Metadata of the chapter that will be visualized in SpringerLink

Book Title	Biometric Recognition	
Series Title		
Chapter Title	Fast and Robust Detection of Anatomical Landmarks Using Cascaded 3D Convolutional Networks Guided by Linear Square Regression	
Copyright Year	2018	
Copyright HolderName	Springer Nature Switzerland AG	
Corresponding Author	Family Name	**Wang**
	Particle	
	Given Name	**Zi-Rui**
	Prefix	
	Suffix	
	Role	
	Division	NELSLIP
	Organization	University of Science and Technology of China
	Address	Hefei, China
	Email	cs211@mail.ustc.edu.cn
Author	Family Name	**Yin**
	Particle	
	Given Name	**Bao-Cai**
	Prefix	
	Suffix	
	Role	
	Division	
	Organization	iFLYTEK AI Research
	Address	Hefei, China
	Email	bcyin@iflytek.com
Author	Family Name	**Du**
	Particle	
	Given Name	**Jun**
	Prefix	
	Suffix	
	Role	
	Division	NELSLIP
	Organization	University of Science and Technology of China
	Address	Hefei, China
	Email	jundu@ustc.edu.cn
Author	Family Name	**Liu**
	Particle	
	Given Name	**Cong**
	Prefix	
	Suffix	

	Role	
	Division	
	Organization	iFLYTEK AI Research
	Address	Hefei, China
	Email	congliu2@iflytek.com
Author	Family Name	**Tao**
	Particle	
	Given Name	**Xiaodong**
	Prefix	
	Suffix	
	Role	
	Division	
	Organization	iFLYTEK AI Research
	Address	Hefei, China
	Email	xdtao@iflytek.com
Author	Family Name	**Hu**
	Particle	
	Given Name	**Guoping**
	Prefix	
	Suffix	
	Role	
	Division	
	Organization	iFLYTEK AI Research
	Address	Hefei, China
	Email	gphu@iflytek.com

Abstract	Detecting anatomical landmarks on structural magnetic resonance imaging (MRI) is an important medical computer-aid technique. However, for some brain anatomical landmarks detection, linear/non-linear registration with skull stripping across subjects is usually unavoidable. In this paper, we propose a novel method. Starting from the original MRI data, a series of 3D convolutional neural networks (cascaded 3D-CNNs) are adopted to iteratively update the predicted landmarks. Specially, the predicted landmarks of each 3D-CNN model are used to estimate the corresponding linear transformation matrix by linear square regression, which is very different from traditional registration methods. Based on the estimated matrix, we can use it to transform the original image for getting the new image for the next 3D-CNN model. With these cascaded 3D-CNNs and linear square regression, we can finally achieve registration and landmark detection.

Keywords (separated by '-')	Anatomical landmark detection - Cascaded 3D-CNNs - Linear square regression - Fast - Robust

Fast and Robust Detection of Anatomical Landmarks Using Cascaded 3D Convolutional Networks Guided by Linear Square Regression

Zi-Rui Wang[1(✉)], Bao-Cai Yin[2], Jun Du[1], Cong Liu[2], Xiaodong Tao[2], and Guoping Hu[2]

[1] NELSLIP, University of Science and Technology of China, Hefei, China
cs211@mail.ustc.edu.cn, jundu@ustc.edu.cn
[2] iFLYTEK AI Research, Hefei, China
{bcyin,congliu2,xdtao,gphu}@iflytek.com

Abstract. Detecting anatomical landmarks on structural magnetic resonance imaging (MRI) is an important medical computer-aid technique. However, for some brain anatomical landmarks detection, linear/non-linear registration with skull stripping across subjects is usually unavoidable. In this paper, we propose a novel method. Starting from the original MRI data, a series of 3D convolutional neural networks (cascaded 3D-CNNs) are adopted to iteratively update the predicted landmarks. Specially, the predicted landmarks of each 3D-CNN model are used to estimate the corresponding linear transformation matrix by linear square regression, which is very different from traditional registration methods. Based on the estimated matrix, we can use it to transform the original image for getting the new image for the next 3D-CNN model. With these cascaded 3D-CNNs and linear square regression, we can finally achieve registration and landmark detection.

Keywords: Anatomical landmark detection · Cascaded 3D-CNNs
Linear square regression · Fast · Robust

1 Introduction

Recently, deep learning [1] is increasingly used for landmarks detection in medical analysis [2–6]. Interesting, a data-driven manner is utilized to generate discriminative landmarks [6] between Alzheimer's disease (AD) [8] and normal control, which can effectively improve the performance of AD diagnosis. This kind of landmark detection is also the topic of this paper.

