

Research Article

Bayesian Fully Convolutional Networks for Brain Image Registration

Kunpeng Cui,^{1,2} Panpan Fu,³ Yinghao Li ,^{3,4} and Yusong Lin ^{2,3,4}

¹School of Information Engineering, Zhengzhou University, Zhengzhou 450001, Henan, China

²Collaborative Innovation Center for Internet Healthcare, Zhengzhou University, Zhengzhou 450052, Henan, China

³School of Software, Zhengzhou University, Zhengzhou 450002, Henan, China

⁴Hanwei IoT Institute, Zhengzhou University, Zhengzhou 450002, Henan, China

Correspondence should be addressed to Yinghao Li; yinghaoli@zzu.edu.cn and Yusong Lin; yslin@ha.edu.cn

Received 28 February 2021; Revised 17 June 2021; Accepted 13 July 2021; Published 27 July 2021

Academic Editor: Jialin Peng

Copyright © 2021 Kunpeng Cui et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The purpose of medical image registration is to find geometric transformations that align two medical images so that the corresponding voxels on two images are spatially consistent. Nonrigid medical image registration is a key step in medical image processing, such as image comparison, data fusion, target recognition, and pathological change analysis. Existing registration methods only consider registration accuracy but largely neglect the uncertainty of registration results. In this work, a method based on the Bayesian fully convolutional neural network is proposed for nonrigid medical image registration. The proposed method can generate a geometric uncertainty map to calculate the uncertainty of registration results. This uncertainty can be interpreted as a confidence interval, which is essential for judging whether the source data are abnormal. Moreover, the proposed method introduces group normalization, which is conducive to the network convergence of the Bayesian neural network. Some representative learning-based image registration methods are compared with the proposed method on different image datasets. Experimental results show that the registration accuracy of the proposed method is better than that of the methods, and its antifolding performance is comparable to that of fast image registration and VoxelMorph. Furthermore, the proposed method can evaluate the uncertainty of registration results.

1. Introduction

Image registration is an image-processing process that aligns two or more images of the same scene captured at different times and different perspectives or by using different sensors [1, 2]. Nonrigid medical image registration is a key step in medical image processing. In clinical diagnosis, it can judge a patient's progress by aligning the brain magnetic resonance images of the patient with Alzheimer's disease at different periods [3, 4]. In tumor surgery, rapid medical image registration can aid doctors in surgical navigation [5–7]. In demography research, image registration is helpful for studying differences in the brain tissue structures of people from different countries.

With the advances in medical image registration technology, various registration methods have been developed,

such as elastic body models [8–10], viscous fluid flow models [11–13], diffusion models [14], curvature registration [15], statistical parameter mapping [16], free-form deformation with b-spline [17], discrete method [18, 19], and demons [20] for registration model construction. Many optimization algorithms have also been devised, such as gradient descent methods [21], conjugate gradient methods [22, 23], Powell's conjugate direction method [24, 25], quasi-Newton methods [26, 27], Gauss-Newton method [28, 29], and stochastic gradient descent methods [30, 31]. Similarity measurement methods, such as the sum of squared differences [32], the sum of absolute differences, cross-correlation [33], and mutual information [34], have been proposed.

However, traditional registration methods face real-time challenges. Large amounts of input data must be processed when performing nonrigid registration modeling on 3D data

with high resolution, a step that requires a long time. The optimization part usually uses an iterative algorithm, thereby further increasing the total time needed to obtain the final result [35].

With the development of deep learning, several researchers have proposed deep neural networks to learn features of unregistered images. Registration methods based on deep learning can be supervised [36, 37] or unsupervised [38–40]. Most supervised registration methods rely on anatomical labels. However, marking anatomical labels is difficult, a step that not only consumes a lot of time of experts but also sometimes hardly guarantees accuracy. In practice, supervised registration methods are restricted. In their place, some scholars have proposed unsupervised medical image registration methods.

Several unsupervised medical image registration methods have been proposed. *VoxelMorph*, a recently proposed unsupervised learning-based method for deformable medical image registration, has a better registration accuracy and a faster speed than other registration methods [41]. Some researchers combined the advantages of classical methods and learning-based methods to produce a probabilistic generative model and derive a diffeomorphic inference algorithm [42]. A registration method called fast image registration (FAIM) for 3D medical image registration has been proposed. Compared with the registration network based on U-net, FAIM has fewer trainable parameters to obtain a higher registration accuracy. In addition, FAIM has less irreversible regions because of the penalty loss for negative Jacobian determinants [43]. Some scholars recently proposed the Probab-Mul registration method, which is a feature-level probability model that can perform regularization on the hidden layers of two deep convolutional neural networks [44].

The focus of the present study is mainly on the accuracy of registration methods and barely on the uncertainty of their registration results. The uncertainty of registration results is very important in clinical applications as it can be used to judge whether the registration result is meaningful. For example, if a model is modeling normal human brain images, it will never see abnormal brain images that have brain tumors, malformations, and edema. When the uncertainty of a registration result is higher than a certain threshold, the source image can be judged as an abnormal brain image. During testing, the Bayesian neural network can obtain the uncertainty of results. Bayesian neural networks are used in autonomous driving, classification tasks, and segmentation tasks. Some researchers recently applied Bayesian neural networks to image registration. Deshpande et al. employed a Bayesian deep learning approach for deformable medical image registration. They reported that this approach has a better performance than existing state-of-the-art approaches [45]. Khawaled et al. developed a fully Bayesian framework for unsupervised deep learning-based deformable image registration. Their approach provided better estimates of the deformation field, thereby improving registration accuracy [46]. However, these aforementioned methods do not sufficiently consider and discuss the

uncertainty of registration results. Furthermore, they are suitable for 2D images only.

In this paper, a method based on Bayesian fully convolutional networks is proposed for image registration. The proposed method generates a geometric uncertainty map to measure the uncertainty of registration results. Thus, when the source image obtained is abnormal data, the model will provide a hint that the source image is problematic instead of immediately accepting the registration result of the model. Group normalization (GN) is also added in networks. GN groups channel similar features into one group. Hence, GN can make the model easier to optimize and converge to improve registration accuracy. The performance of the registration model in evaluating uncertainty is determined.

This paper is organized as follows. Section 2 introduces the principle of the proposed method. Section 3 describes the experimental setup. Section 4 discusses the experimental results. Finally, Section 5 summarizes the results of the study and considers directions for future work.

2. Methods

Figure 1 presents an overview of the proposed method. We used CNN to model the function $g_\gamma(S, T) = u$, where γ is the parameter of the convolutional layers, S is the source image, T is the target image, and u is the displacement field between the source image and the target image. S and T are defined over a 3D spatial domain $\Omega \subset \mathbb{R}^3$. For each voxel $p \in \Omega$, $u(p)$ is the displacement, where the map $\phi = Id + u$ is formed using an identity transform and u . The network takes S and T as the input and uses a set of parameters γ to calculate ϕ . We used a spatial transformation function to warp S to $S \circ \phi$ and evaluate the similarity between $S \circ \phi$ and T . During testing, given the images T and S of the test set, we obtained the registration field by evaluating $g_\gamma(S, T)$.

