Lazy Luna: extendible software for multilevel reader comparison in cardiovascular magnetic resonance imaging

Hadler T., Ammann C., Wetzl J., Viezzer D., Gröschel J., Fenski M., Abazi E., Lange S., Hennemuth A., Schulz-Menger J.

Computer Methods and Programs in Biomedicine
2023 AUG ; 238: 107615
2023 MAY 18 (first published online: final publication)
doi: 10.1016/j.cmpb.2023.107615

Publisher: Elsevier

Contents lists available at ScienceDirect

Computer Methods and Programs in Biomedicine

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

Lazy Luna: Extendible software for multilevel reader comparison in cardiovascular magnetic resonance imaging

Thomas Hadler[a,b,c,*], Clemens Ammann[a,b], Jens Wetzl[d], Darian Viezzer[a,b,c], Jan Gröschel[a,b,c,e], Maximilian Fenski[a,b], Endri Abazi[a,b], Steffen Lange[g], Anja Hennemuth[c,h,i], Jeanette Schulz-Menger[a,b,c,d,e,f]

[a] Working Group on CMR, Experimental and Clinical Research Center, a cooperation between the Max Delbrück Center for Molecular Medicine in the Helmholtz Association and Charité – Universitätsmedizin Berlin, Berlin, Germany
[b] Charité – Universitätsmedizin Berlin, corporate member of Freie Universität Berlin and Humboldt-Universität zu Berlin, Berlin, Germany
[c] DZHK (German Centre for Cardiovascular Research), partner site Berlin, Berlin, Germany
[d] Siemens Healthineers, Erlangen, Germany
[e] Max Delbrück Center for Molecular Medicine in the Helmholtz Association (MDC), Berlin, Germany
[f] Department of Cardiology and Nephrology, HELIOS Hospital Berlin-Buch, Berlin, Germany
[g] Department of Computer Sciences, Hochschule Darmstadt - University of Applied Sciences, Darmstadt, Germany
[h] Institute of Cardiovascular Computer-assisted Medicine, Charité – Universitätsmedizin Berlin, Berlin, Germany
[i] Fraunhofer MEVIS, Bremen, Germany

A R T I C L E I N F O

Article history:
Received 14 November 2022
Revised 27 April 2023
Accepted 16 May 2023

Keywords:
Segmentation
Cardiology
Cardiac magnetic resonance imaging
Magnetic resonance imaging
Software
Quality assurance

A B S T R A C T

Background and objectives: Cardiovascular Magnetic Resonance (CMR) imaging is a growing field with increasing diagnostic utility in clinical routine. Quantitative diagnostic parameters are typically calculated based on contours or points provided by readers, e.g. natural intelligences (NI) such as clinicians or researchers, and artificial intelligences (AI). As clinical applications multiply, evaluating the precision and reproducibility of quantitative parameters becomes increasingly important. Although segmentation challenges for AIs and guidelines for clinicians provide quality assessments and regulation, the methods ought to be combined and streamlined for clinical applications.

The goal of the developed software, Lazy Luna (LL), is to offer a flexible evaluation tool that is readily extendible to new sequences and scientific endeavours.

Methods: An interface was designed for LL, which allows for comparing annotated CMR images. Geometric objects ensure precise calculations of metric values and clinical results regardless of whether annotations originate from AIs or NIs. A graphical user interface (GUI) is provided to make the software available to non-programmers. The GUI allows for an interactive inspection of image datasets as well as implementing tracing procedures, which follow statistical reader differences in clinical results to their origins in individual image contours. The backend software builds on a set of meta-classes, which can be extended to new imaging sequences and clinical parameters. Following an agile development procedure with clinical feedback allows for a quick implementation of new classes, figures and tables for evaluation.

Results: Two application cases present LL's extendibility to clinical evaluation and AI development contexts. The first concerns T1 parametric mapping images segmented by two expert readers. Quantitative result differences are traced to reveal typical segmentation dissimilarities from which these differences originate. The meta-classes are extended to this new application scenario. The second applies to the open source Late Gadolinium Enhancement (LGE) quantification challenge for AI developers "Emidec", which illustrates LL's usability as open source software.

Conclusion: The presented software Lazy Luna allows for an automated multilevel comparison of readers as well as identifying qualitative reasons for statistical reader differences. The open source software LL can be extended to new application cases in the future.

* Corresponding author at: Grabenstr. 39, 12209, Berlin, Germany.
E-mail address: thomas.hadler@charite.de (T. Hadler).

https://doi.org/10.1016/j.cmpb.2023.107615
0169-2607/© 2023 Published by Elsevier B.V.

1. Introduction

Cardiovascular Magnetic Resonance imaging (CMR) is an evergrowing field of imaging and diagnostic techniques that are prominent in research and clinical practice [1]. These imaging techniques include cine imaging to capture motion, Late Gadolinium Enhancement (LGE) for focal scar imaging and parametric mapping techniques (i.e. T1 and T2 mapping) mainly for diffuse fibrosis and edema imaging, respectively. Focal scarring is locally concentrated scar tissue, diffuse fibrosis refers to distributed scar tissue, and edema reflects the water content in myocardial tissue.

CMR is affected by multiple confounders, which influence its processing pipeline, and require standardization and quality management. On the level of image reconstruction, standardization is approached with open-source reconstruction frameworks like Gadgetron [2]. The “Image Biomarker Standardization Initiative” aims for increased reproducibility of radiomics features in image processing [3]. Image segmentation challenges offer quality assessments of post-processing algorithms on openly available datasets [4,5]. However, these challenges often lack rigorousness due to their detachment from clinical reality. Furthermore, recommendations and consensus statements attempt to address confounders in the processing pipeline by summarizing evidence-based best practices and expert agreements [1,6–8]. Nonetheless, such statements stress the need for continuously updating recommendations for higher reproducibility of post-processing and clinical parameters in the future.

CMR diagnostic techniques rely on image processing. Although processing techniques differ in their specifics, they share key characteristics. They build on CMR images, onto which geometrical annotations are drawn. These annotations could be delineations of bloodpools, myocardium or scar tissue, or corresponding anatomical landmarks in an image sequence. Annotated images are used to calculate clinical results (e.g. amounts of fibrosis, cardiac output, etc.).

Many challenges remain for establishing and improving CMR processing methods. Manually annotating images is a laborious undertaking and training new readers is time-intensive; often including supervision by another experienced reader while analysing many training cases to reduce deviations from other experts [9,10]. Experienced readers annotate images according to the SCMR guidelines [1]. Nevertheless, significant inter- and intrareader disagreements remain due to post-processing software, site specific segmentation behaviour, difficult segmentation choices, etc. [11–14]. In order to assess precision and reproducibility of diagnostic techniques, we offer a software package capable of tracing reader differences of image annotations, over calculated clinical parameters to statistical differences (Fig. 1). The software package, called Lazy Luna (LL), allows for the comparison of exactly two readers.

In recent years, convolutional neural networks (CNN) have shown their ability to automatize image annotation tasks [5,10]. Several CNN architectures are optimized for image segmentation, such as the UNet [15] and architectural derivatives [16–18]. CNNs assign classes to image voxels. Thresholding generates segmentation masks as outputs. Although CNNs calculate clinically acceptable parameters, segmentation errors disregarding cardiac geometry remain frequent, diminishing their credibility [19,20]. The quantitative evaluation of CNN contours is habitually based on segmentation metrics like the Dice similarity coefficient (DSC) or the Hausdorff distance (HD). LL is capable of CNN evaluation and comparison as well.

The software was introduced in Hadler et al. [14]. LL was applied to a comparison of two readers on short-axis cine stacks to illustrate the workability of such comparison software. However, this did not delve into the software architecture and the interaction of its classes, or it’s capability to generalize over sequences. In order to make the software replicable and ensure it’s generalizability, this paper aims to formalize and expound on LL’s software architecture. To our knowledge, LL is the only available software package engineered towards dealing with CMR specific QA tasks, that offers a multilevel reader comparison with error-tracing as integrated software.

The aim of this paper is to design, extend and demonstrate Lazy Luna’s ability to generalize to quality assurance tasks in CMR. In order to address these tasks adequately, LL must fulfil the following criteria: LL is accessible to CNN developers and medical professionals. LL is intended as open source software and should be developed with accessible, well-known backend libraries. LL is adaptable to new imaging techniques and scientific endeavours.

2. Requirements

LL is intended as generic software capable of a multilevel reader comparison. Multilevel refers to comparing readers on the annotation level, which consists of comparing the annotations on individual images, while comparing derived clinical parameters on the patient level, and also offering a statistical level of reader comparison, i.e. to determine systematic biases (Fig. 1). LL ought to be capable of tracing differences from the image level to the reader level.

