Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2018) 26: 23 – 34

doi:10.3906/elk-1704-335

Estimating left ventricular volume with ROI-based convolutional neural network

Feng ZHU[1,2,*]

[1]Department of Software Engineering, Faculty of Computer Science and Engineering, Jiangsu University of Science and Technology, Zhenjiang, P.R. China

[2]Department of Medicine, Yong Loo Lin School of Medicine, National University of Singapore, Singapore

Received: 28.04.2017 • **Accepted/Published Online:** 05.09.2017 • **Final Version:** 26.01.2018

Abstract: The volume of the human left ventricular (LV) chamber is an important indicator for diagnosing heart disease. Although LV volume can be measured manually with cardiac magnetic resonance imaging (MRI), the process is difficult and time-consuming for experienced cardiologists. This paper presents an end-to-end segmentation-free method that estimates LV volume from MRI images directly. The method initially uses Fourier transform and a regression filter to calculate the region of interest that contains the LV chambers. Then convolutional neural networks are trained to estimate the end-diastolic volume (EDV) and end-systolic volume (ESV). The resulting models accurately estimate the EDV and ESV with a mean absolute error of 15.83 and 9.82 mL, respectively, and an ejection fraction with root mean square error of 5.56%. The comparison results show that the direct estimation methods possess attractive advantages over the previous segmentation-based estimation methods.

Key words: Convolutional neural network, region of interest, left ventricle volume, magnetic resonance imaging, deep learning

1. Introduction

Cardiovascular diseases, also known as heart diseases, seriously affect our health and lives. In clinical practice, the volume of the human left ventricular (LV) chamber is an important indicator for diagnosing heart diseases. Doctors usually assess the heart's squeezing ability by measuring end-diastolic volume (EDV) and end-systolic volume (ESV) — that is, the volume of the heart when it is contracted and after it is filled with blood. Although we can manually measure the LV volume from magnetic resonance imaging (MRI) slices, the process of measuring the volume of the heart at different stages of a cardiac cycle is difficult and time-consuming, even for experienced cardiologists. Thus, there exists a need for an automatic and robust human LV volume estimation system that can help and enhance doctors' ability to diagnose heart conditions more efficiently.

To estimate cardiac ventricular volumes, a conventional idea is to first segment the cardiac ventricular cavity in each MRI slice [1], then calculate the ventricular volume by summing up the volumes of different slices. Two representative LV segmentation works were based on the level set [2] and graph cut [3] methods, respectively. Unlike segmentation-based methods, segmentation-free methods try to estimate the cardiac ventricular volumes from MRI directly [4–7]. For instance, Afshin et al. explored the problem of direct estimation of LV volumes from image statistics [4]. A limitation of this approach was that it required the user to give two boxes in the reference image when building image statistics. To estimate biventricular volume directly, Wang et al. introduced an

*Correspondence: zhufeng@just.edu.cn

adapted Bayesian model to search for similar images in a set of manually segmented LV/RV templates and then simply calculated the cardiac ventricular volumes as the weighted average over the templates [5]. The method simplified the statistical relationship between image features and ventricular volumes. Therefore, it did not generalize well on more diverse data sets. To tackle the above-mentioned issue, Zhen et al. proposed a direct and learning-based biventricular volume estimation framework [6]. The framework applied a multiscale deep network for unsupervised cardiac image representation learning from unlabeled data and then trained random forests on labeled data for biventricular volume estimation. The proposed framework showed great advantages over previous cardiac ventricular volume estimation methods. However, due to the fact that the result was based on a relatively small data set, the method should be further validated on a larger data set. Meanwhile, since the method did not provide an explicit contour, it would pose a learning curve to those who are used to contouring for validation. Recently, Kabani and Sakka presented a direct method to estimate the LV volume [7]. In their work, they used three different deep convolutional networks to train the LV localization network, the slice localizer network, and the volume estimation network, respectively. The method was effective, but it required the user to input masks and train three networks, which led to overreliance on external resources and a complicated workflow.

Compared with previous segmentation-based methods, segmentation-free methods achieved a more accurate and efficient estimation by avoiding an intermediate segmentation step. However, as discussed above, there are still several problems, such as user input dependency, complex workflow, and a relatively small data set, which often raises questions about the generalization ability that need to be resolved. Additionally, the diversity of the MRI source and the poor quality of the image often affect the accuracy and robustness of the ventricular volume estimation methods. To simplify the workflow and reduce the dependency on user input, we explored an end-to-end segmentation-free LV volume estimation method. The approach implemented ROI-based convolutional neural networks (CNNs) to predict the EDV and ESV from MRI time series data in different axis views. As a powerful deep learning model, CNNs can automatically learn hierarchies of features from many images, which can be applied to various classification and regression tasks [8–10]. Figure 1 shows the overall flowchart of the method.