In [6], Zhang et al. defined discriminative landmarks between AD and healthy control (HC) via a data-driven manner and then used a regression-forest algorithm to identify those landmarks. Furthermore, similar method can be effectively used with the help of longitudinal structural MR images [9]. But, in order

J. Zhou et al. (Eds.): CCBR 2018, LNCS 10996, pp. 1–10, 2018.
https://doi.org/10.1007/978-3-319-97909-0_65

to cover landmarks in different locations, many random forests have to be trained and used. Besides, massive image patches have to be extracted for training these models. Moreover, some additional fine processes have to be employed to make the algorithm have final good performance. As an improved scheme, a fully convolutional network (FCN) has also been adopted to jointly predict all landmarks at the same time in [7]. However, the FCN was completed by two stage training, due to limited medical imaging data. But, in the first training stage, massive image patches still need to be extracted for training.

In addition to the above mentioned problems, as a basic step for brain MRI preprocessing, linear/non-linear registration with skull stripping across subjects is usually unavoidable. To address this critical issue, we propose a novel method to directly work on original MRI data (with little preprocessing), for iteratively estimating landmarks with cascaded 3D-CNNs. In particular, landmarks predicted by each 3D-CNN model are used to estimate the corresponding linear transformation matrix by linear square regression. Based on the estimate transformation matrix, we can use it to transform the image and obtain the new image for the next 3D-CNN model. Finally, a linear transformation matrix can be estimated. This registration strategy is very different from traditional linear registration methods [10]. They usually try to hard search a best transformation which can yield the minimum cost for a given cost function while our registration can be regarded as a learning-based method. The whole procedure can be regarded as fast linear image registration and seamlessly achieve landmark detection at the same time.

2 Materials

All subjects used in this study are obtained from the publicly available dataset, ADNI-1[1]. Totally, there are 199 AD, 229 HCs and 404 MCI subjects, each with 1.5T T1-weighted MR image. All these images are conducted with some necessary preprocessing, i.e., reorientation and resampling to $256 \times 256 \times 256$ (with a voxel resolution $1 \times 1 \times 1\,\mathrm{mm}^3$). A preprocessed MR image and its corresponding landmarks are shown in Fig. 1. As we can see, many landmarks are concentrated around the hippocampus. These areas play important roles in memory and are related to AD [8]. We call these images as original images (ori-images) and the images after linear registration (the Colin27 template [14] for reference) as lin-images.

In experiments, we randomly select 420 subjects from 428 AD and HC subjects as our training set, and use the remaining 8 subjects as validation set to tune the parameters of our networks. Totally, 404 new MCI subjects are used as our testing set to evaluate our approach, demonstrating its robustness. For all the training subjects, their corresponding 100 anatomical landmarks in ori-images (ori-landmarks) and lin-images (lin-landmarks) respectively are defined by [6]. For all the testing subjects, we use their corresponding 100 ori-landmarks

[1] http://adni.loni.usc.edu.

defined by the same way as the ground truth to evaluate the performance of our approach. All these landmarks can be represented as follows:

$$L = [(x_1, y_1, z_1)^{\mathrm{T}}, .(x_i, y_i, z_i)^{\mathrm{T}}.., (x_{100}, y_{100}, z_{100})^{\mathrm{T}}] \quad (1)$$

where $(x_i, y_i, z_i)^{\mathrm{T}}$ denotes the coordinate of the i-th landmark. For simplicity, we use L_{ori} and L_{lin} to represent the ori-landmarks and the lin-landmarks, respectively.