Accessibility, product independence and precision

LL needs an intuitive, generic interface for images and annotations pertaining to them so that readers can be compared, independent of vendors or reader output (polygons for human readers, image masks for CNNs). The software must offer precise calculations of quantitative results. LL is intended as open-source software and should build on available open-source components.

Adaptability and extendibility to new sequences and scientific endeavours

LL requires an understandable backend so that developers can extend the software to new sequences. LL’s core components should be broken down into classes, which allow for a generic and extendable backend.

Usability and target groups

LL should be usable by CNN developers and medical experts alike and address needs of both target groups. CNN developers and medical experts agree on the significance of calculating and displaying clinical results and their statistical evaluation. Clinicians emphasize the importance of visual inspections, whereas CNN developers focus on exact representations of contours and segmentation metrics to assess the effects of post-processing alterations. A graphical user interface (GUI) should be provided in order to allow for an automatic reader comparison with expressive visualizations and statistical analyses, independent of programming familiarity.

3. Computational methods and software architecture

3.1. Data model

In order to make LL’s backend simple to use and generic to different reader types and tasks we use folders as repositories for images and annotations pertaining to these images. The images are kept in the DICOM [21] (Digital Imaging and Communications in Medicine) format, the image annotations are stored as pickle files [22] containing Shapely [23] objects. The following subsections address these decisions separately.

DICOM is the standard for the communication and management of medical imaging information and related data [24]. The library Pydicom [25] offers a simple interface to these data. The

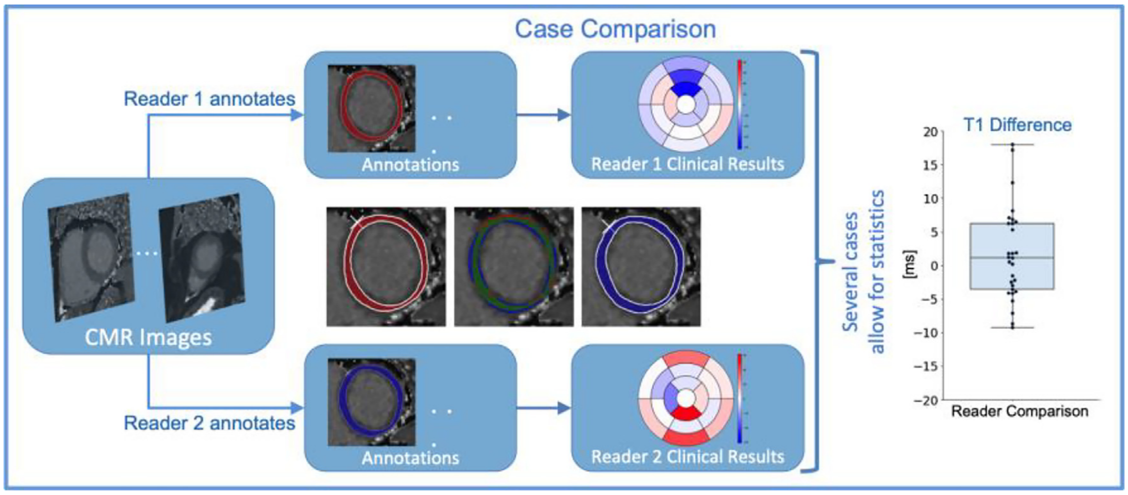


Fig. 1. A case comparison builds on two individual cases that share the same CMR images. Two readers annotate these images (reader 1 top, reader 2 bottom). The contours can be compared to each other visually (centre image) and quantitatively. Clinical results, such as the American Heart Association model segments, are calculated from a stack of contoured images. These can be compared to each other as segment value differences. When two readers analyse several cases, statistical procedures can be performed and visualized (right). Here, the quantitative parameter: T1 difference is shown. The levels of analysis are connected, allowing for the investigation of processing step influence and pipeline sensitivity.
Abbreviations: CMR: Cardiovascular magnetic resonance, AHA: American heart association.

DICOM standard allows for pooling image and image information in the same file such as the RescaleIntercept and RescaleSlope, which convert the stored pixel data to their intended output units by a linear function.

LL requires a DICOM tag for the image type (e.g. SAX T1). This is necessary for clinical practice, in which series are occasionally redone. Labelling allows for identification of the relevant series when several are available.

In order to sort images spatially or temporally, DICOM offers tags such as ImagePosition and InstanceNumber. For area and volume calculation DICOM tags provide PixelSpacing and SliceThickness.

Annotations are stored in a custom format. Every annotated image has an annotation file in pickle format. The file's basename is the referenced DICOM's unique SOPInstanceUID. The pickle file stores a python dictionary of key value pairs. Its keys are contour name strings (i.e. 'lv_myo' for LV myocardium). The keys map to geometrical Shapely objects and auxiliary information for individual contours.

3.2. Annotation processing and precision

Shapely is a Python package for manipulating and analysing geometric objects (i.e. polygons, points) [23,26]. Contours are modelled as Polygons (LV, RV endo- and epicardial contours) or MultiPolygons (papillary muscles); markers are modelled as Points (i.e. insertion points) or MultiPoints (i.e. extent points). Shapely offers precise geometrical operations including calculations, intersections, unions and Hausdorff distance (HD) calculations. Since human readers often segment on a subpixel level Shapely allows for exact geometrical calculations. In order to calculate precise values for segmentation masks (typical outputs for CNNs) the segmented pixels are outlined to produce a polygon. LL uses Rasterio's rasterize function to generate exact outlines from segmentation masks in Shapely format [27]. This permits precise calculations when comparing different reader types.

Metrics

Metrics allow for the comparison of annotations on the image level. For contours the Hausdorff Distance is calculated as the furthest distance of any point on either geometry from its nearest point on the other geometry. The Dice Similarity Coefficient is calculated as two times the intersection area divided by the sum of two Shapely geometries. The millilitres and their differences (ml Diff) are calculated using DICOM tag information on pixel height, width and slice thickness in mm and the contoured areas.

$$ml\ Diff(A,B) = (|A| - |B|) \times area\ per\ pixel\ \times\ slice\ thickness$$

$$Dice(A,B) = \frac{2 \times |A \cap B|}{|A| + |B|}$$

$$HD(A,B) = \max\{max_{a\in cA}(\min_{b\in cB} d(\mathrm{a},\mathrm{b})),\ max_{b\in dc}(\min_{a\in cA} d(\mathrm{a},\mathrm{b}))\}$$

For points, the millimetre distance (mm Diff) is calculated. For connecting lines of specific annotation points, angular differences (angle Diff) may also be calculated:

$$mm\ Diff(p_1,p_2) = |p_1 - p_2|$$

$$angleDiff(p_{1,1},p_{1,2},p_{2,1},p_{2,2}) = arccos(u,v),$$
$$u = \frac{p_{1,1} - p_{1,2}}{|p_{1,1} - p_{1,2}|},\quad v = \frac{p_{2,1} - p_{2,2}}{|p_{2,1} - p_{2,2}|}$$

3.3. Adaptability and extendibility to new image types and scientific endeavours

3.3.1. Software class structure

LL is intended to be accessible and extendable software. LL is written in Python [28] and follows an object oriented programming paradigm. An overview of LL's software class structure and the classes' interactions are illustrated in Fig. 2. The classes are elaborated on in Hadler et al. [14].

Extending Lazy Luna classes to new image techniques

Extending LL to new imaging techniques may require alterations or extensions to several of the above backend classes. The implementation of new Clinical Results or Metrics may require alterations to underlying code, such as the Annotation class or the Category classes. Extendible classes, their main functions and

