Ensemble of Convolutional Neural Networks for Heart Segmentation

Wilson Fok(✉), Kevin Jamart, Jichao Zhao, and Justin Fernandez

Auckland Bioengineering Institute, Auckland, New Zealand
wfok007@aucklanduni.ac.nz

Abstract. Training an ensemble of convolutional neural networks requires much computational resources for a large set of high-resolution medical 3D scans because deep representation requires many parameters and layers. In this study, 100 3D late gadolinium-enhanced (LGE)-MRIs with a spatial resolution of 0.625 mm × 0.625 mm × 0.625 mm from patients with atrial fibrillation were utilized. To contain cost of the training, down-sampling of images, transfer learning and ensemble of network's past weights were deployed. This paper proposes an image processing stage using down-sampling and contrast limited adaptive histogram equalization, a network training stage using a cyclical learning rate schedule, and a testing stage using an ensemble. While this method achieves reasonable segmentation accuracy with the median of the Dice coefficients at 0.87, this method can be used on a computer with a GPU that has a Kepler architecture and at least 3 GB memory.

Keywords: Convolutional neural network · LGE-MRI · Human atria · Pre-trained model · Cyclical learning rate · Ensemble of networks · Dice loss

1 Introduction

In computer vision, major advances in the performance of neural networks comes empirically from an increase in their number of learnable parameters and in floating point operations per seconds [1]. In 2015, an ensemble of two ResNets yielded 3.57% test error in the ImageNet classification challenge. The ensemble is computationally expensive to train as one ResNet, the version of the architecture with 152 parameters layers, contains 60-million parameters [2] and uses 11.3×10^9 floating point operations per second in a single forward pass [3]. Another example is a winning entry for brain tumor segmentation challenge (BRATS 2017); it deployed an ensemble of multiple neural networks with unique architectures that were separately trained and then put together [4].

Segmentation of images of diseased hearts with atrial fibrillation is useful scientifically and clinically. The left atrium can be imaged using 3D late Gadolinium-Enhanced Magnetic Resonance Imaging [5]. However, manual segmentation requires expert knowledge and is laborious and time-consuming. Thus, the task of the segmentation of the left atria can be benefited from the use of automated deep neural networks.

M. Pop et al. (Eds.): STACOM 2018, LNCS 11395, pp. 282–291, 2019.
https://doi.org/10.1007/978-3-030-12029-0_31

However, these advanced networks are computationally expensive to train and deploy. Because of resource constraint, one should not ignore the computational overhead during training and inference. To this end, this paper aims to emphasize the importance of the concept of reuse and to demonstrate its practical value in getting good segmentation performance at a fraction of the overhead. Reuse appears in two ways. One is transfer learning; the other is an ensemble of networks. The goal is to strike a balance between memory usage, floating point operations, and segmentation performance.

Transfer learning is commonly used in object localization/detection and segmentation [6–9]. The theory is that network's weights after training on a large image dataset such as the ImageNet dataset are useful in related tasks such as localization and segmentation. These weights can then be further fine-tuned for other imaging tasks. The advantage of transfer learning is that these weights' values provide better convergence to the objective function of a related task than those that are randomly sampled from a normal distribution. During training, these weights can be tuned, albeit at a lower learning rate, or be frozen altogether. If these weights are held fixed, one eliminates the need to perform operations to update them and to store their error gradient during backpropagation. Furthermore, ensembles of networks typically require separate training on a number of different neural networks with various architectures or different weight initialization [10]. To reduce the computational cost of training such an ensemble, [10] observed that improvement can also be gained by averaging a single network with a fixed architecture but different values of weights that were captured at different points in time during training. It has been shown that DenseNet performance can be boosted by this ensemble technique [10]. During the training phase, the learning rate was perturbed periodically at regular intervals. This is very different from the method of learning rate decay or of setting the learning rate to be dependent on the accumulation of the magnitude of the error gradients over epochs [11, 12]. [13] appears to support cyclical learning rate as it reduces training time for GoogLeNet with inception modules and AlexNet in ImageNet challenge without sacrificing the accuracy. In addition, Cyclic Cosine Annealing, a recently proposed learning rate schedule, may help to move network weight values from one local minimum of the parameter space to the next [14]. An ensemble can improve accuracy for different architectures such as DenseNet and ResNet by averaging its members [10].