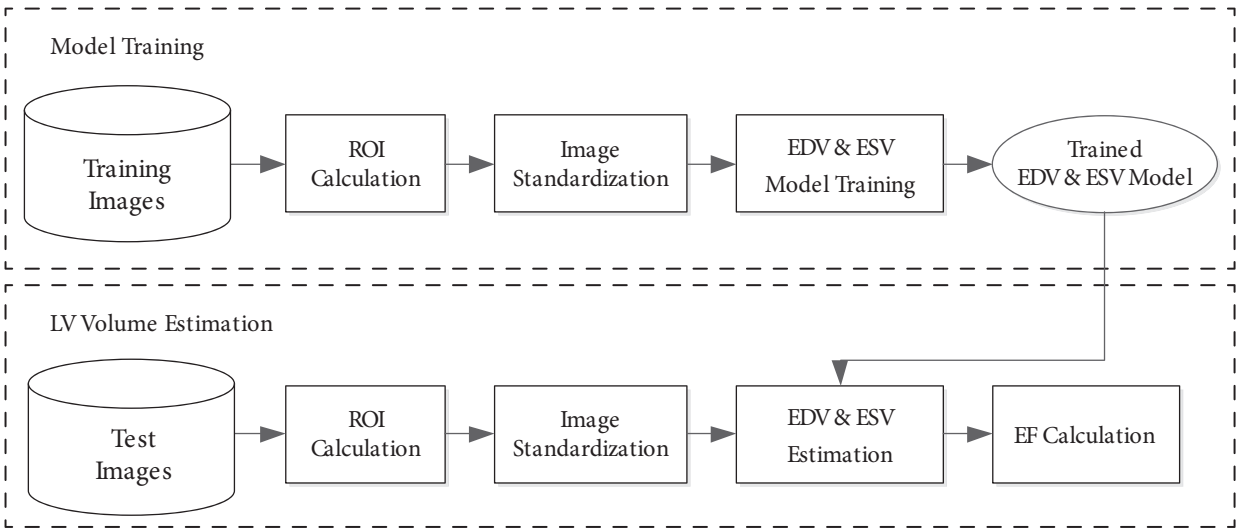


Figure 1. Flowchart of the proposed LV volume estimation method. Upper block: Calculate ROI from training images, standardize the images, and train EDV and ESV model. Bottom block: Calculate ROI from test images, standardize the images, and estimate EDV and ESV according to the corresponding trained model.

The method initially calculated the region of interest (ROI) containing the heart for each slice using the Fourier transform (FT) approach. Next, we standardized the images and fed them into 16-layer CNNs to train the end-systolic and end-diastolic regression models separately. Finally, the method estimated the EDV and ESV according to the corresponding regression model and calculated the ejection fraction (EF) value. The contributions of this work include the design and implementation of a segmentation-free LV volume estimation model using ROI-based CNNs and the simplification of the workflow and reduced dependency on user input.

The remainder of this paper is structured as follows: Section 2 gives a detailed description of the materials and our method. Section 3 shows the experimental results, and Section 4 concludes with a summary and discusses future steps.

2. Materials and methods

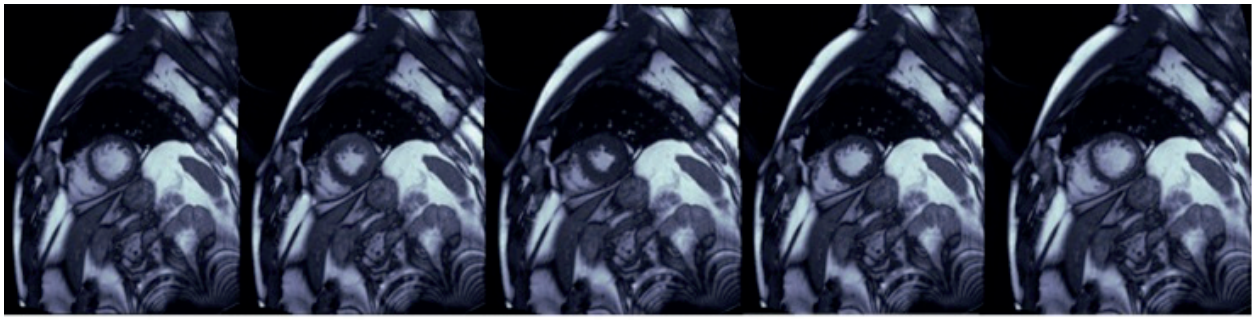
2.1. Data set and features

Our method was implemented and evaluated on the current largest data set released in the second national Kaggle Data Science Bowl. The data set was compiled by the National Institutes of Health and Children's National Medical Center. It has thousands of cardiac MRI images in digital imaging and communications in medicine (DICOM) format for a total of 1140 different patients, including 500 sets of data and their associated systole and diastole volumes for training, 200 for validation, and 440 for testing. For each patient, we were given a total of 30 frame images across a single cardiac cycle. These images were taken in different planes, including a two-chamber view (2Ch), a four-chamber view (4Ch), and a series of longitudinal slices perpendicular to the heart's long axis, known as the short-axis stack (SAX). Figure 2 shows example images of different frames and slices. Several patients had all or more of these slices, while others only had some of them. In addition to these multiview data, the DICOM files contained a lot of metadata. Certain metadata fields, such as pixel spacing, slice thickness, and the patient's age and sex, can provide us with heuristic information to estimate the LV volume.