2.1. Architecture. In this section, the architecture of the convolutional neural network used in the proposed method is described in detail (Figure 2). During training, the moving image and the target image are stacked together as the input fed into the Bayesian fully convolutional network module (BFCNM) [43]. The first layer is inspired by Google's inception module. The purpose of this layer is to compare and capture information on different spatial scales of later registration. Parametric rectified linear unit [47] activation is utilized at the end of each convolution block, and linear activation is employed in the last layer to generate the displacement field. Instead of inserting max-pooling layers, a kernel stride of 2 is used to reduce image size. Three "add" skip connections are present in downsampling and upsampling [43]. The "add" skip connection is conducive to the fusion of upsampling information and its corresponding downsampling information. During the upsampling phase, two Bayesian blocks are used. The Bayesian blocks are composed of a transposed convolutional layer, a convolutional layer, PReLU, a group normalization layer, and a Dropout layer. The detail of the Bayesian block is shown in Figure 2(b). In this paper, Monte Carlo Dropout (MC-

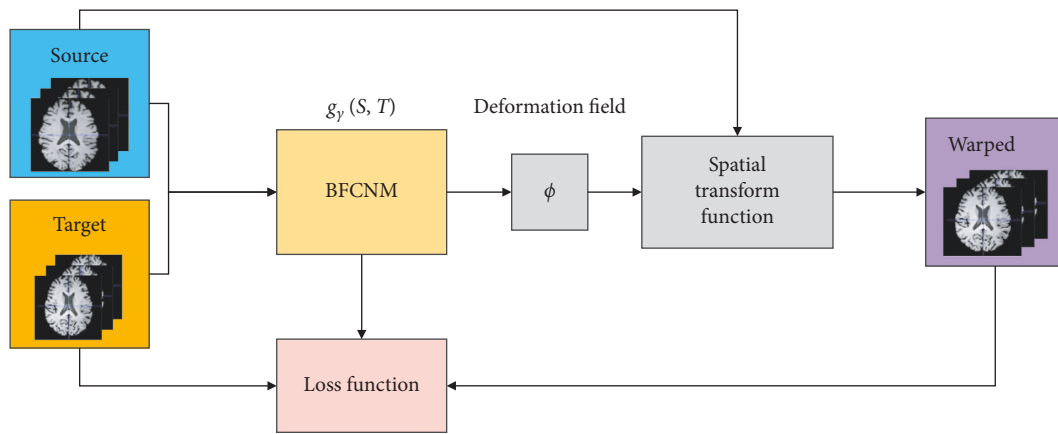


FIGURE 1: Overview of the method.

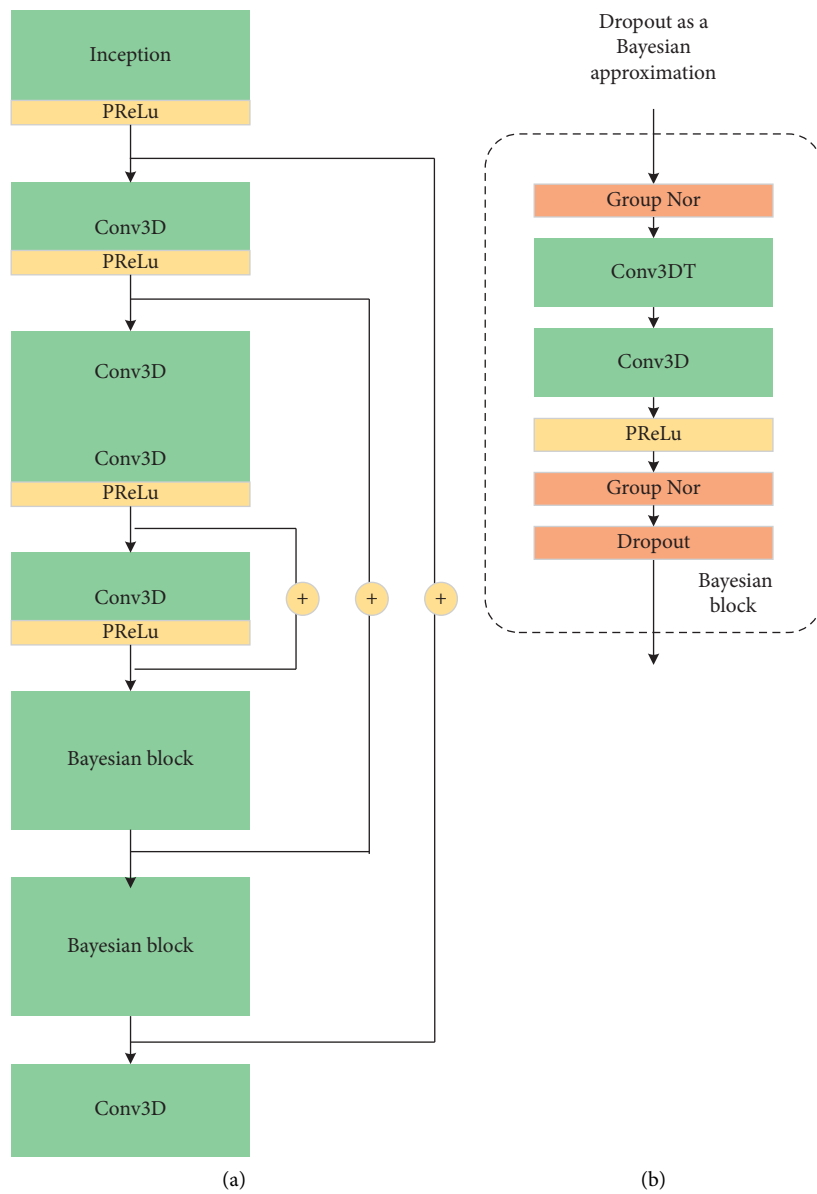


FIGURE 2: Convolution network architecture implementing BFCNM ($g_y(S, T)$). (a) Highest-level view, showing sequential Conv3D and Bayesian block. (b) Details of Bayesian block.

Dropout) is introduced; it is interpreted as a Bayesian approximation of Gaussian processes.

2.2. Spatial Transformation Function. The spatial transformation function uses the ϕ generated by BFCNM to resample S and obtain the warped image $S \circ \phi$. The proposed method learns the optimal parameter values by minimizing the difference between $S \circ \phi$ and T . A differentiable operation is constructed on the basis of the spatial transformer network [41, 48] via the standard gradient-based method to calculate $S \circ \phi$. For each voxel p , a voxel position $p' = p + u(p)$ is calculated in the source image. The image values are only defined in integer positions. Thus, linear interpolation is performed at eight adjacent voxels:

$$S \circ \phi(p) = \sum_{q \in Z(p')} S(q) \prod_{d \in \{x, y, z\}} (1 - |p'_d - q_d|). \quad (1)$$

where $Z(p')$ is the voxel neighbors of p' and d is iterated in the dimension of Ω . Errors can be backpropagated during the optimization process because gradients or subgradients can be calculated.

2.3. Loss Function. The total training loss is the sum of an image dissimilarity term L_{image} and the regularization terms, as shown in Table 1. The loss function [43] is defined as follows:

$$L_{\text{total}} = L_{\text{image}}(S, T) + \alpha R_1(u) + \beta R_2(u). \quad (2)$$

The main loss L_{image} with cross-correlation (CC) in this paper is for the similarity between the warped source image and the target image. The definition of CC is as follows:

$$\text{CC}(t, s \circ \phi) = \frac{(\sum_{x \in \Omega} (t(x) - \bar{t}(x))(s \circ \phi(x) - \overline{s \circ \phi}(x)))^2}{(\sum_{x \in \Omega} (t(x) - \bar{t}(x))^2)(\sum_{x \in \Omega} (s \circ \phi(x) - \overline{s \circ \phi}(x))^2)}, \quad (3)$$

where $t(x)$ is the grey value of the target image, $\bar{t}(x)$ is the average grey value of the target image, $s \circ \phi(x)$ is the grey value of the warped image, and $\overline{s \circ \phi}(x)$ is the average grey value of the warped image.

The first regularization term R_1 regularizes the overall smoothness of the predicted displacements. The parameter of the regular term is α , and its value is always 1. The purpose of the second regularization is to penalize transformations that have many negative Jacobian determinants. The parameter of the regular term is β . The transformations of all nonnegative Jacobian determinants will not be penalized. If the Jacobian determinant is negative, then the transformation result will be folded, which is physically unrealistic.

2.4. Group Normalization. Group normalization (GN) is a feature-normalization technique that is inserted into the architecture of deep neural networks as a trainable process. The purpose of GN is to reduce internal covariant shifts. With training iterations, the distribution of features often continuously changes. Under this condition, the parameters in the

TABLE 1: Loss and regularization functions used.

$L_{\text{image}}(S, T): 1 - \text{CC}(S \circ \phi^{-1}, T)$
Regularization : $R_1(u) = \ Du\ _2$
Regularization : $R_2(u) = 0.5(\det(D\phi^{-1}) - \det(D\phi^{-1}))$

convolutional layer must be continuously updated to adapt to the changes in distribution. GN normalizes the feature to a fixed distribution (mean value is zero, and the standard deviation is 1) and then adjusts the feature to an ideal distribution, which is learned in the training process [48].

