




Cycle-Consistent Training for Reducing Negative Jacobian Determinant in Deep Registration Networks

Dongyang Kuang^{1,2} 

¹ University of Ottawa, Ottawa, Canada
dykuang@outlook.com

² University of Texas – Austin, Austin, Texas, USA

Abstract. Image registration is a fundamental step in medical image analysis. Ideally, the transformation that registers one image to another should be a diffeomorphism that is both invertible and smooth. Traditional methods like geodesic shooting study the problem via differential geometry, with theoretical guarantees that the resulting transformation will be smooth and invertible. Most previous research using unsupervised deep neural networks for registration address the smoothness issue directly either by using a local smoothness constraint (typically, a spatial variation loss), or by designing network architectures enhancing spatial smoothness. In this paper, we examine this problem from a different angle by investigating possible training mechanisms/tasks that will help the network avoid predicting transformations with negative Jacobians and produce smoother deformations. The proposed cycle consistent idea reduces the number of folding locations in predicted deformations without making changes to the hyperparameters or the architecture used in the existing backbone registration network. Code for the paper is available at <https://github.com/dykuang/Medical-image-registration>.

Keywords: Unsupervised registration · Cycle consistent training · Folding deduction

1 Introduction

Image registration is a key element of medical image analysis. Most state-of-the-art registration algorithms, such as ANTs [1], can utilize geometric methods that are guaranteed to produce smooth invertible deformations that are much desired in medical image registrations. A revolution is taking place in the last couple of years in the application of machine learning methods. Especially, the method of convolutional neural networks have made impressive progresses and caused a lot of attentions. While recent registration networks can make predictions of the nonlinear transformation much faster and obtain registration accuracy comparable to or better than traditional methods, they usually do not have theoretical guarantees on the smoothness or invertibility of their predicted deformations.

Supervised methods, such as in [8, 11, 13], learn from known reference deformations for training data – either actual “ground truth” in the case of synthetic image pairs, or deformations computed by other automatic or semi-automatic methods. They usually do not have problems of smoothness, but still rely on other tools such as ANTs running ahead to produce desired transformations. The registration problem is much harder in the setting of unsupervised methods. Most of the early unsupervised approaches like [2, 7, 10, 12, 14] take the idea of spatial transformer (ST) [4]. This spatial transformer used in registration usually consists of two basic functional units: a deformation unit and a sampling unit. With input x (source image) and y (target image) stacked as an ordered pair, the deformation unit produces a static displacement field $\mathbf{u} : \mathcal{R}^3 \rightarrow \mathcal{R}^3$. The warped image \tilde{y} is then constructed in the sampling unit by interpolating the source image with \mathbf{u} via $\tilde{y} = x(\text{Id} + \mathbf{u})$, where Id is the identity map. As a summary, the right action of diffeomorphism ϕ on image x is approximated by $\phi \cdot x = x \circ \phi^{-1} \approx x(\text{Id} + \mathbf{u})$. The smoothness constraint on \mathbf{u} is usually addressed by regularizing its derivative $D\mathbf{u}$. The work [2] is one representative and Fig. 1 shows the work flow of the idea introduced as above. The whole network is trained so that it minimizes the loss: $CC(y, \tilde{y}) + \lambda \|D\mathbf{u}\|_{l_2}$, where CC stands for cross correlation loss and λ is a hyperparameter controlling the strength of the regularization.

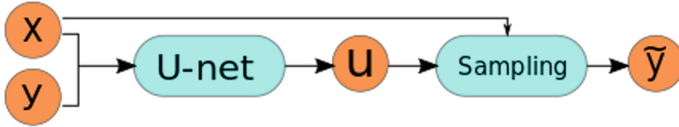


Fig. 1. An overview of the registration network usually used for registration. The popular U-net architecture [9] is used as the deformation unit for generating the displacement field.

These work emphasize more on the accuracy and efficiency of registration when compared to classical methods but usually did not put equal attentions on checking geometric properties such as smoothness, invertibility or orientation preservation for the predicted deformations. Particularly, Jacobian determinant of the predicted transformations i.e. $\det(D\phi^{-1}) \approx \det(\text{Id} + D\mathbf{u})$ from a neural network can very likely be negative at multiple locations. This “folding” issue during prediction may still persist even when one increases regularization strength of $D\mathbf{u}$ (see Fig. 2). Additionally, the value of this hyper-parameter is usually difficult to set in practice in order to reach a good balance between nice geometrical properties¹ and registration accuracy, since larger λ values often cause smaller deformations reducing the accuracy.

