Recursive Cascaded Networks for Unsupervised Medical Image Registration

Shengyu Zhao[2,3]   Yue Dong[2]   Eric I-Chao Chang[3]   Yan Xu[1,3*]

[1]School of Biological Science and Medical Engineering and Beijing Advanced
Innovation Centre for Biomedical Engineering, Beihang University
[2]IIIS, Tsinghua University
[3]Microsoft Research

zsyzzsoft@gmail.com, dongyue8@gmail.com, echang@microsoft.com, xuyan04@gmail.com

Abstract

We present recursive cascaded networks, a general architecture that enables learning deep cascades, for deformable image registration. The proposed architecture is simple in design and can be built on any base network. The moving image is warped successively by each cascade and finally aligned to the fixed image; this procedure is recursive in a way that every cascade learns to perform a progressive deformation for the current warped image. The entire system is end-to-end and jointly trained in an unsupervised manner. In addition, enabled by the recursive architecture, one cascade can be iteratively applied for multiple times during testing, which approaches a better fit between each of the image pairs. We evaluate our method on 3D medical images, where deformable registration is most commonly applied. We demonstrate that recursive cascaded networks achieve consistent, significant gains and outperform state-of-the-art methods. The performance reveals an increasing trend as long as more cascades are trained, while the limit is not observed. Code is available at https://github.com/zsyzzsoft/Recursive-Cascaded-Networks.

1. Introduction

Deformable image registration has been studied in plenty of works and raised great importance. The non-linear correspondence between a pair of images is established by pre-

dicting a deformation field under the smoothness constraint. Among traditional algorithms, an iterative approach is commonly suggested [2, 3, 4, 7, 10, 18, 27, 52], where they formulate each iteration as a progressive optimization problem.

Image registration has drawn growing interests in terms of deep learning techniques. A closely related area is optical flow estimation, which is essentially a 2D image registration problem but the flow fields are discontinuous across objects and the tracking is mainly about motion with rare color difference. Occlusions and folding areas requiring a guess are inevitable in optical flow estimation (but not expected in deformable image registration). Automatically generated datasets (e.g., Flying Chairs [24], Flying Things 3D [41]) are of great help for supervising convolutional neural networks (CNNs) in such settings [24, 29, 30, 54, 55]. Some studies also try to stack multiple networks. They assign different tasks and inputs to each cascade in a non-recursive way and train them one by one [30, 45], but their performance approaches a limit with only a few (no more than 3) cascades. On the other hand, cascading may not help much when dealing with discontinuity and occlusions. Thus by intuition, we suggest that cascaded networks with a recursive architecture fits the setting of deformable registration.

Learning-based methods are also suggested as an approach in deformable image registration. Unlike optical flow estimation, intersubject registration with vague correspondence of image intensity is usually demanded. Some initial works rely on the dense ground-truth flows obtained by either traditional algorithms [14, 56] or simulating intrasubject deformations [36, 53], but their performance is restricted due to the limited quality of training data.

Unsupervised learning methods with comparable performance to traditional algorithms have been presented recently [8, 9, 19, 20, 37, 38]. They only require a similarity measurement between the warped moving image and the fixed image, while the gradients can backpropagate through the differentiable warping operation (a.k.a. spatial trans-
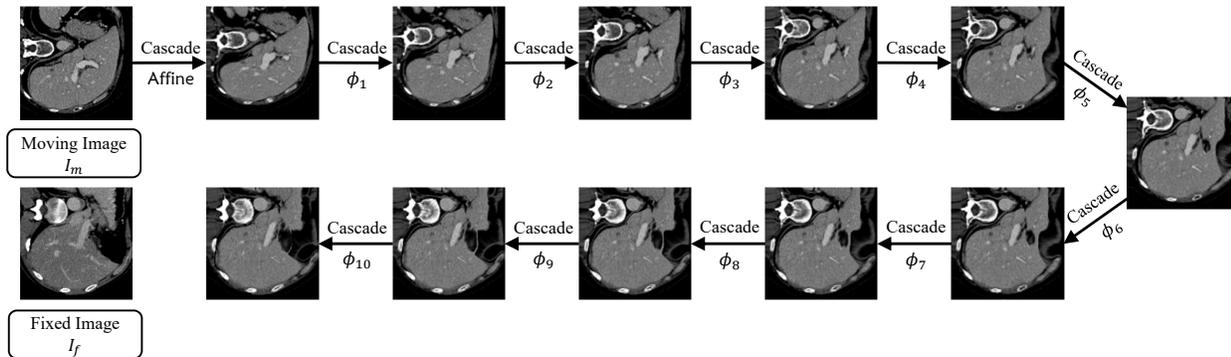
Figure 1. Example of recursive cascaded networks for registering liver CT scans. The moving image is recursively and progressively warped by each of the cascades, finally aligned to the fixed image. Each ϕ_k denotes a predicted flow field, taken both the preceding warped image and the fixed image as inputs. Only image slices are presented but note that the registration is actually performed in 3D.
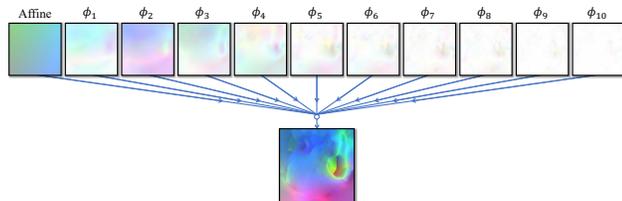

Figure 2. Composition of flow fields, corresponding to the example shown in Figure 1. The final flow prediction is composed of an initial affine transformation and $\phi_1, \ldots, \phi_n$, each of which only performs a rather simple displacement. We can see that the top cascades mainly learn a global alignment, while the bottom cascades play a role of refinement. Flow fields are drawn by mapping the absolute value of the three components (x, y, z) of flow displacements into color channels (R, G, B) respectively. White area indicates zero displacement.

former [32]). However, most proposed networks are enforced to make a straightforward prediction, which proves to be a burden when handling complicated deformations especially with large displacements. DLIR [19] and VTN [37] also stack their networks, though both limited to a small number of cascades. DLIR trains each cascade one by one, i.e., after fixing the weights of previous cascades. VTN jointly trains the cascades, while all successively warped images are measured by the similarity compared to the fixed image. Neither training method allows intermediate cascades to progressively register a pair of images. Those non-cooperative cascades learn their own objectives regardless of the existence of others, and thus further improvement can hardly be achieved even if more cascades are conducted. They may realize that network cascading possibly solves this problem, but there is no effective way of training deep network cascades for progressive alignments.