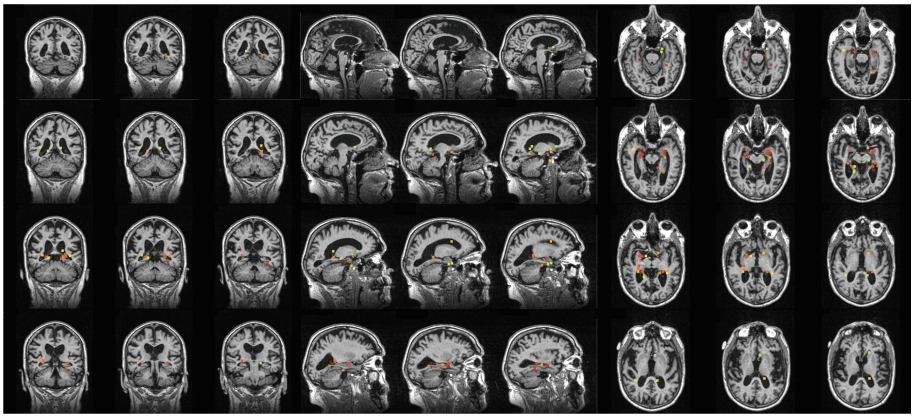

Fig. 1. Example anatomical landmarks on different slices with 3 mm striding.

3 Method

3.1 Overview

The pipeline of proposed method is shown in Fig. 2, which is composed of a number of similar blocks (1,...,n). The input images of each block come from the output images of its previous block. Each block consists of two main parts, namely a 3D-CNN model and a landmark-based linear transformation estimation with linear square regression (LSR), except the last block that only contains a 3D-CNN model. The structure of 3D-CNN is the same for all the blocks. The details of the 3D-CNN and the LSR will be described in Sects. 3.2 and 3.3.

In the training stage, for the 3D-CNN in the first block, we use the ori-images and their corresponding L_{ori} (L^1_{lin}) to train the first 3D-CNN. Once completing this training step, the trained 3D-CNN is used to predict the landmarks L^1_{pre} of the training ori-images. Then, we can conduct the LSR between the L^1_{pre} and their corresponding lin-landmarks L_{lin} and get the corresponding transformation matrix W_1 and biases $\boldsymbol{b}_1$ for each subject in the first block. Based on the estimated matrix W_1, biases $\boldsymbol{b}_1$ and the ori-images, we can get a set of new training data (through linear transformation) for the next 3D-CNN. By applying the same corresponding matrix W_1 and biases $\boldsymbol{b}_1$ to the L_{lin}, the labels

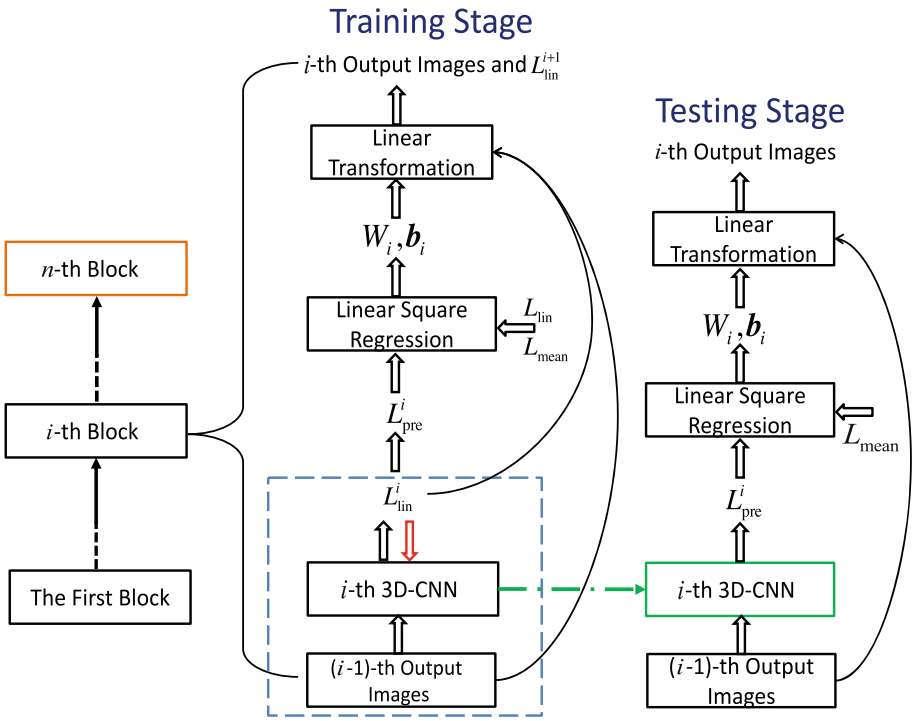

Fig. 2. The pipeline of proposed method.