¹ In the paper, it will mainly refer to smoothness, invertibility and particularly, transformations has positive Jacobian determinant everywhere.

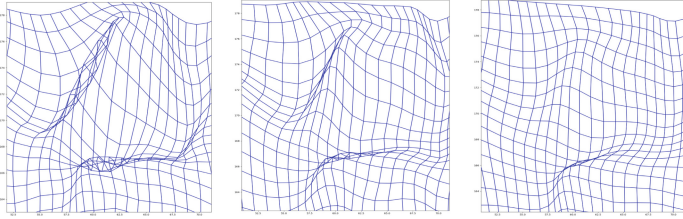


Fig. 2. A snapshot of at the same location of the projected warped grid with different regularization strength. From left to right, the network is trained with $\lambda = 1, 2, 4$ separately. The same location is also used in Fig. 7.

Built upon previous research, state-of-the-art works like [3] proposes a probabilistic VoxelMorph (Prob-VM) design that takes a reparametrization trick and inserts an “integration layer” trying to produce smoother deformation. From modeling point of view, this process-oriented modeling is usually difficult and requires much effort to design a new architecture ahead of time that is proved to be effective later on general data. In order to make an easier modeling process avoiding going inside the box to handcraft an ideal architecture, one can keep the original network with possible flaws untouched but instead seek a different training mechanism/task that is possible to achieve better regularization implicitly. This thought of task-oriented modeling may reveal an alternative way for solving the same problem. In this paper, we take this direction and propose a cycle consistent design for training unsupervised registration networks by assigning an additional task to it. The idea requires no modifications of backbone network’s architectures, form of loss functions or hyperparameters used, hence can be used upon any well-known backbone registration networks. From our experiments with VoxelMorph as the backbone network, the proposed idea reduces chances of negative Jacobian determinant in its predicted transformations and can achieve comparable results with Prob-VM.

2 Related Work

To author’s best knowledge when completing this paper, [15] and [3] are most relevant research in reducing negative Jacobian. Our proposed idea represents a different strategy in solving the problem (see Fig. 3). [15] designed an inverse consistent network and argued adding an explicit “anti-folding constraint” to prevent folding in predicted transformation. Different from his work, we do not create new forms of losses targeting on specific properties, but focuses on discovering possible training mechanisms/tasks that will help better regularize the network in a general way. [3] is developed upon [2] by integrating an variational auto-encoder design and inserting an integration layer that “integrates” initial velocity field to get the final displacements. Unlike their work on modifying backbone architectures for better performance, the cycle consistent idea in this paper leaves the backbone network untouched but achieves regularization implicitly by

adding one more task of recovering the source image from its already predicted image during training. This additional task is meant to help narrow the solution domain where non-smooth or non-invertible transformations are hardly inside during optimization.

3 Proposed Methods

3.1 Cycle Consistent Design

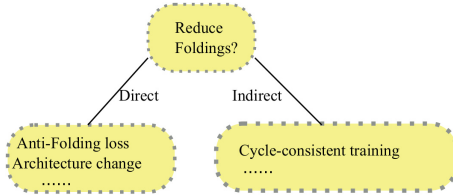


Fig. 3. Two directions for addressing folding issues in prediction.

already been research such as [16] utilizing this idea for better quality control of cross-domain image generations. In their work, they defined two joint cycle consistent loops for better training two separate generative adversarial networks for unpaired image-to-image translation back-and-forth. We use a related idea in a different setting here for regularizing the predicted static displacement field. This “cycle consistent” idea does not involve new form of losses but forces the same network to perform a backward prediction trying to recover the input right after it completes the forward prediction. As seen in Fig. 4, the spatial transformer will first predict a warped image \tilde{y} and the corresponding displacement field $\mathbf{u}_{x \rightarrow \tilde{y}}$ with the stacked source image x and target image y . This predicted warped image \tilde{y} (now as source) is then stacked with the original source image x (now as target). They will be fed into the same spatial transformer to produce a reconstruction \tilde{x} for x and corresponding inverted displacement field $\mathbf{u}_{\tilde{y} \rightarrow \tilde{x}}$. The whole network is trained with the cycle consistent loss:

$$CC(y, \tilde{y}) + \lambda \|D\mathbf{u}_{x \rightarrow \tilde{y}}\|_{l_2} + CC(x, \tilde{x}) + \lambda \|D\mathbf{u}_{\tilde{y} \rightarrow \tilde{x}}\|_{l_2} \quad (1)$$