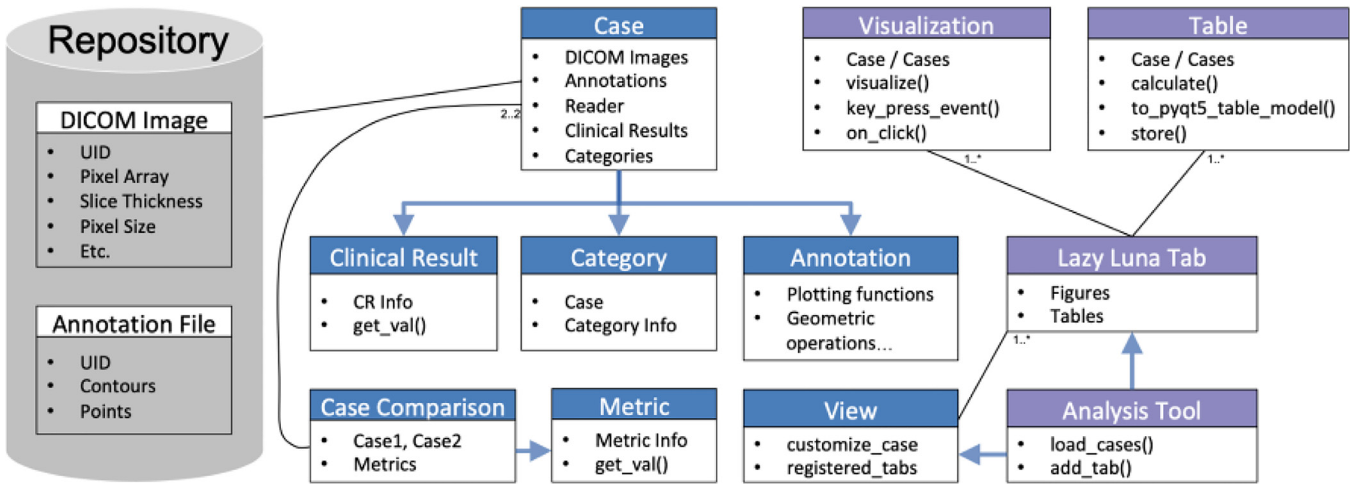


Fig. 2. Class Diagram of Lazy Luna.
Repository (grey).
The repository contains DICOM images and annotation files. The images are sorted into folders by case; the annotation files are sorted into folders by reader and case.
Backend (blue).
The class diagram depicts Lazy Luna's backend classes and how they interconnect. A Case is a container class for DICOM images and annotation files. It makes annotations accessible with an Annotation class, which offers visualization functions and geometric operations. A Case has Categories, which structure the images into slices and phases. Clinical Result classes can be attached to a Case in order to calculate CRs based on the images, annotations and categories. Case Comparisons contain two cases that reference the same images. Metric classes can be attached to a Case Comparison to calculate metric values for the case's annotations. View classes organize the other classes to address the needs of certain series. For example: a SAX Cine View can inform a case to focus on the subset of SAX Cine images and annotations, as well as attaching the respective CRs.
Frontend (purple).
Visualizations inherit their behavior from the Matplot.Figure class, allowing for user interactions. Tables inherit their behavior from the Pandas.DataFrame class. Both classes permit easy integration into PyQt5 tabs. The Analysis Tool is the GUI's central GUI. It allows for loading cases of two different readers, customizing the cases via the views and adding new tabs to the GUI.
Legend: CR: Clinical Result, GUI: graphical user interface, DICOM Image [24], Matplotlib.Figure [29], Pandas.Dataframe [30], PyQt5 [34].

extension possibilities to new endeavours are listed in Table 1. For more practical implementation details we refer to the Github repository and accompanying documentation.

Annotation: is a utility class that handles pickled dictionaries containing Shapely geometries. New geometrical calculations or visualizations would be implemented as Annotation class functions. Categories: sort images and annotations by slice and phase. Further categorizations of images, based on their temporal and spatial relationship to other images, would be implemented as Category class functions. Clinical Results: calculate clinical parameters. New imaging modalities may require new clinical parameters, implemented as individual classes. Metrics: are classes that calculate quantitative comparisons between individual annotations. New scientific endeavours may focus on new comparisons, which are implemented as individual classes. Views: organize cases to address user needs for an imaging modality. A View makes the relevant categories available (such as those that organized the fibrosis images). It connects relevant clinical results to a case. Tabs that were designed for the imaging modality are registered in Views and made available during runtime.

3.3.2. *Visualizations and tables: plug-in scheme*

Matplotlib Figures [29,31] and Pandas DataFrames [30,32] are used as base classes to integrate visualizations and spreadsheets into a GUI written in PyQt5 [33]. Visualizations inherit the functionality to interact with figures via events (e.g. mouse movements or key presses). Matplotlib's FigureCanvas allows for integrating LL's Visualizations into PyQt5 tabs [34]. Tables allow for exports to diverse spreadsheet formats. LL offers an interface class for PyQt5 integration via QTableViews. LL's comparison tool is the Analysis Tool; it builds on PyQt5's QTabWidget, which allows attaching QWidgets. QWidgets can be added to the Analysis Tool and embed visualizations and spreadsheets.

3.3.3. *Error tracing*

LL allows for connecting tables, visualizations and tabs. This enables tracing the consequences of annotation differences from the image level to the patient level. In turn, CR differences on the patient level can be traced to their contour difference origins. Alternatively, entire tabs can be opened when a specific case is to be inspected. For example, by plotting outliers in statistical plots of CRs an outlier case can be investigated for differences of individual contours.

3.3.4. *Extendibility and availability*

In results we demonstrate how LL was extended to fibrosis imaging as well as to focal scar imaging. For the fibrosis imaging extension, this includes devising class extensions, tables and visualizations, as well as producing GUI elements to mirror the requirements.

LL is available as open source code and published on Github. A first-use experience is described for the EMIDEC dataset, in which Late Gadolinium Enhancement was used for focal scar imaging. A use-session video is uploaded as supplementary material.

3.4. *Usability*

Graphical user interface

LL offers a GUI capable of creating and loading cases of two readers in order to compare them to each other.

Automated outputs

LL was developed for quick and extensive reader comparison. In order to accomplish this, each LL View offers a sequence specific storage function. This includes storing spreadsheets of metric values for annotated images, clinical results as well as outputting statistical plots.

Table 1
Class Description and Extension.
The table consists of four columns, the class name, intended class utility, main functions and a description of its extension with potential examples. The main classes are Annotation, Category, Clinical Result, Metric and View.

Class	Use Description	Functions	Extension Description
Annotation	Interface class to geometries: - Getters: for access to geometry objects - Visualizations: of geometries atop matplotlib axes - Helper functions: contain complex geometrical calculations for simple access	Getter functions: - get_contour(cont_name) - get_point(point_name) Visualization functions: - plot_contours(axis, cont_name, color) - plot_points(axis, point_name, color) - plot_face(axis, cont_name, color) - plot_cont_comparison(axis, other_anno, cont_name, colors) Helper functions: - get_contour_as_mask(cont_name)	How to: The class is extended by adding new functions Exemplary helper functions: - Point distances - Angle calculations - Bounding box determination
Category	Sorts images and annotations and offers a simple interface: - Sorting: sorts images and annotations spatially and temporally according to dicom attributes - Getters: simple interface for dicoms, images and annotations - Helper functions: contain calculations requiring sorted dicoms and annotations	Sorting function: - get_sop2depthandtime(sop_uid2filepath) Getter functions: - get_dcm(slice, phase) - get_img(slice, phase) - get_anno(slice, phase) Helper functions: - get_volume(cont_name, phase)	How to: The class is extended by adding new functions Exemplary helper functions: - Cardiac geometry descriptions (such as basal, midventricular, apical slices) - Determining phases in cardiac cycle (like end-systolic phase)
Clinical Result	A Clinical Result calculates a clinical parameters for a case Clinical parameters: calculation of values and differences between readers	Setter: - init(case): sets case, clinical parameter name, measurement unit Getters: - get_val(as_string=False) - get_val_diff(other_clinical_result, as_string=False)	How to: Lazy Luna is extended by clinical results for new imaging modalities by writing new classes Exemplary extension: - Clinical result for calculation of average myocardial voxel intensity
Metric	A Metric quantifies the difference between two annotation geometries with the support of corresponing dicoms Metric values: calculation of metric values	Setter: - init(): sets metric name, measurement unit Getter: - get_val(geo1, geo2, dcm=None, as_string=False)	How to: Lazy Luna is extended by metrics for new imaging modalities by writing new classes Exemplary extension: - Difference in number of pixels within contour type for two readers
View	A view structures cases by appending relevant categories, clinical results and tabs	Setter: - init(): sets the view name, tabs for inidividual case_comparisons and lists of case comparisons Adjust case: - initialize_case(case): calculates information requiring only one calculation - customize_case(case): connects the view's categories and clinical results Store information function: - store(case_comparisons)	How to: Extending Lazy Luna to a new imaging modality requires the implementation of a custom View class Exemplary extension: - A View for focal scar imaging

Development strategy and code maintenance

Agile development

LL's current outputs and tabs were developed in an agile development environment with clinicians in iterative feedback loops for the individual sequences. Following the implementation of the underlying backend (load images, sort them by space and acquisition time, calculate volumes from annotations), new functionalities were discussed, implemented and finally integrated or discarded in iterative loops of clinical feedback (Fig. 3).

Code maintenance

LL's code is published on Github as open source software and maintained by the first author. Github's Issues functionality allows for tracking bugs, and other forms of feedback for incremental code improvements. LL builds on several open source packages that are community maintained. As the underlying code and API of these underlying software packages evolve, LL's backend code will require adjustments that can also be effectively pursued as issues.