Therefore, we propose the recursive cascade architecture, which encourages the unsupervised training of an un-

limited number of cascades that can be built on existing base networks, for advancing the state of the art. The difference between our architecture and existing cascading methods is that each of our cascades commonly takes the current warped image and the fixed image as inputs (in contrast to [30, 45]) and the similarity is only measured on the final warped image (in contrast to [19, 37]), enabling all cascades to learn progressive alignments cooperatively. Figure 1 shows an example of applying the proposed architecture built on 10 deformable cascades of the base network VTN.

Conceptually, we formulate the registration problem as determining a parameterized flow prediction function, which outputs a dense flow field based on the input of an image pair. This function can be recursively defined on the warped moving image with essentially the same functionality. Instead of training the function in a straightforward way, the final prediction can be considered a composition of recursively predicted flow fields, while each cascade only needs to learn a simple alignment of small displacement that can be refined by deeper recursion. Figure 2 verifies our conception. Our method also enables the use of shared-weight cascades, which potentially achieves performance gains without introducing more parameters.

To summarize, we present a deep recursive cascade architecture for deformable image registration, which facilitates the unsupervised end-to-end learning and achieves consistent gains independently of the base network; shared-weight cascading technique with direct test-time improvement is developed as well. We conduct extensive experiments based on diverse evaluation metrics (segmentations and landmarks) and multiple datasets across image types (liver CT scans and brain MRIs).

2. Related Work

Cascade approaches have been involved in a variety of domains of computer vision, e.g., cascaded pose regression
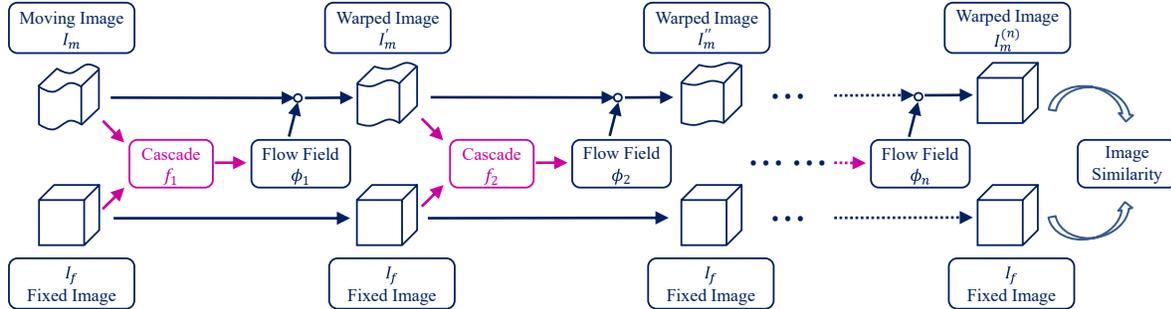
Figure 3. Illustration of our recursive cascade architecture. Circle denotes a composition, where the preceding warped image ($I_m^{(k-1)}$) is reconstructed by the predicted flow field (ϕ_k), resulting in the successive warped image ($I_m^{(k)}$). The unsupervised end-to-end learning is only guided by the image similarity between $I_m^{(n)}$ and I_f, in contrast to previous works.

progressively refines a pose estimation learned from supervised training data [23], cascaded classifiers speed up the process of object detection [25].

Deep learning also benefits from cascade architectures. For example, deep deformation network [57] cascades two stages and predicts a deformation for landmark localization. Other applications include object detection [13], semantic segmentation [17], and image super-resolution [16]. There are also several works specified to medical images, e.g., 3D image reconstruction for MRIs [6, 49], liver segmentation [46] and mitosis detection [15]. Note that shallow, non-recursive network cascades are usually proposed in those works.

In respect of registration, traditional algorithms iteratively optimize some energy functions in common [2, 3, 4, 7, 10, 18, 27, 52]. Those methods are also recursive in general, i.e., similarly functioned alignments with respect to the current warped images are performed during iterations. Iterative Closest Point is an iterative, recursive approach for registering point clouds [12, 58], where the closest pairs of points are matched at each iteration and a rigid transformation that minimizes the difference is solved. In deformable image registration, most traditional algorithms basically works like this but in a much more complex way. Standard symmetric normalization (SyN) [4] maximizes the cross-correlation within the space of diffeomorphic maps during iterations. Optimizing free-form deformations using B-spline [48] is another standard approach.

Learning-based methods are presented recently. Supervised methods entail much effort on the labeled data that can hardly meet the realistic demands, resulting in the limited performance [14, 56, 36, 53]. Unsupervised methods are proposed to solve this problem. Several initial works shows the possibility of unsupervised learning [19, 20, 38, 50], among which DLIR [20] performs on par with the B-spline method implemented in SimpleElastix [40] (a multilanguage extension of Elastix [35], which is selected as one

of our baseline methods). VoxelMorph [8] and VTN [37] achieve better performance by predicting a dense flow field using deconvolutional layers [44], whereas DLIR only predicts a sparse displacement grid interpolated by a third order B-spline kernel. VoxelMorph only evaluates their method on brain MRI datasets [8, 9], but shown deficiency on other datasets such as liver CT scans by later work [37]. Additionally, VTN proposes an initial convolutional network which performs an affine transformation before predicting deformation fields, leading to a truly end-to-end framework by substituting the traditional affine stage.

State-of-the-art VTN and VoxelMorph are selected as our base networks, and the suggested affine network is also integrated as our top-level cascade. To our knowledge, none of those work realizes that training deeper cascades advances the performance for deformable image registration.

3. Recursive Cascaded Networks

Let I_m, I_f denote the moving image and the fixed image respectively, both defined over d-dimensional space Ω. A flow field is a mapping $\phi : \Omega \to \Omega$. For deformable image registration, a reasonable flow field should be continuously varying and prevented from folding. The task is to construct a flow prediction function F which takes I_m, I_f as inputs and predicts a dense flow field that aligns I_m to I_f.

We cascade this procedure by recursively performing registration on the warped image. The warped image I_m' is exactly the composition of the flow field and the moving image, namely

$$I_m' = \phi \circ I_m. \tag{1}$$

Conceptually,

$$F(I_m, I_f) = \phi \circ F_1(\phi \circ I_m, I_f), \tag{2}$$

where F_1 may be the same as F, but in general a different flow prediction function. This recursion can be infinitely applied in theory.