While it is straightforward that this design directly addresses the invertibility of the network, the cycle constraint task also contributes to the task of learning a smooth solution in an indirect way: the design regularizes the network by forcing the spatial transformer to learn a solution and its possible inverse at the same time. This helps the network rule out transformations that are not cycle consistent during optimization. This design also does not add any additional learnable parameters to the original spatial transformer and can be trained as equally efficient.

From the mathematical point of view, the transformations used in registration tasks should ideally be diffeomorphisms so that topological properties are not changed during the transformations. In order to approximate the ideal deformation, training of the network should also respect this invertibility property. In fields such as computer vision, there have

Though similar, this idea is also different from bi-directional registrations where the target image will also be warped towards the source image during optimization. In our design, the target image will never be warped. To be more specific, given loss function L and input source-target image pair (x, y) , the neural network with parameters θ learns the mapping f to transform x towards y : $y \approx f(\theta; (x, y))$. The two optimization problems can be vaguely summarized as below in Table 1:

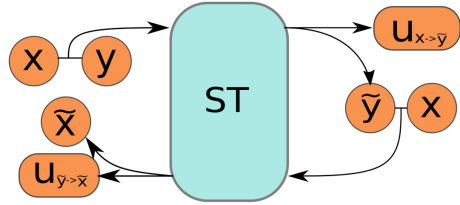


Fig. 4. A diagram illustrating the cycle consistent design.

Table 1. Different object function optimization formulations between bi-directional registration and cycle-consistent training.

Methods	Formulations of optimization
Bi-direction:	$\arg \min_{\theta} L(y, f(\theta; x, y)) + L(x, f(\theta; y, x))$
Cycle-consistent:	$\arg \min_{\theta} L(y, f(\theta; x, y)) + L(x, f(\theta; f(\theta; x, y), x))$

Bi-direction registration uses both pairs (x, y) and (y, x) as inputs, while the cycle-consistent training only uses (x, y) . They are equivalent if there exists a “perfect” deformation that aligns the registration pair and this transformation f can be learned with parameters θ during training: $y = f(\theta; (x, y))$.

4 Experiment

4.1 Dataset

We used MindBoggle101 dataset [6] for experiments. Details of data collection and processing, including atlas creation, are described in [6]. In the present paper, we used brain volumes consisting of the following three named subsets of Mindboggle101:

- NKI-RS-22: “Nathan Kline Institute/Rockland sample”
- NKI-TRT-20: “Nathan Kline Institute/Test-Retest”
- OASIS-TRT-20: “Open Access Series of Imaging Studies/ Test-Retest”.

Each image has a dimension of $182 \times 218 \times 182$, we truncated the margin reducing the size to $144 \times 180 \times 144$. These images are already linearly aligned to MNI152 space. We also normalized the intensity of each brain volume to $[0, 1]$ by its maximum voxel intensity. Figure 5 shows one subject of the dataset with two annotated labels. Labels used in Mindboggle101 data set are cortex surface labels. Their geometrical complexity leads to more challenging registration

tasks, especially for neural network approaches. In the following experiments, the original VoxelMorph network [2] will be used as the backbone network. This backbone network alone, it with cycle consistent design and the probabilistic VoxelMorph will be compared. The backbone method and the method with cycle consistent design are trained with $\lambda = 1$. Unless specifically stated, epochs = 10 and “Adam” optimizer [5] with learning rate 10^{-4} are used for all the three networks.