4. Results

The Results section will be divided into subsections corresponding to the Requirements paragraphs.

4.1. Accessibility, product independence and precision

Precision

As described in "Computational Methods and Software Architecture" and presented in Hadler et al. [14] Lazy Luna's backend offers precise calculations for annotation comparisons and calculation of clinical parameters. Polygonal contour calculations allow for geometrical accuracy, while sorting slice positions and interpolating missing slices guarantee clinical results with numerical precision. LL's backend was used for extensive comparisons of CNN architectures to a medical expert on SAX Cine images with different acquisition techniques [35].

Hardware specifications

LL was tested for 64-Bit systems of macOS Mojave 10.14.6, Windows 10 Home and Ubuntu 20.04. LL has been tested for Python 3.6 – 3.8.

Accessibility, open source software and product independence

LL builds on several open-source libraries, as described in "Computational Methods and Software Architecture". LL is available as open-source software on Github: https://github.com/thadler/LazyLuna.

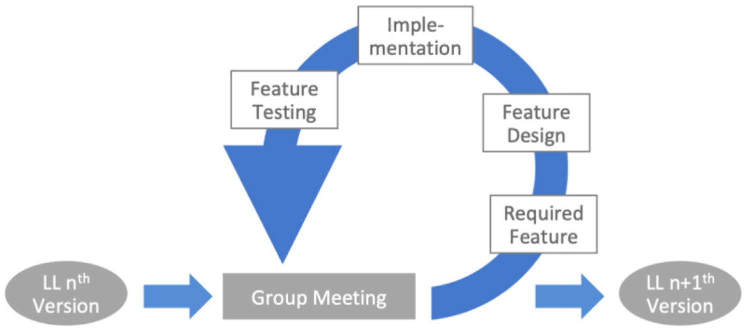


Fig. 3. Agile Software Development for Lazy Luna.
Starting at version n of LL, the incremental development consisted of team meetings with users who described concrete features they required, followed by abstractions of these to implementable features, their implementation and testing. These in turn were then kept or discarded according to the utility clinicians/AI developers saw in them in the next group meeting. After one or several such development loops a next version n+1 of LL was installed for trainee or AI evaluation.

4.2. Usability and target groups

LL offers a GUI as presented in Figs. 5,6,7,8 making it available to programmers, researchers and clinicians alike, regardless of programming skills. The software has been used in several settings to assert its utility, including trainee comparison and AI assessment, as follows. The software's graphical user interface has been used by clinicians in the working group for comparison between trainees and expert readers to provide quality assurance and standardization of contouring techniques in research and clinical routine. LL was also used for an extensive comparison between two readers on an in-house dataset [14]. LL's backend was used for extensive comparisons of AIs with different backend CNN architectures to a medical expert on SAX Cine images with different acquisition techniques [35].

The rest of the Results section will focus on the requirement of LL's adaptability and extendibility by presenting different steps of an extension to fibrosis and edema imaging (T1 & T2 mapping) as well as focal scar imaging (LGE).

4.3. Adaptability and extendibility to new sequences and scientific endeavours

LL's class structure is extendable to new sequences. By inheriting from the base classes for specific purposes LL can be extended. We demonstrate such extensions, first, for fibrosis imaging, second for focal scar imaging.

4.3.1. Extending Lazy Luna to fibrosis imaging

The adjustments largely mirror typical requirements of T1 mapping [36]. This type of parametric technique often entails locating quantitative difference within the cardiac geometry (Fig. 4).

LL's extension will reflect this in the extended classes below:

- Annotation: T1 mapping requires contours of the myocardium and an insertion point, which demarcates where the right ventricle connects to the left ventricular (LV) myocardium. The insertion point and the centroid of the LV endocardium divide the LV myocardium into segments by degrees. The Annotation class allows for calculating arbitrary numbers of segments
- Category: Images are typically acquired as stacks that span the length of the heart from base to apex. Sorting images allows for locating abnormalities/pathologies and defining segments according to the American Heart Association model [37]. The Category class must be extended to sort the images according to cardiac location
- Clinical Result: The average of T1 values in ms for annotated images
- Metrics: The Dice Similarity Coefficient and the Hausdorff metric are transferred from the SAX Cine applications. The myocardium's average pixel values and their differences are added for T1 mapping, as well as the insertion point to LV centroid angle differences between both readers
- Visualizations: One figure visualizes segmentation and insertion point differences between readers by plotting a coloured comparison of differences (Fig. 6). A second figure produces the AHA model according to annotations of the respective readers. A third figure presents histograms for different segment averages, depending on the insertion point positioning
- Tables: LL offers a table of T1 value averages and average differences for myocardial segments for both readers
- Tabs: A tab for an average AHA model for all cases and their differences is offered (Fig. 7). Another tab for individual cases allows the inspection of the effect of segmentation differences and insertion point differences on the T1 averages of arbitrary numbers of segments (Fig. 8).

Plug-in concept for tables and visualizations

A case overview tab was implemented. It shows a table of the case's clinical results and images with annotations plotted on top. The user can click through the image stack with up/down arrow keys. The Table uses LL's Clinical Result classes. The visualization builds on a T1 mapping Category for slice sorting/accessing images by slice, image correction by rescaling the image values according to DICOM attributes and the Annotation class' support functions for contour plotting (Fig. 5).

Metrics, qualitative comparisons and error tracing

LL admits for tracing reader differences from the patient level to the image level by connecting statistical analysis tabs to case specific tabs. Statistical visualizations, which plot cases as points (e.g. Bland-Altman, correlation plots, etc.), can interactively open tabs with additional information pertaining to the case (Fig. 6).

AHA model and error tracing

The 16-segment AHA model is a popular geometrical abstraction of the heart that is used in the clinical environment to present local average T1 mapping values. It sorts the image stack into basal, midventricular and apical slices, and divides individual slices into six (for basal, midventricular) and four (for apical) segments. This allows for tracing the global value of T1 parametric values from the patient level to the position of T1 value outliers. In order to consider image segments the Annotation class was extended to allow for their calculation. This requires the myocardial contour and the reference point.

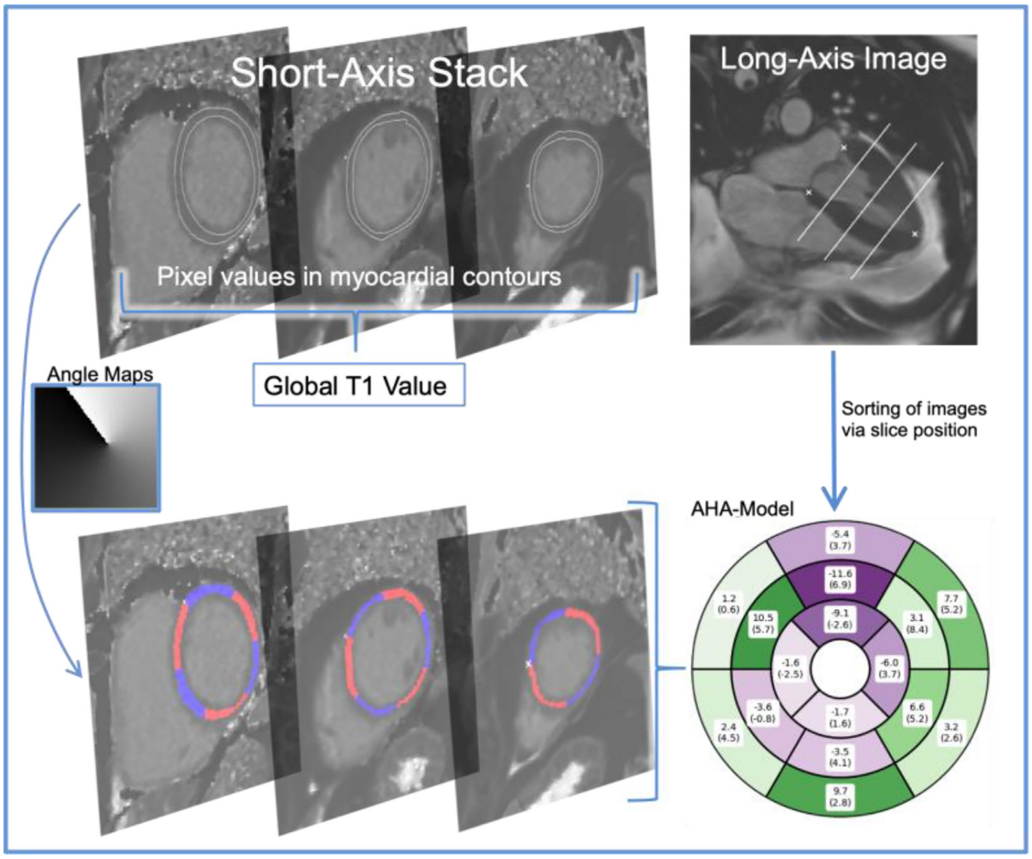


Fig. 4. Pipeline on calculating the AHA model from DICOM images with Annotations.
At the top left there are T1 parametric mapping images with contours and insertion points pertaining to them. The myocardium is divided into segments (red, blue) via angle maps, which are derived from the annotations (lower left). The LVM mask is generated from contours by selecting all pixels whose centre is within the polygon or by Bresenham's line algorithm. The individual images are assigned to a basal, midventricular or apical location depending on the their spatial relationship to the extent and apical points in a long-axis view of the heart (top right). On the bottom right the AHA model is calculated by assigning sorting the segments into their respective bins and calculating the average. The rings correspond to the basal, midventricular and apical positions (outside to inner).
Legend: AHA: American Heart Association, LVM: Left ventricular myocardium.

