Computers in Biology and Medicine 111 (2019) 103334

Contents lists available at ScienceDirect

Computers in Biology and Medicine

journal homepage: www.elsevier.com/locate/compbiomed

EVCMR: A tool for the quantitative evaluation and visualization of cardiac MRI data

Yoon-Chul Kim[a], Khu Rai Kim[b], Kwanghee Choi[c], Minwoo Kim[c], Younjoon Chung[c], Yeon Hyeon Choe[d,*]

[a] Clinical Research Institute, Samsung Medical Center, Sungkyunkwan University School of Medicine, Seoul, South Korea
[b] Department of Electronic Engineering, Sogang University, Seoul, South Korea
[c] Department of Computer Science and Engineering, Sogang University, Seoul, South Korea
[d] Department of Radiology and HVSI Imaging Center, Heart Vascular Stroke Institute, Samsung Medical Center, Sungkyunkwan University School of Medicine, Seoul, South Korea

ARTICLE INFO

Keywords:
MRI
Python
Heart
Image segmentation
Visualization
Deep learning

ABSTRACT

Quantitative evaluation of diseased myocardium in cardiac magnetic resonance imaging (MRI) plays an important role in the diagnosis and prognosis of cardiovascular disease. The development of a user interface with state-of-the-art techniques would be beneficial for the efficient post-processing and analysis of cardiac images. The aim of this study was to develop a custom user interface tool for the quantitative evaluation of the short-axis left ventricle (LV) and myocardium. Modules for cine, perfusion, late gadolinium enhancement (LGE), and T1 mapping data analyses were developed in Python, and a module for three-dimensional (3D) visualization was implemented using PyQtGraph library. The U-net segmentation and manual contour correction in the user interface were effective in generating reference myocardial segmentation masks, which helped obtain labeled data for deep learning model training. The proposed U-net segmentation resulted in a mean Dice score of 0.87 (±0.02) in cine diastolic myocardial segmentation. The LV mass measurement of the proposed method showed good agreement with that of manual segmentation (intraclass correlation coefficient = 0.97, mean difference and 95% Bland-Altman limits of agreement = 4.4 ± 12.2 g). C+ + implementation of voxel-wise T1 mapping and its binding via pybind11 led to a significant computational gain in calculating the T1 maps. The 3D visualization enabled fast user interactions in rotating and zooming-in/out of the 3D myocardium and scar transmurality. The custom tool has the potential to provide a fast and comprehensive analysis of the LV and myocardium from multi-parametric MRI data in clinical settings.

1. Introduction

Cardiac magnetic resonance (CMR) involves no ionizing radiation and utilizes a variety of imaging sequence mechanisms with an aim to comprehensively evaluate in vivo cardiac morphology and function [1]. In particular, cardiac cine, perfusion, late gadolinium enhancement (LGE), and T1 mapping are the imaging sequences often used in clinical CMR settings [2]. Quantification of cardiac MRI elicits the extraction of important imaging biomarkers, including end-diastolic volume, wall thickness, ejection fraction, myocardial upslope, scar volume, and myocardial extracellular volume fraction (ECV) [3]. Comprehensive analysis of multi-parametric cardiac MRI data is often performed in clinical research. Quantitative assessment in cardiac MRI relies on commercial software tools, provided mainly by the MRI scanner vendors. The primary limitation of commercial software is the lack of information about what is inside the function. Due to this “black-box” characteristic, researchers resort to custom in-house tools or publicly available software tools, which are more transparent in terms of the visibility of the source code than commercial software.

Open-source CMR post-processing tools were demonstrated in the literature to overcome the limitation of commercial software tools. Segment is a comprehensive Matlab-based software package, developed to cover almost all of the cardiovascular image post-processing tasks on MRI data [4]. Recently, Segment extended its features to the image processing of cardiac computed tomography (CT) and single photon emission computed tomography (SPECT) data. Although it provides a

[*] Corresponding author. Department of Radiology and HVSI Imaging Center, Heart Vascular Stroke Institute, Samsung Medical Center, Sungkyunkwan University School of Medicine, 81 Ilwon-ro, Gangnam-gu, Seoul, 06351, South Korea.
E-mail address: yhchoe@skku.edu (Y.H. Choe).

https://doi.org/10.1016/j.compbiomed.2019.103334
Received 15 March 2019; Received in revised form 24 May 2019; Accepted 17 June 2019

Table 1
Comparison of the proposed tool with other publicly available cardiac MR analysis tools and highlights of the features in the proposed tool.

	Segment	MRmap	EVCMR
Operating system	Windows	Windows, Linux, Mac	Windows
Programming language	Matlab 2014a	IDL 7.0	Python 3.6
Function support	Cine, perfusion, LGE, T1/T2/T2* mapping, flow, etc.	T1/T2/T2* mapping	Cine, perfusion, LGE, T1 mapping
Deep learning-based segmentation	No	No	Yes
3D visualization	Yes	No	Yes
Export of segmentation masks	No	No	Yes

variety of powerful functionalities, it is sophisticated such that non-expert users may experience difficulties in using the tool. MRmap is an open-source software package for the measurement of cardiac T1, T2, and T2* relaxation maps for myocardial tissue characterization and was written in Interactive Data Language (IDL) 7.0 [5]. It is straightforward to use, but its tasks are limited to the measurement of relaxation maps. However, to our knowledge, Python-based open-source software tools have not been demonstrated for cardiac MRI analysis of the myocardium, although Python is gaining popularity with its characteristic of rapid development of advanced medical image processing methods [6]. Table 1 compares publicly available CMR analysis software tools.

The purpose of this paper was to demonstrate a graphical user interface (GUI) software tool, named EVCMR (quantitative Evaluation and Visualization of Cardiac short-axis Magnetic Resonance imaging data). EVCMR aims to provide 1) user-interactive analysis of cardiac cine, perfusion, LGE, and T1 mapping data, 2) automatic myocardial segmentation based on deep learning, and 3) accelerated T1 mapping based on pybind11 and C++. In addition, the implementations of three-dimensional (3D) visualization of the myocardium, 3D mapping of scar transmurality, and the multi-plane view of the myocardial upslope or ECV maps are described, and the visualization results are demonstrated.

2. Methods

Fig. 1 illustrates the schematics of CMR image analysis pipeline, the scope of the functionalities, and features of the tool. Importantly, as shown in Fig. 1b, the interface tool helps collect the labeled data in a convenient and user-interactive fashion and feed them as input to the deep learning algorithm, while the deep learning training can provide the interface tool with an improved segmentation model. We present data acquisition protocols and associated imaging parameters, followed by the development and validation of deep learning segmentation models. Operational procedures and underlying methodologies of the user interface tool are presented for image analysis and visualization.

2.1. CMR data acquisition

Cardiac MRI scans were performed on a 1.5T scanner (Siemens Avanto, Erlangen, Germany). Clinical MR examinations were approved by our institutional review board, and informed consent was obtained from the subjects prior to MR examinations. Details of the four imaging sequences relevant to our image analysis development are described below.

In cardiac cine imaging, a balanced steady-state free-precession (SSFP) sequence was used to acquire 10–12 contiguous short-axis slices for a complete coverage of the LV. Each slice was acquired per breath-hold and contained 30 dynamic cardiac phases after retrospective electrocardiogram (ECG) triggering. Imaging parameters were as follows: slice thickness = 6 mm, echo time (TE) = 1.3 ms, field of view (FOV) = 350 mm × 300 mm, pixel size = 1.29 mm × 1.29 mm, and spacing between slices = 10 mm.

In cardiac perfusion imaging, the subjects were scanned immediately after the injection of a bolus of 0.1 mmol/kg gadobutrol (Gadovist; Bayer

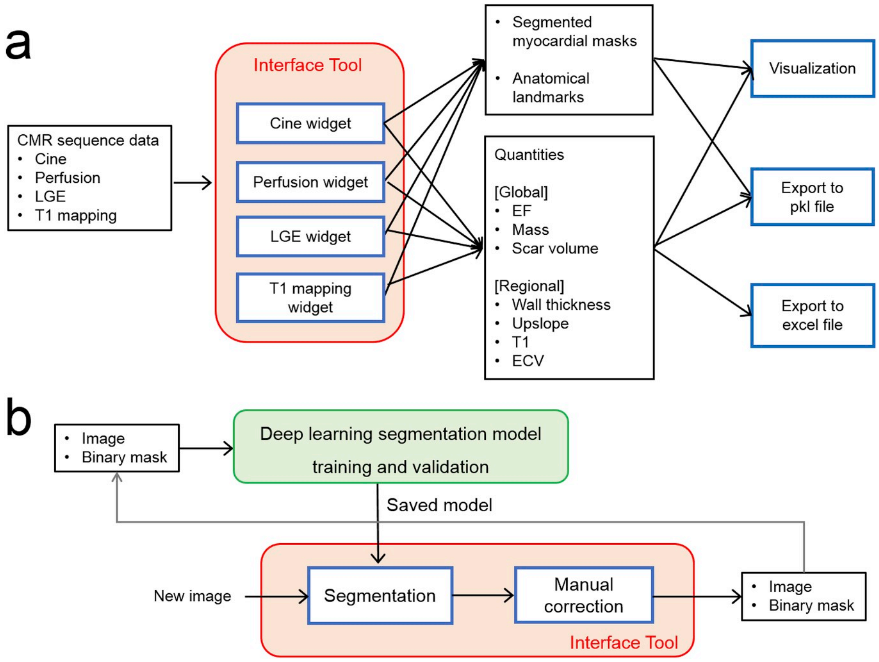


Fig. 1. Schematic diagram of the EVCMR. (a) Data processing flow and scope of the interface tool functionalities. (b) Main characteristic of the tool as a generator of labeled training data for deep learning segmentation. A deep learning segmentation model trained with a larger dataset can provide the tool with improved segmentation accuracy and robustness. AbbreviationsCMR = cardiac magnetic resonance; LGE = late gadolinium enhancement; EF = ejection fraction; ECV = extracellular volume fraction.