L_{lin}^2 of the second 3D-CNN in the second block can be obtained. Next, we keep training the second 3D-CNN. Especially, the weights of the first 3D-CNN are used to initialize the weights of the second 3D-CNN, which is very important as shown in Sect. 4.3. But, there is a problem in the testing stage, since we don't know their lin-landmarks L_{lin}, to address this problem, we use the average landmarks L_{mean} of all training lin-landmarks L_{lin} as our targets in the LSR for every testing subject. This could cause a potential issue since the targeted landmarks are changing across subjects. To further handle this problem and get better performance, we propose a novel data augmentation, i.e., we also use the L_{mean} as targets in the LSR to augment our training data for the second 3D-CNN, which helps our model to see more data based on different transformations and thus enhance the generalization ability of our model. In this way, the amount of training data of the second 3D-CNN are also doubled, compared with the training data of the first 3D-CNN, since no data augmentation is used for the first 3D-CNN. The same training/testing strategies are used for the blocks 3 to n.

3.2 3D-CNN

The conventional convolutional neural network (CNN) [11] consists of successively stacked convolutional layers, optionally followed by spatial pooling, one or more fully-connected layers and a softmax layer. For convolutional and pooling layers in the usual 2-dimensional CNN (2D-CNN), each layer is a three-dimensional tensor organized by a set of planes called feature maps, while convolutional and pooling layers in the 3D-CNN are the four-dimensional tensors organized by a set of cubes, which is naturally suitable to deal with 3D MR images. The fully-connected layer and the softmax layer are the same with the common deep neural network (DNN). Inspired by the locally-sensitive, orientation-selective neurons in the visual system of cats, each unit in a feature map is constrained to connect a local region in the previous layer, which is called the local receptive field. Two contiguous local receptive fields are usually shifted for s pixels (referred as stride) along a certain direction. All units in the same feature map of a convolutional layer share a set of weights, each computing a dot product between its weights and local receptive field in the previous layer and then followed by nonlinear activation functions (e.g., rectifier). Meanwhile, the units in a pooling layer perform a spatial average or max operation for their local receptive field to reduce spatial resolution and noise interferences. Accordingly, the key information for identifying the pattern can be retained. We formalize the convolution operation in 3D-CNN as:

$$Y_{i,j,h,k} = \sum_{m,n,q,l} X_{(i-1)\times s+m,(j-1)\times s+n,(h-1)\times s+q,l} K_{m,n,q,k,l}$$

where $X_{i,j,h,k}$ is the value of the input unit in feature map k at row i, column j and height h, while $Y_{i,j,h,k}$ is corresponding to the output unit. $K_{m,n,q,k,l}$ is the connection weight between a unit in feature map k of the output and a unit in channel l of the input, with an offset of m rows, n columns and q heights between the output unit and the input unit. Similarly, the pooling operation can be conducted by using a max operation in this study. The 3D-CNN model used in this paper is shown in Fig. 3.

3.3 Linear Square Regression

As shown in Fig. 4, given D source points (blue points) $\mathrm{S}_d = (x_{\mathrm{s\_}d}, y_{\mathrm{s\_}d}, z_{\mathrm{s\_}d})^{\mathrm{T}}$ $(d = 1, ..., D)$ and their corresponding target points (red points) $\mathrm{T}_d = (x_{\mathrm{t\_}d}, y_{\mathrm{t\_}d}, z_{\mathrm{t\_}d})^{\mathrm{T}}$ $(d = 1, ..., D)$, the LSR tries to find a 3×3 matrix W and a 3-dimensional vector $\boldsymbol{b}$ to minimize the Eq. (2) so that the new transformed points Tr_d (green points) can get close to the target points.

$$Loss = \frac{1}{2} \sum_{d=1}^{D} |W\mathrm{S}_d + \boldsymbol{b} - \mathrm{T}_d|^2 \tag{2}$$

namely:

$$(W^*, \boldsymbol{b}^*) = \arg\min_{(W,\boldsymbol{b})} Loss \tag{3}$$

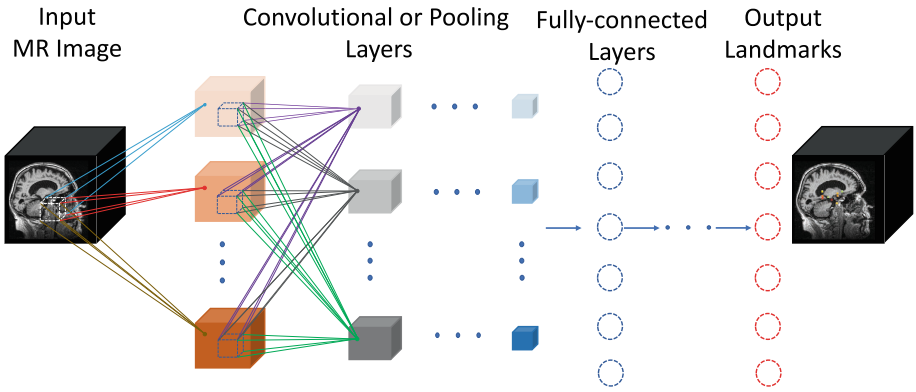

Fig. 3. Schematic diagram of 3DCNN for landmark detection.