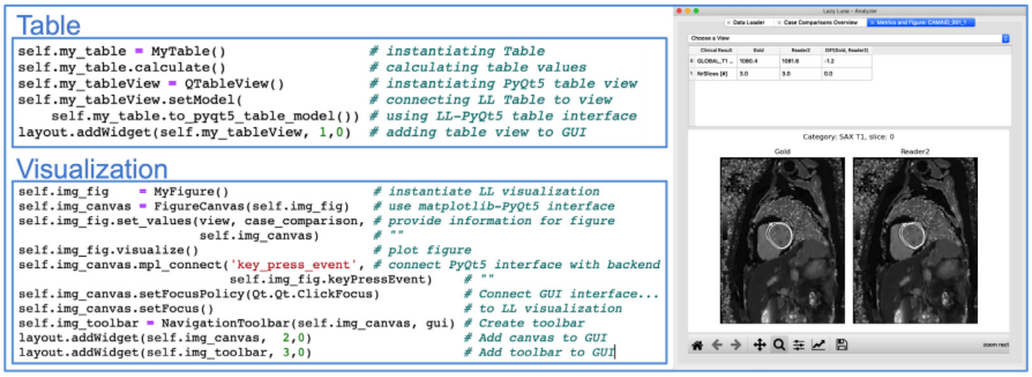


Table
Visualization
Lazy Luna - Analyzer
Data Loader
Case Comparisons Overview
Metrics and Figure: CAMAID_001_1
Choose a View
Category: SAX T1, slice: 0
Gold
Reader2

```
self.my_table = MyTable()                         # instantiating Table
self.my_table.calculate()                         # calculating table values
self.my_tableView = QTableView()                  # instantiating PyQt5 table view
self.my_tableView.setModel(                       # connecting LL Table to view
    self.my_table.to_pyqt5_table_model())         # using LL-PyQt5 table interface
layout.addWidget(self.my_tableView, 1,0)          # adding table view to GUI
```

```
self.img_fig    = MyFigure()                       # instantiate LL visualization
self.img_canvas = FigureCanvas(self.img_fig)       # use matplotlib-PyQt5 interface
self.img_fig.set_values(view, case_comparison,     # provide information for figure
                        self.img_canvas)           # ""
self.img_fig.visualize()                           # plot figure
self.img_canvas.mpl_connect('key_press_event',     # connect PyQt5 interface with backend
                            self.img_fig.keyPressEvent)     # ""
self.img_canvas.setFocusPolicy(Qt.Qt.ClickFocus)            # Connect GUI interface...
self.img_canvas.setFocus()                                  # to LL visualization
self.img_toolbar = NavigationToolbar(self.img_canvas, gui)  # Create toolbar
layout.addWidget(self.img_canvas,  2,0)                     # Add canvas to GUI
layout.addWidget(self.img_toolbar, 3,0)                     # Add toolbar to GUI
```

	Clinical Result	Gold	Reader2	Diff(Gold, Reader2)
0	GLOBAL_T1 ...	1080.4	1081.6	-1.2
1	NrSlices [#]	3.0	3.0	0.0

Fig. 5. Adding Tables and Visualizations to LL Analysis Tool.
On the left code samples are exemplified; on the right the resulting tab of the LL Analysis Tool is presented. On the upper left, code for the integration of an LL Table into the Analysis Tool is shown. After implementing an LL Table it can be instantiated and added to a tab by using QTableViews as an interface. On the lower left, code for the integration of an LL Visualization into the Analysis Tool is presented. After writing an LL Visualization, it can be instantiated and added to a tab by using the FigureCanvas class as an interface to PyQt5. The GUI can be connected to the visualization's key press events with the function mpl_connect. On the right, the Analysis tool is presented with the tab containing the table and visualization. Further code examples can be viewed in github online.
Legend: LL: Lazy Luna, QTableView [34], FigureCanvas [29], GUI: graphical user interface, Global_T1: Average of T1 values inside myocardium for all slices in image stack.

In order to compare two readers at assessing the reproducibility of the AHA segments, LL provides AHA calculations and visualizations for individual cases as well as reader differences by segment. Analogously, for patient cohorts, averaged AHA models and averages of the differences between both readers' segment values are presentable (Fig. 7).

Number of myocardial segments and insertion point and error tracing

As above, calculating average T1 values in myocardial segments of images is central to post-processing of fibrosis images to locate abnormal ranges. Average segment value differences between readers can be caused by contour deviations or different insertion

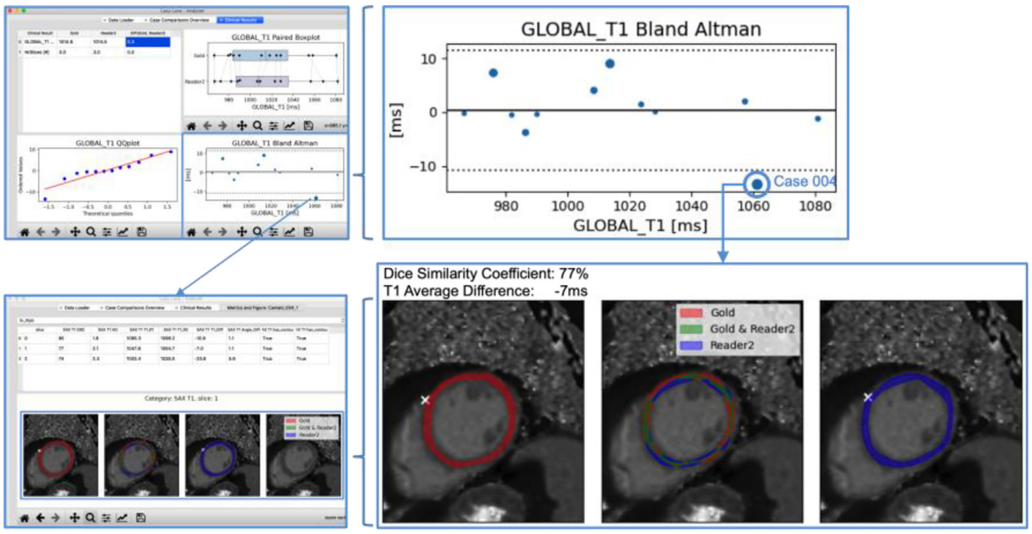


Fig. 6. Error Tracing and Differences.
On the left two, Lazy Luna tabs are presented. On the right, outlined GUI parts are expanded. The upper left tab provides a table of average CR values and statistical plots of the analysed cases, including a paired boxplot, a QQ-plot and a Bland-Altman plot of global T1 values. The Bland-Altman plot is magnified with the circled case being the largest outlier (upper right). By clicking the case the lower tab was opened for an in-depth inspection (lower left). A table of metric values per slice is presented on the top, and a visualization of reader contours and their agreement/disagreement below (gold reader red, reader 2 blue, reader agreement green). The reader differences, as quantified by the DSC and the T1 Average Difference are caused by these contour differences, revealing the origin of the global T1 value difference of the outlier case.
Legend: LL: Lazy Luna, QTableView[32], FigureCanvas[30], GUI: graphical user interface, Global_T1: Average of T1 values inside myocardium for all slices in image stack.