Following this recursion, the moving image is warped successively, enabling the final prediction (probably with large displacement) to be decomposed into cascaded, progressive refinements (with small displacements). One cascade is basically a flow prediction function (f_k), and the k-th cascade predicts a flow field of

$$\phi_k = f_k(I_m^{(k-1)}, I_f). \tag{3}$$

$I_m^{(k)}$ denotes the moving image warped by the first k cascades. Figure 3 depicts the proposed architecture. Assuming for n cascades in total, the final output is a composition of all predicted flow fields, i.e.,

$$F(I_m, I_f) = \phi_n \circ \cdots \circ \phi_1, \tag{4}$$

and the final warped image is constructed by

$$I_m^{(n)} = F(I_m, I_f) \circ I_m. \tag{5}$$

3.1. Subnetworks

Each f_k is implemented as a convolutional neural network in this paper. Every network is designed to predict a deformable flow field on itself based on the input warped image and the fixed image. $f_1, \ldots f_n$ can be different in network architecture, but surely using a common base network is well-designed enough for convenience. Those cascades may learn different network parameters on each, since one cascade is allowed to learn a part of measurements or perform some type of alignment specifically. Note that the images input to the networks are discretized and so are the output flow fields, thus we treat them by multilinear interpolation (or simply trilinear interpolation for 3D images), and out-of-bound indices by nearest-point interpolation [37].

An architecture similar to the U-Net [31, 47] is widely used for deformable registration networks, such as VTN [37] and VoxelMorph [8]. Such network consists of encoders followed by decoders with skip connections. The encoders help to extract features, while the decoders perform upsampling and refinement, ending with a dense prediction.

For medical images, it is usually the case that two scans can be roughly aligned by an initial rigid (or affine) transformation. VoxelMorph [8] assumes that input images are pre-affined by an external tool, whereas VTN [37] integrates an efficient affine registration network which outperforms the traditional stage. As a result, we also embed the affine registration network as our top-level cascade, which behaves just like a normal one except that it is only allowed to predict an affine transformation rather than general flow fields.

3.2. Unsupervised End-to-End Learning

We suggest that all cascades can be jointly trained by merely measuring the similarity between $I_m^{(n)}$ and I_f to-

gether with regularization losses. Enabled by the differentiable composition operator (i.e., warping operation), recursive cascaded networks can learn to perform progressive alignments cooperatively without supervision. To our knowledge, no previous work achieves good performance by stacking more than 3 deformable registration networks, partly because they train them one by one [19] (then the performance can hardly improve) or they measure the similarity on each of the warped images [37] (then the networks can hardly learn progressive alignments).

Regularization losses are basically the smooth terms of $\phi_1, \ldots, \phi_n$, and thus are necessary. Every predicted flow field is penalized by an L2 variation loss as done in [8, 37]. The affine cascade works with its own regularization losses introduced in VTN [37].

3.3. Shared-Weight Cascading

One cascade can be repetitively applied during recursion. I.e., multiple cascades can be shared with the same parameters, and that is called shared-weight cascading.

After an n-cascade network is trained, we can still possibly apply additional shared-weight cascades during testing. For example, we may replicate all cascades as an indivisible whole by the end of $I_m^{(n)}$, i.e., totally $2n$ cascades are associated with flow prediction functions $f_1, \ldots, f_n, f_1, \ldots, f_n$ respectively. We develop a better approach by immediately inserting one or more shared-weight cascades after each, i.e., totally $r \times n$ cascades are constructed by substituting each f_k by r times of that. This approach will be proved to be effective later in the experiments.

Shared-weight cascading during testing is an option when the quality of output flow fields can be improved by further refinement. However, we note that this technique does not always get positive gains and may lead to over deformation. Recursive cascades only ensure an increasing similarity between the warped moving image and the fixed image, but the aggregate flow field becomes less natural if the images are too perfectly matched.

The reason we do not use shared-weight cascading in training is that shared-weight cascades consume extra GPU memory as large as non-shared-weight cascades during gradient backpropagation in the platform we use (Tensorflow [1]). The number of cascades to train is constrained by the GPU memory, but they would perform better with the allowance of learning different parameters when the dataset is large enough to avoid overfitting.

4. Experiments

4.1. Experimental Settings

We build our recursive cascaded networks mainly based on the network architecture of VTN [37], which is a state-of-the-art method for deformable image registration. Note

that VTN already stacks a few cascades of their deformable subnetworks, and a single cascade is being used as our base network. Up to 10-cascade VTN (excluding the affine cascade) is jointly trained using our proposed method. To show the generalizability of our architecture, we also choose VoxelMorph [9] as another base network. We train up to 5-cascade VoxelMorph, because each cascade of VoxelMorph consumes more resources.

We evaluate our method on two types of 3D medical images: liver CT scans and brain MRI scans. For liver CT scans, we train and test recursive cascaded networks for pairwise, subject-to-subject registration, which stands for a general purpose of allowing the fixed image to be arbitrary. For brain MRI scans, we follow the experimental setup of VoxelMorph [8], where each moving image is registered to a fixed atlas, called atlas-based registration. Both settings are common in medical image registration.

Implementation. Inherited from the implementation of VTN [37] using Tensorflow 1.4 [1] built with a custom warping operation, the correlation coefficient is used as the similarity measurement, while the ratios of regularization losses are kept the same as theirs. We train our model using a batch size of 4, on 4 cards of 12G NVIDIA TITAN Xp GPU. The training stage runs for 10^5 iterations with the Adam optimizer [33]. The learning rate is initially 10^{-4} and halved after 6×10^4 steps and again after 8×10^4 steps.

Baseline Methods. VTN [37] and VoxelMorph [8] are state-of-the-art learning-based methods. We cascade their base networks and also compare with the original systems. Besides, we also compare against SyN [4] (integrated in ANTs [5] together with the affine stage) and B-spline [48] (integrated in Elastix [35] together with the affine stage), which are shown to be the top-performing traditional methods for deformable image registration [8, 34, 37]. We run ANTs SyN and Elastix B-spline with the parameters recommended in VTN [37].

Evaluation Metrics. We quantify the performance by the Dice score [22] based on the segmentation of some anatomical structure, between the warped moving image and the fixed image, as done in [8, 19]. The Dice score of two regions A, B is formulated as

$$\mathrm{Dice}(A, B) = 2 \cdot \frac{|A \cap B|}{|A| + |B|}. \tag{6}$$

Perfectly overlapped regions come with a Dice score of 1. The Dice score explicitly measures the coincidence between two regions and thereby reflects the quality of registration. If multiple anatomical structures are annotated, we compute the Dice score with respect to each and take an average.

In addition, landmark annotations are available in some datasets and can be utilized as an auxiliary metric. We compute the average distance between the landmarks of the fixed image and the warped landmarks of the moving image, also introduced in VTN [37].

4.2. Datasets

For liver CT scans, we use the following datasets:

- MSD [42]. This dataset contains various types of medical images for segmenting different target objects. CT scans of liver tumours (70 scans excluding LiTS), hepatic vessels (443 scans), and pancreas tumours (420 scans) are selected since liver is likely to be included.

- BFH (introduced in VTN [37]), 92 scans.

- SLIVER [28], 20 scans with liver segmentation ground truth. Additionally, 6 anatomical keypoints selected as landmarks are annotated by 3 expert doctors, and we take their average as ground truth.

- LiTS [39], 131 scans with liver segmentation ground truth.

