpse representation is composed by several cropped images with different glimpse scales extracted from an image at the same location. In the above experiments, a glimpse has three scales, of which the smallest scale has the highest resolution but smallest view range, while the largest scale has the largest view range but lowest resolution. As shown in Table 6, we observe a larger degradation in performance when we only use the high resolution scale. Similarly, using only the low resolution scale is slightly worse than combining all scales together, although there is a smaller gap in performance.

We designed another experiment on the MNIST



(a) 3D Visualization (b) 3D Realignment (c) 3D Coregistration (d) 3D Normalization

Figure 6: Visualization of results achieved by DRN on three different 3D registrations of neuroimaging data.

Table 6: Discussion on glimpse scale on brain image.

	Average Error		
Glimpse scale	Translation	Rotation	Scaling
High Resolution	0.25	0.42	0.99
Low Resolution	0.15	0.20	0.74
All Resolution	0.12	0.26	0.72

Table 7: Discussion on glimpse scale on digit image.

	Average Error	
Glimpse scale	Translation	Rotation
High Resolution	0.30	0.87
Low Resolution	0.11	1.48
All Resolution	0.15	0.91

dataset. In this experiment, every digit is embedded in a 96×96 background and then stacked 96 times to form a 3D digit, then a random rigid transformation is performed. Unlike the brain 3D image, the object in the 3D digit image is small compared to the size of the background. The average errors of DRN using different levels of glimpse are shown in Table 7. It illustrates that only using the high resolution scale can achieve small error on rotation but high error on translation. On the contrary, only using low resolution scale can achieve high error on rotation but low error on translation, due to it has large view range. Combining the different scales together can reduce both errors on translation and rotation and is a good middle ground.

5 Related Work

Many different algorithms have been designed for image registration. In earlier work, algorithms for this problem used similarity measures based on the difference in pixel values [5, 21]. Due to the multi-modal nature of coregistration for human neuroimaging data, new similarity measures had to be defined. Collignon et al. [7] proposed an information theoretic approach to solve 3D rigid coregistration by using mutual information as a matching criteria. Ashburner and Friston [2] proposed a unified framework for coregistration and tissue segmentation. Meanwhile, Gartus et al. [9] conducted experiments to compare the output of automatic methods with that of human experts. Several feature-based algorithms were also proposed by Rangarajan et al. [28], Pang et al. [27], and Ma et al. [22]. Finally, Benjemaa and Schmitt [6], Williams [31], Arun et al. [1] and Guo et al. [12] proposed pattern recognition-based methods.

In recent years, deep learning has achieved great success in numerous tasks. In 2012, Krizhevsky et al. [16] proposed a method using CNNs for image classification which demonstrated outstanding results. More recently, researchers have begun to explore CNN-based approaches for neuroimaging data. Nie et al. [26] used CNNs for survival time prediction of brain tumor patients. Lee et al. [18] proposed a CNN-based approach for classification of fMRI time sequences. Hosseini et al. [15] used a CNN to extract high level features from fMRI. Recently, Liao et al. [19] proposed an approach based on deep supervised learning. It achieves state-of-the-art performance on the registration of 3D medical images.

Deep attention models have been proposed for various tasks in computer vision [3, 8, 17, 30] and natural language processing [4, 33]. RAM [25] is a recurrent attention model which was proposed to reduce the computational complexity associated with processing large image data. Recently, Haque et al. used the RAM method for Depth-Based Person Identification [13]. Multiple work have also been published introducing various types of attention with RNN [10, 11, 20, 36].

6 Conclusion

We proposed a deep dual-attention recurrent model as a computationally-efficient solution for various affine registration problems in neuroimaging. In particular, the dual-attention mechanism is able to handle inhomogeneity between the raw image and the template. Experimental results on two different datasets evaluating three different types of image registrations show that our proposed method can outperform the state-of-the-art while keeping computational and memory costs low.

7 Acknowledgement

This work is supported in part by NSF through grant IIS-1718310.

References

[1] K. S. Arun, T. S. Huang, and S. D. Blostein. Least-squares fitting of two 3-D point sets. *IEEE Trans-*

actions on Pattern Analysis and Machine Intelligence, 9(5):698–700, 1987.

[2] J. Ashburner and K. Friston. Multimodal image coregistration and partitioning-a unified framework. *Neuroimage*, 6(3):209–217, 1997.

[3] J. Ba, V. Mnih, and K. Kavukcuoglu. Multiple object recognition with visual attention. *arXiv:1412.7755*, 2014.

[4] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *arXiv:1409.0473*, 2014.

[5] D. I. Barnea and H. F. Silverman. A class of algorithms for fast digital image registration. *IEEE Transactions on Computers*, C-21(2):179–186, 1972.

[6] R. Benjemaa and F. Schmitt. A solution for the registration of multiple 3D point sets using unit quaternions. In *Proc. of ECCV*, pages 34–50, 1998.

[7] A. Collignon et al. Automated multi-modality image registration based on information theory. *Information Processing in Medical Imaging*, 3(6):263–274, 1995.