Here, x is the feature computed by a layer, and i is an index. In the case of 3D images, $i = (i_N, i_C, i_D, i_H, i_W)$ is a 5D vector indexing the features in (N, C, D, H, W) order, where N is the batch axis; C is the channel axis; and D, H , and W are the spatial depth, height, and width axes, respectively.

Formally, the group normalization layer must compute for mean μ and standard deviation σ in a set S_i . S_i is a group and defined as follows:

$$S_i = \left\{ k | k_N = i_N, \lfloor \frac{k_C}{C/G} \rfloor = \lfloor \frac{i_C}{C/G} \rfloor \right\}, \quad (4)$$

where G is the number of groups, which is a predefined hyperparameter; C/G is the number of channels in each group; $\lfloor \cdot \rfloor$ represents floor operation; $\lfloor k_C / (C/G) \rfloor = \lfloor i_C / (C/G) \rfloor$ means that the indexes i and k are in the same group of channels, assuming that each group of channels is stored in sequential order along the C axis; and S_i contains all voxels along the (D, H, W) axes and along with a group of (C/G) channels.

The mean μ_i and standard deviation σ_i of S_i are computed as follows:

$$\begin{aligned} \mu_i &= \frac{1}{m} \sum_{k \in S_i} x_k, \\ \sigma_i &= \sqrt{\frac{1}{m} \sum_{k \in S_i} (x_k - \mu_i)^2 + \varepsilon}, \end{aligned} \quad (5)$$

where ε is a small constant, and m is the size of set S_i . GN then performs the following computation:

$$\hat{x}_i = \frac{1}{\sigma_i} (x_i - \mu_i), \quad (6)$$

GN learns a per-channel linear transform to compensate for the possible loss of representational ability:

$$y_i = \gamma \hat{x}_i + \beta, \quad (7)$$

where γ and β are trainable scale and shift, respectively. Given the S_i in (4), the GN layer is defined by equations (5)–(7). Specifically, the voxels in the same group are normalized by the same μ_i and σ_i . GN also learns the γ and β of each channel.

2.5. Bayesian Neural Network. In this section, the registration network based on Bayesian inference is introduced. The credibility of the results is important in solving medical

problems. Several researchers proposed a Bayesian neural network by studying the uncertainty of deep learning [49, 50]. The Bayesian neural network is a statistical model derived from the perspective of probability. The parameters in the model are initialized by an a priori distribution, and the parameters are further optimized by Bayesian inference. In the Bayesian neural network, given the data set D and weight W , the dataset D contains data X and label Y . The goal of Bayesian neural network training is to optimize the parameters, that is, to seek the posterior distribution of weight W . According to Bayesian criterion, the posterior distribution of weight W is written as

$$\begin{aligned} p(W|D) &= \frac{p(D|W)p(W)}{p(D)} \\ &= \frac{p(y|x, W)p(W)}{p(y|x)}, \end{aligned} \quad (8)$$

where x and y are the data in the training set and the corresponding label, respectively, and $p(W)$ is the initial value of the parameter (i.e., the prior distribution). The posterior distribution of the labels predicted by the Bayesian neural network can then be calculated as follows [51]:

$$\begin{aligned} p(y^*|x^*, D) &= \mathbb{E}_{p(W|D)} [p(y^*|x^*, W)] \\ &= \int p(y^*|x^*, W)p(W|D)dW. \end{aligned} \quad (9)$$

In equation (9), the weight parameter W in the network is used to predict the unknown distribution of label y^* . In the Bayesian neural network model, the solution of the posterior distribution $p(W|D)$ of the parameters is the key to the entire model. However, this solution is computationally intractable for neural networks of any size. Therefore, many researchers use approximate methods to obtain the solution [52, 53].

A common approach is to use variational inference to approximate the posterior distribution of the weights. This method introduces the variational distribution $q_\theta(W)$ of weight w , which is parameterized on θ . The approximate posterior distribution $q_\theta(W)$ is obtained by minimizing the Kullback-Leibler (KL) divergence between $q_\theta(W)$ and the true posterior distribution $p(W|D)$.

$$\text{KL}(q_\theta(W) \| p(W|D)). \quad (10)$$

Minimizing KL divergence is equivalent to minimizing the Negative Evidence Lower Bound (NELBO):

$$\begin{aligned} \text{NELBO} &= \mathbb{E}_{q_\theta} [-\log p(Y|X, W)] \\ &\quad + \text{KL}(q_\theta(W) \| p(W)) \\ &= - \int q_\theta(W) \log p(Y|X, W) dW \\ &\quad + \text{KL}(q_\theta(W) \| p(W)), \end{aligned} \quad (11)$$

with respect to the variational parameter θ . The first term (commonly referred to as the expected log-likelihood) encourages $q_\theta(W)$ to place its mass on the configurations

of the latent variable that explains the observed data. However, the second term (referred to as prior KL) encourages $q_\theta(W)$ to be similar to the prior distribution $p(W)$, preventing the model from overfitting. The goal is to develop an explicit and accurate approximation for the expectation.

Our approach uses Bernoulli approximating variational inference and Monte Carlo sampling [54]. In practice, Dropout is used for Bayesian neural network approximation.

When Dropout [55] is applied to the output of a layer, the output can be written as

$$a_i^{\text{DO}} = \sigma(z_i \odot (W_i v)). \quad (12)$$

where, for a single K_{i-1} dimensional input v , the i^{th} layer of a neural network with K_i units would output a K_i dimensional activation vector; w_i is the $K_i \times K_{i-1}$ weight matrix; $\sigma(\cdot)$ is the nonlinear activation function; \odot signifies the Hadamard product; z_i is a K_i dimensional binary vector with its elements drawn independently from $z_i^{(k)} \sim \text{Bernoulli}(p_i)$ $k=1, \dots, K_i$; and p_i is the probability of keeping the output activation.

The solution of the posterior distribution $p(W|D)$ of the parameters is further improved after introducing the Bernoulli distribution into the weight parameters of our model. The Monte Carlo sampling method is used to estimate the first item in (11):

$$\begin{aligned} \mathbb{E}_{q_\theta} \log p(Y|X, W) &= \sum_{n=1}^N \int q_\theta(W) \log p(y_n|x_n, W) \\ &= \frac{1}{N} \sum_{n=1}^N \log p(y_n|x_n, \hat{W}_n), \end{aligned} \quad (13)$$

where \hat{W}_n is not the maximum posterior estimation but the random variable realizations from the Bernoulli distribution; and $\hat{W}_n \sim q_\theta(W)$, which is the same as applying Dropout to the weights of the network. For the second item in equation (11) (i.e., KL term), the approximate solution is given in the literature [56]. The KL term has been shown to be equivalent to $\sum_{i=1}^L \|W_i\|_2^2$. Thus, equation (11) can be rewritten as

$$\text{NELBO} = -\frac{1}{N} \sum_{n=1}^N \log p(y_n|x_n, \hat{W}_n) + \sum_{i=1}^L \|W_i\|_2^2. \quad (14)$$

Equation (14) is the unbiased estimation of equation (11). Interestingly, it is the same as the loss function used in standard neural networks with L_2 weight regularization, and Dropout is applied to all weights of the network. Therefore, training such a neural network with stochastic gradient descent has the same effect as minimizing the KL term in (10). This scheme is similar to a Bayesian neural network and can generate a set of parameters that can best explain the observed data while preventing overfitting.

Predictions in this model follow (9) replacing the posterior $p(W|D)$ with the approximate posterior $q_\theta(W)$. The integral can be approximated with Monte Carlo integration [51, 54]:

$$\begin{aligned}
p(y^*|x^*, \mathcal{D}) &\approx \int p(y^*|x^*, W)q_\theta(W)dW \\
&\approx \frac{1}{T} \sum_{t=1}^T \log p(y_n|x_n, \hat{W}_t) \\
&\approx p_{MC}(y^*|x^*),
\end{aligned} \tag{15}$$

where $\hat{W}_t \sim q_\theta(W)$, which means that, at test time, the Dropout layers are kept active to keep the Bernoulli distribution over the network weights. This integration is referred to as the Monte Carlo Dropout.