- LSPIG (Liver Segmentation of Pigs, provided by the First Affiliated Hospital of Harbin Medical University), containing 17 pairs of CT scans from pigs, along with liver segmentation ground truth. Each pair comes from one pig with (perioperative) and without (preoperative) 13 mm Hg pneumoperitoneum pressure.

Unsupervised methods are trained on the combination of MSD and BFH with 1025^2 ($1025 = 70 + 443 + 420 + 92$) image pairs in total. SLIVER (20×19 image pairs) and LiTS (131×130 image pairs) are used for regular evaluation, while LSPIG is regarded as a challenging dataset which entails generalizability. Only 34 intrasubject image pairs in LSPIG, each of which comes from a same pig (preoperative to perioperative, or vice versa), are evaluated.

For brain MRI scans, we use the following datasets:

- ADNI [43], 66 scans.

- ABIDE [21], 1287 scans.

- ADHD [11], 949 scans.

- LPBA (LONI Probabilistic Brain Atlas) [51]. This dataset contains 40 scans, each of which comes with segmentation ground truth of 56 anatomical structures.

ADNI, ABIDE, ADHD are used for training, and LPBA for testing. All 56 anatomical structures are evaluated by an average Dice score. For atlas-based registration, the first scan in LPBA is fixed as the atlas in our experiments, which

Method	SLIVER		LiTS	LSPIG	LPBA	Time (sec)	
	Dice	Lm. Dist.	Dice	Dice	Avg. Dice	GPU	CPU
ANTs SyN [4, 5]	0.895 (0.037)	12.2 (5.7)	0.862 (0.055)	0.825 (0.063)	0.708 (0.015)	-	748
Elastix B-spline [35, 48]	0.910 (0.038)	12.6 (6.6)	0.863 (0.059)	0.825 (0.059)	0.675 (0.013)	-	115
VoxelMorph[1] [9]	0.883 (0.034)	14.0 (4.6)	0.831 (0.061)	0.715 (0.090)	0.685 (0.017)	0.20	17
VoxelMorph (reimplem.)[2]	0.913 (0.025)	13.1 (4.7)	0.870 (0.048)	0.833 (0.057)	0.688 (0.015)	0.15	14
5-cascade VoxelMorph	0.944 (0.017)	12.4 (4.9)	0.903 (0.055)	0.849 (0.062)	0.708 (0.015)	0.41	69
3×5-cascade VoxelMorph	0.950 (0.014)	11.9 (4.9)	0.905 (0.065)	0.842 (0.066)	0.715 (0.014)	1.09	201
VTN (ADDD)[3] [37]	0.942 (0.020)	12.0 (4.9)	0.897 (0.049)	0.846 (0.064)	0.701 (0.014)	0.13	26
10-cascade VTN	0.953 (0.014)	10.8 (4.9)	**0.909 (0.060)**	**0.855 (0.060)**	0.716 (0.013)	0.25	87
2×10-cascade VTN	**0.956 (0.012)**	**10.2 (4.7)**	0.908 (0.070)	0.849 (0.063)	**0.719 (0.012)**	0.42	179

Table 1. Comparison among traditional methods (ANTs SyN and Elastix B-spline), our baseline networks (VoxelMorph and VTN), and our proposed recursive cascaded networks with and without shared-weight cascading. $r \times n$-cascade means that every deformable cascade is repetitively applied for r times during testing, using our proposed shared-weight cascading method. For liver datasets (SLIVER, LiTS, and LSPIG), the Dice score measures the overlap of liver segmentations, and Lm. Dist. means an average distance among 6 annotated landmarks. Avg. Dice means an average Dice score among all 56 segmented anatomical structures for the brain dataset LPBA. Standard deviations across instances are in parentheses.

[1] Images for training and testing are pre-affined (as required in VoxelMorph [9]) using ANTs [5].

[2] Reimplemented with an integrated affine network and trained using our method.

[3] Denotes one affine registration subnetwork plus three dense deformable subnetworks [37].

is shown to be without loss of generality later in the atlas analysis.

We carry out standard preprocessing steps referring to VTN [37] and VoxelMorph [8]. Raw scans are resampled into $128 \times 128 \times 128$ voxels after cropping unnecessary area around the target object. For liver CT scans, a simple threshold-based algorithm is applied to find a rough liver bounding box for cropping. For brain MRI scans, skulls are first removed using FreeSurfer [26]. The volumes are visualized for quality control so that seldom badly processed images are manually removed. (An overview of the evaluation datasets is provided in the supplementary material.)

4.3. Results

Table 1 summarizes our overall performance compared with state-of-the-art methods. Running times are approximately the same across datasets, so we test them on SLIVER, with an NVIDIA TITAN Xp GPU and an Intel Xeon E5-2690 v4 CPU. No GPU implementation of ANTs or Elastix has been found, nor in previous works [5, 8, 19, 35, 37]. Figure 4 visualizes those methods on an example in the brain dataset LPBA. (See the supplementary material for more examples.)

As shown in Table 1, recursive cascaded networks outperform the existing methods in all our datasets with significant gains. More importantly, the proposed architecture is independent of the base network, not limited to VTN [37] and VoxelMorph [8]. Although the number of cascades causes linear increments to the running times, a 10-cascade VTN still runs in a comparable (GPU) time to the baseline networks, showing the efficiency of our architecture.
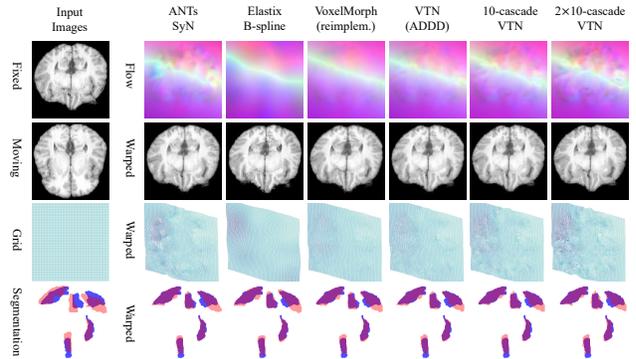

Figure 4. Visualization of an example in the brain dataset LPBA. Grids of deeper color represent lower height. Segmentations of 5 chosen anatomical structures are presented by projection. Blue areas stand for the segmentations of the fixed image, and red areas for the moving image or the warped images.

Number of Cascades. Table 2 presents the results with respect to different number of recursive cascades, choosing either VTN or VoxelMorph as our base network. As shown in the table, recursive cascaded networks achieve consistent performance gains independently of the base network. Our 3-cascade VTN (in Table 2) already outperforms VTN (ADDD) (in Table 1) although they have similar network architectures, mainly because our intermediate cascades learn progressive alignments better with only the similarity loss drawn on the final warped image. Figure 5 plots our results for better illustrating the increasing trend. Note that our architecture requires a linear time increment, but cascading a small-size base network like VTN is quite efficient.