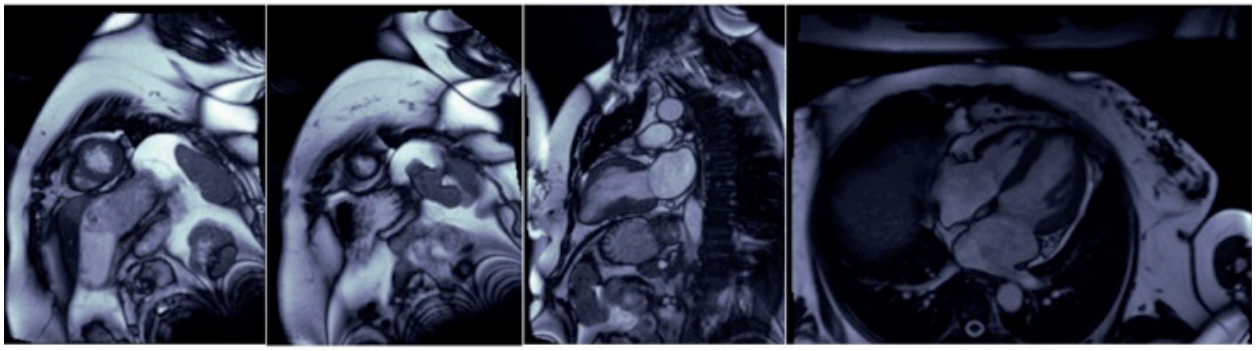
Although the number of data was significantly higher than before, their inconsistency made it difficult to robustly estimate the LV volume. First, the data varied widely in terms of both number of slices per study and number of images per slice. For example, there were 22 SAX slices in Study 436 and only one in Study 499. Similarly, there were 330 images in Study 334 SAX_29 and only 22 in Study 416 SAX_9. In addition to the dimensional variance, the size of the images, pixel resolution, and slice thickness varied across studies. Furthermore, the LV images had varying levels of brightness and contrast. All these factors increased the difficulty of further network training. In our approach, the original 500 training studies and 200 validation studies were merged in a training data set, where 84% were used for training and the remaining 16% were used for validation. In each study, 2Ch and 4Ch view slices were ignored and only SAX view slices were used for training. To maintain the consistency of the training data, all slices that had more than 30 frames in the training set were removed. In this way, more than 200,000 images were eventually used for model training.

2.2. ROI calculation and image standardization

ROI calculation is a critical stage in cardiac volume estimation. It can give unexpected boosts to both processing speed and accuracy for our CNN model. Since most blood flows through the left ventricle in the heartbeat cycle, the LV location is expected to have the highest variation throughout the time series of the slice. This feature allows us to distinguish between the heart and other tissues in the thorax by analyzing the intensity values at pixel locations that change over time. Figure 3 shows changes of pixel intensities over time in positions P1 and P2. Apparently, the pixel intensity of P1 changes significantly in comparison with P2.



(a) Study 1 SAX_13 Frame 1, 6, 12, 18, 30



(b) Study 1 SAX_13, SAX_15, 2Ch, 4Ch

Figure 2. Multiview cardiac MRI images: a) example images of different frames in Study 1 SAX_13, b) example images of different view slices in Study 16.

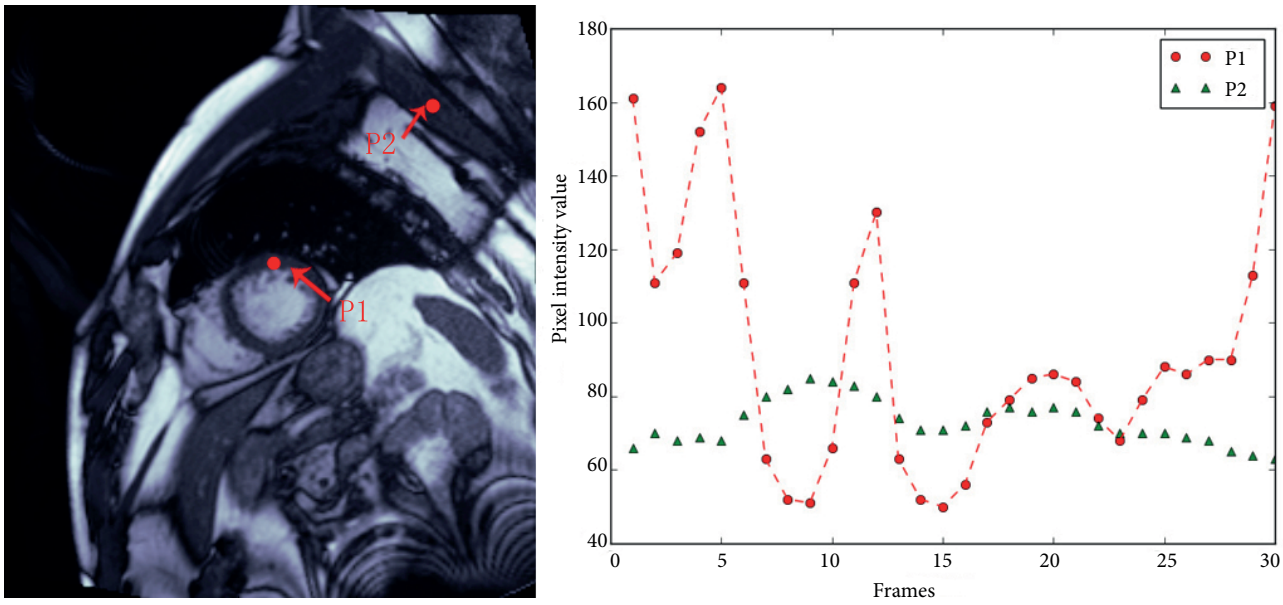


Figure 3. Illustration of pixel intensity change over time. The left part is the original cardiac MRI image with P1 and P2 at different locations. The right part indicates changes of pixel intensities over time in positions P1 and P2.