The Monte Carlo Dropout reflects the need to conduct multiple forward propagation processes on the same input. In this manner, the output of “different network structures” can be obtained under the action of Dropout during testing. The prediction results and the uncertainty of the model can be obtained by calculating the average and statistical variance of these outputs. The advantage of Bayesian deep learning is that Monte Carlo Dropout can give a prediction value and the confidence of the predicted value.

2.6. Measuring Model Uncertainty. Uncertainties in a network are a measure of how certain the model is with its prediction. In general, Bayesian modeling has two types of uncertainty. Model uncertainty, also known as Epistemic uncertainty, measures what the model does not know owing to the lack of training data. This uncertainty can be reduced with more data. During testing, model uncertainty can measure whether the testing data exists in the distribution of the training data. Aleatoric uncertainty measures the noise inherent in the observation data and cannot be reduced by collecting more data [51].

By computing the result of stochastic forward passes of the Bayesian neural network, the model’s confidence of its output can be estimated. In this paper, the mean μ and the standard deviation σ of all displacements produced by Monte Carlo sampling are calculated. The mean μ is used in the registration image, whereas the standard deviation σ provides an estimate of the uncertainty of registration results. The mean μ of the displacement fields is calculated as follows:

$$\mu = \frac{1}{M} \sum_{i=1}^M y_i, \tag{16}$$

where M represents the number of Monte Carlo sampling ($M = 48$ in this paper). y_i represents the displacement field sampled by i th. After calculating the mean value of the displacement fields, the standard variance of the displacement fields can be calculated as follows:

$$\sigma = \sqrt{\frac{1}{M-1} \sum_{i=1}^M (y_i - \mu)^2}, \tag{17}$$

where σ can be expressed as the uncertainty of registration results.

3. Experiment

3.1. Experimental Setup. The dataset we adopted herein was created by Arno et al. who based it on a collection of 101 T1-weighted MRIs from healthy subjects [57]. In this paper, we used brain images from the four subsets of Mindboggle101, namely, NKI-RS-22, NKI-TRT-20, MMRR-21, and OASIS-TRT-20, for a total of 83 images. These images are already warped to MNI152 space. Each image had a dimension of $182 \times 218 \times 182$, each of which we truncated to $144 \times 180 \times 144$. In the preprocessing stage, we utilized the FMRIB Software Library (FSL) to perform affine registration on NKI-RS-22, NKI-TRT-20, MMRR-21, and OASIS-TRT-20. We initially normalized the voxel intensity of each brain image and then normalized voxel intensity to 0–255. Finally, we performed a registration test on the five main anatomical regions of the cerebral cortex.

3.2. Evaluation Metrics

3.2.1. Dice Scores. If the registration field ϕ represents an accurate correspondence, then the corresponding anatomical regions in T and $S \circ \phi$ should overlap well. Therefore, we evaluated registration accuracy by using the Dice score. The Dice score is defined as follows [43]:

$$\text{DICE} = \frac{2 * |X \cap Y|}{|X| + |Y|}. \tag{18}$$

3.2.2. Regularized Penalty Folding. We also evaluated the regularity of deformation fields. Specifically, the Jacobian matrix captures the local properties of ϕ around voxel p . We counted all nonbackground voxels where the Jacobian determinant $\det(\nabla\phi) < 0$ is negative [43]:

$$N := \sum \delta(\det(D\phi) < 0), \tag{19}$$

where $\delta(\cdot)$ indicates that if it is true, then the return value is 1.

3.2.3. Uncertainty Evaluation Metrics. We adopted the method proposed in the literature to evaluate uncertainty performance [51]. We used metrics that incorporate the ground truth label, model prediction, and uncertainty value to evaluate the performance of such models in estimating uncertainty. Figure 3 shows the required processing steps to prepare these quantities for our metrics in a registration example. We computed the map of correct and incorrect values by matching the ground truth labels and the model predictions. We converted the uncertainty map into a map of certain and uncertain predictions by setting the uncertainty threshold T , which varies between the minimum and the maximum uncertainty values in the entire test set. The following indicators can reflect the characteristics of a good uncertainty estimator.

Negative predictive value (NPV): in the output of certain results by the model, NPV is the percentage of voxels that is

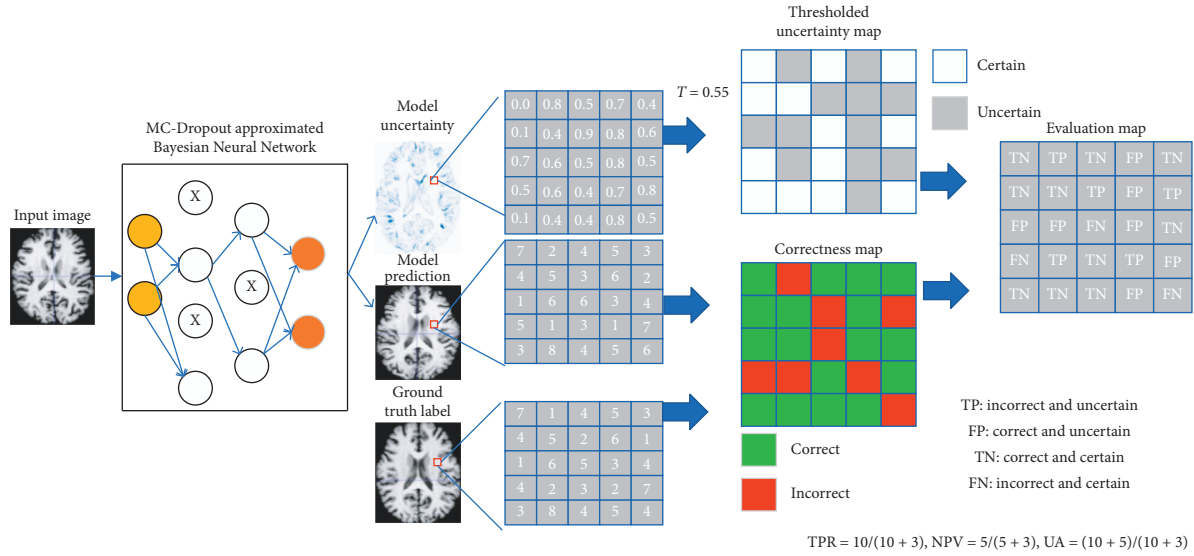


FIGURE 3: Overview of the metrics for the evaluation of the uncertainty quality in a registration example.

correctly predicted and can be written as a conditional probability:

$$NPV = \frac{P(\text{correct, certain})}{P(\text{certain})} = \frac{TN}{TN + FN}. \quad (20)$$

True positive rate (TPR): if a model is making an incorrect prediction, then the proportion of uncertain voxels is called TPR. TPR can be written as a conditional probability:

$$TPR = \frac{P(\text{uncertain, incorrect})}{P(\text{incorrect})} = \frac{TP}{TP + FN}. \quad (21)$$

Uncertainty accuracy (UA): UA is the overall accuracy of uncertainty estimation and can be measured as the ratio of the desired cases explained above (TP and TN) over all possible cases:

$$UA = \frac{P(\text{correct, certain}) + P(\text{uncertain, incorrect})}{P(\text{correct}) + P(\text{incorrect})} \quad (22)$$

$$= \frac{TP + TN}{TP + TN + FP + FN}.$$

Clearly, in all the metrics proposed above, higher values indicate that the model performs better. The values of these metrics depend on the uncertainty threshold.

3.3. Baseline Methods. In the comparative study, we used FSL, a comprehensive library of analytical tools for fMRI, MRI, and diffusion tensor imaging brain imaging data, as the baseline to perform an affine registration experiment with 12 degrees of freedom on the test set. We used the second baseline symmetric normalization (SyN) with mutual information as a similarity measure in the publicly available Advanced Normalization Tools (ANTs) software package [58]. We also tested the recently developed CNN-based methods, namely, VoxelMorph [41], FAIM [43], and Probab-Mul [44], and compared their performance with that of

the proposed method. The hyperparameters of the CNN-based methods were consistent. Finally, we adopted various methods for ablation study. The method that only adds GN was denoted as Our-GN, and the method that only adds Dropout was labeled as Our-DO. These two methods were consistent with our method in terms of hyperparameter settings.