Architecture	SLIVER		LiTS	LSPIG	LPBA	Time (sec)	
	Dice	Lm. Dist.	Dice	Dice	Avg. Dice	GPU	CPU
Affine only	0.794 (0.042)	14.8 (4.7)	0.754 (0.059)	0.727 (0.054)	0.628 (0.017)	0.08	0.4
1-cascade VoxelMorph	0.913 (0.025)	13.1 (4.7)	0.867 (0.050)	0.833 (0.057)	0.688 (0.015)	0.15	14
2-cascade VoxelMorph	0.933 (0.021)	12.8 (4.8)	0.888 (0.048)	0.845 (0.057)	0.699 (0.014)	0.21	27
3-cascade VoxelMorph	0.940 (0.018)	12.6 (5.0)	0.897 (0.049)	0.849 (0.060)	0.706 (0.014)	0.28	40
4-cascade VoxelMorph	0.943 (0.017)	12.5 (5.1)	0.900 (0.052)	**0.851 (0.058)**	0.707 (0.014)	0.35	54
5-cascade VoxelMorph	**0.944 (0.017)**	**12.4 (4.9)**	**0.903 (0.055)**	0.849 (0.062)	**0.708 (0.015)**	0.41	69
1-cascade VTN	0.914 (0.025)	13.0 (4.8)	0.870 (0.048)	0.833 (0.054)	0.686 (0.014)	0.10	10
2-cascade VTN	0.935 (0.020)	12.2 (4.7)	0.891 (0.045)	0.843 (0.061)	0.697 (0.014)	0.12	18
3-cascade VTN	0.943 (0.018)	11.8 (4.7)	0.900 (0.045)	0.850 (0.060)	0.703 (0.014)	0.13	26
4-cascade VTN	0.948 (0.016)	11.6 (4.8)	0.906 (0.047)	0.852 (0.063)	0.708 (0.014)	0.15	35
5-cascade VTN	0.949 (0.015)	11.5 (4.8)	0.908 (0.051)	0.853 (0.064)	0.709 (0.014)	0.17	47
6-cascade VTN	0.951 (0.015)	11.3 (4.9)	**0.910 (0.050)**	0.852 (0.064)	0.712 (0.014)	0.18	57
7-cascade VTN	0.951 (0.015)	11.2 (4.9)	0.908 (0.055)	0.852 (0.061)	0.712 (0.013)	0.20	65
8-cascade VTN	0.952 (0.014)	11.1 (4.7)	**0.910 (0.056)**	0.854 (0.059)	0.714 (0.013)	0.22	75
9-cascade VTN	**0.953 (0.014)**	10.9 (4.7)	**0.910 (0.059)**	0.851 (0.064)	**0.716 (0.013)**	0.23	90
10-cascade VTN	**0.953 (0.014)**	**10.8 (4.9)**	0.909 (0.060)	**0.855 (0.060)**	**0.716 (0.013)**	0.25	87

Table 2. Comparison among different number of recursive cascades. n-cascade means n recursive cascades of the base network, excluding the affine cascade. Standard deviations across instances are in parentheses.

(a) Dice scores on liver datasets.   (b) Landmark distances on SLIVER.   (c) Dice scores on LPBA.
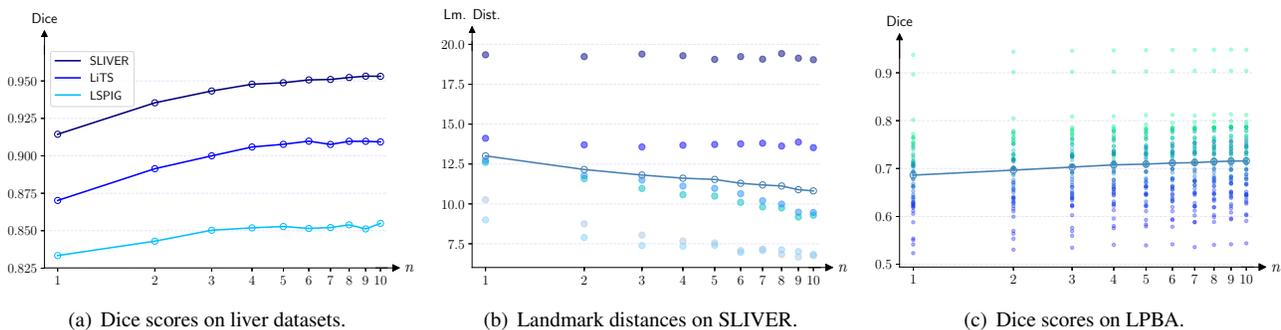
Figure 5. Plot of our results with respect to the number of cascades (n) of the base network VTN, corresponding to the data in Table 2. The x-axes are in log scale since it better reflects the trends. (a) plots the Dice scores evaluated on the liver datasets (SLIVER, LiTS, and LSPIG). (b) plots the landmark distances evaluated on SLIVER, while the distances (average across instances) of 6 landmarks are scattered with respective colors and the line stands for the mean values. (c) plots the Dice scores evaluated on the brain dataset LPBA, while the Dice scores (average across instances) of 56 anatomical structures are scattered respectively and the line stands for the mean values.

Shared-Weight Cascading. Deeper cascades can be directly constructed using weight sharing. As we suggest, an $r{\times}n$-cascade network successively repeats each of the jointly trained n cascades for r times during testing. A linear time increment is also required. This technique ensures an increasing similarity between the warped moving image and the fixed image, but we note that it does not always get positive performance gains.

Table 3 presents the results of shared-weight cascaded networks, together with the image similarity (correlation coefficient is used in this paper). The image similarity is always increasing as we expect. Shallower cascaded networks benefit more from this technique relatively to the deeper ones, since the images are still not well-registered (with relatively low similarity, as shown in the table). Less

expected results on LiTS and LSPIG datasets may imply that this additional technique has a limited generalizability.

Note that shared-weight cascades generally perform worse than their jointly trained counterparts. More than 3 times of shared-weight cascades are very likely to deteriorate the quality (which partly coincides with previous studies), further proving the end-to-end learning to be vital.

Cascades vs. Channels vs. Depth. VoxelMorph (VM) [9] suggests that the number of channels in the convolutional layers can be doubled for a better performance. We compare this variant (VM x2) against the jointly trained 2-cascade VM as well as a shared-weight 2×1-cascade VM, shown in Table 4. VM x2 performs better than the original one as they suggest, but worse than both of our cascade methods.