As was suggested in [11], there were three critical steps to determine the ROI. Figure 4 gives the flowchart of ROI calculation.

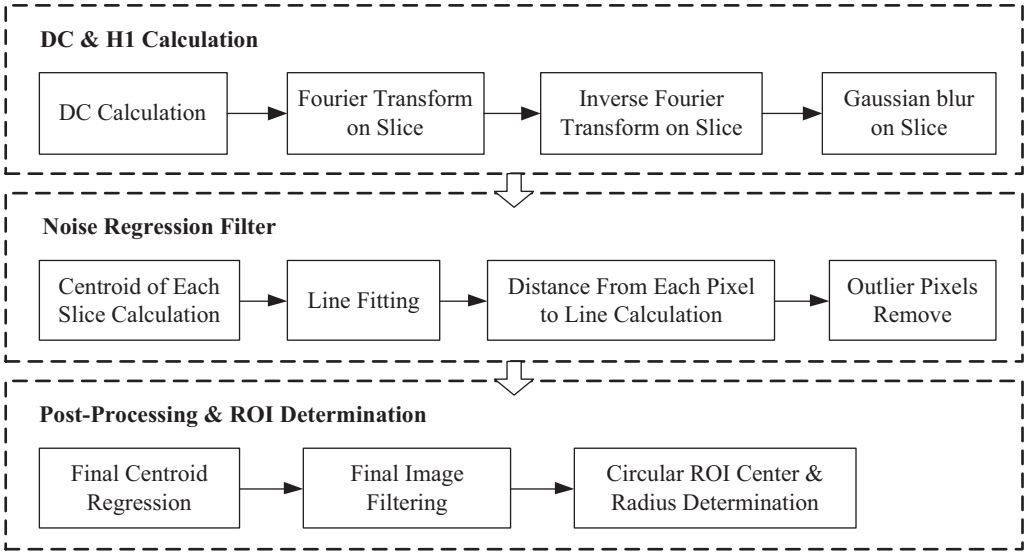


Figure 4. Flowchart of the ROI calculation. Upper block: Calculate the DC images using the average grayscale value of each pixel in each slice, and then compute the H1 images using FT to extract images that capture the maximal activity at the corresponding heartbeat frequency. Middle block: Calculate the centroids from the H1 images and fit a 3D line to these centroids, then compute a weighted distance from each pixel to the line and remove the pixels from the center of the heart. Bottom block: Perform the centroid regression and image filtering again to obtain the final filtered H1 images and the parameters of the final 3D line, and then compute the radius to determine the circular ROI.

The first step was to calculate the DC and H1 components for each slice. The DC component of the signal was equivalent to the average grayscale value of each pixel in each slice over time. Figure 5a shows one example of a DC image. The calculation of the first harmonic (H1) component was somewhat complicated. Since each slice can be seen as a separate 2D + T signal, the n-dimensional fast Fourier transform (FFT) was applied to each slice. Then the H1 component of the transform was converted into our original signal data by inverse FFT. When all the H1 components were obtained, a Gaussian blur operation was performed to reduce the effect of the signal from noncardiac structures. Specifically, all pixels that were less than 5% of the maximum value of all H1 components were set to zero. Figure 5b shows one example of a final H1 image.

After obtaining the DC and H1 images, there were still some nonzero pixels in the H1 image because the aorta in the thorax will contract and relax with the heartbeat. Therefore, the next step was to remove the pixels from the center of the heart with an iterative regression strategy. In each iteration, the algorithm first computed the centroids from the H1 image for each slice and then fit a 3D line to the centroids of all slices by linear least squares. Afterwards, a weighted distance from each pixel to the line was calculated. Finally, all pixels falling in the outliers were removed from the H1 images.

The last step was postprocessing and ROI determination. In postprocessing, the final centroid regression and image filter were performed again on H1 images. Then the circular ROIs were determined on the final filtered H1 images and each DC image. The center of each ROI was the point where the 3D line intersected with the slice, and the estimated radius was based on the criterion of maximizing the proportion of nonzero pixels in the circle. Figure 5c shows the final circular ROI image.

(a) (b) (c)

Figure 5. Example images in determining circular ROI: a) example of the DC image, b) example of the H1 image, c) example of the final circular ROI image.

After determining the ROI, the images were resized using the pixel spacing metafield in the DICOM files to ensure that the LV cavity area was consistent across all images. Then each image was cropped to 64 $\times$ 64 pixels and all the slices were stacked together to create a single input for each study.

2.3. Model training

Deep learning architectures, such as deep neural networks, CNNs, and recurrent neural networks have been applied to fields including computer vision, speech recognition, natural language processing, biomedical image processing, and computational biology [12]. Among these architectures, CNNs are one of the most popular architectures, owing to their outstanding performance in the various tasks of object recognition and image classification. The basic structure of CNNs consists of convolutional layers, nonlinear layers, pooling layers, and fully connected layers. The convolutional layer is the core building block of a CNN. The layer's parameters consist of a set of learnable filters (or kernels), which have a small receptive field but extend through the full depth of the input volume. After each convolutional layer, it is common to apply a nonlinear layer (or activation layer) to increase the nonlinear properties of the model. After several nonlinear layers, it is common to apply a pooling layer (or downsampling layer). There are several nonlinear functions to implement pooling, among which max-pooling is the most common. Finally, after several convolutional and max-pooling layers, the high-level reasoning in the neural network is performed by fully connected layers.