3.4. Implementation. We divided the data set into training and test image sets. The training set consisted of all ordered brain image pairs from the union of the NKI-RS-22, NKI-TRT-20, and MMRR-21 subsets, which comprised 3906 pairs in total. The test set consisted of all ordered pairs from the OASIS-TRT-20 subset with 380 pairs in total. We trained FAIM, VoxelMorph, Probab-Mul, and our method on all pairs of images from the training set and then examined their predicted deformations by using the pairs of images from the testing set.

We implemented our method using Keras [59] with a Tensorflow backend [60]. We used the Adam optimizer. We trained three networks with the same hyperparameters: batch size = 1, learning rate = 10^{-4} , epochs = 10, and $\alpha = 1$.

4. Results and Discussion

In this experiment, we separately trained the proposed networks with different β values. We optimized the parameters by the validation set and reported results in our test set. The predicted deformation field could not guarantee diffeomorphism; therefore, the transformation of irreversible regions caused an image to “fold” on itself. In these regions, the determinant of the Jacobian matrix of the deformation field was negative (Figure 4). However, spatial folding is physically impossible; hence, this phenomenon causes registration errors in clinical applications. The frequency of such errors limits the application of neural networks in image registration.

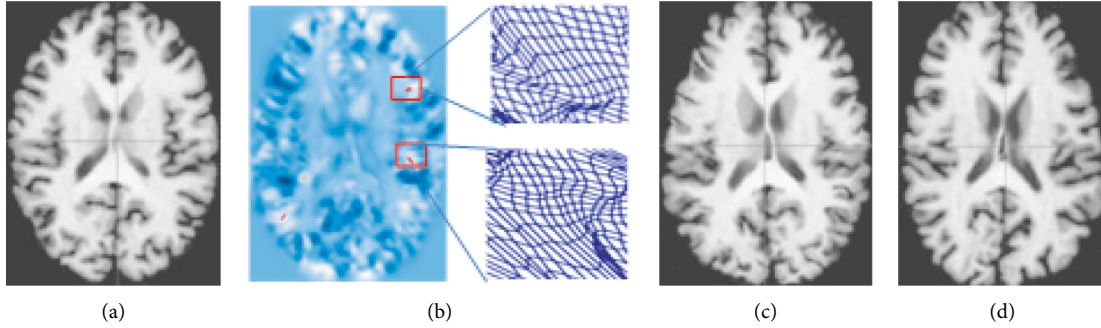


FIGURE 4: The first and last images are the source and target images, respectively, and the third image is the deformed image produced by the method. The second image shows the values of the Jacobian determinant of the predicted deformation with “folding” locations (negative determinant) marked in red. The deformed grids illustrate parts of the deformation. (a) Source, (b) deformation, (c) deformed, and (d) target.

4.1. Dice Scores. The mean Dice scores of the different methods across all predicted labels with their corresponding target labels are shown in Table 2. We selected five scales of regularization strength β from 0 to 10^{-2} . Results showed that FSL was not suitable for fine registration because of its few registration parameters in the affine registration with 12 degrees of freedom. ANT (SyN) is a nonrigid registration method, and its registration accuracy was found to be higher than that of affine registration. The Dice scores of FAIM slightly decreased as β increased, and its Dice scores were higher than those of VoxelMorph. The registration accuracy of Probab-Mul was slightly better than that of FAIM. The proposed method achieved the highest registration accuracy under all β values.

The results of the ablation study revealed that the Dice score of Our-GN was higher than FAIM by 2.8% on average (Table 2). Experimental results showed that GN could improve the accuracy of registration. Moreover, the registration accuracy of Our-DO was slightly lower than that of FAIM (Table 2), illustrating that adding a Dropout layer had little impact on registration accuracy.

When β takes 0, 10^{-5} , 10^{-4} , 10^{-3} , and 10^{-2} , the Dice scores of our method were higher than those of FAIM by 2.63%, 2.80%, 2.84%, 2.50%, and 3.37%, respectively, higher than those of Probab-Mul by 2.31%, 2.49%, 2.55%, 2.05%, and 2.88%, respectively, and higher than those of VoxelMorph by 4.78%, 5.17%, 5.27%, 5.21%, and 5.90%, respectively (Table 2). This result implied that inserting GN layers into the network architecture could indeed enable the network to learn better parameters and could make the network easier to optimize and converge, thereby improving registration accuracy. During training, we set epochs = 10, and each epoch took about 100 min to perform 2900 iterations.

Figure 5 presents the boxplot of the Dice scores of the five main anatomical regions of the cerebral cortex when $\beta = 10^{-3}$. The Dice scores of ANTs in each label were quite different, indicating that ANTs were unstable. The flatness of the boxplots indicated that the stability of the proposed method was comparable to that of other deep learning methods. The proposed method achieved the highest registration accuracy in the five regions of the cerebral cortex.

TABLE 2: Mean Dice scores with different β values.

Mean Dice	$\beta = 0$	$\beta = 10^{-5}$	$\beta = 10^{-4}$	$\beta = 10^{-3}$	$\beta = 10^{-2}$
FSL (Affine)	0.4357	—	—	—	—
ANTs(SyN)	0.5139	—	—	—	—
VoxelMorph	0.5255	0.5203	0.5165	0.5091	0.4908
FAIM	0.5470	0.5440	0.5408	0.5362	0.5161
Probab-Mul	0.5502	0.5471	0.5437	0.5407	0.5210
Our-DO	0.5459	0.5421	0.5380	0.5323	0.5149
Our-GN	0.5729	0.5709	0.5679	0.5591	0.5410
Our method	0.5733	0.5720	0.5692	0.5612	0.5498

Bold values mean the optimal dice score of all methods at the same β value.

Figure 6 shows the mean Dice scores corresponding to the different β values of all methods. The accuracy of the proposed method was relatively consistent with different β values and was higher than that of the other methods.

4.2. Data-Regularized Penalty Folding. Figure 7 visualizes the effects of the second regularization term $R_2(u)$, which directly penalized “foldings” during training. $\beta = 0$ means the regularization was not used, and multiple locations were visible in the transformation whose Jacobian determinants were negative. The number of “foldings” voxels greatly reduced when $\beta = 10^{-5}$. Only several “folding” voxels were observed when $\beta = 10^{-4}$. The number of “folding” voxels was almost eliminated when $\beta = 10^{-3}$.

We listed the mean values of the number of voxels of different methods whose Jacobian determinants were negative in Table 3. The proposed method had a lower number of “foldings” in the predicted deformations as β increased.

4.3. Uncertainty Measure. In this section, the performance of the registration model in estimating uncertainty was evaluated. Figure 8 shows the results of uncertainty evaluation by the proposed method. In the experiment, the Dropout rate of the Dropout layer was set to 0.5. During the test, the Dropout layer was always on, and 48 Monte Carlo samplings were performed. The threshold T of the uncertainty map had an impact on the uncertainty measure. We set the threshold between 0 and 1 with an interval of 0.1. As the threshold increased, the proportion of the uncertain part