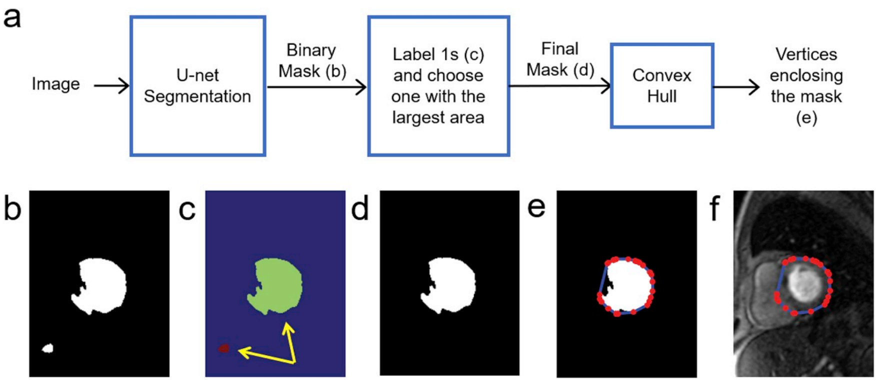

a
Image
U-net Segmentation
Binary Mask (b)
Label 1s (c) and choose one with the largest area
Final Mask (d)
Convex Hull
Vertices enclosing the mask (e)
b
c
d
e
f

Fig. 2. Segmentation workflow and an example of image post-processing in the case of segmentation error, characterized by having more than one binary objects. (a) A flowchart of segmentation and image post-processing. (b) Binary epicardial segmentation. There are two separate 1s in the binary epicardial image. (c) Automatic labeling of the 1s results in the green-colored island and the red-colored island (see yellow arrows). (d) The final epicardial mask is obtained by choosing the island with the largest area. (e) Convex hull of the final epicardial mask. The vertices are shown in red dots, and the line segments connecting the vertices are indicated by the blue line. (f) Final epicardial vertices and line segments are overlaid to the input image.

Healthcare, Berlin, Germany). They were instructed to hold their breath as long as possible during the scan. Saturation preparation pulse, followed by a fast gradient echo pulse sequence, was used. Imaging parameters were in-plane resolution = 2.5 mm × 3.0 mm, slice thickness = 8 mm, FOV = 400 mm × 315 mm, image matrix = 320 × 252, image pixel spacing = 1.25 mm × 1.25 mm, and slice-to-slice acquisition time interval = 170 ms. Four short-axis slice locations were prescribed sequentially from the basal to the apical slice levels with a slice interval of 14.4–17.6 mm. Four slice image data were acquired within every R-R interval. The imaging lasted 80 R-R intervals. Motion correction was performed to align the short axis view of the heart across the frames [7].

In LGE imaging, an inversion time (TI) scout scan was performed to optimize the inversion time for effective nulling of the healthy myocardium. After the choice of optimal inversion time, 9–10 short-axis slices were acquired during breath-holds in diastole, 15 min after a bolus injection of the contrast agent. An inversion recovery sequence was used for magnetization preparation, and a balanced SSFP sequence was used for the readout. Imaging parameters were as follows: slice thickness = 6 mm, TE = 1.2 ms, inversion time (TI) = 280–360 ms, FOV = 400 mm × 362 mm, image matrix = 256 × 232, pixel size = 1.56 mm × 1.56 mm, and spacing between slices = 10 mm.

The modified Look-Locker inversion-recovery (MOLLI) sequence was used for T1 mapping [8,9]. Pre- and post-contrast T1 mapping data were obtained in three slices: basal, mid, and apical levels. Post-contrast data were obtained 10 min after a bolus injection of the contrast agent. Imaging parameters were as follows: slice thickness = 8 mm, TE = 1.01 ms, spacing between slices = 20 mm, number of phase encoding steps = 104, pixel bandwidth = 1085 Hz, acquisition matrix = 192 × 120, image matrix = 384 × 288, pixel spacing = 0.9375 mm × 0.9375 mm, and FOV = 360 mm × 270 mm. MOLLI 5 (3)3 sequence was used for the pre-contrast T1 mapping, and MOLLI 4 (1)3 (1)2 sequence was used for the post-contrast T1 mapping. Images were acquired in diastole. Motion correction was performed in order to align all the TI images [10].

Image data obtained using the above-mentioned four imaging sequences were exported to DICOM files, which were used for the development of the tool.

2.2. Deep learning segmentation

We adopted a U-net architecture [11] for simultaneous segmentation of epicardial and endocardial masks, as U-net has proven effective in medical image segmentation. For U-net training, the publicly available York [12] data (n = 33) and our internal data (n = 77) were used. After data augmentation using grid deformation, the total number of images were 17,683, including the entire slice range (from apical to basal) and systolic/diastolic frames. A total of 13,535 images from 88 subjects were used for training, and 4,148 images from 22 subjects were used for validation. The combined use of the data helped increase the volume of our training data and reduce the time for data labeling. The dimensions of cardiac cine images, endocardial masks, and epicardial masks used for U-net training were 256 × 216 after cropping the periphery of the image. The image intensity was scaled to [0, 255], and each image was saved in .png format. The U-net model implementation was modified from Dong et al.‘s work [13] in TensorLayer and was tested on an 8 GB GPU (NVIDIA GeForce GTX 1080). For training, we used the following parameters: epochs = 100, batch size = 4, Adam optimizer with different learning rates = [0.000001, 0.00001, 0.0001, 0.0002], and beta = 0.9. The training and validation loss functions were each plotted with respect to the epoch number. The learning rate of 0.0001 showed relatively lower values in the validation than other learning rates and was chosen as the U-net model parameter for testing. The epoch number of 40 was chosen because it resulted in a relatively low loss value in the validation. As the epoch number was higher than 40, the validation loss showed an increasing pattern, which is indicative of overfitting in the training. After training, we saved the U-net model and used it for the segmentation task in our user interface.

To test the U-net model, we used cardiac MR data from 32 subjects (26M/6F; age 62 ± 9 years; 8 patients with coronary artery disease (CAD), 8 healthy subjects, 8 patients with aortic stenosis (AS), and 8 patients with hypertrophic cardiomyopathy (HCM)). Manual segmentation served as a reference. The input to the U-net segmentation model was a short-axis slice image, and the output was endocardial and epicardial binary masks. Additional post-processing was performed after the U-net segmentation. The U-net segmentation might produce more than one binary objects, which are not connected with respect to each other (Fig. 2b and c). We applied the automatic labeling of each binary object and calculated the area of each labeled mask. The mask with the largest area was selected as the final endocardial (or epicardial) mask. The U-net segmentation accuracy was measured using the Dice similarity coefficient metric:

$$\mathrm{DSC}(\mathrm{A},\ \mathrm{B}) = \frac{2\cdot \mathrm{n}(\ \mathrm{A}\cap\ \mathrm{B})}{\mathrm{n}(\mathrm{A}) + \mathrm{n}(\mathrm{B})} \tag{1}$$

where A and B are sets of labeled segmentation masks, and $\mathrm{n}(\mathrm{S})$ is the number of labeled voxels in a set S. $\mathrm{DSC}(\mathrm{A},\ \mathrm{B}) = 1$ indicates a complete overlap between A and B, while $\mathrm{DSC}(\mathrm{A},\ \mathrm{B}) = 0$ indicates no overlap. We evaluated the Dice similarity scores in myocardial segmentations of cine, perfusion, and LGE datasets in test subjects.

From the epi- or endo-cardial binary mask generated by the U-net segmentation model, we constructed a polygon using convex hull (Fig. 2e). The vertices comprising the polygon can serve as points adjustable by user interaction, which helps manually correct the epi- or endo-cardial contours, in case of segmentation error, in the user interface.

2.3. GUI implementation

The method was implemented using PyQT libraries [14] in a computer environment of 64-bit Microsoft Windows 10 operating system