The VGG net is one of the most influential CNN architectures due to its neat structure and satisfactory performance in many localization and recognition studies [9]. In the VGG net structure, all the convolution kernels are of size 3 $\times$ 3. A max-pooling layer is made after 2 or 3 layers of convolutions, and the number of filters is doubled after each max-pooling. Taking into account the advantages of the VGG net, the VGG16-like CNN, which has 13 convolution layers and three full connected layers, was designed to train the model. As illustrated in Figure 6, the model took 30 frames in a single SAX slice as 30 separate input channels to the network and trained two different networks: one for EDV estimation and the other for ESV estimation.

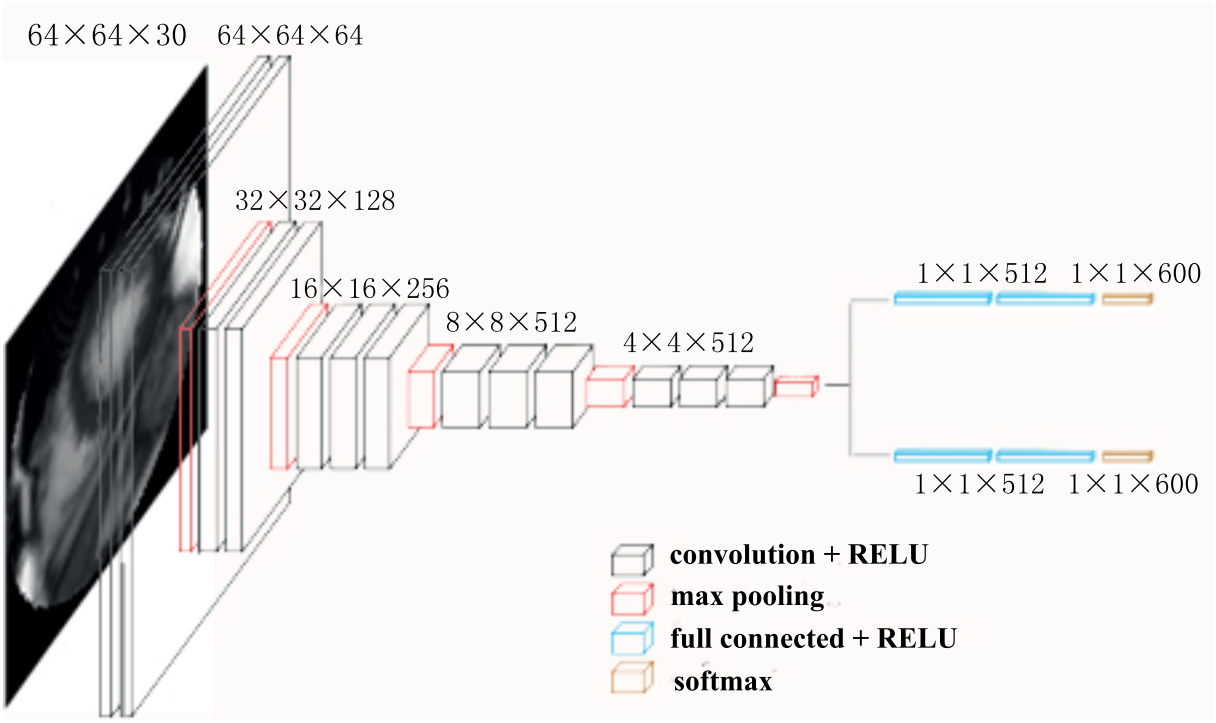


Figure 6. CNN architecture for EDV and ESV model training. The input is 30 images with 64 × 64 pixels in a single SAX slice. The output is a 600-way softmax.

Considering the efficiency of training, the two networks shared the convolution layers and pooling layers; however, the full connected layers were separated. The convolution layers applied a specified number of convolution filters to extract feature maps with filter size 3 × 3. To introduce nonlinearity to the network, a ReLU activation layer was used after each convolution layer. Then a max-pooling layer with the same padding and a stride of 1 was applied to output the maximum number in every subregion that the filter convolved around. Next, every node in the full connected layer was connected to every node in the preceding layer. The output of each network was a 600-way softmax, followed by a cumulative sum layer. Table 1 gives detailed settings for the parameters of each layer in the network.

Due to the small size of the training data, it is better to perform data augmentation to boost the performance of our deep learning model. In our work, different types of affine transforms were applied to the data augmentation while keeping the geometry of the image unchanged. It is found that data augmentation can effectively improve the generalization ability of the model. Unlike traditional data augmentation, which increased the data set before training, we augmented the data during training, which allowed us to combine the image rescaling and augmentation into a single affine transform. The augmentation parameters are as follows:

- Rotation: random with angle between –180 and 180 degrees
- Translation x-axis: random with shift between –8 and 8 mm
- Translation y-axis: random with shift between –8 and 8 mm
- Flipping: flipping images left–right or up–down with probability 0.5

Table 1. Detailed settings for the parameters of each layer.