We access the accuracy of predicted registration via dice score between ROI labels/masks. For image pair (x, y) , each indexed label L_x^i associated with x will be warped with the deformation ϕ predicted from the registration network, dice score is then calculated. A higher dice score usually indicates a better registration.

$$Dice((\phi \cdot L_x^i), L_y^i) = \frac{2|(\phi \cdot L_x^i) \cap L_y^i|}{|\phi \cdot L_x^i| + |L_y^i|} \quad (2)$$

We first visualize this metric on test set (OASIS-TRT-20) in Fig. 6. It gives a detailed summary of dice scores on separate regions for registration. All the three neural network approaches appear to provide similar dice scores for most regions and slightly outperform the non-neural-network-based method such as Ants’ SyNQuick algorithm. As will be illustrated later in details, these similar dice scores are actually results of deformations that have different Jacobian properties. The foldings of the deformation is accessed via examining locations where negative Jacobian determinants happen. Let \mathcal{P} be defined as the percentage of voxel locations where the Jacobian determinant is negative over all voxels V , i.e.

$$\mathcal{P} := \frac{\sum \delta(\det(D\phi^{-1}) < 0)}{V}.$$

The ideal transformation predicted should have this number as small as possible. To better access the general performance of our proposed methods, we perform a 3-fold validation² with the 3 datasets at hand. We summarize this number from different methods into Table 2 for comparison. The author reminds readers that Table 2 is not for the purpose of competing with Prob-VoxelMorph or Ants’ SyNQuick, but simply a demonstration that an indirect task oriented method such as the proposed cycle-consistent training can also achieve comparable registration quality with state-of-the-art method such as Prob-VoxelMorph. To support this, results from some statistical hypothesis tests are organized in Table 3.

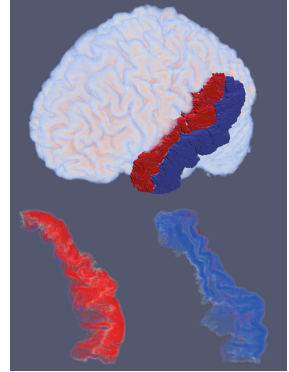


Fig. 5. One sample with two ROI labels shown. Bottom: the two labels viewed from a different angle

² Each fold will use 2 of the 3 datasets for forming training set and test on the third. Figures 6 and 7 are from the fold when pairs from OASIS dataset are used as test. This experiment has 1722 training pairs and 380 test pairs.

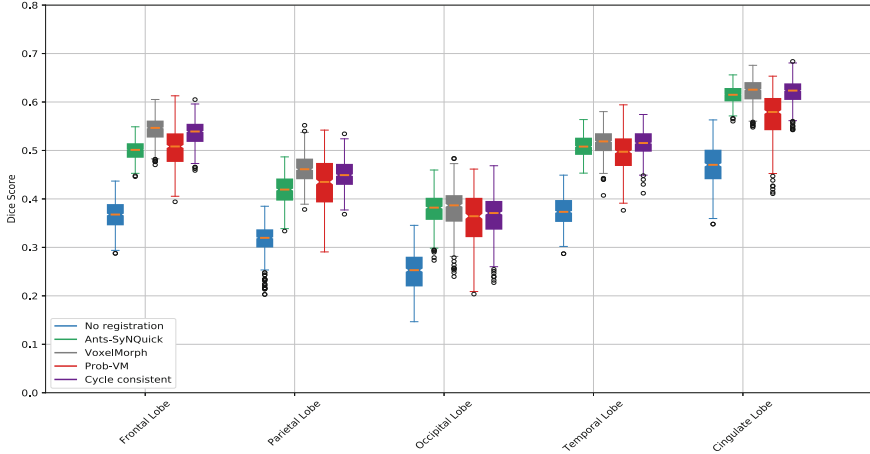


Fig. 6. Mean dice scores of different methods on selected regions. Each point is the mean dice score averaged over corresponding ROI labels per registration pair instead of over the union of labels in that region. Results from SyNQuick algorithm in the ANTs package are also listed as an example for better interpreting these dice scores, but not for the purpose of comparison.