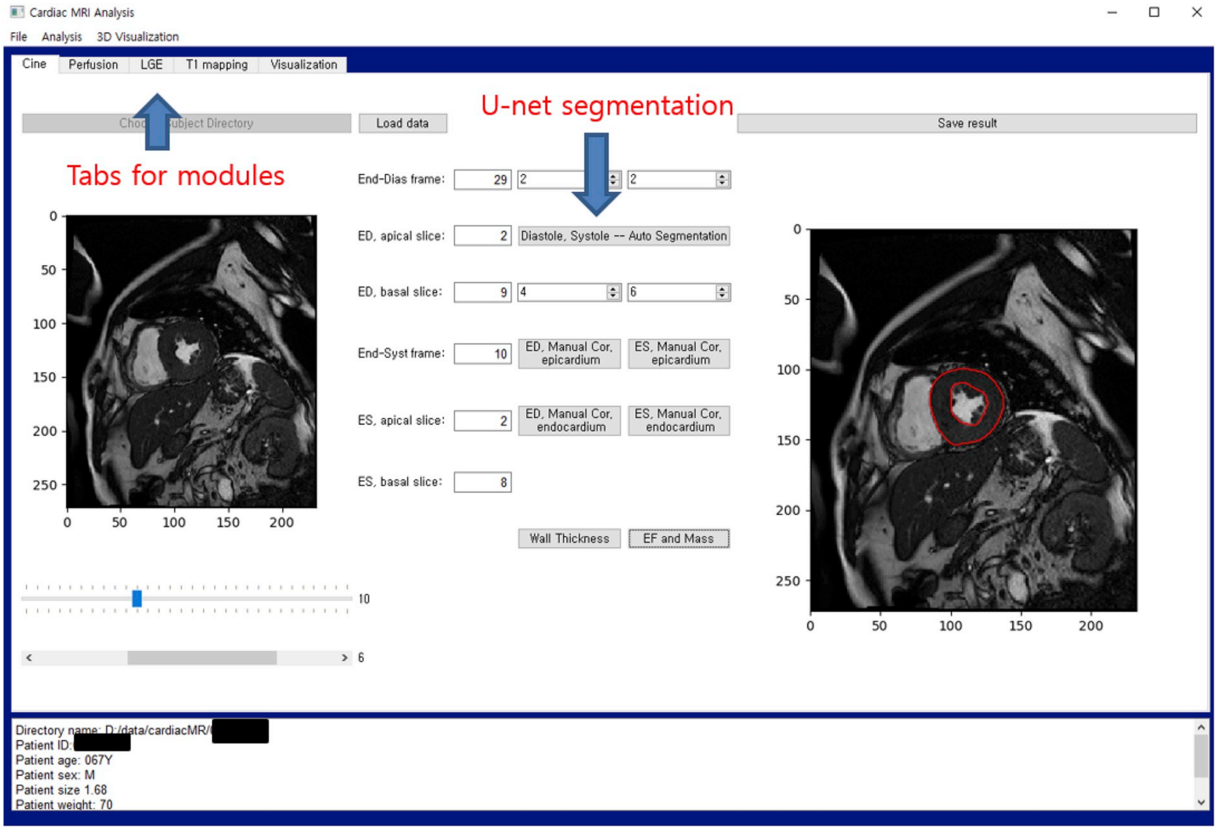
Table 2
The main Python libraries used for development.

	Usage	Classes/Functions	Libraries
User interface	Layout	QMainWindow, QAction, QTabWidget, QWidget, QSlider, QPushButton, QSpinBox	PyQt5 [14]
Display	Image display	FigureCanvasQTAgg, Polygon	Matplotlib [15]
Segmentation	Myocardial segmentation in cine, perfusion, and LGE	InputLayer, Conv2d, MaxPool2d, Deconv2d, ConcatLayer	tensorlayer.layers [35]
	Post processing	Regionprops	Skimage.measure [36]
		label	Scipy.ndimage.measurements [37]
		ConvexHull	Scipy.spatial
Perfusion	Upslope calculation	polyfit, poly1d	Numpy [38]
	Colormap overlay onto gray-scale image	ma.masked_where	Numpy
T1 mapping	T1 fitting	t1_fit (custom function)	Pybind11 [19]
	Non-rigid registration	BSplineTransformInitializer, SetOptimizerAsLBFGSB	SimpleITK [39]
Visualization	polygonal mesh	measure.marching_cubes	Scikit-image [36]
	Visualization of 3d mesh	opengl.GLMeshItem	PyQtGraph [40]
	Visualization of multiple planes	opengl.GLSurfacePlotItem	PyQtGraph
	Slice interpolation	ndimage.morphology.distance_transform_edt	Scipy [37]
	Scar transmurality calculation	ndimage.map_coordinates	Scipy
	colormap	GradientEditorItem.colorMap	PyQtGraph

with AMD Eight-Core CPU @ 3.60 GHz and 16 GB RAM. In this study, U-net segmentation was used for cine, perfusion, and LGE myocardial segmentation. Display of figures on the GUI was handled using the Matplotlib [15]. When the user clicked and selected a directory named with subject number, the directory was treated as a root directory. The 'cine', 'perf', 'lge', and 't1mapping' sub-directories were created under the root directory. Each sub-directory had DICOM files which were to be read in for image analysis. Table 2 lists the Python libraries used for the development. The software tool is available at https://sites.google.com/site/yoonckim1/software/evcmr.

2.3.1. *Cine analysis*

The user scrolls the sliders for the slice and frame navigation to identify diastolic and systolic frames and their corresponding slice range covering the LV myocardium (Fig. 3). Automatic myocardial segmentation is performed using the trained U-net model. A developed manual contour correction algorithm, which is compatible with the PyQt user interface, enables manual adjustments of the vertices comprising a polygon. In our case, the endocardial mask (or epicardial mask) has a polygonal shape. Subtraction of the endocardial mask from the epicardial mask results in a myocardial mask. The user identifies some of the vertices that contribute to the segmentation errors and



End-Dias frame: 29 2 2
ED, apical slice: 2
ED, basal slice: 9 4 6
End-Syst frame: 10
ES, apical slice: 2
ES, basal slice: 8

Directory name: D:/data/cardiacMR/[redacted]
Patient ID: [redacted]
Patient age: 067Y
Patient sex: M
Patient size 1.68
Patient weight: 70

Fig. 3. Layout of the cardiac cine analysis interface. The interface allows for automatic myocardial segmentation in end-diastole and end-systole by loading a trained U-net model. It also supports the manual adjustment of the boundaries. After the segmentation of the epicardial and endocardial borders in systole and diastole, the diastolic wall thickness, ejection fraction, and cardiac mass can be computed.

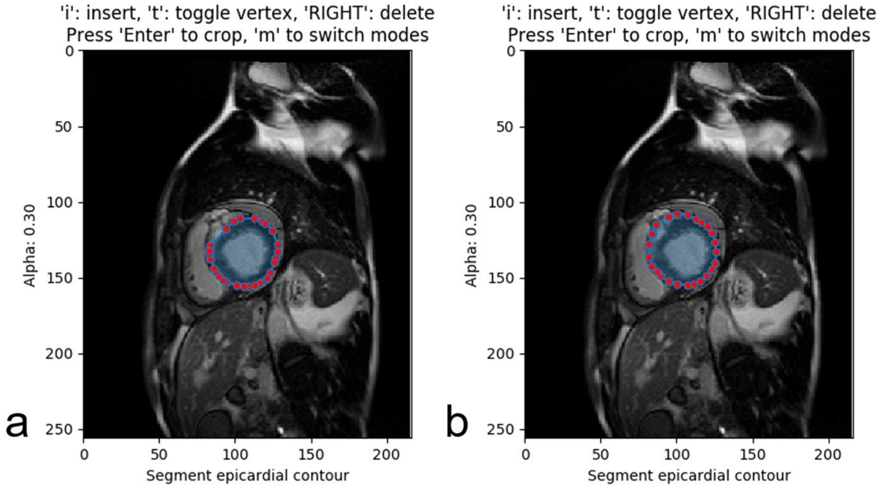

'i': insert, 't': toggle vertex, 'RIGHT': delete
Press 'Enter' to crop, 'm' to switch modes
Alpha: 0.30
0
50
100
150
200
250
Segment epicardial contour
a
b

Fig. 4. Manual contour adjustment in myocardial segmentation. (a) An example of the epicardial contour after automatic U-net segmentation. The vertices extracted from the epicardial binary mask require the user's adjustment of the vertices for correction of the border. (b) After manual adjustment, the vertices are correctly located on the epicardial border. After pressing the 'Enter' key, the corrected epicardial border is updated.

places them on the border (Fig. 4). The manual correction routine is also available in perfusion, LGE, and T1 mapping analyses.

After segmentation of all slices, the LV ejection fraction (EF), LV end-diastolic volume (EDV), LV end-systolic volume (ESV), and LV mass are calculated as described in Eqs (2)–(5).

$$\text{LV EDV} = \sum_{z=1}^{N} N_{\text{LV, ED}}(z)\cdot V_{\text{vox}} \tag{2}$$

$$\text{LV ESV} = \sum_{z=1}^{N} N_{\text{LV, ES}}(z)\cdot V_{\text{vox}} \tag{3}$$

$$\text{LV EF} = \frac{\text{LV EDV} - \text{LV ESV}}{\text{LV EDV}}\cdot 100\,(\%) \tag{4}$$

$$\text{LV mass} = \rho\cdot \sum_{z=1}^{N} N_{\text{myo, ED}}(z)\cdot V_{\text{vox}} \tag{5}$$

Here, z is slice index, N is the number of slices, and V_{vox} is voxel volume. $N_{\text{LV,ED}}(z)$ is the number of end-diastolic LV cavity voxels at slice z. $N_{\text{LV,ES}}(z)$ is the number of end-systolic LV cavity voxels at slice z. $N_{\text{myo,ED}}(z)$ is the number of end-diastolic myocardial voxels at slice z. ρ is myocardial density, and $\rho = 1.05$ (g/mL) was used in this study. The results of LV EDV and LV ESV are expressed in milliliters (mL) while the results of LV mass are expressed in grams (g). The final myocardial masks can be saved and used for other tasks such as 3D visualization and supervised machine learning for segmentation.

The quantitative results of the LV EDV, ESV, EF, and mass obtained from the proposed interface were compared with those obtained from the commercial software (Argus, Siemens Healthcare, Erlangen, Germany) and from manual segmentation. Intraclass correlation coefficient and Bland-Altman statistics of mean difference and 95% limits of agreement were calculated for comparisons of two measurement methods.

2.3.2. Perfusion analysis

The perfusion analysis interface provides a viewer for checking dynamic perfusion images via the slider (Fig. 5). The user adjusts the slider and identifies a frame of interest showing good contrast between the myocardium and blood. The user mouse-clicks the LV center point and the RV insertion point and clicks the "Auto Myo Segment" button to automatically segment the myocardium using the trained U-net model. Incorrect endocardial and epicardial contours can be corrected manually. Based on the locations of LV center point and RV insertion point, six myocardial segments are automatically determined. The start and end frame numbers for the LV blood pool signal and the myocardial signal are entered to calculate the slope of LV blood signal and the slope of myocardial signal. The myocardial upslope is calculated as the ratio of the LV blood slope to the myocardial slope in each myocardial segment as described in Ref. [16]. This procedure is repeated until all short axis slices are processed.