The Eq. 3 has closed solution. So the transformed points:

$$\mathrm{Tr}_d = W\mathrm{S}_d + \boldsymbol{b} \quad (d = 1, ..., D) \tag{4}$$

In the i-th block, for the input image j, the source points are the predicted landmarks $L^{i,j}_{\mathrm{pre}}$ of the i-th 3D-CNN while the target points are the lin-landmarks L^{j}_{lin} or the average landmarks L_{mean}. Based on the transformation matrix W^j_i and the bias vector $\boldsymbol{b}^j_i$, we can transform the input image j and the corresponding landmarks $L^{i,j}_{\mathrm{lin}}$ for training the $(i+1)$-th 3D-CNN in the training stage. Obviously, the transformed landmarks $L^{(i+1),j}_{\mathrm{lin}}$ can be directly obtained by using the Eq. (4), and the transformed image can also be obtained by using this equation. As shown in Fig. 5, we can assume a point $\mathrm{P_T}$ in the transformed image, corresponds to its corresponding location $\mathrm{P_S}$ in the image j, which can be obtained by tracing back based on the W^j_i and $\boldsymbol{b}^j_i$, thus, we can get the intensity value for the point $\mathrm{P_T}$ ($\mathrm{P_S}$) according to image interpolation.

4 Experiments

4.1 Evaluation Criteria

In the experiments, we focus on evaluating our approach in detecting anatomical landmarks for the ori-images. For example, for the testing subject j, based on the predicted landmarks $L^{n,j}_{\mathrm{pre}}$ of the last 3D-CNN model and a series of estimated transformation matrices W^j_i $(i = 1, ..., n-1)$ and biases $\boldsymbol{b}^j_i$ $(i = 1, ..., n-1)$, the final predicted landmarks in the ori-images can be traced back according to the following formula:

$$L^{i,j}_{\mathrm{pre}} = (W^j_i)^{-1} * (L^{(i+1),j}_{\mathrm{pre}} - \boldsymbol{b}^j_i) \quad (i = n-1, ..., 1)$$

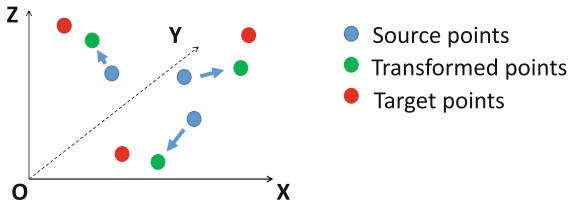

Fig. 4. The diagram of the linear square regression. (Color figure online)

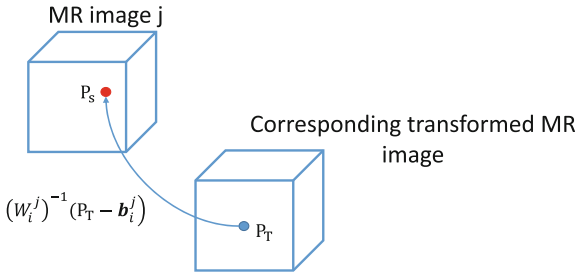

Fig. 5. The diagram about how to find corresponding points between MR image and transformed MR image by using the transformation matrix and bias.

To quantitatively evaluate detection performance, we use the mean Euclidean distance to measure the detection error (DE) between all final predicted landmarks and the ground truth landmarks in the ori-images:

$$\mathrm{DE} = \sum_{p=1}^{P} \sqrt{(x_p - \tilde{x}_p)^2 + (y_p - \tilde{y}_p)^2 + (z_p - \tilde{z}_p)^2}$$

(x_p, y_p, z_p) is the ground truth landmark and $(\tilde{x}_p, \tilde{y}_p, \tilde{z}_p)$ is the predicted landmark. There are P (40400) landmarks in all for 404 MCI subjects, each with 100 landmarks.