Cascade	SLIVER			LiTS		LSPIG		LPBA	
	Dice	Lm. Dist.	Similarity	Dice	Similarity	Dice	Similarity	Avg. Dice	Similarity
1×1	0.914 (0.025)	13.0 (4.8)	0.7458 (0.0396)	0.870 (0.048)	0.7386 (0.0468)	0.833 (0.054)	0.7527 (0.0515)	0.686 (0.014)	0.9814 (0.0021)
2×1	0.932 (0.020)	12.6 (5.0)	0.8108 (0.0289)	0.886 (0.048)	0.8045 (0.0376)	0.840 (0.057)	0.8162 (0.0392)	0.694 (0.014)	0.9845 (0.0016)
3×1	0.937 (0.019)	12.5 (5.1)	0.8333 (0.0248)	0.888 (0.050)	0.8272 (0.0336)	0.839 (0.057)	0.8369 (0.0338)	0.695 (0.013)	0.9854 (0.0014)
4×1	0.938 (0.018)	12.5 (5.2)	0.8444 (0.0227)	0.887 (0.053)	0.8381 (0.0314)	0.837 (0.057)	0.8467 (0.0305)	0.692 (0.013)	0.9857 (0.0011)
5×1	0.939 (0.018)	12.5 (5.2)	0.8510 (0.0214)	0.886 (0.056)	0.8446 (0.0300)	0.835 (0.058)	0.8518 (0.0289)	0.686 (0.013)	0.9857 (0.0010)
1×2	0.935 (0.020)	12.2 (4.7)	0.8270 (0.0297)	0.891 (0.045)	0.8209 (0.0367)	0.843 (0.061)	0.8435 (0.0369)	0.697 (0.014)	0.9854 (0.0017)
2×2	0.947 (0.017)	11.6 (4.8)	0.8779 (0.0198)	0.900 (0.049)	0.8715 (0.0282)	0.847 (0.063)	0.8919 (0.0243)	0.701 (0.014)	0.9885 (0.0011)
3×2	0.948 (0.016)	11.5 (4.8)	0.8930 (0.0171)	0.900 (0.054)	0.8865 (0.0254)	0.845 (0.063)	0.9039 (0.0211)	0.697 (0.014)	0.9895 (0.0008)
1×3	0.943 (0.018)	11.8 (4.7)	0.8584 (0.0245)	0.900 (0.045)	0.8535 (0.0318)	0.850 (0.060)	0.8774 (0.0282)	0.703 (0.014)	0.9876 (0.0014)
2×3	0.951 (0.015)	11.2 (4.8)	0.8977 (0.0168)	0.905 (0.052)	0.8927 (0.0246)	0.852 (0.061)	0.9102 (0.0210)	0.710 (0.014)	0.9904 (0.0009)
3×3	0.951 (0.015)	11.1 (4.9)	0.9088 (0.0146)	0.904 (0.058)	0.9037 (0.0225)	0.850 (0.062)	0.9189 (0.0188)	0.711 (0.014)	0.9916 (0.0007)
1×5	0.949 (0.015)	11.5 (4.8)	0.8926 (0.0186)	0.908 (0.051)	0.8893 (0.0254)	0.853 (0.063)	0.9088 (0.0223)	0.709 (0.014)	0.9894 (0.0010)
2×5	0.954 (0.013)	10.8 (4.9)	0.9215 (0.0131)	0.908 (0.061)	0.9184 (0.0198)	0.851 (0.063)	0.9334 (0.0164)	0.715 (0.013)	0.9921 (0.0006)
3×5	0.954 (0.013)	10.6 (5.0)	0.9308 (0.0115)	0.906 (0.067)	0.9278 (0.0182)	0.845 (0.065)	0.9406 (0.0145)	0.715 (0.013)	0.9930 (0.0005)
1×10	0.953 (0.014)	10.8 (4.9)	0.9163 (0.0145)	**0.909 (0.060)**	0.9129 (0.0211)	**0.855 (0.059)**	0.9290 (0.0174)	0.716 (0.013)	0.9918 (0.0008)
2×10	**0.956 (0.012)**	**10.2 (4.7)**	0.9384 (0.0106)	0.908 (0.070)	0.9355 (0.0171)	0.849 (0.062)	0.9471 (0.0132)	**0.719 (0.012)**	0.9942 (0.0005)
3×10	**0.956 (0.012)**	**10.2 (4.7)**	**0.9461 (0.0094)**	0.905 (0.076)	**0.9434 (0.0158)**	0.841 (0.068)	**0.9534 (0.0112)**	0.717 (0.012)	**0.9951 (0.0004)**

Table 3. Results of recursive cascaded networks built on the base network VTN, with different times (1×, 2×, 3×, or more) of shared-weight cascades. Similarity is measured by the correlation coefficient between the warped moving image and the fixed image.

Architecture	SLIVER		LiTS	LSPIG	LPBA
	Dice	Lm. Dist.	Dice	Dice	Avg. Dice
VoxelMorph	0.913 (0.025)	13.1 (4.7)	0.867 (0.050)	0.833 (0.057)	0.688 (0.015)
VM x2	0.922 (0.024)	13.0 (4.9)	0.879 (0.047)	0.839 (0.058)	0.691 (0.015)
VM-double	0.919 (0.025)	12.9 (4.9)	0.877 (0.048)	0.833 (0.059)	0.689 (0.015)
VM xx2	0.925 (0.023)	**12.8 (4.9)**	0.881 (0.047)	0.843 (0.057)	0.693 (0.014)
2×1-cascade VM	0.930 (0.021)	12.8 (4.8)	0.883 (0.051)	0.840 (0.060)	0.697 (0.014)
2-cascade VM	**0.933 (0.021)**	12.8 (4.8)	**0.888 (0.048)**	**0.845 (0.057)**	**0.699 (0.014)**

Table 4. Comparison against other variants of VoxelMorph (VM), including VM x2 (doubling the feature counts of every convolutional layer), VM-double (doubling the number of convolutional layers at each level), and VM xx2 (doubling the encoder-decoder architecture cascade-like).

Method	Avg. Dice		
	Atlas1	Atlas2	Atlas3
ANTs SyN	0.708 (0.015)	0.717 (0.011)	0.707 (0.015)
Elastix B-spline	0.675 (0.013)	0.684 (0.011)	0.670 (0.013)
VoxelMorph	0.688 (0.015)	0.694 (0.010)	0.678 (0.015)
5-cascade VoxelMorph	0.708 (0.015)	0.714 (0.011)	0.702 (0.014)
3×5-cascade VoxelMorph	0.715 (0.014)	0.721 (0.012)	0.713 (0.013)
VTN (ADDD)	0.701 (0.014)	0.709 (0.011)	0.695 (0.015)
10-cascade VTN	0.716 (0.013)	0.723 (0.010)	0.712 (0.013)
2×10-cascade VTN	**0.719 (0.012)**	**0.725 (0.011)**	**0.716 (0.013)**

Table 5. Experiments on different atlases in LPBA.