2.3.3. LGE analysis

The LGE analysis interface provides a viewer for checking both the cine and the LGE images via the slider (Fig. 6). Our approach to LGE myocardial segmentation is to apply the trained U-net segmentation model to each short-axis LGE image slice. The LGE myocardial contours, if not perfectly aligned, are corrected by the user's manual correction. The cine image serves as a reference, since it shows a clear depiction of the epicardial and endocardial borders, whereas scar tissue in the LGE image typically has brightness similar to nearby tissues, making the identification of the myocardial borders difficult. After the myocardial segmentation, scar segmentation is performed using the n-SD method in which a pixel is classified as scar when its intensity is higher than $m + n\sigma$ [17,18].

$$I(x, y) = \begin{cases} \text{scar}, & \text{if } I(x, y) > m + n\sigma \\ \text{no scar}, & \text{otherwise} \end{cases} \tag{6}$$

Here, $I(x, y)$ is image intensity at a voxel location (x,y) and the m and σ are the mean and standard deviation of a remote healthy myocardial region of interest (ROI). The default value of n is set to 5. The user is required to choose a slice and segment a remote healthy myocardial ROI. The LGE infarct volume (or scar volume) is calculated and is expressed in milliliters (mL).

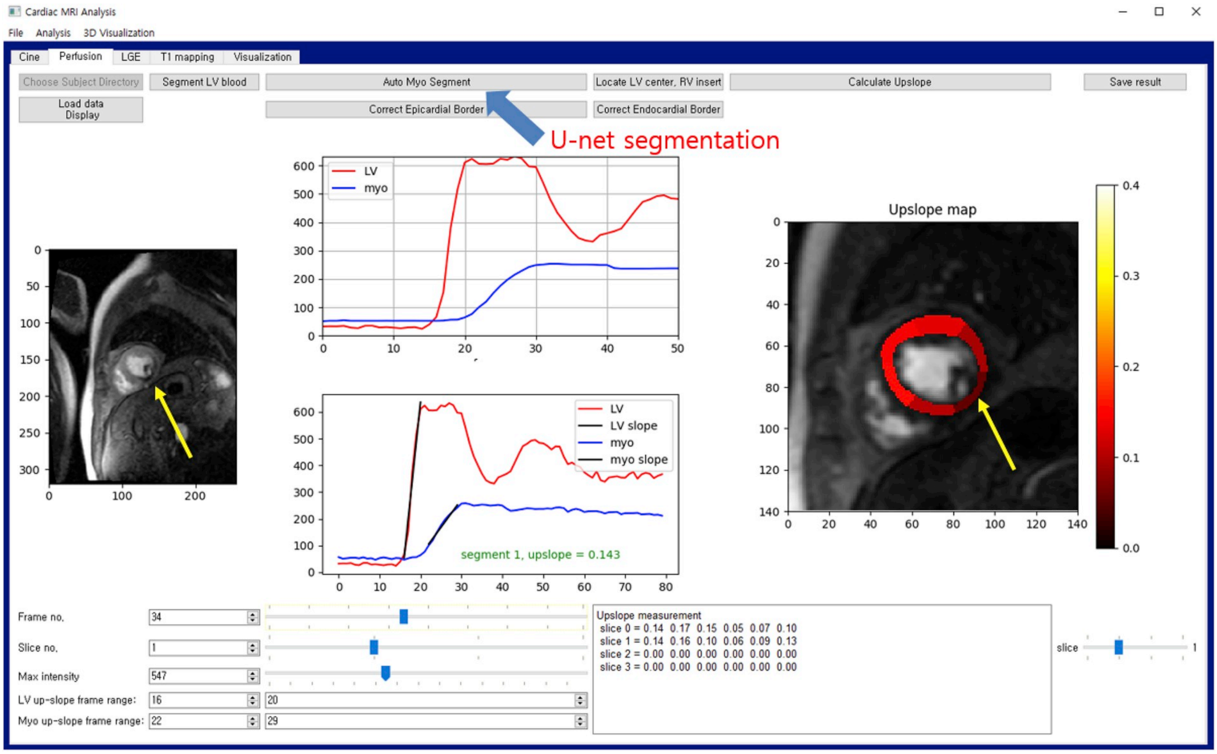
$$\text{Scar volume} = \sum_{z=1}^{N} N_{\text{scar}}(z)\cdot V_{\text{vox}} \tag{7}$$

where z is slice index, N is the number of slices, $N_{\text{scar}}(z)$ is the number of scar voxels at slice z, and V_{vox} is voxel volume.

2.3.4. T1 mapping analysis

The T1 mapping interface provides a viewer for checking the pre and post TI images and for computing the T1 and ECV maps in all three slices (Fig. 7). DICOM datasets with images of different TIs were already motion-corrected in pre- and post-contrast, separately. The pixel-wise T1 map calculation involved curve fitting for T1 parameter estimation at every pixel.

We implemented T1 map calculation in C++ and used pybind11 [19] to make the compiled C++ code compatible with the Python environment. C++ implementation was performed on a Windows OS, Microsoft Visual Studio 2017 platform. For the T1 parameter estimation, we used the solve_least_squares_lm function of the Dlib library [20]. Computational time was measured and compared between



segment 1, upslope = 0.143

Frame no. 34

Slice no. 1

Max intensity 547

LV up-slope frame range: 16 20

Myo up-slope frame range: 22 29

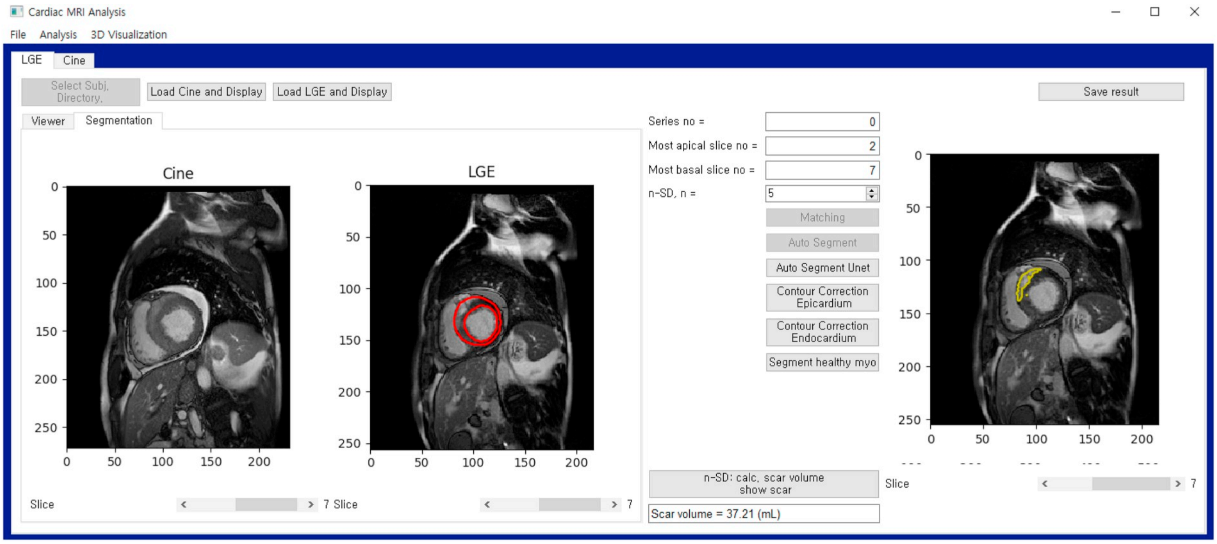
```
Upslope measurement
 slice 0 = 0.14  0.17  0.15  0.05  0.07  0.10
 slice 1 = 0.14  0.16  0.10  0.06  0.09  0.13
 slice 2 = 0.00  0.00  0.00  0.00  0.00  0.00
 slice 3 = 0.00  0.00  0.00  0.00  0.00  0.00
```

Fig. 5. Layout of the cardiac perfusion analysis interface. The frame number slider allows the user to select the frame which shows high image contrast between the blood and myocardium. The chosen frame is used to automatically segment the myocardium using the trained U-net model. When the user specifies the LV slope and the myocardium slope frame range values and presses the "Calculate Upslope" button, the upslope measurement is automatically performed. A myocardial upslope color-coded map is overlaid to the image. This process is repeated until all slices are processed. The yellow arrows indicate a perfusion defect located in the right coronary artery (RCA) territory.

EVCMR and MRmap.

Patient's motion between pre- and post-contrast T1 mapping can result in a mis-registration error when calculating an ECV map. The interface provides a ComboBox widget for selecting 'no registration' or 'registration'. In the case where the user selects 'registration', a deformation field is estimated after a non-rigid registration between the last TI pre-contrast image and the last TI post-contrast image as described by Kellman et al. [21]. The non-rigid registration provided by the Simple ITK library [22] involves B-spline transformation and L-BFGS-B optimization. The deformation field is applied to the TI post-contrast images.