This paper proposes an algorithm that combines (1) ideas from transfer learning to use weights from ResNet (networks trained on ImageNet dataset for image classification task), and (2) techniques to create an ensemble of convolutional neural networks. The method section provides information on the data used in the heart segmentation challenge, and training and evaluation methods of the ensemble of neural networks. The result section delineates the progress of training, validation and testing. The discussion section summarizes the key findings, and it shows that the lessons learned from classification tasks can be carried forward nicely to segmentation tasks. The conclusion describes future directions of this paper.

2 Methods

2.1 Data Source

The dataset contains 100 cases of 3D late gadolinium-enhanced (LGE)-MRIs with a spatial resolution of 0.625 mm × 0.625 mm × 0.625 mm. These images have 88 slices in the z direction and either 640 × 640 or 576 × 576 in the x and y directions respectively. The edges of the 640 × 640 were trimmed down to 576 × 576. No additional data source was used except this MRI dataset. These images have detailed structural information of the diseased hearts, showing atria and its scar. Heart functions are linked to their structures [15, 16]. All the images of the hearts are in grey-scale, and come with a ground truth mask that outlines the left atria.

2.2 Data Processing

As the intensity contrast of the MRIs was low, contrast limited adaptive historgram equalization technique was used to enhance its contrast [17]. This technique worked by locally enhancing the contrast of the images. It used a pixel neighborhood in which the intensity was adjusted. One of the crucial parameters was thus the size of this neighborhood/ kernel. As the kernel size varied, the output image differed. A range of kernel sizes was tested from ¼ down to 1/128 of the image's height. The final selection included sizes of 1/8 and 1/32 on an ad hoc basis. These two kernels operated on different scales and thus revealed different boundaries of the part to be segmented.

2.3 Image Processing and Augmentation

The original resolution of the MRI scans was down-sampled to 88 by 112 by 112 using a bi-cubic interpolation at each depth to save memory usage. Out of these 100 cases, training took 70%; validation 10%; testing 20%.

To augment the dataset, image transformations were performed, including histogram matching and various image transformations (rotation, cropping and translation, zooming, skewing, shearing, randomly erasing patches, tilting, flipping and reversing stack orders). The rotation operation randomly rotated the images at an angle between 0-360°. The cropping and translation operation randomly cut out 2/3 of the stack of images using a square bounding box that was randomly placed inside the scans. The zooming operation randomly zoomed in and out based on a ratio between 1.5 and 0.5. The perspective skewing calculated a transformation plane that, out of four corners (upper left, upper right, lower left or lower right), skewed toward one corner. The tilting operation calculated the transformation plane that, out of four sides (top, bottom, left and right), tilted toward one side. The shearing operation transformed the image in either x or y direction at an angle that was chosen randomly between ±25°. The erasing operation erased a rectangular patch whose width and height ranged from 10% to 100% of the input's width and height. As the width and height were chosen randomly and independently, the shape of the patch was usually a rectangle, even though a square was possible. The reverse of stack order meant the slices in the depth direction were sorted back to front. The flipping operation flipped the images either sideway or

upside down. The histogram matching first normalized the cumulative distribution of pixel intensity of a source and a template, both sampled at random from the cases. A source was to be matched to a template. The quantiles of the source's pixel intensity were mapped and adjusted to the corresponding values dictated by the template's distribution. If the amount of free hard disk space is of concern, one can opt for augmenting dataset during training on demand rather than storing the transformed images and masks on hard disk. As a result of the augmentation procedure, the training dataset increased to 11250 cases while the validation and the testing datasets both increased to 3750.

2.4 Network Training, Validation and Testing

Figure 1 is the schematic of the proposed architecture of our deep convolutional neural network. The inputs were two stacks of the contrast-enhanced MRI scans from an identical case. The 2D feature maps produced by pre-trained ResNet34 were represented by flat squares. The layers of ResNet that could extract highly expressive features were kept in the original sequential or hierarchical order, and the rest (classification layer and the pooling layer) were discarded. Specifically, these layers were layer 0–2, 4, 5, 6, 7 in Pytorch implementation. The feature maps were then concatenated to show the two input stacks at each level. Subsequently, 3D convolution was performed on them to account for the fact that the object to-be-segmented was 3D.