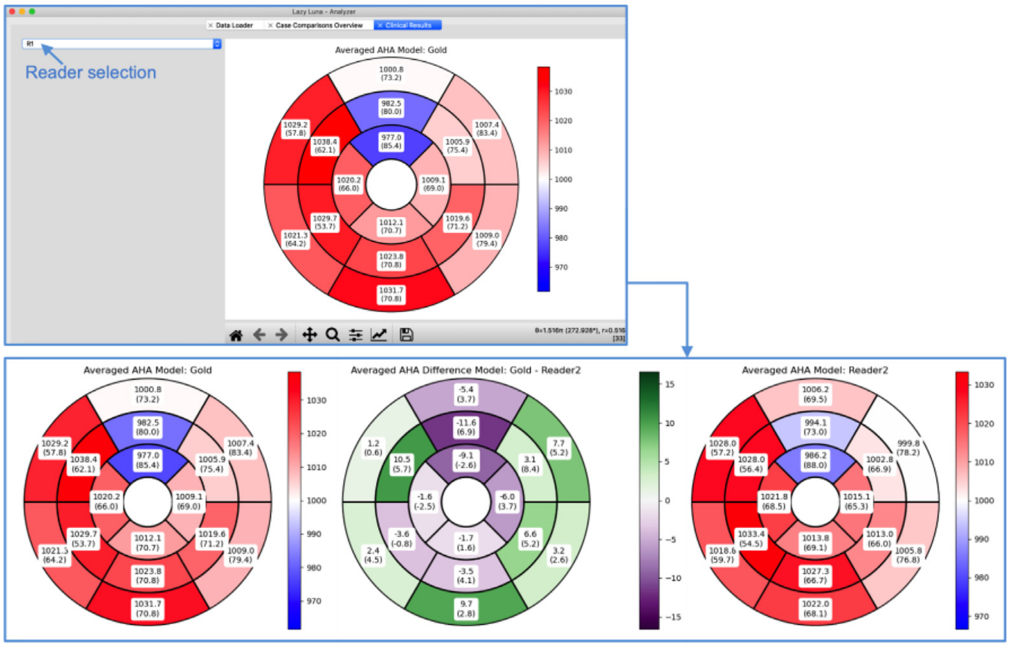
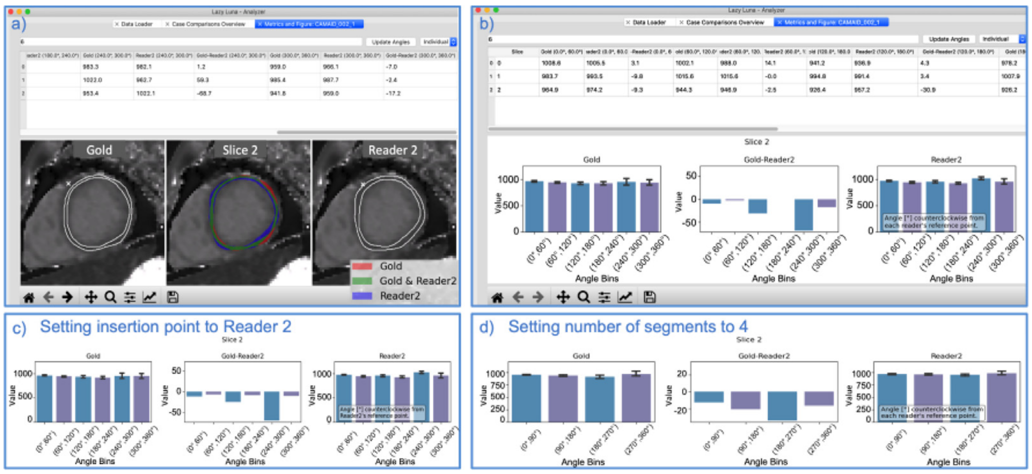

Lazy Luna - Analyzer
Data Loader
Case Comparisons Overview
Clinical Results
Reader selection
Averaged AHA Model: Gold
Averaged AHA Difference Model: Gold - Reader2
Averaged AHA Model: Reader2

Fig. 7. AHA model.
On the top the Average AHA Model tab for several patients shows the average (and the standard deviation) of segment values of a reader for all provided cases. The reader was selected on the upper left of the GUI. Below three figures generated from this tab are presented (from left to right): First, the average AHA model for all cases for the first reader is presented. Second, the average of differences between the first and second reader is visualized. Third, the average AHA model for all cases for the second reader.
Legend: AHA: American heart association, GUI: graphical user interface.

points, which position the segments along the myocardium. Varying numbers of segments may affect the value averages since the number of pixels per segment decreases when the number of segments grows. LL's backend classes offer functions to assess these segment-level confounders. LL's GUI allows the user to examine the effects of these confounders (Fig. 8).

4.3.2. *Extending Lazy Luna to focal scar imaging and CNN outputs*

An LL jupyter notebook was used to convert Emidec's image dataset from Nifti format to DICOM format and the segmentation masks to LL's annotation format. The Nifti format consists of a header with the image's meta information and the image data. The header contains relevant information such as voxel width, height,


a)
Lazy Luna - Analyzer
Data Loader
Case Comparisons Overview
Metrics and Figure: CAMAID_002_1
6
Update Angles
Individual
Gold
Slice 2
Reader 2
Gold
Gold & Reader2
Reader2
b)
Slice 2
Gold
Gold-Reader2
Reader2
Value
Angle Bins
Angle [°] counterclockwise from each reader's reference point.
c) Setting insertion point to Reader 2
d) Setting number of segments to 4

	ader2 (180.0°, 240.0°)	Gold (240.0°, 300.0°)	Reader2 (240.0°, 300.0°)	Gold-Reader2 (240.0°, 300.0°)	Gold (300.0°, 360.0°)	Reader2 (300.0°, 360.0°)	Gold-Reader2 (300.0°, 360.0°)
0		983.3	982.1	1.2	959.0	966.1	-7.0
1		1022.0	962.7	59.3	985.4	987.7	-2.4
2		953.4	1022.1	-68.7	941.8	959.0	-17.2

	Slice	Gold (0.0°, 60.0°)	ader2 (0.0°, 60.0	-Reader2 (0.0°, 6	old (60.0°, 120.0°	ader2 (60.0°, 120.	Reader2 (60.0°, 1	old (120.0°, 180.0	Reader2 (120.0°, 180.0°)	Gold-Reader2 (120.0°, 180.0°)	Gold (18
0	0	1008.6	1005.5	3.1	1002.1	988.0	14.1	941.2	936.9	4.3	978.2
1	1	983.7	993.5	-9.8	1015.6	1015.6	-0.0	994.8	991.4	3.4	1007.9
2	2	964.9	974.2	-9.3	944.3	946.9	-2.5	926.4	957.2	-30.9	926.2

Fig. 8. Tracing Segment Differences.
The Segments and Insertion Point tab allows to switch between two different visualizations (a). The first line of the tab consists of the (left to right) selecting the number of segments for the myocardium, an update button for the table and figure below, and a selection option for the insertion point (first reader's insertion point, the second reader's, or each individual reader's point). The second row is a table of segment values; three consecutive columns concern the first reader's segment average, the second reader's and their difference. The main figure (row three) shows contours and insertion points for both readers, which are plotted on the left and right image, respectively. The centre image presents a contour comparison of both readers. In b) the figure of tab a) was switched (via shift key) to histograms of the segments' average values: The first and third histograms concern the readers' average segment values; the middle histogram represents the average segment value differences. In c) histograms were created according to the second reader's insertion point for both readers, which led to minor histogram value deviations. In d) a histogram was calculated with four instead of six histograms, which "averaged away" 30ms of reader differences.

and depth. The image data has a shape, which provides the image sizes, the number of phases and slices. With this information a Dicom file is constructed. The annotation files were generated from the Nifti file masks, stored in the same format as the images. The voxel segmentations in Nifti format were outlined as polygons and stored in LL annotation format as pickle files, mapping contour names to shapely geometries. The exact code is published in the repository as a Jupyter Notebook and may guide new users to extending LL to new data formats.

The annotation class required no changes to generalize to focal scar imaging. Several new clinical results were implemented as classes, including the LV volume, LVM volume and mass, scar volume, mass and fraction, excluded volume as mass as well as no reflow volume. LL also required a new View class, the SAX_LGE_View to refocus the cases on the images and clinical parameters.

A UNet predicted segmentation masks for myocardium, scar and no reflow tissue. We used this under-trained network to predict the Emidec segmentations for the training set. We used LL to produce the reader comparisons in Fig. 9 and offer an investigative video in supplementary material.

5. Discussion

Lazy Luna (LL) is a software package that offers a multilevel comparison of readers on different CMR techniques. It is available to clinicians and programmers alike. LL's extendibility to new image types and scientific endeavours was illustrated by presenting a step-by-step extension of LL to fibrosis imaging. In results this allowed for an illustrative reader comparison on fibrosis imaging cases. LL has been used in other settings to assert its functionality and utility. The GUI has successfully been used for comparison between trainees and expert readers to provide quality assurance and standardization of contouring techniques in research and clinical routine. LL was also used for an extensive comparison between two readers on an in-house dataset [14].