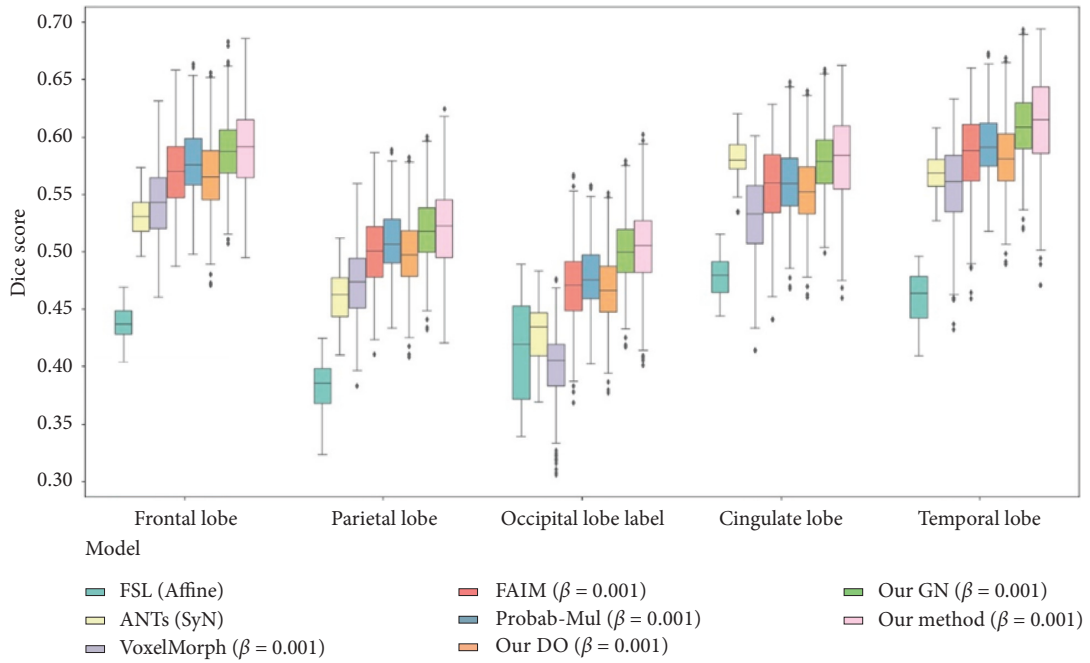


FIGURE 5: Boxplots of Dice scores for five main anatomical structures of the cerebral cortex.

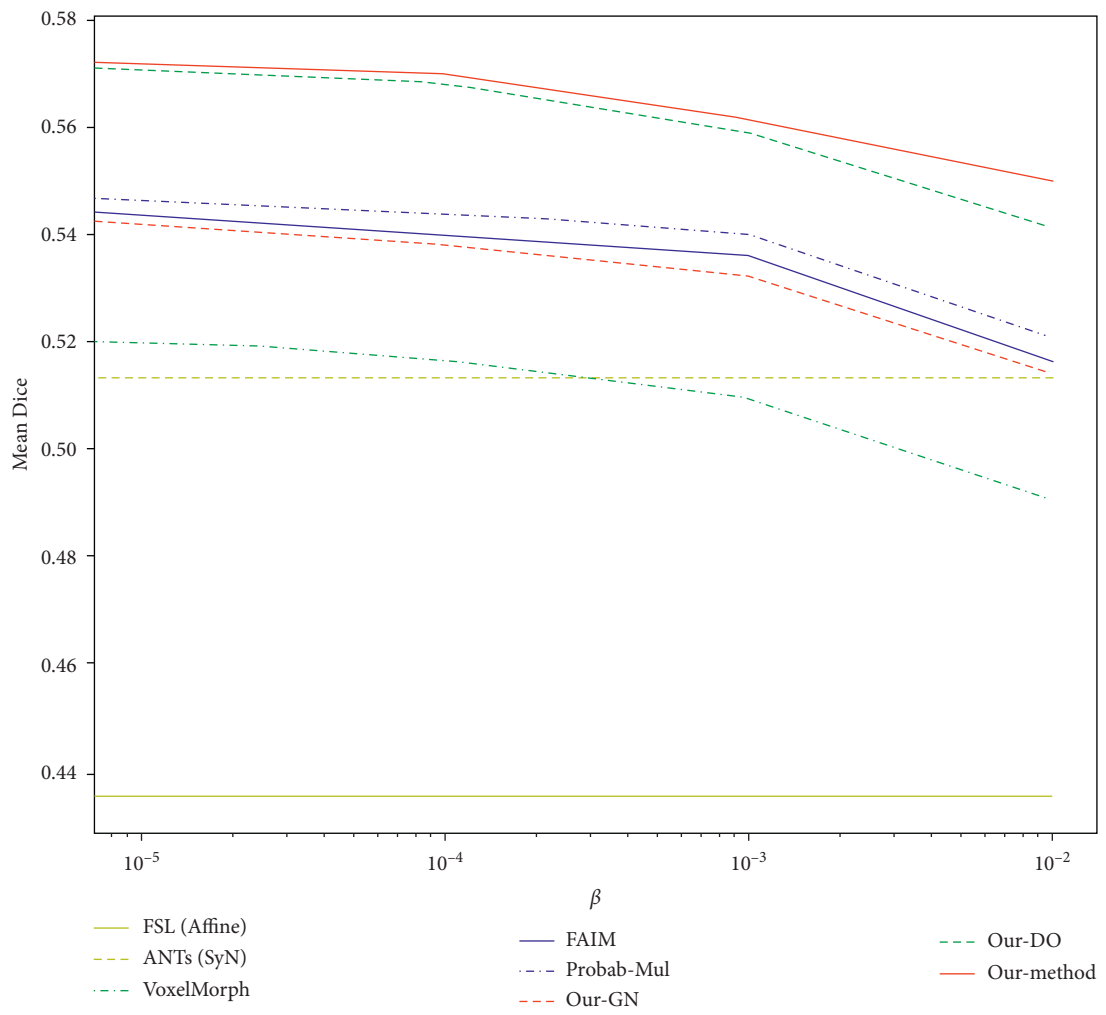


FIGURE 6: Mean Dice score corresponding to the different β of all methods.

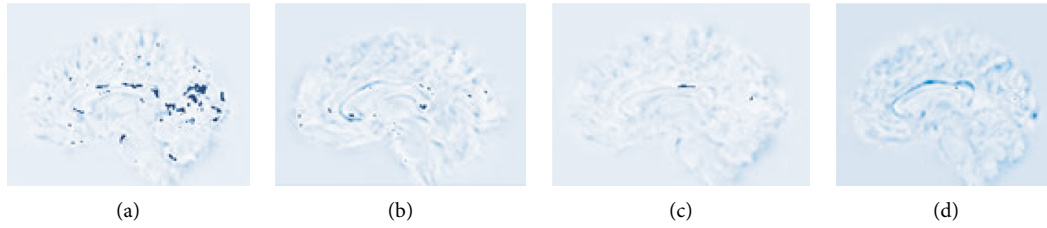


FIGURE 7: Locations where $\det(\nabla\phi) < 0$ (marked in dark blue) with different β shown on one slice. Predictions were done using the proposed method. (a) 0, (b) $\beta = 10^{-5}$, (c) $\beta = 10^{-4}$, and (d) $\beta = 10^{-3}$.

TABLE 3: Mean number of “folding” locations with different β values.

Mean N	$\beta = 0$	$\beta = 10^{-5}$	$\beta = 10^{-4}$	$\beta = 10^{-3}$	$\beta = 10^{-2}$
VoxelMorph	33733	1400	232	60	13
FAIM	39377	1531	234	26	3
Probab-Mul	39905	1700	241	28	6
Our method	39842	1680	240	25	3

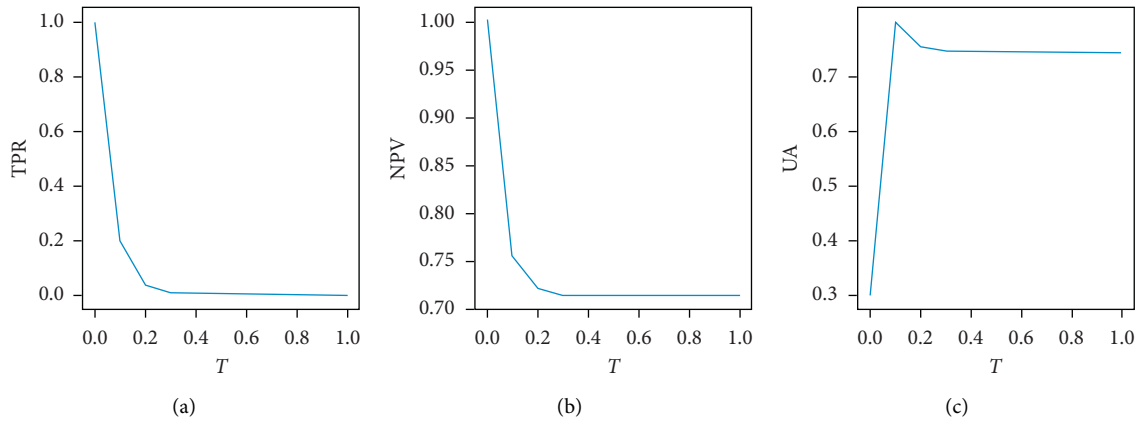


FIGURE 8: Quantitative uncertainty estimation performance for registration task using the evaluation metrics. The abscissa is the threshold (T). The ordinate is negative predictive value (NPV), true positive rate (TPR), and uncertainty accuracy (UA), respectively.

of the uncertainty map decreased; hence, the TPR curve gradually decreased. The maximum value of 0.756 in the NPV measurement was obtained at the threshold of 0.1. This value slightly decreased as the threshold further increased but still greater than 0.72. If the model was certain about its prediction, then the accuracy of the prediction was higher. Uncertainty accuracy was also the largest (0.77) when the threshold was 0.1 and slightly declined as the threshold further increased. It remained greater than 0.712. Uncertainty accuracy is the overall accuracy of uncertainty measurements. It shows the ratio of the cases we desired in all possible cases. The uncertainty accuracy of our model was relatively high (Figure 8).