[8] M. Denil, L. Bazzani, H. Larochelle, and N. de Freitas. Learning where to attend with deep architectures for image tracking. *Neural Computation*, 24(8):2151–2184, 2012.

[9] A. Gartus et al. Comparison of fMRI coregistration results between human experts and software solutions in patients and healthy subjects. *European Radiology*, 17(6):1634–1643, 2007.

[10] A. Graves. Generating sequences with recurrent neural networks. *arXiv:1308.0850*, 2013.

[11] K. Gregor et al. Draw: A recurrent neural network for image generation. *arXiv:1502.04623*, 2015.

[12] Y. Guo et al. An accurate and robust range image registration algorithm for 3D object modeling. *IEEE Transactions on Multimedia*, 16(5):1377–1390, 2014.

[13] A. Haque, A. Alahi, and L. Fei-Fei. Recurrent attention models for depth-based person identification. In *Proc. of CVPR*, pages 1229–1238, 2016.

[14] D. L. Hill, C. Studholme, and D. J. Hawkes. Voxel similarity measures for automated image registration. In *Proc. of VBC*, pages 205–217, 1994.

[15] M.-P. Hosseini et al. Deep learning with edge computing for localization of epileptogenicity using multimodal rs-fMRI and EEG big data. In *Proc. of ICAC*, pages 83–92, 2017.

[16] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. In *Proc. of NeurIPS*, pages 1097–1105, 2012.

[17] H. Larochelle and G. E. Hinton. Learning to combine foveal glimpses with a third-order Boltzmann machine. In *Proc. of NeurIPS*, pages 1243–1251, 2010.

[18] J. B. Lee, X. Kong, Y. Bao, and C. Moore. Identifying deep contrasting networks from time series data: Application to brain network analysis. In *Proc. of SDM*, pages 543–551, 2017.

[19] R. Liao et al. An artificial agent for robust image registration. In *Proc. of AAAI*, pages 4168–4175, 2017.

[20] J. Liu et al. Global context-aware attention LSTM networks for 3D action recognition. In *Proc. of CVPR*, pages 3671–3680, 2017.

[21] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proc. of IJCAI*, pages 674–679, 1981.

[22] J. Ma et al. Robust feature matching for remote sensing image registration via locally linear transforming. *IEEE Transactions on Geoscience and Remote Sensing*, 53(12):6469–6481, 2015.

[23] A. Mensch, G. Varoquaux, and B. Thirion. Compressed online dictionary learning for fast resting-state fMRI decomposition. In *Proc. of ISBI*, pages 1282–1285, 2016.

[24] S. Miao, Z. J. Wang, and R. Liao. A CNN regression approach for real-time 2D/3D registration. *IEEE Transactions on Medical Imaging*, 35(5):1352–1363, 2016.

[25] V. Mnih, N. Heess, A. Graves, and K. Kavukcuoglu. Recurrent models of visual attention. In *Proc. of NeurIPS*, pages 2204–2212, 2014.

[26] D. Nie et al. 3D deep learning for multi-modal imaging-guided survival time prediction of brain tumor patients. In *Proc. of MICCAI*, pages 212–220, 2016.

[27] S. Pang, J. Xue, Q. Tian, and N. Zheng. Exploiting local linear geometric structure for identifying correct matches. *Computer Vision and Image Understanding*, 128(1):51–64, 2014.

[28] A. Rangarajan, H. Chui, and J. S. Duncan. Rigid point feature registration using mutual information. *Medical Image Analysis*, 3(4):425–440, 1999.

[29] Y. Sun, B. Bhanu, and S. Bhanu. Automatic symmetry-integrated brain injury detection in MRI sequences. In *Proc. of CVPRW*, pages 79–86, 2009.

[30] Y. Tang, N. Srivastava, and R. R. Salakhutdinov. Learning generative models with visual attention. In *Proc. of NeurIPS*, pages 1808–1816, 2014.

[31] J. Williams and M. Bennamoun. Simultaneous registration of multiple corresponding point sets. *Computer Vision and Image Understanding*, 81(1):117–142, 2001.

[32] R. J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. In *Reinforcement Learning*, pages 5–32. Springer, 1992.

[33] K. Xu et al. Show, attend and tell: Neural image caption generation with visual attention. In *Proc. of ICML*, pages 2048–2057, 2015.

[34] Y. Zhang et al. Detection of subjects and brain regions related to Alzheimer's disease using 3D MRI scans based on eigenbrain and machine learning. *Frontiers in Computational Neuroscience*, 9(1):66, 2015.

[35] L. Zhou et al. Discriminative brain effective connectivity analysis for Alzheimer's disease: a kernel learning approach upon sparse Gaussian Bayesian network. In *Proc. of CVPR*, pages 2243–2250, 2013.

[36] Z. Zhou et al. See the forest for the trees: Joint spatial and temporal recurrent neural networks for video-based person re-identification. In *Proc. of CVPR*, pages 6776–6785, 2017.