On the other hand, the number of parameters in VM x2 is 4 times as large as that in VoxelMorph (as well as 2×1-cascade VM), and 2 times as large as that in 2-cascade VM.

However, one may wonder that whether simply deeper networks would do the trick. To this end, we construct VM-double by doubling the number of convolutional layers at each U-net level, and also an encoder-decoder-encoder-decoder architecture denoted VM xx2, which looks similar to a 2-cascade VM except the explicit warping. They have approximately the same amount of parameters compared to the 2-cascade VM, but are outperformed by a considerable margin. This experiment implies that our improvements are essentially based on the proposed recursive cascade architecture rather than simply introducing more parameters.

Atlas Analysis. The performance for atlas-based registration may vary depending on the chosen atlas. As a comparison, we retrain the models on two more (the second and the third) atlases in the LPBA dataset, shown in Table 5. These results indicate that our performance is consistent and robust to the choice of atlas.

5. Discussion

Recursive cascaded networks are quite simple to implement, and also easy to train. We do not tune the ratios of losses when training more cascades, nor the training schedule, showing the robustness of our architecture. If more resources are available or a distributed learning platform is being used, we expect that the performance can be further improved by deeper cascades, and also, training or fine-tuning shared-weight cascades would be an alternative choice. A light-weight base network is also worth an exploration.

A possible limitation of this work would be on the smoothness of the composed field. Theoretically, recursive cascaded networks preserve the image topology as long as every subfield does. However, folding area is common in currently proposed methods and may be amplified during recursion, which brings challenges especially for the use of weight sharing techniques. This problem can be reduced by taking a careful look on the regularization terms, or designing a base network that guarantees invertibility.

6. Conclusion

We present a deep recursive cascade architecture and evaluate its performance in deformable medical image registration. Experiments based on diverse evaluation metrics demonstrate that this architecture achieves significant gains over state-of-the-art methods on both liver and brain datasets. With the superiority of good performance, the general applicability of the unsupervised method, and being independent of the base network, we expect that the proposed architecture can potentially be extended to all deformable image registration tasks.

References

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, pages 265–283, 2016. 4, 5

[2] John Ashburner. A fast diffeomorphic image registration algorithm. *Neuroimage*, 38(1):95–113, 2007. 1, 3

[3] John Ashburner and Karl J Friston. Voxel-based morphometrythe methods. *Neuroimage*, 11(6):805–821, 2000. 1, 3

[4] Brian B Avants, Charles L Epstein, Murray Grossman, and James C Gee. Symmetric diffeomorphic image registration with cross-correlation: evaluating automated labeling of elderly and neurodegenerative brain. *Medical image analysis*, 12(1):26–41, 2008. 1, 3, 5, 6

[5] Brian B Avants, Nick Tustison, and Gang Song. Advanced normalization tools (ants). *Insight j*, 2:1–35, 2009. 5, 6

[6] Khosro Bahrami, Islem Rekik, Feng Shi, and Dinggang Shen. Joint reconstruction and segmentation of 7t-like mr images from 3t mri based on cascaded convolutional neural networks. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 764–772. Springer, 2017. 3

[7] Ruzena Bajcsy and Stane Kovačič. Multiresolution elastic matching. *Computer vision, graphics, and image processing*, 46(1):1–21, 1989. 1, 3

[8] Guha Balakrishnan, Amy Zhao, Mert R Sabuncu, John Guttag, and Adrian V Dalca. An unsupervised learning model for deformable medical image registration. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9252–9260, 2018. 1, 3, 4, 5, 6

[9] Guha Balakrishnan, Amy Zhao, Mert R Sabuncu, John Guttag, and Adrian V Dalca. Voxelmorph: a learning framework for deformable medical image registration. *IEEE transactions on medical imaging*, 2019. 1, 3, 5, 6, 7

[10] M Faisal Beg, Michael I Miller, Alain Trouvé, and Laurent Younes. Computing large deformation metric mappings via geodesic flows of diffeomorphisms. *International journal of computer vision*, 61(2):139–157, 2005. 1, 3

[11] Pierre Bellec, Carlton Chu, Francois Chouinard-Decorte, Yassine Benhajali, Daniel S Margulies, and R Cameron Craddock. The neuro bureau adhd-200 preprocessed repository. *Neuroimage*, 144:275–286, 2017. 5

[12] Paul J Besl and Neil D McKay. Method for registration of 3-d shapes. In *Sensor Fusion IV: Control Paradigms and Data Structures*, volume 1611, pages 586–607. International Society for Optics and Photonics, 1992. 3

[13] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6154–6162, 2018. 3

[14] Xiaohuan Cao, Jianhua Yang, Jun Zhang, Dong Nie, Minjeong Kim, Qian Wang, and Dinggang Shen. Deformable image registration based on similarity-steered cnn regression. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 300–308. Springer, 2017. 1, 3

[15] Hao Chen, Qi Dou, Xi Wang, Jing Qin, and Pheng Ann Heng. Mitosis detection in breast cancer histology images via deep cascaded networks. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016. 3

[16] Zhen Cui, Hong Chang, Shiguang Shan, Bineng Zhong, and Xilin Chen. Deep network cascade for image super-resolution. In *European Conference on Computer Vision*, pages 49–64. Springer, 2014. 3

[17] Jifeng Dai, Kaiming He, and Jian Sun. Instance-aware semantic segmentation via multi-task network cascades. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3150–3158, 2016. 3

[18] Christos Davatzikos. Spatial transformation and registration of brain images using elastically deformable models. *Computer Vision and Image Understanding*, 66(2):207–222, 1997. 1, 3

[19] Bob D de Vos, Floris F Berendsen, Max A Viergever, Hessam Sokooti, Marius Staring, and Ivana Išgum. A deep learning framework for unsupervised affine and deformable image registration. *Medical image analysis*, 52:128–143, 2019. 1, 2, 3, 4, 5, 6

[20] Bob D de Vos, Floris F Berendsen, Max A Viergever, Marius Staring, and Ivana Išgum. End-to-end unsupervised deformable image registration with a convolutional neural network. In *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, pages 204–212. Springer, 2017. 1, 3

[21] Adriana Di Martino, Chao-Gan Yan, Qingyang Li, Erin Denio, Francisco X Castellanos, Kaat Alaerts, Jeffrey S Anderson, Michal Assaf, Susan Y Bookheimer, Mirella Dapretto, et al. The autism brain imaging data exchange: towards a large-scale evaluation of the intrinsic brain architecture in autism. *Molecular psychiatry*, 19(6):659, 2014. 5

[22] Lee R Dice. Measures of the amount of ecologic association between species. *Ecology*, 26(3):297–302, 1945. 5

[23] Piotr Dollár, Peter Welinder, and Pietro Perona. Cascaded pose regression. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1078–1085. IEEE, 2010. 3