5. Conclusion

We developed an unsupervised 3D medical image registration method that uses Bayesian fully convolutional networks for registration. The proposed method introduces probability distributions for network weights and obtains the uncertainty of registration results. We introduced GN into the neural network architecture, which is conducive to

the optimization and convergence of the neural network. The experimental results showed that the proposed method can obtain higher registration Dice scores than other state-of-the-art models and achieve an antifolding performance comparable to that of FAIM and VoxelMorph. The proposed method can also estimate the uncertainty of registration results. Although penalty folding can reduce the irreversible area of registration result, it cannot guarantee that the irreversible area is zero. Thus, the nonrigid registration of diffeomorphism with high accuracy is one of our research directions in the future.

Data Availability

All datasets used to support the findings of this study were supplied by the publicly available Mindboggle101 database. The URL to access this data is <https://osf.io/yhkde/>.

Conflicts of Interest

The authors declare that they have no conflicts of interest regarding the publication of this work.

Acknowledgments

The authors thank the National Supercomputing Center in Zhengzhou for the computing resources they provided. This study was funded by the National Natural Science Foundation of China under Grant no. 81772009, Scientific and Technological Research Project of Henan Province under Grant no. 182102310162, Key Research Project of Education Department of Henan Province under Grant no. 21A520042, the Research Fund for Young Scholars of Zhengzhou University under Grant no. F0001297, and Collaborative Innovation Major Project of Zhengzhou under Grant nos. 20XTZX06013 and 20XTZX05015.

References

- [1] B. Zitová and J. Flusser, "Image registration methods: a survey," *Image and Vision Computing*, vol. 21, no. 11, pp. 977–1000, 2003.
- [2] M. L. Uss, B. Vozel, S. K. Abramov, and K. Chehdi, "Selection of a similarity measure combination for a wide range of multimodal image registration cases," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 99, pp. 1–16, 2002.
- [3] A. D. Leow, I. Yanovsky, N. Parikshak et al., "Alzheimer's disease neuroimaging initiative: a one-year follow up study using tensor-based morphometry correlating degenerative rates, biomarkers and cognition," *NeuroImage*, vol. 45, no. 3, pp. 645–655, 2019.
- [4] X.-F. Hu, J.-Q. Zhang, X.-M. Jiang et al., "Amplitude of low-frequency oscillations in Parkinson's disease," *Chinese Medical Journal*, vol. 128, no. 5, pp. 593–601, 2015.
- [5] P. Markelj, D. Tomaževič, B. Likar, and F. Pernuš, "A review of 3D/2D registration methods for image-guided interventions," *Medical Image Analysis*, vol. 16, no. 3, pp. 642–661, 2012.
- [6] Y. Otake, M. Armand, R. S. Armiger et al., "Intraoperative image-based multiview 2D/3D registration for image-guided orthopaedic surgery: incorporation of fiducial-based C-arm tracking and GPU-acceleration," *IEEE Transactions on Medical Imaging*, vol. 31, no. 4, pp. 948–962, 2012.
- [7] M. Ferrant, "Serial registration of intraoperative MR images of the brain," *Medical Image Analysis*, vol. 6, no. 4, pp. 337–359, 2002.
- [8] I. Yanovsky, C. L. Guyader, A. Leow, P. Thompson, and L. Vese, "Unbiased volumetric registration via nonlinear elastic regularization," in *Proceedings of the Miccai Workshop on Mathematical Foundations of Computational Anatomy*, New York, NY, USA, October 2008.
- [9] C. Le Guyader and L. A. Vese, "A combined segmentation and registration framework with a nonlinear elasticity smoother," *Computer Vision and Image Understanding*, vol. 115, no. 12, pp. 1689–1709, 2011.
- [10] C. Davatzikos, "Spatial transformation and registration of brain images using elastically deformable models," *Computer Vision and Image Understanding*, vol. 66, no. 2, pp. 207–222, 1997.
- [11] Y. Wang and L. H. Staib, "Physical model-based non-rigid registration incorporating statistical shape information," *Medical Image Analysis*, vol. 4, no. 1, pp. 7–20, 2000.
- [12] E. D'Agostino, F. Maes, D. Vandermeulen, and P. Suetens, "A viscous fluid model for multimodal non-rigid image registration using mutual information," *Medical Image Analysis*, vol. 7, no. 4, pp. 565–575, 2003.
- [13] M. C. Chiang, D. Leow, D. Klunder et al., "Fluid registration of diffusion tensor images using information theory," *IEEE Transactions on Medical Imaging*, vol. 7, no. 4, pp. 42–456, 2008.
- [14] H. Lombaert, L. Grady, X. Pennec, N. Ayache, and F. Chriet, "Spectral log-demons: diffeomorphic image registration with very large deformations," *International Journal of Computer Vision*, vol. 107, no. 3, pp. 254–271, 2014.
- [15] N. D. Cahill, J. A. Noble, and D. J. Hawkes, "Demons algorithms for fluid and curvature registration," in *Proceedings of the IEEE International Symposium on Biomedical Imaging: from Nano to Macro*, Boston, MA, USA, July 2009.
- [16] J. Ashburner and K. J. Friston, "Voxel-based morphometry-the methods," *NeuroImage*, vol. 11, no. 6, pp. 805–821, 2000.
- [17] D. Rueckert, L. I. Sonoda, C. Hayes, D. L. G. Hill, and M. O. Leach, "Nonrigid registration using free-form deformations: application to breast MR images," *IEEE Transactions on Medical Imaging*, vol. 18, no. 8, pp. 712–721, 2002.
- [18] A. V. Dalca, A. Bobu, N. S. Rost, and P. Golland, "Patch-based discrete registration of clinical brain images," in *Proceedings of the International Workshop on Patch-based Techniques in Medical Imaging*, Athens, Greece, October 2016.
- [19] B. Glocker, N. Komodakis, G. Tziritas, N. Navab, and N. Paragios, "Dense image registration through MRFs and efficient linear programming," *Medical Image Analysis*, vol. 12, no. 6, pp. 731–741, 2008.
- [20] X. Pennec, P. Cachier, and N. Ayache, "Understanding the 'demon's algorithm': 3D non-rigid registration by gradient descent," in *Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention*, Cambridge, UK, September 1999.
- [21] H. J. Johnson and G. E. Christensen, "Consistent landmark and intensity-based image registration," *IEEE Transactions on Medical Imaging*, vol. 21, no. 5, pp. 450–461, 2002.
- [22] G. Postelnicu, L. Zollei, and B. Fischl, "Combined volumetric and surface registration," *IEEE Transactions on Medical Imaging*, vol. 28, no. 4, pp. 508–522, 2019.
- [23] A. A. Joshi, D. W. Shattuck, P. M. Thompson, and R. M. Leahy, "Surface-constrained volumetric brain registration using harmonic mappings," *IEEE Transactions on Medical Imaging*, vol. 26, no. 12, pp. 1657–1669, 2007.
- [24] J. P. W. Pluim, J. B. A. Maintz, and M. A. Viergever, "Information measures in medical image registration," *IEEE Transactions on Medical Imaging*, vol. 23, no. 12, pp. 1508–1516, 2004.
- [25] R. Gan, A. Chung, and S. Liao, "Maximum distance-gradient for robust image registration," *Medical Image Analysis*, vol. 12, no. 4, pp. 452–468, 2008.
- [26] M. C. Seiler and F. A. Seiler, "Numerical recipes in C: the art of scientific computing," *Risk Analysis*, vol. 9, no. 3, pp. 415–416, 1989.
- [27] R. H. Byrd, J. Nocedal, and Y. X. Yuan, "Global convergence of a CASS of Quasi-Newton methods on convex problems," *SIAM Journal on Numerical Analysis*, vol. 24, no. 5, pp. 171–1190, 1987.
- [28] B. T. T. Yeo, T. Vercauteren, P. Fillard, J. M. Peyrat, and O. Clatz, "DT-REF in D: diffusion tensor registration with exact finite-strain differential," *IEEE Transactions on Medical Imaging*, vol. 28, no. 12, pp. 1914–1928, 2019.
- [29] B. T. T. Yeo, M. R. Sabuncu, T. Vercauteren, N. Ayache, B. Fischl, and P. Golland, "Spherical demons: fast diffeomorphic landmark-free surface registration," *IEEE Transactions on Medical Imaging*, vol. 29, no. 3, pp. 650–668, 2010.