Layer type	Filter number, size	Number of parameters	Output shape
Input layer	30 filters, 3 × 3	0	(32, 30, 64, 64)
Convolution	64 filters, 3 × 3	17,344	(32, 64, 64, 64)
Convolution	64 filters, 3 × 3	36,928	(32, 64, 64, 64)
Max-pooling		0	(32, 64, 32, 32)
Convolution	128 filters, 3 × 3	73,856	(32, 128, 32, 32)
Convolution	128 filters, 3 × 3	147,584	(32, 128, 32, 32)
Max-pooling		0	(32, 128, 16, 16)
Convolution	256 filters, 3 × 3	295,168	(32, 256, 16, 16)
Convolution	256 filters, 3 × 3	590,080	(32, 256, 16, 16)
Convolution	256 filters, 3 × 3	590,080	(32, 256, 16, 16)
Max-pooling		0	(32, 256, 8, 8)
Convolution	512 filters, 3 × 3	1,180,160	(32, 512, 8, 8)
Convolution	512 filters, 3 × 3	2,359,808	(32, 512, 8, 8)
Convolution	512 filters, 3 × 3	2,359,808	(32, 512, 8, 8)
Max-pooling		0	(32, 512, 4, 4)
Convolution	512 filters, 3 × 3	2,359,808	(32, 512, 4, 4)
Convolution	512 filters, 3 × 3	2,359,808	(32, 512, 4, 4)
Convolution	512 filters, 3 × 3	2,359,808	(32, 512, 4, 4)
Max-pooling		0	(32, 512, 2, 2)
Full connected	512 units	1,049,088	(32, 512)
Full connected	512 units	262,656	(32, 512)
Full connected	600 units	307,800	(32, 600)

3. Experiments and results

3.1. Implementation details

All the experiments were performed at the National Supercomputing Centre Singapore (NSCC) with a configuration of 12 cores CPU E5-2690v3, NVIDIA Tesla K40 GPU, and 128 GB RAM per node. Next, the specific implementation of the model is discussed.

The weights of the network were trained using the Adam stochastic gradient descent optimization algorithm [13]. First, the Adam algorithm updates exponential moving averages of the gradient (m_t) and the squared gradient (v_t) with Eqs. (1)–(3).

$$m_t = \beta_1 m_{t-1} + (1-\beta_1)\, g_t \tag{1}$$

$$v_t = \beta_2 v_{t-1} + (1-\beta_2)\, g_t^2 \tag{2}$$

$$g_t = \nabla_\theta f_t(\theta_{t-1}) \tag{3}$$

Here, the hyperparameters β_1, $\beta_2 \in [0, 1)$ are exponential decay rates for the moment estimates; $f(\theta)$ is a stochastic objective function with parameters θ; and g_t denotes the gradient, i.e. the vector of partial derivatives of f_t, w.r.t θ evaluated at time step t. With m_t and v_t, the algorithm counteracts these biases by computing bias-corrected first estimates ($\hat{m}_t$) and second-moment estimates ($\hat{v}_t$) with Eqs. (4) and (5).

$$\hat{m}_t = \frac{m_t}{1-\beta_1^t} \tag{4}$$

$$\hat{v_t} = \frac{v_t}{1 - \beta_2^t} \tag{5}$$

Finally, the algorithm uses Eq. (6) to update the parameters.

$$\theta_t = \theta_{t-1} - \frac{\alpha}{\sqrt{\hat{v_t} + \varepsilon}} \hat{m}_t \tag{6}$$

In our experiments, the settings of the parameters were $\alpha = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\varepsilon =$ 1e-8. All parameters were set empirically using the default values from [13] and it was shown that the values work well in practice.

Overfitting is another common problem in network training, especially in the case of fewer training data. In our experiments, the original 500 training data and 200 validation data were combined. Then 83% of the training data set was used for training and the remaining 17% of the data were used for validation. In this way, we prevented overfitting by monitoring the loss in the validation set. Furthermore, the dropout technique was used to prevent our models from overfitting [14]. This technique can be viewed as mathematically equivalent to an approximation to the probabilistic deep Gaussian process [15]. During the training, several neurons and their connections were randomly dropped in every forward pass. This prevented the network from getting too fitted to the training data and thus helped alleviate the overfitting problem. In the experiments, the dropout ratio was set to 0.5. During the training, weights for the best iteration were saved in PKL files so that they could be loaded later for prediction.

As mentioned in the Kaggle contest, the evaluation metric that was used for our model is the continuous ranked probability score (CRPS). The CRPS is interpreted as the average squared distance between the predicted cumulative distribution function (CDF) and the ground truth distribution. The CRPS is computed as follows:

$$C = \frac{1}{600N} \sum_{m=1}^{N} \sum_{n=0}^{599} \left(P\left(y \leq n\right) - H\left(n - V_m\right) \right)^2 \tag{7}$$

Here, P is the predicted distribution (i.e. the predicted CDF of the volume). N is the number of rows in the test set, V is the actual LV volume ($V \in \{0, 1, 2, \ldots, 599\}$), and H(x) is the Heaviside step function (H(x) = 1 for x $\geq$ 0 and zero otherwise). Because no ground truth value was larger than 600 mL, the support of the distribution is taken to be between 0 and 599 mL.

3.2. Results and analysis

In our experiments, different hyperparameters were selected to train the model. After that, we showed and compared the results to other segmentation-based and direct estimation methods.