4.2 Experimental Setup

All 3D-CNN models in our approach have the same structure, i.e., five convolutional layers in the front. Their local receptive fields, feature maps and strides are [(5,5,5),24,1], [(5,5,5),64,1], [(5,5,5),128,1], [(3,3,3),256,1], [(3,3,3),512,1], respectively. The first four convolutional layers are followed by a pooling layer while the last convolutional layer is followed by two fully-connected layers with 1024 neural nodes. All convolutional layers are equipped with batch normalization. All pooling layers have local receptive field (2,2,2) and stride is 2, except the first pooling layer which has local receptive field (3,3,3) and stride 3. The activation functions of the convolutional and fully-connected layers are ReLU, while the output

layer with 300 neural nodes uses a sigmoid function to predict landmarks. In the training of 3D-CNN models, the common stochastic gradient descent (SGD) algorithm with a momentum coefficient is used.

Table 1. The DE (mm) comparison of different training strategies and different total numbers of blocks.

	Block1	Block2	Block3	Block4	Block5
Structure1	3.20	2.86	2.72	2.66	2.66
Structure2	3.20	2.82	2.68	2.57	2.57
Structure3	3.20	2.70	2.27	2.02	**1.95**

4.3 Results

Table 1 lists the DE comparison of different training strategies and different total numbers of blocks. For structure1, we don't use data augmentation and all CNN models are randomly initialized. That is, in the training stage, only lin-landmarks L_{lin} are used as targets in the LSR while both L_{lin} and L_{mean} are used in the structure2. Further more, in the structure3, not only the data augmentation strategy is used in the training stage, but also the weights of the $(i-1)$-th 3D-CNN are used to initialize the weights of the i-th 3D-CNN. For all structures (1–3), with the increase of the number of blocks, the performance of our approach can become better and better. In the structure3, the DE can be reduced to 1.95 mm (block5) from 3.20 mm (block1), which shows the usefulness of the cascaded 3D-CNNs in this task. When the number of blocks is increased to 5, the performance keeps unchanging (for structure1 and structure2) or just small increment (i.e., for structure3). Moreover, when the number of blocks is set to 5, compared with the structure1, the DE of the structure2 can be reduced to 2.57 mm from 2.66 mm due to the use of data augmentation strategy, which enhances the generalization ability of our model. Finally, the improvement of the structure3 is significant, compared with the structure2, due to two main reasons. First, the i-th 3D-CNN can get better initial weights than random initialization and thus can be trained easily. Second, more importantly, we think the training data of the i-th 3D-CNN is augmented based on the training data of the $(i-1)$-th 3D-CNN, which allows our model to cover more possible transformations. All these make our 3D-CNN models improve the performance gradually.

Table 2 lists the average detection time for the whole process in our proposed approach, using a computer with the processor of Intel(R) Xeon(R) E5-2650 2.20 GHz and NVIDIA GPU P40. It should be noted that each individual time includes the time cost of reading and writing data. As we can see, our proposed approach is free of traditional image registration and has very low detection error.

We tentatively only use the traditional linear registration (TLR) method including skull stripping to detect landmarks. First, all testing images are skull-stripped and then the FLIRT [13] is used to achieve linear registration based

on the Colin27 template. Finally, we take the average landmarks of all training lin-landmarks as our detected landmarks in lin-images for every testing image and then their landmarks in corresponding ori-images can be found by utilizing linear registration matrix. In Table 3, we can see the detection result 2.93 mm is unsatisfactory if we only use TLR. Further more, if we train a 3D-CNN model based on these lin-images, the detection result can reduce to 2.05 mm while the 3D-CNN model based on ori-images only can achieve 3.20 mm as shown in Table 1, which means linear registration is very important. Comparing with our method, the traditional linear registration with skull-stripping needs more processes and gets higher detection error than our proposed method. So we think our method has more advantages and the whole procedure can be regarded as fast linear registration and seamlessly achieve landmark detection at the same time.

Table 2. The average detection time of the whole process of our proposed system.

Procedure	Implementation	Individual time	Total time
3D-CNNs (1–5) prediction	C++/MXNet [12]	13.3 s	15.9 s
LSR (block 1–4)	Python	2.3 s	
Final prediction	Python	0.3 s	

Table 3. The comparison of different methods.