Once the pre- and post-T1 maps are calculated and co-registered, the means of the segmented LV blood pools are computed for the pre- and the post-contrast T1 map. Finally, the pixel-wise ECV map is



Series no = 0

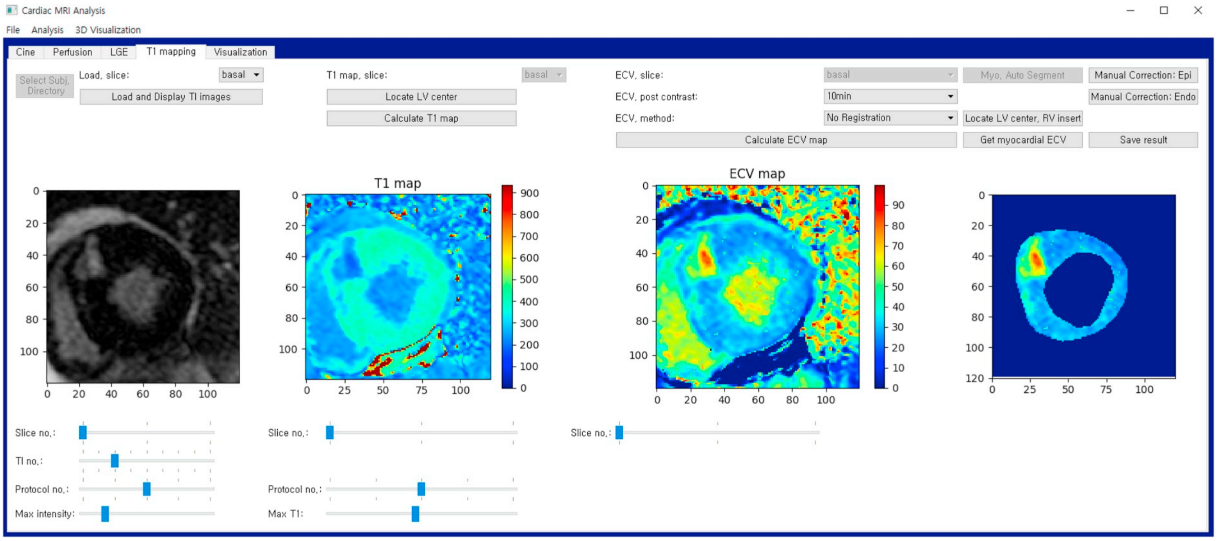
Most apical slice no = 2

Most basal slice no = 7

n-SD, n = 5

Scar volume = 37.21 (mL)

Fig. 6. LGE image analysis layout. The cine and LGE datasets are loaded and displayed. The user scrolls the slice sliders and determines the slice range for the LV scar quantification. The cine image can also serve as a reference for an LGE image when it is difficult to identify the myocardium due to similar brightness between the scar myocardium and the surrounding tissues. The user identifies a slice and segments a healthy myocardial ROI for the scar segmentation. When the user presses the "n-SD: calc. Scar volume, show scar" button, the total scar volume is computed and displayed. The segmented scar is overlaid to the image (yellow contour).



Load, slice: basal

T1 map, slice: basal

ECV, slice: basal

ECV, post contrast: 10min

ECV, method: No Registration

Fig. 7. T1 mapping image analysis layout. The TI image datasets are loaded for the pre- and post-contrast sequence data. Pre- and post-T1 maps are calculated from the pre- and post-contrast TI images, respectively. Once the post-T1 map is registered to the pre-T1 map, an ECV map is calculated. The user can segment the myocardial contour to obtain pixel-wise myocardial ECV maps for the basal, mid, and apical slices.

calculated [21]. The ECV map is expressed as percent (%).

$$\mathrm{ECV}(x,\,y) = (100\text{–}\mathrm{hct})\cdot\frac{\mathrm{R1}_{\mathrm{post}}(x,\,y) - \mathrm{R1}_{\mathrm{pre}}(x,\,y)}{\mathrm{R1}_{\mathrm{post,blood}} - \mathrm{R1}_{\mathrm{pre,blood}}} \tag{8}$$

Here, hct is the hematocrit, the proportion of red blood cells in blood, and is expressed as a percentage. $\mathrm{R1}_{\mathrm{pre,\ blood}}$ and $\mathrm{R1}_{\mathrm{post,\ blood}}$ are the mean R1 (= the reciprocal of T1) values of the segmented LV blood pool from the pre- and post-contrast T1 maps, respectively. $\mathrm{R1}_{\mathrm{pre}}(x, y)$ and $\mathrm{R1}_{\mathrm{post}}(x, y)$ are the R1 value at the location (x,y) from the pre- and post-contrast T1 maps, respectively. This procedure is repeated until all the three slices (i.e., apical, mid, and basal) are processed.

2.3.5. *Visualization*

PyQtGraph [23] was adopted for 3D visualization in the GUI. The 3D visualization is realizable after the myocardial segmentation results are saved in. pkl format. When the user mouse-clicks the “Generate Surface Rendering” button, the following procedures are performed to produce the 3D rendering of the myocardium. From the endocardial and epicardial masks segmented from all slices in the cine dataset, the myocardial binary masks are stacked along the slice axis. Slice resolution is coarser than in-plane resolution, and shape-based slice interpolation is performed to interpolate the myocardial binary masks in the slice direction based on the signed distance function [24]. The slice interpolation increases the number of slices and results in an identical voxel size in 3D. A 3D surface mesh is created after the application of the Marching cubes algorithm [25].

When the user mouse-clicks the “Visualize Scar Transmurality” button, the following procedures are performed. The segmentation results of the myocardium and scar from the LGE images are used for scar transmurality calculation and visualization. Scar transmurality is obtained by computing the ratio of the scar thickness to the myocardial thickness along each radial line [26]. The scar transmurality value in each radial line is projected onto the neighborhood of a position on a 3D epicardial surface. The ‘GLMeshItem’ function provided by PyQtGraph is used for the surface visualization. The colormap ‘jet’ and its associated colorbar are used for display.

When the user mouse-clicks the “Show Multi-Planes – Perfusion, Upslope” button (or the “Show Multi-Planes – ECV” button), the following procedures are performed. The multi-plane visualization is performed with the perfusion upslope map (or ECV map). After image translation, the LV center points are aligned along the slice axis. After image rotation, the RV anterior insertion points are aligned in the same (x, y) position. The alignment helps better visualize the orientation of the myocardium in all slices. The colormap and its associated colorbar are implemented for display. The colormap “hot” is used for perfusion upslope map display, whereas the colormap “jet” is used for the ECV map display.

3. Results

Fig. 8 shows the learning curves for the training and validation for four different learning rate choices. For a given learning rate, the time to finish up to 100 epochs in the U-net model training took approximately 20 h. The learning rate of 10^{-6} showed a slowly decreasing pattern of training learning curve and a Dice loss of validation learning curve higher than those from the other learning rates. Based on the validation loss function results, we chose an epoch = 40 and a learning rate = 0.0001, which showed a stable learning pattern with a relatively low Dice loss value.

Mean Dice similarity scores of the U-net myocardial segmentation

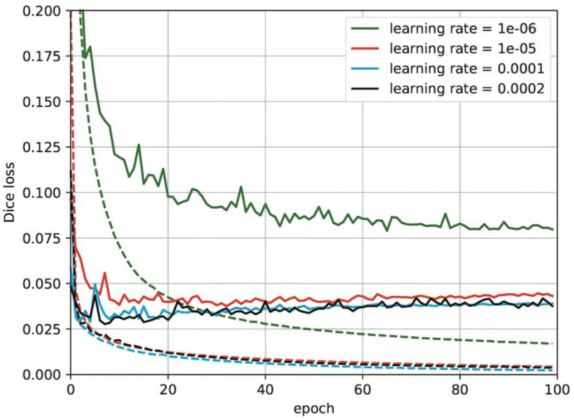


Fig. 8. Learning curves of training (dashed line) and validation (sold line) for four different learning rates. Among the learning rates, the learning rate of 0.0001 results in low Dice loss in training and validation, as indicated by the light blue and black curves. It shows less fluctuation in the validation learning curve than the learning rate of 0.0002. As the number of epochs increases, the Dice loss of validation shows a slightly increasing trend, which is a sign of overfitting. In our study, we chose an epoch number of 40 and a learning rate of 0.0001.

Table 3
Mean (± SD) of Dice similarity coefficients in myocardial segmentation of the test subjects' data.

Group	Cine		Perfusion	LGE
	Diastole	Systole		
Healthy (n = 8)	0.86 (±0.03)	0.89 (±0.05)	0.83 (±0.05)	0.86 (±0.04)
CAD (n = 8)	0.87 (±0.01)	0.89 (±0.02)	0.80 (±0.08)	0.80 (±0.08)
AS (n = 8)	0.87 (±0.01)	0.90 (±0.02)	0.81 (±0.05)	0.85 (±0.06)
HCM (n = 8)	0.89 (±0.02)	0.92 (±0.01)	0.82 (±0.14)	0.87 (±0.04)
Overall (n = 32)	0.87 (±0.02)	0.90 (±0.03)	0.82 (±0.08)	0.84 (±0.06)

Abbreviations: CAD = coronary artery disease, AS = aortic stenosis, HCM = hypertrophic cardiomyopathy, LGE = late gadolinium enhancement.

for four different disease groups in test subjects are shown in Table 3. In cine image evaluation, the average Dice scores of the short-axis myocardial segmentation in diastole and systole were 0.87 ± 0.02 and 0.90 ± 0.03, respectively. In perfusion image evaluation, the average Dice scores were 0.82 ± 0.08 in all test subjects. The CAD group showed the lowest mean Dice score of 0.80 among the four groups, while the healthy volunteer group showed the highest mean Dice score of 0.83. In LGE image evaluation, the average Dice scores were 0.84 ± 0.06 in all test subjects. The CAD group showed the lowest mean Dice score of 0.80, while the HCM group showed the highest mean Dice score of 0.87. Fig. 9 displays the cine myocardial segmentation results from a patient with aortic stenosis. The U-net myocardial segmentation had a large degree of overlap with manual myocardial segmentation in all short-axis slices. Fig. 10 shows representative automatic segmentation results in cardiac perfusion image data. The automatic segmentation results were obtained after the user chose an image frame with good contrast between the myocardium and the LV blood pool from the user interface.