Figure 7 shows CRPS losses for both diastolic and systolic models. With the validation data, the lowest diastole and systole CRPS loss was 0.0279 and 0.0177, respectively. As shown in Figure 7, it can be found that either in the training set or the validation set the CRPS loss of systolic model was always less than the loss of the diastolic model. We believe that this was because the size of the systolic volume changes less than that of the diastolic volume, which made systolic volume easier to estimate. The training took about 7 h and reached optimal CRPS loss at 0.0228, which was the average of the lowest diastole and systole CRPS loss. The LV volume estimation error was 0.014 $\pm$ 0.015.

Figure 8 gives the results for our diastolic and systolic models fitting the test data. The vertical distance between the line and dots or triangles represents the estimation error for that patient. The resulting models accurately estimated the EDV and ESV with a mean absolute error of 15.83 and 9.82 mL, respectively. As illustrated in Figure 8, several outliers were present in either the EDV or the ESV estimation. For example, No. 81 in estimated diastole and No. 96 in estimated systole both refer to patient No. 756. Their estimated volumes are significantly lower than the actual volume. By observing the original image data, the patient was diagnosed with severe hypertrophic cardiomyopathy cardiovascular disease. The patient's heart wall was thicker, which caused the model to be unable to learn features well. It was also possible that the image in the training set was unbalanced, resulting in a relatively large bias in the estimation. Moreover, the model was not ideal for estimating extremely large and small volumes. These conditions showed that the robustness of the model needed improvement.

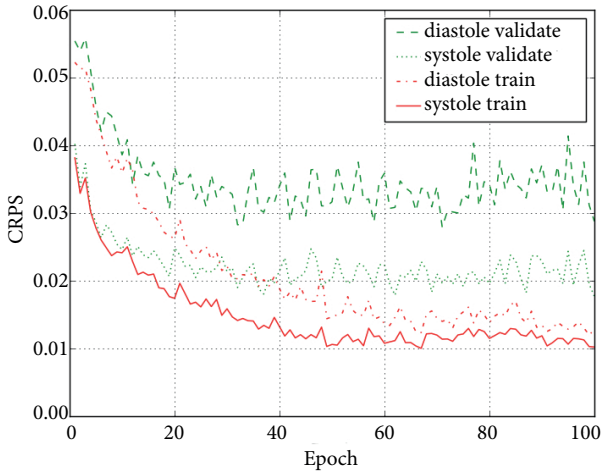


Figure 7. CRPS losses for both diastolic and systolic models. The red solid line and red dotted line represent systolic and diastolic training losses, respectively. The green dashed line and green dotted line represent systolic and diastolic validating losses, respectively.

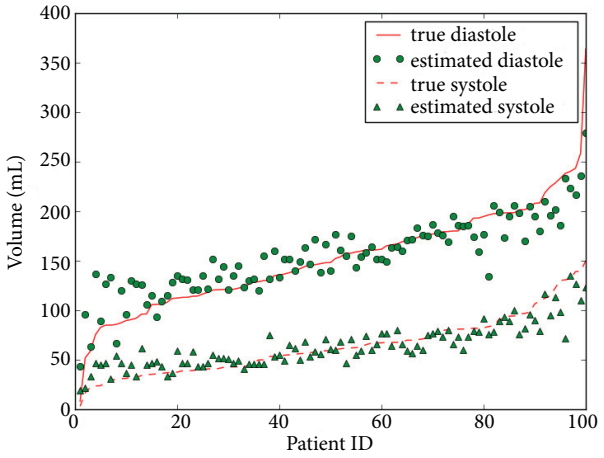


Figure 8. Diastolic and systolic models fitting on the test data. The red lines are ground truth EDVs and ESVs, which are plotted in increasing order. The green dots are estimated EDVs and the green triangles are estimated ESVs.

Ejection fraction, i.e. the amount or percentage of blood ejected from the left ventricle with each heartbeat, is a key indicator of the heart-squeezing function. A low EF value is often an early sign of heart failure or other types of heart disease. The EF value can be calculated as in Eq. (8):

$$EF = 100 \times \frac{V_D - V_S}{V_D} \tag{8}$$

As shown in Figure 9, the red lines show ground truth EFs and green dots indicate estimated EFs. The RMSE of the estimated EF is 5.56%. It can be seen from the figure that most EF values are concentrated between 50% and 70%. However, several EF values were less than 50% or higher than 70%, which can help the cardiologist to examine these values further.

In [6], the authors made a detailed comparison of their method with other existing estimation methods. The comparison results showed that the direct estimation methods possessed attractive advantages over the previous segmentation-based estimation methods. As illustrated in Table 2, our experiment results validated this view.

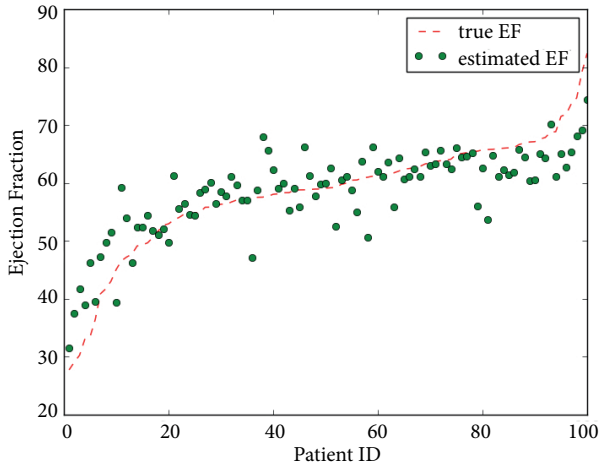


Figure 9. Estimated EFs fitting on the ground truth EFs. The red dashed line shows ground truth EFs and the green dots indicate estimated EFs.