LL provides error tracing by connecting different analysis levels. We demonstrated this on an AHA model difference for individual case comparison and an average of AHA model differences for two readers. By doing this, the relevance of contouring differences can be traced from individual images and segments to reader trends, while offering insights into how segmentation difficulty and cardiac geometry interact.

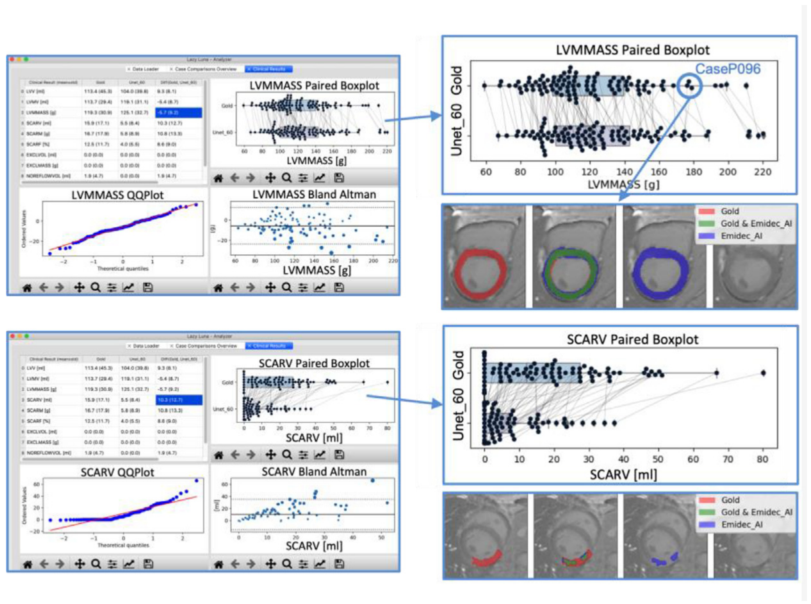
Increasingly, AIs are being applied to multiple sequences to quantify several cardiac parameters simultaneously: such as SAX Cine imaging, LAX Cine-, fibrosis-, edema- and scar imaging [38,39]. As this becomes more prominent, full sequence comparisons as offered by LL should become more typical. LL is provided as open-source software. Its usability and general concept can be verified quickly with the EMIDEC dataset on Github. LL should allow AI developers to evaluate the quality of their algorithms on several levels of analysis simultaneously, satisfying the interest of CNN developers while addressing the clinical relevance of differences.

In recent years, CNN developers have argued that their algorithms are within the range of interobserver reproducibility and thus also in typical variability of clinical routine parameters [5,16]. The equivalence of different contouring software has been assessed by testing that confidence intervals were within defined tolerance ranges [11]. Such ranges are necessarily parameter specific as different confounders have different effect-sizes. Assessing and defining tolerable biases between CNNs and clinicians as well as limiting the proportion of cases, which lie outside of such tolerance ranges, may be the topic of another work. LL should be extended to test for, and visualize, reader equivalence according to well-defined criteria.

Likewise, the training and education of readers is a time-intensive task, based on curriculums and proficiency assessments [10,40]. One-on-one teacher-student training is deemed most advantageous, but is also most resource absorbing [9,10]. Training and education has been shown to improve LV volume reproducibility [9]. LL could be used in training settings to automate difference assessment as well as offering illustrations of annotation differences between teacher and student.

Outlook

LL can analyse and characterise reader differences in order to standardize contouring methods for more reproducible clinical results. AIs are increasingly relevant for quantifying CMR images. LL has been used to investigate different CNN architectures' (UNet



	Clinical Result (mean±std)	Gold	Unet_60	Diff(Gold, Unet_60)
0	LVV [ml]	113.4 (45.3)	104.0 (39.8)	9.3 (8.1)
1	LVMV [ml]	113.7 (29.4)	119.1 (31.1)	-5.4 (8.7)
2	LVMMASS [g]	119.3 (30.9)	125.1 (32.7)	-5.7 (9.2)
3	SCARV [ml]	15.9 (17.1)	5.5 (8.4)	10.3 (12.7)
4	SCARM [g]	16.7 (17.9)	5.8 (8.9)	10.8 (13.3)
5	SCARF [%]	12.5 (11.7)	4.0 (5.5)	8.6 (9.0)
6	EXCLVOL [ml]	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)
7	EXCLMASS [g]	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)
8	NOREFLOWVOL [ml]	1.9 (4.7)	0.0 (0.0)	1.9 (4.7)

Fig. 9. Emidec CNN Analysis.
The top three sub-figures compare the UNet's LVM estimation to the gold standard's. The Clinical Results tab (top left) offers an overview of all clinical result averages of the gold standard and the UNet_60 as well as their differences. The focus lies on the paired boxplot enlarged on the upper right. Here, the two readers (Gold top, Unet_60 bottom) are presented as boxplots with the case's LVM values plotted as dots on top. Lines connect the case dots in order to visualize the case-specific reader differences. Below the reader contours of CaseP096 were presented, showing the constant overestimation of the epicardial contour, providing a qualitative clue to the LVM difference. The bottom three sub-figures provide an analogous analysis concerning the UNet's SCARV estimation. The paired boxplot reveals that the UNet consistently underestimates the scar volume drastically. The lower plot reveals that the UNet has not yet learned to estimate the full scar but only fragments of it.
Legend: LVMMASS: left ventricular mass, SCARV: scar volume, ml: millilitre, g: gram.

[15], FCN [41], Dilated UNet [42], MultiResUnet [43]) performances against expert readers [35]. Several working groups have enhanced CNN performance by integrating cardiac geometry assumptions or plausibility checks of the heart's blood pool volumes over the cardiac cycle into the overall segmentation pipeline [44–46]. Whether AI performance improves by integrating cardiac geometry information should be investigated.

We based LL on a class diagram for explainability. The general idea is intuitive: Cases contain images and annotations, categories manage the images and annotations, and the images and annotations can be combined to calculate clinical results per case. When two cases reference the same images the annotations can be compared to each other on the image level, and the clinical results on patient level. This applies to CMR as we could generalize to different sequences in Results. Furthermore, this principle applies to any conceivable 2D imaging modality, such as Neuro MRI, CT scans or echocardiography.

Releasing the software as open source code should allow for more feedback and a sharing of maintenance costs for LL. This should provide incentives to improve and adjust the software to a variety of needs.

Limitations

LL requires user intervention to recognize image types (such as SAX Cine or T1 Mapping images). Currently, this recognition is semi-automated by a DICOM tag focussed image-type suggestion. However, users have described it as tedious and the task ought to be automatized in future work. The current version of LL allows for the comparison between exactly two readers (regardless of them being human or AI). Comparing multiple readers to another reader (often the gold standard reader) would be a useful extension to LL. This study is limited to the analysis of LL's software architecture. Case studies that would illustrate the utility of LL in QA scenarios are out of the scope of this work's scope.

6. Conclusion

The presented software Lazy Luna allows for a multilevel reader comparison on several sequences typical in the clinical domain, while remaining flexible and extendible to new scientific undertakings. Extending LL to T1 parametric mapping demonstrated the software's flexibility. LL will enhance reader comparisons by merging AI analysis with clinical analysis and contribute to standardization and reproducibility in clinical routine.

Author contributions

All co-authors provided input to the project. CA, JW, DV, SL, AH and JSM provided advice and support on software development. MF and EA provided the contours of the presented cases. JG and JSM provided feedback on the software's utility in practice. TH implemented the software and carried out the data analysis. All co-authors reviewed and approved the final manuscript.

Ethics declarations

The local ethics committee of Charité Medical University Berlin gave ethics approval for the original study (approval number EA1/323/15). All patients gave their written informed consent before participating in the study.

Data availability

We have all source images and workspaces. They are saved at a central server of the Charité as well as locally on a working group specific server. We could make access possible following dedicated request after communication with the legal department as there are special rules based on the EU law and the rules of the Berlin data officer rules. All the data generated during this study are included in this published article and its supplementary information files.

Declaration of Competing Interest

The authors declare no competing interests.

Acknowledgements including Declarations

We wish to thank the members of the WG CMR for feedback and input. TH receives funding from the German Research Foundation (GRK2260, BIOQIC). At time of the software's conception TH was a Master's student employee at Siemens Healthineers, Erlangen Germany.

Supplementary materials

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.cmpb.2023.107615.