Intraclass correlation and Bland-Altman statistics between two measurement methods in the test data are shown in Table 4. The comparisons were made on the imaging biomarkers of LV EDV, LV ESV, LV EF, and LV mass, which were calculated from cardiac cine image analysis. The LV EDV, LV ESV, LV EF, and LV mass measured by the proposed method were highly correlated with manual segmentation, having intraclass correlation coefficients of 0.98, 0.93, 0.86, and 0.97, respectively. For LV EDV, the proposed U-net method showed the smallest mean difference of −4.9 mL when compared with manual segmentation. For LV ESV, the proposed method showed the smallest mean difference of −2.9 mL when compared with commercial software (Argus, Siemens Healthcare, Erlangen Germany). For LV EF, the proposed method showed the smallest mean difference of 3.1% when compared with commercial software. For LV mass, the proposed method showed the smallest mean difference of 4.4 g when compared with manual segmentation. Overall, the proposed method and manual segmentation showed the narrowest 95% limits of agreement in LV EDV, LV ESV, LV EF, and LV mass.

In T1 mapping image analysis, myocardial segmentation was performed to obtain segmental myocardial ECV values, but the U-net model was not considered because the T1 mapping images did not look similar to the cine images with regard to image intensity in the

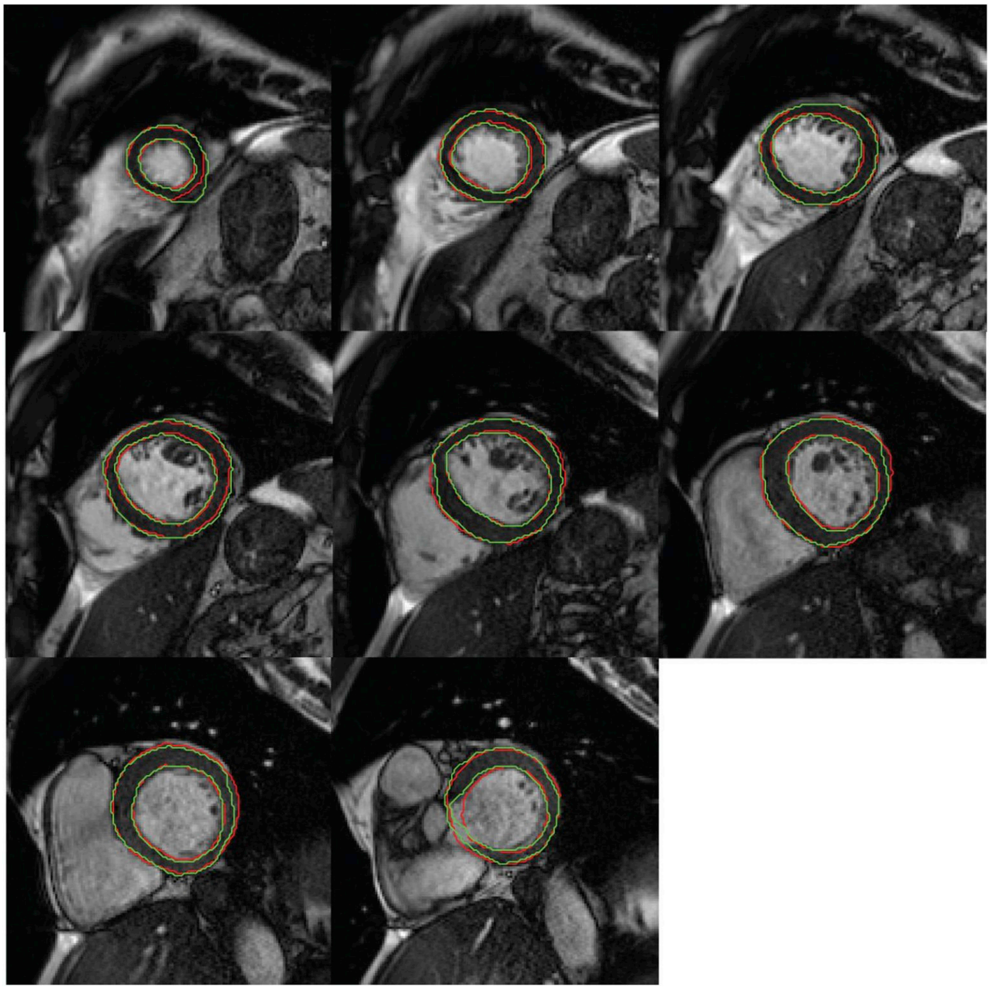


Fig. 9. An example of segmentation contours in a patient with aortic stenosis for diastole. Manually annotated myocardial contours are indicated by red, and the automatically segmented myocardial contours using U-net are indicated by green.

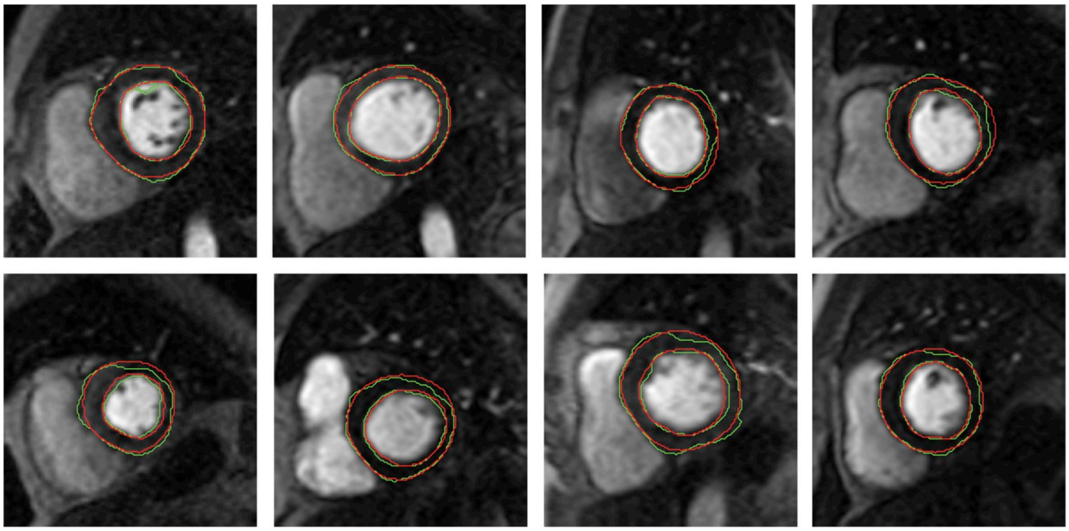


Fig. 10. Representative automatic segmentation results in cardiac perfusion data when the U-net model, trained using cardiac cine data, was used. Manually annotated myocardial contours are indicated by red, and the automatically segmented myocardial contours using U-net are indicated by green. Although image contrast between the myocardium and the surrounding tissue can vary across subjects, the U-net segmentation produced myocardial masks, most of which were comparable to manual segmentation and resulted in a mean Dice score of 0.82 (± 0.08) in 32 subjects.

Table 4
Intraclass correlation and Bland-Altman statistics between two measurement methods in the test data.

		LV EDV	LV ESV	LV EF	LV mass
Commercial - Manual	ICC	0.89	0.93	0.85	0.75
	95% CI	0.78 – 0.94	0.86 – 0.96	0.72 – 0.93	0.55 – 0.87
	Mean diff.	−14.5 mL	−6.2 mL	1.4%	21.9 g
	95% limits of agreement	−32.0 – 3.0 mL	−21.0 – 8.5 mL	−7.9 – 10.6%	−1.7 – 45.5 g
Proposed - Manual	ICC	0.98	0.93	0.86	0.97
	95% CI	0.95 – 0.99	0.86 – 0.97	0.74 – 0.93	0.94 – 0.99
	Mean diff.	−4.9 mL	−8.3 mL	4.5%	4.4 g
	95% limits of agreement	−17.0 – 7.2 mL	−17.9 – 1.3 mL	−0.1 – 9.0%	−7.9 – 16.6 g
Proposed - Commercial	ICC	0.95	0.96	0.82	0.83
	95% CI	0.89 – 0.97	0.91 – 0.98	0.66 – 0.91	0.69 – 0.91
	Mean diff.	8.2 mL	−2.9 mL	3.1%	−18.2 g
	95% limits of agreement	−7.0 – 23.3 mL	−15.7 – 9.9 mL	−5.5 – 11.6%	−37.0 – 0.7 g

Abbreviations: ICC = intraclass correlation coefficient, CI = confidence interval, LV = left ventricle, EDV = end-diastolic volume, ESV = end-systolic volume, EF = ejection fraction, Commercial = commercially available software, Proposed = EVCMR with U-net segmentation, Manual = manual segmentation.

myocardium, blood, and surrounding tissues. When the pixel-wise T1 mapping was implemented in C++ via pybind11, it took approximately 3 s to obtain a T1 map. The computational time was significantly shorter than MRmap, which took approximately 70 s and 150 s in pre- and post-contrast T1 mapping, respectively.

The GUI provided a 3D visualization widget for visualizing the myocardial wall and scar transmurality from the segmented myocardium in cine and LGE data. The perfusion and T1 mapping datasets consisted of only four and three short axis slices, respectively. For visualization of the perfusion upslope maps and ECV maps, a multiple plane visualization scheme was adopted (Fig. 11). Fig. 11 compares the visualization results between an HCM patient (top row) and a CAD patient with right coronary artery (RCA) stenosis (bottom row). From the leftmost column, it is observed that the septal HCM case clearly shows thicker myocardium in the septum than the patient with RCA stenosis. From the 2nd leftmost column, it is observed that high values of scar transmurality are observed in the anteroseptal region in the HCM patient, while high values of scar transmurality are observed in the inferior region in the CAD patient with RCA stenosis.