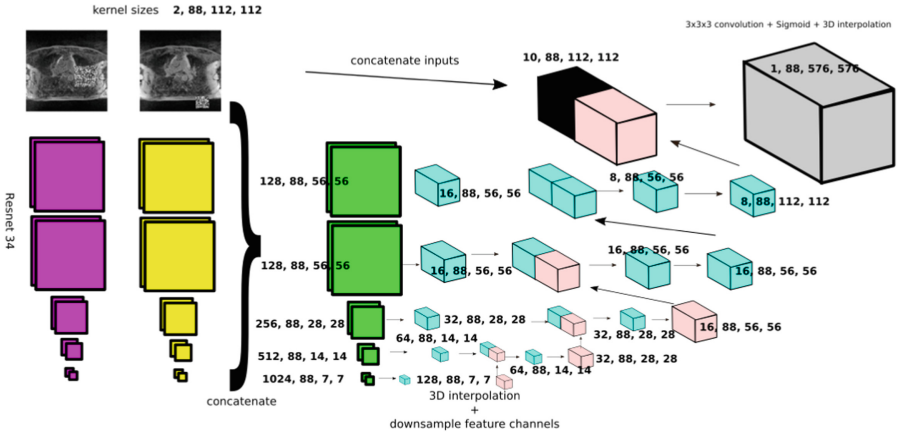

Fig. 1. An overview of the neural network architecture. The input is contrast-enhanced with 1/8 and 1/32 kernels. The weights from the ResNet34 process the stacks slice by slice in 2D. The resultant feature maps are then convolved with 3D filters at their corresponding levels (horizontal arrows). The feature cubes are interpolated to match the dimension of the upper levels and then concatenated with those higher up. Inputs are reused and concatenated to the highest level. The network predicts the mask by $3 \times 3 \times 3$ convolution, followed by sigmoid activations and interpolation. The size of the squares and cubes alludes to the change in dimension. The detailed dimension (number of feature channels, depth, height, width) is written in bold near to the layers.

3D Convolution was followed by batch-norm [18] whose outputs were passed through a rectified linear unit [19]. The network resembled U-Net or V-Net on the ascending arm [20, 21]. It expanded its resolution and joined the lower level to the upper level level-by-level. In order to match the dimensions between the upper and lower levels, interpolation was used to obtain the exact size and to minimize misalignment. Finally, the features went through a 3D $3 \times 3 \times 3$ convolution with sigmoid activation units for binary classification or segmentation. An additional interpolation pushed the predicted mask to the dimension back to the original scans. Unlike U-Net, this network's architecture is a hybrid with the descending arm composed of 2D convolutions from a pre-trained ResNet34 and ascending arm composed of 3D convolutions.

Dice loss as in [21] was used, for it seemed to be more suitable than cross-entropy loss for segmentation. It penalized both false positive and false negative and thus was more similar to that of Dice coefficient, the scoring criterion. The total loss was calculated as a sum of the Dice losses on all individual slices. On segmenting, this automatically put more weights on the smaller parts than the larger parts. Misclassification of small regions as appeared on the MR images became costlier mathematically because of the ratios among true positive, false positive and false negative in Dice loss. As the region shrank, any misclassified voxel would give rise to an increase in the loss.

Training began by randomly shuffling the training set. The learnable weights' values were initialized as in [22]. The mini-batch size was 5. Adam algorithm [11] optimized the network.

Learning rate, one of the hyper-parameters, is hard to optimize. A learning rate test as in [13] was run to estimate a suitable learning rate value range. In that paper, the suggestion is to try a wide range of learning rates for any particular configuration (such as minibatch size, dataset, objective function, and network architecture). The learning rate test began training a network from the slowest to the fastest rate, and the accuracy for each level of the rates was recorded. The optimal range was characterized by the rise in the accuracy. In this study, the learning rate test examined log10 of learning rates spanning -4 to 0. At each level, 10 weight updates were performed and the means and standard deviations of the Dice coefficients were recorded (Fig. 2a). The optimal range was later found to be between 0.003–0.03. Throughout the course of training, the learning rate schedule was a triangular waveform with a period of 50 weight updates and a peak of 0.03 and a trough of 0.003. The weights were saved at checkpoints which were set at every 200 weight updates.