- [30] P. Unser and M. Unser, "Optimization of mutual information for multiresolution image registration," *IEEE Transactions on Image Processing*, vol. 9, no. 12, pp. 2083–2099, 2000.
- [31] S. Klein, J. P. W. Pluim, M. Staring, and M. A. Viergever, "Adaptive stochastic gradient descent optimisation for image registration," *International Journal of Computer Vision*, vol. 81, no. 3, pp. 227–239, 2009.
- [32] S. Ekström, F. Malmberg, H. Ahlström, J. Kullberg, and R. Strand, "Fast graph-cut based optimization for practical dense deformable registration of volume images," *Computerized Medical Imaging and Graphics*, vol. 84, Article ID 101745, 2020.
- [33] B. Avants, C. Epstein, M. Grossman, and J. Gee, "Symmetric diffeomorphic image registration with cross-correlation: evaluating automated labeling of elderly and neurodegenerative brain," *Medical Image Analysis*, vol. 12, no. 1, pp. 26–41, 2008.
- [34] H. Y. Zhang, J. W. Zhang, and J. Z. Sun, "Registration method for CT-MR image based on mutual information," *Transactions of Tianjin University*, vol. 13, no. 3, pp. 226–230, 2007.
- [35] V. Villena-Martinez, S. Oprea, and M. Saval-Calvoet, "When deep learning meets data alignment: a review on deep registration networks (DRNs)," 2020, <https://arxiv.org/abs/2003.03167>.
- [36] K. A. J. Eppenhof and J. P. W. Pluim, "Pulmonary CT registration through supervised learning with convolutional neural networks," *IEEE Transactions on Medical Imaging*, vol. 38, no. 5, pp. 1097–1105, 2019.
- [37] B. Wang, Y. Lei, S. Tian et al., "Deeply supervised 3D fully convolutional networks with group dilated convolution for automatic MRI prostate segmentation," *Medical Physics*, vol. 46, no. 4, pp. 1707–1718, 2019.
- [38] B. D. de Vos, F. F. Berendsen, M. A. Viergever, H. Sokooti, M. Staring, and I. Išgum, "A deep learning framework for unsupervised affine and deformable image registration," *Medical Image Analysis*, vol. 52, pp. 128–143, 2019.
- [39] V. Kearney, S. Haaf, A. Sudhyadhom, G. Valdes, and T. D. Solberg, "An unsupervised convolutional neural network-based algorithm for deformable image registration," *Physics in Medicine and Biology*, vol. 63, no. 18, Article ID 185017, 2018.
- [40] G. Wu, M. Kim, Q. Wang, B. C. Munsell, and D. Shen, "Scalable high-performance image registration framework by unsupervised deep feature representations learning," *IEEE Transactions on Biomedical Engineering*, vol. 63, no. 7, pp. 1505–1516, 2016.
- [41] G. Balakrishnan, A. Zhao, M. R. Sabuncu, J. Guttag, and A. V. Dalca, "VoxelMorph: a learning framework for deformable medical image registration," *IEEE Transactions on Medical Imaging*, vol. 38, no. 8, pp. 1788–1800, 2019.
- [42] A. V. Dalca, G. Balakrishnan, J. Guttag, and M. R. Sabuncu, "Unsupervised learning of probabilistic diffeomorphic registration for images and surfaces," *Medical Image Analysis*, vol. 57, pp. 226–236, 2019.
- [43] D. Kuang and T. Schmäh, "FAIM-a ConvNet method for unsupervised 3D medical image registration," *Machine Learning in Medical Imaging*, Springer, Berlin, Germany, 2019.
- [44] L. H. Liu, X. W. Hu, L. Zhu, and P. A. Heng, "Probabilistic multilayer regularization network for Unsupervised3D brain image registration," in *Proceedings of the Medical Image Computing and Computer Assisted Intervention-MICCAI 2019*, Shenzhen, China, October 2019.
- [45] V. S. Deshpande and J. S. Bhatt, "Bayesian deep learning for deformable medical image registration," in *Proceedings of the International Conference on Pattern Recognition and Machine Intelligence*, Tezpur, India, December 2019.
- [46] S. Khawaled and M. Freiman, "Unsupervised deep-learning based deformable image registration: a Bayesian framework," 2020, <https://arxiv.org/abs/2008.03949>.
- [47] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: surpassing human-level performance on ImageNet classification," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Santiago, Chile, December 2015.
- [48] Y. Wu and K. He, "Group normalization," *International Journal of Computer Vision*, vol. 128, no. 3, pp. 742–755, 2020.
- [49] R. M. Neal, "Bayesian learning for neural networks," *Lectures Notes in Statistics*, Springer, New York, NY, USA, 1996.
- [50] A. Kendall and Y. Gal, "What uncertainties do we need in Bayesian deep learning for computer vision?" *Advances in Neural Information Processing Systems*, pp. 5574–5584, Curran Associates, Inc., New York, NY, USA, 2017.
- [51] A. Mobiny, H. V. Nguyen, S. Moulik, N. Garg, and C. C. Wu, "DropConnect is effective in modeling uncertainty of Bayesian deep networks," 2019, <https://arxiv.org/abs/1906.04569>.
- [52] D. J. C. Mackay, "Probable networks and plausible predictions—a review of practical Bayesian methods for supervised neural networks," *Network: Computation in Neural Systems*, vol. 6, no. 3, pp. 469–505, 1995.
- [53] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, "Weight uncertainty in neural networks," 2015, <https://arxiv.org/abs/1505.05424>.
- [54] Y. Gal and Z. Ghahramani, "Bayesian convolutional neural networks with Bernoulli approximate variational inference," 2015, <https://arxiv.org/abs/1603.04467>.
- [55] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from over-fitting," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.
- [56] Y. Gal and Z. Ghahramani, "Dropout as a Bayesian approximation: representing model uncertainty in deep learning," 2015, <https://arxiv.org/abs/1506.02142>.
- [57] K. Arno and T. Jason, "101 labeled brain images and a consistent human cortical labeling protocol," *Frontiers in Neuroence*, vol. 6, no. 171, p. 171, 2012.
- [58] B. B. Avants, N. J. Tustison, G. Song, P. A. Cook, A. Klein, and J. C. Gee, "A reproducible evaluation of ants similarity metric performance in brain image registration," *NeuroImage*, vol. 54, no. 3, pp. 2033–2044, 2011.
- [59] A. Rayne, "Keras," *The School Librarian*, vol. 24, no. 8, pp. 259–261, 2013.
- [60] M. Abadi, A. Agarwal, P. Barham et al., "TensorFlow: large-scale machine learning on heterogeneous distributed systems," 2016, <https://arxiv.org/abs/1603.04467>.