[24] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2758–2766, 2015. 1

[25] Pedro F Felzenszwalb, Ross B Girshick, and David McAllester. Cascade object detection with deformable part models. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2241–2248. IEEE, 2010. 3

[26] Bruce Fischl. Freesurfer. *Neuroimage*, 62(2):774–781, 2012. 6

[27] Ben Glocker, Nikos Komodakis, Georgios Tziritas, Nassir Navab, and Nikos Paragios. Dense image registration through mrfs and efficient linear programming. *Medical image analysis*, 12(6):731–741, 2008. 1, 3

[28] Tobias Heimann, Bram Van Ginneken, Martin A Styner, Yulia Arzhaeva, Volker Aurich, Christian Bauer, Andreas Beck, Christoph Becker, Reinhard Beichel, György Bekes, et al. Comparison and evaluation of methods for liver segmentation from ct datasets. *IEEE transactions on medical imaging*, 28(8):1251–1265, 2009. 5

[29] Tak-Wai Hui, Xiaoou Tang, and Chen Change Loy. Liteflownet: A lightweight convolutional neural network for optical flow estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8981–8989, 2018. 1

[30] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2462–2470, 2017. 1, 2

[31] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017. 4

[32] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. In *Advances in neural information processing systems*, pages 2017–2025, 2015. 2

[33] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[34] Arno Klein, Jesper Andersson, Babak A Ardekani, John Ashburner, Brian Avants, Ming-Chang Chiang, Gary E Christensen, D Louis Collins, James Gee, Pierre Hellier, et al. Evaluation of 14 nonlinear deformation algorithms applied to human brain mri registration. *Neuroimage*, 46(3):786–802, 2009. 5

[35] Stefan Klein, Marius Staring, Keelin Murphy, Max A Viergever, and Josien PW Pluim. Elastix: a toolbox for intensity-based medical image registration. *IEEE transactions on medical imaging*, 29(1):196–205, 2010. 3, 5, 6

[36] Julian Krebs, Tommaso Mansi, Hervé Delingette, Li Zhang, Florin C Ghesu, Shun Miao, Andreas K Maier, Nicholas Ayache, Rui Liao, and Ali Kamen. Robust non-rigid registration through agent-based action learning. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 344–352. Springer, 2017. 1, 3

[37] Tingfung Lau, Ji Luo, Shengyu Zhao, Eric I Chang, Yan Xu, et al. Unsupervised 3d end-to-end medical image registration with volume tweening network. *arXiv preprint arXiv:1902.05020*, 2019. 1, 2, 3, 4, 5, 6

[38] Hongming Li and Yong Fan. Non-rigid image registration using fully convolutional networks with deep self-supervision. *arXiv preprint arXiv:1709.00799*, 2017. 1, 3

[39] LiTS. Liver tumor segmentation challenge, 2018. Available at https://competitions.codalab.org/competitions/15595. 5

[40] Kasper Marstal, Floris Berendsen, Marius Staring, and Stefan Klein. Simpleelastix: A user-friendly, multi-lingual library for medical image registration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 134–142, 2016. 3

[41] Nikolaus Mayer, Eddy Ilg, Philip Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4040–4048, 2016. 1

[42] MSD. Medical segmentation decathlon, 2018. Available at https://decathlon-10.grand-challenge.org/. 5

[43] Susanne G Mueller, Michael W Weiner, Leon J Thal, Ronald C Petersen, Clifford R Jack, William Jagust, John Q Trojanowski, Arthur W Toga, and Laurel Beckett. Ways toward an early diagnosis in alzheimers disease: the alzheimers disease neuroimaging initiative (adni). *Alzheimer's & Dementia*, 1(1):55–66, 2005. 5

[44] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. Learning deconvolution network for semantic segmentation. In *Proceedings of the IEEE international conference on computer vision*, pages 1520–1528, 2015. 3

[45] Jiahao Pang, Wenxiu Sun, Jimmy SJ Ren, Chengxi Yang, and Qiong Yan. Cascade residual learning: A two-stage convolutional neural network for stereo matching. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 887–895, 2017. 1, 2

[46] Hariharan Ravishankar, Rahul Venkataramani, Sheshadri Thiruvenkadam, Prasad Sudhakar, and Vivek Vaidya. Learning and incorporating shape models for semantic segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 203–211. Springer, 2017. 3

[47] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 4

[48] Daniel Rueckert, Luke I Sonoda, Carmel Hayes, Derek LG Hill, Martin O Leach, and David J Hawkes. Nonrigid registration using free-form deformations: application to breast mr images. *IEEE transactions on medical imaging*, 18(8):712–721, 1999. 3, 5, 6

[49] Jo Schlemper, Jose Caballero, Joseph V Hajnal, Anthony N Price, and Daniel Rueckert. A deep cascade of convolutional neural networks for dynamic mr image reconstruction. *IEEE transactions on Medical Imaging*, 37(2):491–503, 2018. 3

[50] Thilo Sentker, Frederic Madesta, and René Werner. Gdl-fire 4d: Deep learning-based fast 4d ct image registration. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 765–773. Springer, 2018. 3

[51] David W Shattuck, Mubeena Mirza, Vitria Adisetiyo, Cornelius Hojatkashani, Georges Salamon, Katherine L Narr, Russell A Poldrack, Robert M Bilder, and Arthur W Toga. Construction of a 3d probabilistic atlas of human cortical structures. *Neuroimage*, 39(3):1064–1080, 2008. 5

[52] Dinggang Shen and Christos Davatzikos. Hammer: hierarchical attribute matching mechanism for elastic registration. In *Proceedings IEEE Workshop on Mathematical Methods*

in Biomedical Image Analysis (MMBIA 2001), pages 29–36. IEEE, 2001. 1, 3

[53] Hessam Sokooti, Bob de Vos, Floris Berendsen, Boudewijn PF Lelieveldt, Ivana Išgum, and Marius Staring. Nonrigid image registration using multi-scale 3d convolutional neural networks. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 232–239. Springer, 2017. 1, 3

[54] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8934–8943, 2018. 1

[55] Jia Xu, René Ranftl, and Vladlen Koltun. Accurate optical flow via direct cost volume processing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1289–1297, 2017. 1

[56] Xiao Yang, Roland Kwitt, Martin Styner, and Marc Niethammer. Quicksilver: Fast predictive image registration–a deep learning approach. *NeuroImage*, 158:378–396, 2017. 1, 3

[57] Xiang Yu, Feng Zhou, and Manmohan Chandraker. Deep deformation network for object landmark localization. In *European Conference on Computer Vision*, pages 52–70. Springer, 2016. 3

[58] Zhengyou Zhang. Iterative point matching for registration of free-form curves and surfaces. *International journal of computer vision*, 13(2):119–152, 1994. 3