4. Discussion

A custom cardiac MR image analysis tool was designed and implemented in Python. Python open-source libraries were helpful in developing the automatic myocardial segmentation, curve fitting, registration, and 3D visualization methods. The user interface provides multiple tabs to support analysis of the images obtained from a variety of cardiac imaging sequences. In this study, we considered image data from four cardiac imaging sequences: cine, perfusion, LGE, and T1 mapping. The development of T2/T2* mapping analysis modules was not considered in this study since image data were not available. Given the fact that a widget is easily integrated as a tab in the interface, the addition of widgets for the analysis of other cardiac MR sequence datasets, such as T2 weighted images for the segmentation of area at risk [27], myocardial tagging of images for strain analysis [28], and 4D flow images for blood flow measurement [29], can be straightforward in this framework.

Other publicly available software tools deserve to be mentioned. Segment is an open-source platform which supports the image analysis of MRI, CT, and SPECT and is widely used in cardiovascular image analysis [4]. Segment is a Matlab-based software package where the source code works reliably on the Matlab R2014a version (https://github.com/Cardiac-MR-Group-Lund/segment-open). Although it is open-source, some parts of the code are not accessible and are protected with Matlab p-files. In addition, Matlab is a commercial product which requires the purchase of relevant Matlab toolboxes, in cases where specific functions provided by toolboxes are need. MRmap is a relatively light-weight package which supports T1, T2, and T2* mapping

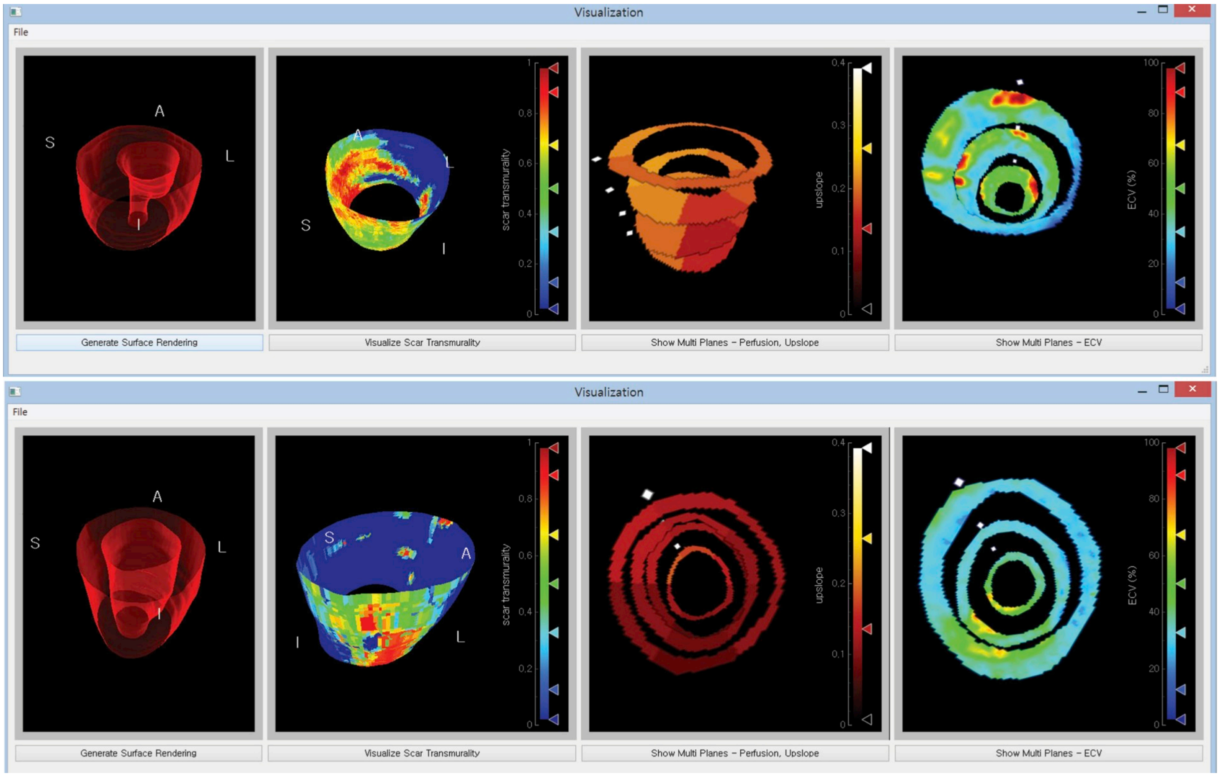


Fig. 11. Visualization layout. Top: Results from a patient with septal hypertrophic cardiomyopathy. Bottom: Results from a patient with coronary artery disease in the right coronary artery. From left to right: 1) 3D surface rendering of the LV epicardial and endocardial walls, 2) the 3D scar transmurality map, 3) the multi-plane display of the segmental upslope measurements in myocardial perfusion, and 4) the multi-plane display of the ECV measurements in T1 mapping data. The letters A, S, I, and L indicate the anterior, septal, inferior, and lateral position of the heart, respectively. The white square for the upslope and the ECV maps indicates the anterior right ventricle (RV) insertion points.

(https://sourceforge.net/projects/mrmap/) [5]. It was written in Interactive Data Language (IDL). The drawback is that it is significantly slower in T1 mapping, compared to our C++ based T1 map calculation.

In EVCMR, the user can inspect acquired images and qualitatively assess the image quality. This is important since quality control of the acquired image data is necessary before the data are considered for diagnosis. It facilitates data rejection when the image quality is not acceptable for further quantitative analysis. For example, severe motion artifacts can be noted in one slice of the cine images in a patient. Motion artifacts can be attributed to a patient's inability to hold their breath or to arrhythmia. Also, the normal myocardial regions may not be effectively nulled in the LGE images of a patient's data. The ineffective nulling may be attributed to the diffused pattern of myocardial fibrosis in the patient's LGE data. This type of data can be considered as not acceptable for further quantitative analysis.

The U-net model trained using cardiac cine datasets worked reasonably well on the cardiac perfusion datasets, but the selection of an appropriate time frame in the dynamic perfusion images influenced the segmentation results. The user interface allowed the user to test the U-net segmentation with different choices of frame numbers and to select an appropriate myocardial segmentation result. The mean Dice score of 0.84 of LGE data was lower than that of 0.87 of cine diastolic data. The degradation of the performance in LGE data was most pronounced in the CAD group (0.80 in LGE vs. 0.87 in cine). This is attributed to the fact that some CAD patients had a large scar burden in the myocardium. It was observed that in LGE image segmentation the U-net segmentation model produced segmentation errors, especially when scar tissue of bright signal intensity was dominant within the myocardium. Manual correction was necessary to include the scar tissue within the myocardium. Importantly, the segmented contours of the U-net segmentation followed by manual contour correction provided by the interface can be re-used for supervised segmentation model training.

The main disadvantage of Python is known to be its slow computation speed, especially when the code involves the Python optimization functions (e.g., the curve_fit function available from the scipy.optimize library) within the *for* loop. The pixel-wise estimation of the T1 values corresponds to this time-consuming scenario. EVCMR demonstrated the effectiveness of pybind11 in accelerating the pixel-wise T1 mapping using the pre-compiled version of C++ code. This approach is also applicable to other time-consuming tasks, such as the voxel-wise deconvolution in tissue perfusion quantification [30].

The current GUI involves manual procedures such as 1) landmark localization of the LV center point and the anterior RV insertion point for the segmental analysis of the myocardium and 2) slice range specification for the cine and LGE image segmentation. Automation of the procedures remains as future work. The manual annotation can be replaced with automatic landmark localization. Machine learning methods may be suitable for automatic landmark detection in cardiac MR [31,32]. Short-axis slice range selection for cine image analysis can be performed automatically with a method similar to the study in Ref. [33]. In case of failure, it may be necessary to have the user check the results of the automatic algorithms and correct them in cases of inaccurate results.

Our study has limitations. First, the test data used in the software tool included only DICOM files from a single vendor (i.e., Siemens). Although it is a straightforward task to handle DICOM header files in other vendor data, the use of multi-vendor data would strengthen the versatility of the software tool. Second, the software tool only supports short-axis slice images for LV myocardial analysis. Long-axis cardiac images provide additional information and may be helpful in determining the range of the left ventricle. Third, motion compensation along the slice direction in the short-axis cardiac cine was not investigated in this study. This would help improve the 3D reconstruction of the left ventricle [34].

5. Conclusion

We designed a CMR image post-processing, quantification, and visualization tool in Python and demonstrated the quantification and visualization results in patients suspected with heart disease. We implemented a U-net segmentation model and applied it to our custom Python GUI. The U-net segmentation was effective in automatic segmentation of the myocardium for cardiac cine, perfusion, and LGE analyses in the user interface. Also, the interface tool with its deep learning-based segmentation and manual contour correction was beneficial in efficiently obtaining labeled data for the training of segmentation models. Modules were developed to analyze images in a user-interactive fashion from the cine, perfusion, LGE, and T1 mapping image data. The interface tool has the potential to provide comprehensive 3D visualization of the diseased myocardium from multi-parametric MRI data in patients with cardiovascular disease. Future directions include 1) further improvement of automatic myocardial segmentation in cardiac perfusion and LGE datasets and 2) development of modules for the analyses of T2-weighted images and 4D flow datasets.

Acknowledgements

This work was supported by the Basic Science Research Program through the National Research Foundation of Korea funded by the Ministry of Science, ICT, and Future Planning (Grant numbers: NRF-2015 R1C1A1A02036340, NRF-2018 R1D1A1B07042692).