It is clear that Table 2 suggests there are differences of the underlying transformation in terms of the measures introduced as above. From the cross validation results, the baseline method has a mean value of 1.97% locations where Jacobian determinants are negative. When the cycle consistent design is applied, this value drops to 0.13%. In other words, more than 90% of the unsatisfactory locations happening in the baseline prediction are eliminated (H_0 can be rejected with p -value = 0.02 in test I). This result is very close to the performance of probabilistic VoxelMorph with 0.03% improvement in $\mu(P)$ (whether to adopt or reject H_0 will depend on one’s confidence level with p -value = 0.05 in test II) and 0.9% “higher” mean dice score (H_0 cannot be rejected with such a large

Table 2. Summary of metrics with the 3-fold validation, mean (μ) and the standard deviation (σ) calculated over the 3 folds are shown. Since Ants’ SyNQuick method does not require training set to register a pair of images, folds split is not appropriate for its evaluation. We only record mean values from registering all the pairs in the whole dataset for comparison.

Method	$\mu(P)$	$\sigma(P)$	$\mu(Dice)$	$\sigma(Dice)$
Ants’ SyNQuick	≈ 0		47.63%	
VoxelMorph	1.97%	0.59%	49.83%	0.17%
Prob-VoxelMorph	0.16%	0.05%	47.25%	0.19%
Cycle consistent	0.13%	0.04%	48.10%	0.86%

Table 3. Some hypothesis tests results summarized from the 3-fold experiments. Abbreviations used: CC for “VoxelMorph with Cycle-Consistent training”, VM for “VoxelMorph without Cycle-Consistent training” and PVM for the “Prob-VoxelMorph”.

Null Hypothesis H_0 :	Test performed	p -value
I: $\mu(\mathcal{P}) _{CC} \geq \mu(\mathcal{P}) _{VM}$	One tailed paired t-test	0.02
II: $\mu(\mathcal{P}) _{CC} \geq \mu(\mathcal{P}) _{PVM}$	One tailed paired t-test	0.05
III: $\mu(Dice) _{CC} = \mu(Dice) _{PVM}$	Two tailed paired t-test	0.23

p -value in test III of Table 3, hence this improvement is not statistical significant, the two methods are comparable in this measure). As a summary, these results suggest the two different directions (direct ways as Prob-VoxelMorph and indirect approaches as Cycle-Consistent training) have comparable effects in terms of reducing foldings locations while maintaining registration accuracy.

For better visualization, we also put one slice of the Jacobian determinant map and the projected warped grid on the same slice in Fig. 7. The transformation for visualization used in the figure is predicted on the pair formed by subject OASIS-TRT-3 (source) and subject OASIS-TRT-8 (target).

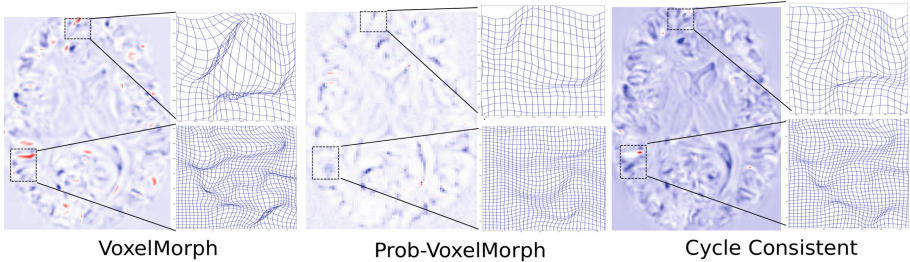


Fig. 7. Determinant of Jacobian map and the warped grid projected on the same slice. From left to right: the baseline VoxelMorph prediction, the Probabilistic VoxelMorph and baseline with cycle consistent design. Locations where determinants are negative are shown in red. (Color figure online)

Figure 7 shows an example of locations with negative Jacobian determinants. This help give an intuitive view of what happened behind the curtain. From the warped grid columns, one can clearly see networks with cycle consistent design did not change much in locations where the baseline prediction are already smooth but put attentions on foldings and “unfold” them to produce a smoother transformation. Note that the grid shown in the upper right corner of cycle consistent result is smoother compared to the grid shown in the middle of Fig. 2 where the regularization strength is doubled (i.e. $\lambda = 2$). The color map of Prob-VoxelMorph looks pale because there exists at least one location with a large

Jacobian determinant value in this random example. In this case, most locations with Jacobian determinants relatively smaller will be renormalized close to zero during the normalization step when creating the color map.