Cessation of improvement was determined by a paired sample t-test. 20 randomly selected samples in the validation dataset were fed to the networks at different checkpoints. Training was terminated when t-tests showed no improvement in the validation loss. While t-test may not be necessary for determining whether difference in losses exists, it can still be useful as it quantifies the lack of difference by its p-value.

Amongst the checkpoints, six were selected based on their low validation losses. Afterwards, an ensemble of networks was formed by selecting a number of members up to all the six checkpoints. Once the ensemble was created, the ensemble's prediction was a simple average of all the predictions made by the members. To identify an ensemble that yielded low validation loss, several combinations were tried and the one with the lowest loss was kept aside. The ensemble's prediction, by nature, was

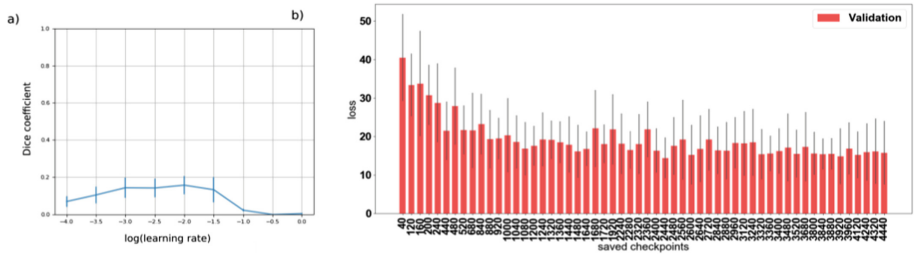

Fig. 2. (a) shows the results of a learning rate test. The optimal range for the learning rate spans over the hump (rise in Dice coefficient). The range of −1.5–−2.5 was used in training in the form of a cyclical learning rate scheme. (b) shows validation loss at different checkpoints. The validation loss rose and fell, like a ripple, as a consequence of the learning rate scheme, and it had shown little or no improvement after the ripple settled down. Those checkpionts, with lower losses, were selected in a list of candidates for an ensemble.

continuous, between 0–1, owing to the sigmoid activation function. To obtain a binary mask, the prediction was thresholded at a cutoff value. By comparing changes in Dice coefficients of predictions given by the earlier ensemble at different cutoffs, the optimal threshold was the one that yielded the highest Dice coefficient.

Testing was conducted on 63 combinations of ensembles of neural networks using up to the six aforementioned checkpoints. The smallest ensemble included a single network that was loaded with weights at a single checkpoint; the largest ensemble included six networks with weights from all the six checkpoints. Each test ran on 50 randomly selected samples in the augmented test set. The performance was quantified by the median of the Dice coefficients.

3 Experimental Results

The results are delineated in the order of training, validation and testing. Figure 2a plots the results of the learning rate test. Between –4–0, an optimal range of values sits between −1.5–−2.5 (log values)/0.003–0.03 because of higher average Dice coefficients. This optimal range was set to be the peak and tough of the cyclical learning rate schedule. Figure 2b shows the Dice loss of validation dataset at various checkpoints. The rhythmic rise and fall of the loss mirrors the cyclical learning rate. The peaks of the learning rate waveform have likely assisted the network in exploring the possible parameters' space, thereby dislodging the weights from one local minimum to the next. The price tag is a temporary increase in loss. Paired sample t-test reveals no significant evidence that the loss of checkpoint 4440 is different from that of checkpoint 3920 (p-value: 0.72). This suggests further training may not be needed. The binary mask was obtained by thresholding the network's prediction. The optimal threshold is at 0.37, giving a Dice coefficient of 0.85.

The evaluation of the performance of the ensembles was performed in two ways, using the test set. The first is a graphic method in which the network's mistake and

correctness were plotted against the ground truth. The second is a quantitative way in which the median of Dice coefficients was calculated.

Figure 3 shows a side-by-side comparison between a ground truth and a predicted mask estimated by an ensemble of a test case. While the region of true positive highly aligns with that of the ground truth, the regions of the false negative and false positive are small and very small respectively. Figure 4 summarizes the top 16 medians of the Dice coefficients produced by ensembles of different combinations of neural networks. The performance of the ensembles is sensitive to group size and combinations of networks with a specific selection of weights. It is worth pointing out that only the first and the seventh columns represent the score given by a single network. The rest are ensembles of two or more networks.