References

[1] J. Schulz-Menger, D.A. Bluemke, J. Bremerich, S.D. Flamm, M.A. Fogel, M.G. Friedrich, R.J. Kim, F. von Knobelsdorff-Brenkenhoff, C.M. Kramer, D.J. Pennell, S. Plein, E. Nagel, Standardized image interpretation and post processing in cardiovascular magnetic resonance: society for Cardiovascular Magnetic Resonance (SCMR) board of trustees task force on standardized post processing, J. Cardiovasc. Magn. Reson. : official journal of the Society for Cardiovascular Magnetic Resonance 15 (2013) 35.

[2] C.M. Kramer, J. Barkhausen, S.D. Flamm, R.J. Kim, E. Nagel, P. Society for cardiovascular magnetic resonance board of trustees task force on standardized, standardized cardiovascular magnetic resonance (CMR) protocols 2013 update, J. Cardiovasc. Magn. Reson. : official journal of the Society for Cardiovascular Magnetic Resonance 15 (2013) 91.

[3] A.K. Attili, A. Schuster, E. Nagel, J.H. Reiber, R.J. van der Geest, Quantification in cardiac MRI: advances in image acquisition and processing, Int. J. Cardiovasc. Imaging 26 (Suppl 1) (2010) 27–40.

[4] E. Heiberg, J. Sjogren, M. Ugander, M. Carlsson, H. Engblom, H. Arheden, Design and validation of Segment–freely available software for cardiovascular image analysis, BMC Med. Imaging 10 (2010) 1.

[5] D.R. Messroghli, A. Rudolph, H. Abdel-Aty, R. Wassmuth, T. Kuhne, R. Dietz, J. Schulz-Menger, An open-source software tool for the generation of relaxation time maps in magnetic resonance imaging, BMC Med. Imaging 10 (2010) 16.

[6] M.D. Blackledge, D.J. Collins, D.M. Koh, M.O. Leach, Rapid development of image analysis research tools: bridging the gap between researcher and clinician with pyOsiriX, Comput. Biol. Med. 69 (2016) 203–212.

[7] H. Xue, S. Zuehlsdorff, P. Kellman, A. Arai, S. Nielles-Vallespin, C. Chefdhotel, C.H. Lorenz, J. Guehring, Unsupervised inline analysis of cardiac perfusion MRI, Medical image computing and computer-assisted intervention : MICCAI, International Conference on Medical Image Computing and Computer-Assisted Intervention, vol. 12, 2009, pp. 741–749.

[8] D.R. Messroghli, A. Radjenovic, S. Kozerke, D.M. Higgins, M.U. Sivananthan, J.P. Ridgway, Modified Look-Locker inversion recovery (MOLLI) for high-resolution T1 mapping of the heart, Magn. Reson. Med. 52 (2004) 141–146.

[9] P. Kellman, M.S. Hansen, T1-mapping in the heart: accuracy and precision, J. Cardiovasc. Magn. Reson. : official journal of the Society for Cardiovascular Magnetic Resonance 16 (2014) 2.

[10] H. Xue, S. Shah, A. Greiser, C. Guetter, A. Littmann, M.P. Jolly, A.E. Arai, S. Zuehlsdorff, J. Guehring, P. Kellman, Motion correction for myocardial T1 mapping using image registration with synthetic image estimation, Magn. Reson. Med. 67 (2012) 1644–1655.

[11] O. Ronneberger, P. Fischer, T. Brox, U-net, Convolutional networks for biomedical image segmentation, International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer, 2015, pp. 234–241.

[12] A. Andreopoulos, J.K. Tsotsos, Efficient and generalizable statistical models of shape and appearance for analysis of cardiac MRI, Med. Image Anal. 12 (2008) 335–357.

[13] H. Dong, G. Yang, F. Liu, Y. Mo, Y. Guo, Automatic Brain Tumor Detection and Segmentation Using U-Net Based Fully Convolutional Networks Medical Image Understanding and Analysis, Springer, 2017, pp. 506–517 arxiv:1705.03820.

[14] M. Summerfield, Rapid GUI Programming with Python and Qt: The Definitive Guide to PyQt Programming, Pearson Education, 2007.

[15] J.D. Hunter, Matplotlib: a 2D graphics environment, Comput. Sci. Eng. 9 (2007) 90–95.

[16] Y.C. Kim, S.M. Kim, Y.H. Choe, Robust semi-automated quantification of cardiac MR perfusion using level set: application to hypertrophic cardiomyopathy patient data, Comput. Biol. Med. 71 (2016) 162–173.

[17] A.S. Flett, J. Hasleton, C. Cook, D. Hausenloy, G. Quarta, C. Ariti, V. Muthurangu, J.C. Moon, Evaluation of techniques for the quantification of myocardial scar of differing etiology using cardiac magnetic resonance, JACC. Cardiovascular imaging 4 (2011) 150–156.

[18] Y. Mikami, L. Kolman, S.X. Joncas, J. Stirrat, D. Scholl, M. Rajchl, C.P. Lydell, S.G. Weeks, A.G. Howarth, J.A. White, Accuracy and reproducibility of semi-automated late gadolinium enhancement quantification techniques in patients with hypertrophic cardiomyopathy, J. Cardiovasc. Magn. Reson. : official journal of the Society for Cardiovascular Magnetic Resonance 16 (2014) 85.

[19] W. Jakob, J. Rhinelander, D. Moldovan, pybind11–seamless Operability between C+ + 11 and python, (2016).

[20] D.E. King, Dlib-ml: a machine learning toolkit, J. Mach. Learn. Res. 10 (2009) 1755–1758.

[21] P. Kellman, J.R. Wilson, H. Xue, M. Ugander, A.E. Arai, Extracellular volume fraction mapping in the myocardium, part 1: evaluation of an automated method, J. Cardiovasc. Magn. Reson. : official journal of the Society for Cardiovascular Magnetic Resonance 14 (2012) 63.

[22] K. Marstal, F. Berendsen, M. Staring, S. Klein, SimpleElastix: a user-friendly, multilingual library for medical image registration, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (2016) 134–142.

[23] PyQtGraph, Scientific Graphics and GUI Labrary for Python, www.pyqtgraph.org.

[24] S.P. Raya, J.K. Udupa, Shape-based interpolation of multidimensional objects, IEEE Trans. Med. Imaging 9 (1990) 32–42.

[25] W.E. Lorensen, H.E. Cline, Marching cubes: a high resolution 3D surface construction algorithm, ACM siggraph computer graphics 21 (1987) 163–169.

[26] M.C. Carminati, C. Boniotti, L. Fusini, D. Andreini, G. Pontone, M. Pepi, E.G. Caiani, Comparison of image processing techniques for nonviable tissue quantification in late gadolinium enhancement cardiac magnetic resonance images, J. Thorac. Imaging 31 (2016) 168–176.

[27] A.E. Arai, S. Leung, P. Kellman, Controversies in cardiovascular MR imaging: reasons why imaging myocardial T2 has clinical and pathophysiologic value in acute myocardial infarction, Radiology 265 (2012) 23–32.

[28] N.F. Osman, W.S. Kerwin, E.R. McVeigh, J.L. Prince, Cardiac motion tracking using CINE harmonic phase (HARP) magnetic resonance imaging, Magn. Reson. Med. 42 (1999) 1048–1060.

[29] M. Bustamante, S. Petersson, J. Eriksson, U. Alehagen, P. Dyverfeldt, C.J. Carlhall, T. Ebbers, Atlas-based analysis of 4D flow CMR: automated vessel segmentation and flow quantification, J. Cardiovasc. Magn. Reson. : official journal of the Society for Cardiovascular Magnetic Resonance 17 (2015) 87.

[30] M. Jerosch-Herold, Quantification of myocardial perfusion by cardiovascular magnetic resonance, J. Cardiovasc. Magn. Reson. : official journal of the Society for Cardiovascular Magnetic Resonance 12 (2010) 57.

[31] O. Oktay, W. Bai, R. Guerrero, M. Rajchl, A. de Marvao, D.P. O'Regan, S.A. Cook, M.P. Heinrich, B. Glocker, D. Rueckert, Stratified decision forests for accurate anatomical landmark localization in cardiac images, IEEE Trans. Med. Imaging 36 (2017) 332–342.

[32] Y.-C. Kim, Y. Chung, Y.H. Choe, Automatic localization of anatomical landmarks in cardiac MR perfusion using random forests, Biomed. Signal Process. Control 38 (2017) 370–378.

[33] M. Paknezhad, S. Marchesseau, M.S. Brown, Automatic basal slice detection for cardiac analysis, J. Med. Imaging 3 (2016) 034004.

[34] M.C. Carminati, F. Maffessanti, E.G. Caiani, Nearly automated motion artifacts correction between multi breath-hold short-axis and long-axis cine CMR images, Comput. Biol. Med. 46 (2014) 42–50.

[35] H. Dong, A. Supratak, L. Mai, F. Liu, A. Oehmichen, S. Yu, Y. Guo, TensorLayer: a versatile library for efficient deep learning development, Proceedings of the 2017 ACM on Multimedia Conference, ACM, 2017, pp. 1201–1204.

[36] S. Van der Walt, J.L. Schönberger, J. Nunez-Iglesias, F. Boulogne, J.D. Warner, N. Yager, E. Gouillart, T. Yu, scikit-image: image processing in Python, PeerJ 2 (2014) e453.

[37] T.E. Oliphant, SciPy: open source scientific tools for Python, Comput. Sci. Eng. 9 (2007) 10–20.

[38] S.v.d. Walt, S.C. Colbert, G. Varoquaux, The NumPy array: a structure for efficient numerical computation, Comput. Sci. Eng. 13 (2011) 22–30.

[39] B.C. Lowekamp, D.T. Chen, L. Ibáñez, D. Blezek, The design of SimpleITK, Front. Neuroinf. 7 (2013) 45.

[40] L. Campagnola, PyQtGraph-scientific graphics and GUI library for python, Available from: http://www.pyqtgraph.org/.