Table 2. Comparison results of estimation errors for LV volumes.

Methods	LV volume estimation errors
Our method	0.014 ± 0.015
Zhen et al. [6]	0.010 ± 0.011
Wang et al. [5]	0.016 ± 0.019
Ayed et al. [3] (level set)	0.036 ± 0.025
Ayed et al. [2] (graph cut)	0.029 ± 0.027

In contrast with [6], although the results of our approach were not the best among the direct estimation methods, our experiments were carried out on a larger data set. Additionally, the workflow of our method is simple and does not need much user intervention. Furthermore, the results were compared with the method of [7] for the same data set. As shown in Table 3, our method achieved a degree of improvement in both ESV and EF metrics.

Table 3. Comparison of EDV, ESV, and EF errors.

Methods	EDV error (mL)	ESV error (mL)	EF error (%)
Our method	15.83	9.82	5.56
Kabani et al. [7]	15.82	11.16	5.64

To verify the effect of the layer number on the performance of the model, we implemented a 9-layer network. The preliminary study showed that the 16-layer net achieved better results than the 9-layer net.

4. Conclusion

This paper has described an end-to-end segmentation-free method to estimate the LV EDV and ESV from cardiac MRI images. ROIs were calculated with FT to improve the performance of the model. CNNs were used to extract features and combine information from different axes views to predict volumes. ROI calculation and model optimization are two critical processes for the final result. The former is responsible for keeping constant the input dimension and eliminating noise, whereas the latter is in charge of improving the accuracy of the model. In the future, accuracy and robustness will be further improved by ensembling different models.

Acknowledgments

This research was supported by the Jiangsu Overseas Research and Training Program and was partially funded by a PhD research project at Jiangsu University of Science and Technology. The author thanks the Genome Institute of Singapore, the National Supercomputing Center of Singapore, and the National University of Singapore for providing scientific research resources. The author also thanks the Kaggle website for providing the experimental data.

References

[1] Petitjean C, Dacher JN. A review of segmentation methods in short axis cardiac MR images. Med Image Anal 2011; 15: 169-184.

[2] Ayed BI, Li S, Ross I. Embedding overlap priors in variational left ventricle tracking. IEEE T Med Imaging 2009; 28: 1902-1913.

[3] Ayed BI, Punithakumar K, Li S, Islam A, Chong J. Left ventricle segmentation via graph cut distribution matching. In: ACM 2009 Medical Image Computing and Computer-Assisted Intervention; 20–24 September 2009; London, UK. New York, NY, USA: ACM. pp. 901-909.

[4] Afshin M, Ayed BI, Punithakumar K, Law M, Islam A, Goela A, Peters T, Li S. Regional assessment of cardiac left ventricular myocardial function via MRI statistical features. IEEE T Med Imaging 2014; 33: 481-494.

[5] Wang Z, Salah BM, Gu B, Islam A, Goela A, Li S. Direct estimation of cardiac bi-ventricular volumes with an adapted Bayesian formulation. IEEE T Bio-Med Eng 2014; 61: 1251-1260.

[6] Zhen X, Wang Z, Islam A, Bhaduri M, Chan I, Li S. Multi-scale deep networks and regression forests for direct bi-ventricular volume estimation. Med Image Anal 2016; 30: 120-129.

[7] Kabani AW, El-Sakka MR. Estimating ejection fraction and left ventricle volume using deep convolutional networks. Lect Notes Comp Sci 2016; 9730: 678-686.

[8] Krizhevsky A, Sutskever I, Hinton GE. ImageNet classification with deep convolutional neural networks. In: NIPS 2012 Advances in Neural Information Processing Systems Conference; 3–8 December 2012; Lake Tahoe, NV, USA. La Jolla, CA, USA: NIPS. pp. 1097-1105.

[9] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. arXiv preprint 2014; 1409.1556.

[10] Szegedy C, Liu W, Jia YQ, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke Vet, Rabinovich A. Going deeper with convolutions. In: IEEE 2015 Conference on Computer Vision and Pattern Recognition; 7–12 June 2015; Boston, MA, USA. New York, NY, USA: IEEE. pp. 1-9.

[11] Lin X, Cowan B, Young A. Automated detection of left ventricle in 4D MR images: experience from a large study. In: ACM 2006 Medical Image Computing and Computer-Assisted Intervention Conference; 1–6 October 2006; Copenhagen, Denmark. New York, NY, USA: ACM. pp. 728-735.

[12] LeCun Y, Bengio Y, Hinton G. Deep learning. Nature 2015; 521: 436-444.

[13] Kingma D, Ba J. Adam: A method for stochastic optimization. arXiv preprint 2014; 1412.6980.

[14] Srivastava N, Hinton GE, Krizhevsky A, Sutskever I, Salakhutdinov R. Dropout: a simple way to prevent neural networks from overfitting. J Mach Learn Res 2014; 15: 1929-1958.

[15] Gal Y, Ghahramani Z. Dropout as a Bayesian approximation: Representation model uncertainty in deep learning. In: ACM 2016 International Conference on Machine Learning; 19–24 June 2016; New York, NY, USA. New York, NY, USA: ACM. pp. 1050-1059.