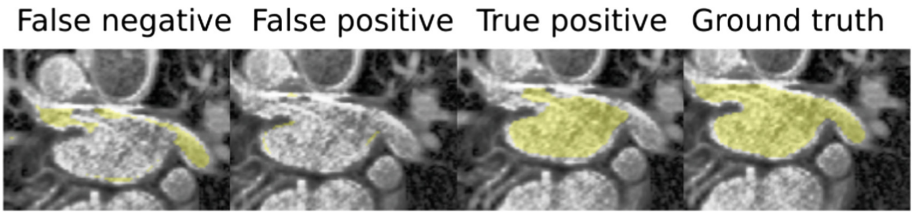

Fig. 3. A side-by-side comparison between the ground truth mask and a predicted mask estimated by an ensemble during testing. This case obtained a Dice coefficient of 0.87. Yellow region highlights the false positive, the false negative, and the true positive of the prediction as well as the ground truth. The shape formed by true-positive pixels resembles the shape of the ground truth. (Color figure online)

4 Discussion

The findings of this paper align with the existing view of the behavior, in general, of ensembles of neural networks and with the effect of using cyclical learning rate schedules in training. However, it is important to emphasize the difference between this work and the existing literature. The existing view is that ensemble outperforms individuals in terms of robustness and accuracy [10, 23]. The performance does not guarantee to improve by a merely increase in size. The optimal combinations for an ensemble are normally found by trial and error. The finding in this paper confirms this view. The ensemble of networks in Fig. 4 often produces slightly better results than its individual members because, amongst the 16, only two were single networks. As the Dice coefficient rises, it is hard to identify a clear pattern of combination of networks that would work best. The performance gain is hypothesized to be a result of averaging predictions by various sets of weights at saddle points or local minimums on the path of parameter optimization during training.

The cyclical learning rate schedule has been observed to produce an oscillating pattern [13, 14]. [10] deployed similar schedules for training and had found that the oscillation was a consistent hallmark. The exploration using fast learning rate can be beneficial in the long run despite temporary increase in loss as observed in this work.

Fig. 4. The size and membership of an ensemble can often produce variation in the level of accuracy. The above plot highlights the median of Dice coefficients produced by ensembles of different combinations of neural networks at different checkpoints. A black square means the inclusion of that checkpoint. A white square means the opposite. The series of black squares down each column represents the configuration of that ensemble. The x axis shows the Dice coefficient in an ascending order. The y axis shows which checkpoints were used in the ensemble of networks' weights.

What makes this work novel is that the ensemble's objective is to segment images whereas the existing work in the literature concerns image classification. Although segmentation can be regarded as a classification task between yes or no, it is still non-identical to classifying many classes of objects such as 1000 different classes in ImageNet Classification Challenge. Nonetheless, the finding of this work indicates that the existing tricks for training neural networks also apply in segmentation tasks. The benefit of using transfer learning, cyclical learning rate schedule and ensembles is reduced computational overhead. In this work, while training needs only 2.4 GB of memory on GPU, testing requires no more than 3.0 GB. Consequently, it is possible to implement this work on a computer with a GPU that has a Kepler architecture and at least 3 GB memory.

5 Conclusions and Future Work

This work demonstrates a feasible pipeline that helps train an ensemble with limited computational resources. It suggests that the tricks used in image classification can be carried forward to segmentation. The future work will include evaluations of different learning rate schedules, and investigations of the benefits and harms of various image augmentation techniques. As learning rate is often set to monotonically decay during training, future studies should compare how the generalization of ensembles improves under various learning rate decay schedules and other cyclical learning rate schedules with different waveforms, periods, and peak and tough values. The speed of network training will also be explored under the influence of different schedules. Moreover, this preliminary work deploys many image transformation techniques, but their potential impacts on ensemble's performance are unclear. In other words, how the transformation techniques work to boost the ensemble's ability to achieve good segmentation in unseen patient cases should be considered. Future work will investigate the balance

between the computational overhead on the image augmentation and performance gains, and will consider ways to fine tune or adjust parameters of these techniques as to better reflect the characteristics of MRI data of the hearts.