Method	DE
TLR	2.93 mm
TLR+3D-CNN	2.05 mm
Our proposed method	**1.95** mm

5 Conclusion

In this study, we propose a novel and effective approach to fast detect anatomical landmarks and estimate linear transformation for image registration. It's interesting we find a clever and effective way to combine landmarks detection and linear registration. Our experimental results show a competitive advantage. For the future work, we will verify our algorithm on other anatomical landmarks and enhance our deep learning model.

Acknowledgments. This work was supported in part by the National Key R&D Program of China under contract No. 2017YFB1002202, in part by the National Natural Science Foundation of China under Grants 61671422 and U1613211, in part by the MOE-Microsoft Key Laboratory of USTC. The authors would like to thank Dr. Dinggang Shen for the contributions on implementation.

References

1. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**, 436–444 (2015)
2. Riegler, G., Urschler, M., Ruther, M., Bischof, H., Stern, D.: Anatomical landmark detection in medical applications driven by synthetic data. In: IEEE International Conference on Computer Vision Workshops, pp. 12–16 (2015)
3. Zheng, Y., Liu, D., Georgescu, B., Nguyen, H., Comaniciu, D.: 3D deep learning for efficient and robust landmark detection in volumetric data. In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (eds.) MICCAI 2015. LNCS, vol. 9349, pp. 565–572. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-24553-9_69
4. Payer, C., Štern, D., Bischof, H., Urschler, M.: Regressing heatmaps for multiple landmark localization using CNNs. In: Ourselin, S., Joskowicz, L., Sabuncu, M.R., Unal, G., Wells, W. (eds.) MICCAI 2016. LNCS, vol. 9901, pp. 230–238. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46723-8_27
5. Ghesu, F.C., Georgescu, B., Mansi, T., Neumann, D., Hornegger, J., Comaniciu, D.: An artificial agent for anatomical landmark detection in medical images. In: Ourselin, S., Joskowicz, L., Sabuncu, M.R., Unal, G., Wells, W. (eds.) MICCAI 2016. LNCS, vol. 9902, pp. 229–237. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46726-9_27
6. Zhang, J., Gao, Y., Gao, Y., Munsell, B.C., Shen, D.: Detecting anatomical landmarks for fast Alzheimers disease diagnosis. IEEE Trans. Med. Imaging **35**(12), 2524–2533 (2016)
7. Zhang, J., Liu, M., Shen, D.: Detecting anatomical landmarks from limited medical imaging data using two-stage task-oriented deep neural networks. IEEE Trans. Image Process. **26**(10), 4753–4764 (2017)
8. Hyman, B.T., Van Hoesen, G.W., Damasio, A.R., Barnes, C.L.: Alzheimer's disease: cell-specific pathology isolates the hippocampal formation. Science **225**, 1168–1171 (1984)
9. Zhang, J., Liu, M., An, L., Gao, Y., Shen, D.: Alzheimer's disease diagnosis using landmark-based features from longitudinal structural MR images. IEEE J. Biomed. Health Inform. **21**(6), 1607–1616 (2017)
10. Jenkinson, M., Bannister, P., Michael, B., Stephen, S.: Improved optimization for the robust and accurate linear registration and motion correction of brain images. Neuroimage. **17**(2), 825–841 (2002)
11. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proc. IEEE **86**(11), 2278–2324 (1998)
12. Chen, T., Mu, L., et al.: Mxnet: a flexible and efficient machine learning library for heterogeneous distributed systems. arXiv preprint arXiv:1512.01274 (2015)
13. Fischer, B., Modersitzki, J.: FLIRT: a flexible image registration toolbox. In: Gee, J.C., Maintz, J.B.A., Vannier, M.W. (eds.) WBIR 2003. LNCS, vol. 2717, pp. 261–270. Springer, Heidelberg (2003). https://doi.org/10.1007/978-3-540-39701-4_28
14. Holmes, C.J., Hoge, R., Collins, L., Woods, R., Toga, A.W., Evans, A.C.: Enhancement of MR images using registration for signal averaging. J. Comput. Assist. Tomogr. **22**(2), 324–333 (1998)

Author Queries

Chapter 65

Query Refs.	Details Required	Author's response
AQ1	This is to inform you that corresponding author has been identified as per the information available in the Copyright form.	
AQ2	Per Springer style, both city and country names must be present in affiliations. Accordingly, we have inserted the city and country name "Hefei and China" in affiliations "1 and 2". Please check and confirm if the inserted city and country name is correct. If not, please provide us with the correct city and country name.	