5 Conclusion

We contribute the idea of cycle-consistent training for reducing locations of negative Jacobian determinants occurred in deformations when a deep neural network is used for unsupervised registration tasks. Unlike most other approaches that address the problem directly by creating new losses or developing new architectures for regularization, this paper focuses on another direction that could bring improvements implicitly by adopting different training mechanisms. The idea does not require changing anything from the backbone network and hence can be used upon arbitrary registration networks. Heuristically, the additional cycle-consistent task during training forces the network to learn recovery transformations at the same time, hence help narrow down the solution domain during optimization. While the theoretical support for this idea still needs to be investigated as part of future research, experiments have shown that this indirect approach is capable of obtaining comparable results with state-of-the-art methods in terms of reducing negative Jacobian determinants while maintaining registration accuracy.

References

1. Avants, B.B., Tustison, N.J., Song, G., Cook, P.A., Klein, A., Gee, J.C.: A reproducible evaluation of ants similarity metric performance in brain image registration. *Neuroimage* **54**(3), 2033–2044 (2011)
2. Balakrishnan, G., Zhao, A., Sabuncu, M.R., Guttag, J., Dalca, A.V.: An unsupervised learning model for deformable medical image registration. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9252–9260 (2018)
3. Dalca, A.V., Balakrishnan, G., Guttag, J., Sabuncu, M.R.: Unsupervised learning for fast probabilistic diffeomorphic registration. In: Frangi, A.F., Schnabel, J.A., Davatzikos, C., Alberola-López, C., Fichtinger, G. (eds.) *MICCAI 2018*. LNCS, vol. 11070, pp. 729–738. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-00928-1_82
4. Jaderberg, M., Simonyan, K., Zisserman, A., et al.: Spatial transformer networks. In: *Advances in Neural Information Processing Systems*, pp. 2017–2025 (2015)
5. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014)
6. Klein, A., Tourville, J.: 101 labeled brain images and a consistent human cortical labeling protocol. *Front. Neurosci.* **6**, 171 (2012)
7. Li, H., Fan, Y.: Non-rigid image registration using fully convolutional networks with deep self-supervision. *arXiv preprint arXiv:1709.00799* (2017)

8. Rohé, M.-M., Datar, M., Heimann, T., Sermesant, M., Pennec, X.: SVF-Net: learning deformable image registration using shape matching. In: Descoteaux, M., Maier-Hein, L., Franz, A., Jannin, P., Collins, D.L., Duchesne, S. (eds.) MICCAI 2017. LNCS, vol. 10433, pp. 266–274. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-66182-7_31
9. Ronneberger, O., Fischer, P., Brox, T.: U-Net: convolutional networks for biomedical image segmentation. In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (eds.) MICCAI 2015. LNCS, vol. 9351, pp. 234–241. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-24574-4_28
10. Shan, S., et al.: Unsupervised end-to-end learning for deformable medical image registration. arXiv preprint [arXiv:1711.08608](https://arxiv.org/abs/1711.08608) (2017)
11. Sokooti, H., de Vos, B., Berendsen, F., Lelieveldt, B.P.F., Išgum, I., Staring, M.: Nonrigid image registration using multi-scale 3D convolutional neural networks. In: Descoteaux, M., Maier-Hein, L., Franz, A., Jannin, P., Collins, D.L., Duchesne, S. (eds.) MICCAI 2017. LNCS, vol. 10433, pp. 232–239. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-66182-7_27
12. Wang, S., Kim, M., Wu, G., Shen, D.: Scalable high performance image registration framework by unsupervised deep feature representations learning. In: Deep Learning for Medical Image Analysis, pp. 245–269. Elsevier (2017)
13. Yang, X., Kwitt, R., Styner, M., Niethammer, M.: Quicksilver: fast predictive image registration—a deep learning approach. *NeuroImage* **158**, 378–396 (2017)
14. Yoo, I., Hildebrand, D.G.C., Tobin, W.F., Lee, W.-C.A., Jeong, W.-K.: ssEMnet: serial-section electron microscopy image registration using a spatial transformer network with learned features. In: Cardoso, M.J., et al. (eds.) DLMIA/ML-CDS -2017. LNCS, vol. 10553, pp. 249–257. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-67558-9_29
15. Zhang, J.: Inverse-consistent deep networks for unsupervised deformable image registration. arXiv preprint [arXiv:1809.03443](https://arxiv.org/abs/1809.03443) (2018)
16. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. arXiv preprint (2017)