References

[1] J. Schulz-Menger, et al., Standardized image interpretation and post-processing in cardiovascular magnetic resonance - 2020 update : Society for Cardiovascular Magnetic Resonance (SCMR): Board of Trustees Task Force on Standardized Post-Processing, J. Cardiovasc. Magn. Reson. Off. J. Soc. Cardiovasc. Magn. Reson. 22 (2020) 19.
[2] M.S. Hansen, T.S. Sørensen, Gadgetron: An open source framework for medical image reconstruction: Gadgetron, Magn. Reson. Med. 69 (2013) 1768–1776.
[3] A. Zwanenburg, S. Leger, M. Vallières, S. Löck, Image biomarker standardisation initiative, Radiology 295 (2020) 328–338.
[4] Left Ventricle Full Quantification Challenge MICCAI 2019. https://lvquan19.github.io/.
[5] O. Bernard, et al., Deep learning techniques for automatic MRI cardiac multi-structures segmentation and diagnosis: is the problem solved? IEEE Trans. Med. Imaging 37 (2018) 2514–2525.
[6] P. Haaf, et al., Cardiac T1 Mapping and Extracellular Volume (ECV) in clinical practice: a comprehensive review, J. Cardiovasc. Magn. Reson. 18 (2017) 89.
[7] D.R. Messroghli, et al., Clinical recommendations for cardiovascular magnetic resonance mapping of T1, T2, T2* and extracellular volume: a consensus statement by the Society for Cardiovascular Magnetic Resonance (SCMR) endorsed by the European Association for Cardiovascular Imaging (EACVI), J. Cardiovasc. Magn. Reson. Off. J. Soc. Cardiovasc. Magn. Reson. 19 (2017) 75.
[8] on behalf of SCMR Clinical Trial Writing Group et al. Society for Cardiovascular Magnetic Resonance (SCMR) expert consensus for CMR imaging endpoints in clinical research: part I - analytical validation and clinical qualification. *J. Cardiovasc. Magn. Reson.* 20, 67 (2018).
[9] T.D. Karamitsos, L.E. Hudsmith, J.B. Selvanayagam, S. Neubauer, J.M. Francis, Operator induced variability in left ventricular measurements with cardiovascular magnetic resonance is improved after training, J. Cardiovasc. Magn. Reson. Off. J. Soc. Cardiovasc. Magn. Reson. 9 (2007) 777–783.
[10] E. Hedström, et al., The effect of initial teaching on evaluation of left ventricular volumes by cardiovascular magnetic resonance imaging: comparison between complete and intermediate beginners and experienced observers, BMC Med. Imaging 17 (2017) 33.
[11] L. Zange, et al., Quantification in cardiovascular magnetic resonance: agreement of software from three different vendors on assessment of left ventricular function, 2D flow and parametric mapping, J. Cardiovasc. Magn. Reson. Off. J. Soc. Cardiovasc. Magn. Reson. 21 (2019) 12.
[12] S. Marchesseau, J.X.M. Ho, J.J. Totman, Influence of the short-axis cine acquisition protocol on the cardiac function evaluation: a reproducibility study, Eur. J. Radiol. Open 3 (2016) 60–66.
[13] J. Mullally, et al., Marked variability in published CMR criteria for left ventricular basal slice selection - impact of methodological discrepancies on LV mass quantification, J. Cardiovasc. Magn. Reson. 15 (2013) P101.
[14] T. Hadler, et al., Introduction of Lazy Luna an automatic software-driven multilevel comparison of ventricular function quantification in cardiovascular magnetic resonance imaging, Sci. Rep. 12 (2022) 6629.
[15] Ronneberger, O., Fischer, P. & Brox, T. U-Net: convolutional networks for biomedical image segmentation. *ArXiv150504597 Cs* (2015).
[16] W. Bai, et al., Automated cardiovascular magnetic resonance image analysis with fully convolutional networks, J. Cardiovasc. Magn. Reson. Off. J. Soc. Cardiovasc. Magn. Reson. 20 (2018) 65.
[17] Isensee, F. et al. nnU-Net: Self-adapting Framework for U-Net-Based Medical Image Segmentation. *ArXiv180910486 Cs* (2018).
[18] J. Duan, et al., Automatic 3D Bi-ventricular segmentation of cardiac images by a shape-refined multi- task deep learning approach, IEEE Trans. Med. Imaging 38 (2019) 2151–2164.
[19] Shwartzman, O., Gazit, H., Shelef, I. & Riklin-Raviv, T. The Worrisome Impact of an Inter-rater Bias on Neural Network Training. *ArXiv190611872 Cs Eess* (2020).
[20] J. Sander, B.D. De Vos, J.M. Wolterink, I. Išgum, Towards increased trustworthiness of deep learning segmentation methods on cardiac MRI, Med. Imaging 2019 Image Process 44 (2019), doi:10.1117/12.2511699.
[21] DICOM. *DICOM* https://www.dicomstandard.org.
[22] G. Van Rossum, The Python Library Reference, release 3.8.2, Python Software Foundation, 2020.
[23] Gillies, S. & others. Shapely: manipulation and analysis of geometric objects. (2007).
[24] M. Mustra, K. Delac, M. Grgic, Overview of the DICOM standard, in: 2008 50th International Symposium ELMAR, 1, 2008, pp. 39–44.
[25] D. Mason, SU-E-T-33: Pydicom: an Open Source DICOM Library, Med. Phys. 38 (2011) 3493.
[26] The Shapely User Manual — Shapely 1.8.0 documentation. https://shapely.readthedocs.io/en/latest/manual.html.
[27] Gillies, S. & others. Rasterio: geospatial raster I/O for Python programmers. (2013).
[28] G. Van Rossum, F.L. Drake, Python 3 Reference Manual, CreateSpace, 2009.
[29] matplotlib.figure.Figure — Matplotlib 3.3.4 documentation. https://matplotlib.org/3.3.4/api/_as_gen/matplotlib.figure.Figure.html.
[30] pandas.DataFrame — pandas 1.4.1 documentation. https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.html.
[31] J.D. Hunter, Matplotlib: A 2D Graphics Environment, Comput. Sci. Eng. 9 (2007) 90–95.
[32] team, T. pandas development. pandas-dev/pandas: Pandas. (2020) doi:10.5281/zenodo.3509134.
[33] Qt 5.15. https://doc.qt.io/qt-5/.
[34] PyQt - QTab Widget. https://www.tutorialspoint.com/pyqt/pyqt_qtabwidget.htm.
[35] Hadler, T., Amman, C., Gröschel, J. & Schulz-Menger, J. Multilevel comparison of neural networks for ventricular function quantification in CMR accelerated by compressed sensing. *ISMRM - Int. Soc. Magn. Reson. Med.*
[36] D.R. Messroghli, et al., Clinical recommendations for cardiovascular magnetic resonance mapping of T1, T2, T2* and extracellular volume: a consensus statement by the Society for Cardiovascular Magnetic Resonance (SCMR) endorsed by the European Association for Cardiovascular Imaging (EACVI), J. Cardiovasc. Magn. Reson. 19 (2017) 75.
[37] Standardized Myocardial Segmentation and Nomenclature for Tomographic Imaging of the Heart. 4.
[38] Multi-sequence myocardium segmentation with cross-constrained shape and neural network-based initialization, Comput. Med. Imaging Graph 71 (2019) 49–57.
[39] A Chartsias, et al., Disentangle, align and fuse for multimodal and semi-supervised image segmentation, IEEE Trans. Med. Imaging 40 (2021) 781–792.
[40] E.A. Ruden, D.P. Way, R.W. Nagel, F. Cheek, A.J. Auseon, Best practices in teaching echocardiography to cardiology fellows: a review of the evidence, Echocardiogr. Mt. Kisco N 33 (2016) 1634–1641.
[41] Long, J., Shelhamer, E. & Darrell, T. Fully Convolutional Networks for Semantic Segmentation. *ArXiv14114038 Cs* (2015).
[42] Wang, S. et al. U-Net Using Stacked Dilated Convolutions for Medical Image Segmentation. 8.
[43] N. Ibtehaz, M.S. Rahman, MultiResUNet : rethinking the U-Net architecture for multimodal biomedical image segmentation, Neural Netw. 121 (2020) 74–87.
[44] R. Robinson, et al., Automated quality control in image segmentation: application to the UK Biobank cardiovascular magnetic resonance imaging study, J. Cardiovasc. Magn. Reson. Off. J. Soc. Cardiovasc. Magn. Reson. 21 (2019) 18.
[45] B. Ruijsink, et al., Quality-aware semi-supervised learning for CMR segmentation, Stat. Atlases Comput. Models Heart STACOM Workshop (2020) 97–107 2020.
[46] Chen, C. et al. Learning Shape Priors for Robust Cardiac MR Segmentation from Multi-view Images. in vol. 11765 523–531 (2019).