References

1. Canziani, A., Paszke, A., Culurciello, E.: An analysis of deep neural network models for practical applications. arXiv preprint arXiv:1605.07678 (2016)
2. Boulch, A.: ShaResNet: reducing residual network parameter number by sharing weights. arXiv preprint arXiv:1702.08782 (2017)
3. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
4. Kamnitsas, K., et al.: Ensembles of multiple models and architectures for robust brain tumour segmentation. arXiv preprint arXiv:1711.01468 (2017)
5. McGann, C., et al.: Atrial fibrillation ablation outcome is predicted by left atrial remodeling on MRI. Circ. Arrhythmia Electrophysiol. **7**(1), 23–30 (2014)
6. Girshick, R.: Fast R-CNN, arXiv preprint arXiv:1504.08083 (2015)
7. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask R-CNN. In: 2017 IEEE International Conference on Computer Vision (ICCV), pp. 2980–2988 (2017)
8. Iandola, F., Moskewicz, M., Karayev, S., Girshick, R., Darrell, T., Keutzer, K.: Densenet: implementing efficient convnet descriptor pyramids. arXiv preprint arXiv:1404.1869 (2014)
9. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 779–788 (2016)
10. Huang, G., Li, Y., Pleiss, G., Liu, Z., Hopcroft, J.E., Weinberger, K.Q.: Snapshot ensembles: train 1, get M for free. arXiv preprint arXiv:1704.00109 (2017)
11. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
12. Tieleman, T., Hinton, G.: Lecture 6.5 - RMSProp, COURSERA: Neural Networks for Machine Learning. Technical report (2012)
13. Smith, L.N.: Cyclical learning rates for training neural networks. In: 2017 IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 464–472 (2017)
14. Loshchilov, I., Hutter, F.: SGDR: Stochastic Gradient Descent with Warm Restarts. arXiv preprint arXiv:1608.03983 (2016)
15. Hansen, B.J., et al.: Atrial fibrillation driven by micro-anatomic intramural re-entry revealed by simultaneous sub-epicardial and sub-endocardial optical mapping in explanted human hearts. Eur. Heart J. **36**(35), 2390–2401 (2015)
16. Zhao, J., et al.: Three-dimensional integrated functional, structural, and computational mapping to define the structural 'Fingerprints' of heart-specific atrial fibrillation drivers in human heart ex vivo. J. Am. Heart Assoc. **6**(8), e005922 (2017)
17. Zuiderveld, K.: Contrast limited adaptive histogram equalization. In: Heckbert, P.S. (eds.) Graphics Gems, pp. 474–485. Academic Press, Cambridge (1994)
18. Ioffe, S., Szegedy, C.: Batch normalization: accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167 (2015)
19. Nair, V., Hinton, G.E.: Rectified linear units improve restricted boltzmann machines. In: Proceedings of the 27th International Conference on Machine Learning (ICML-10), pp. 807–814 (2010)

20. Çiçek, Ö., Abdulkadir, A., Lienkamp, S.S., Brox, T., Ronneberger, O.: 3D U-Net: learning dense volumetric segmentation from sparse annotation. In: International Conference on Medical Image Computing and Computer-Assisted Intervention, pp. 424–432 (2016)
21. Milletari, F., Navab, N., Ahmadi, S.-A.: V-net: fully convolutional neural networks for volumetric medical image segmentation. In: 2016 Fourth International Conference on 3D Vision (3DV), pp. 565–571 (2016)
22. He, K., Zhang, X., Ren, S., Sun, J.: Delving deep into rectifiers: surpassing human-level performance on imagenet classification. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1026–1034 (2015)
23. Pereira, S., Pinto, A., Alves, V., Silva, C.A.: Brain tumor segmentation using convolutional neural networks in MRI images. IEEE Trans. Med. Imag. **35**(5), 1240–1251 (2016)
24. Xiong, Z., Fedorov, V., Fu, X., Cheng, E., Macleod, R., Zhao, J.: Fully automatic left atrium segmentation from late gadolinium enhanced magnetic resonance imaging using a dual fully convolutional neural network. IEEE Trans. Med. Imag. (2018) (in Press)