Exploring a Modular Approach for Deploying and Testing Cardiac Image Processing and Analysis Methods in the Clinical Workflow

João Abrantes[1], Nuno Almeida[1,2], and Samuel Silva[1,2(✉)]

[1] Institute for Electronics and Informatics Engineering of Aveiro (IEETA), University of Aveiro, Aveiro, Portugal
jgabrantes@ua.pt

[2] Department of Electronic, Telecommunications, and Informatics (DETI), University of Aveiro, Aveiro, Portugal
{nunoalmeida,sss}@ua.pt

Abstract. Cardiac health is a big concern, worldwide, with cardiovascular disease (CVD) being the most predominant cause of death, making early diagnosis a top priority in public health. The technology supporting the diagnosis of CVD, e.g., cardiac computerized tomography angiography (CTA), has been evolving at a fast pace providing a wide range of data on the anatomy and function of the heart. When developing novel processing and analysis methods to tackle this data, one important challenge concerns how to make them available for clinicians to test. The aim would be to enable full exploration of these methods in the clinical workflow, i.e., supporting all the standard image visualization and analysis features provided by clinical workstations, from early on, to foster insight over their features and usefulness to inform development. Additionally, with the advances of technology, mobile devices, beyond the traditional workstations, have gained importance, e.g., during consultation. However, they have limited processing resources. In this regard, this work explores a modular multiplatform solution to support the early deployment of cardiac analysis methods to the clinician's office. To this effect, we define the requirements for such a platform and present a first instance of its development exploring the Open Health Imaging Foundation resources. At its current stage, we demonstrate the feasibility of the approach and the integration of simple image processing modules in the pipeline.

Keywords: image processing and analysis · modular software architecture · multiplatform approaches · cardiac CT angiography

1 Introduction

According to the World Health Organization, cardiovascular diseases (CVDs) are the cause of death for an estimated 17.9 million people, every year, representing

A. Cunha et al. (Eds.): MobiHealth 2022, LNICST 484, pp. 241–254, 2023.
https://doi.org/10.1007/978-3-031-32029-3_22

32% of all deaths worldwide [19]. In this regard, it is important to continuously improve the capabilities of clinical diagnosis [11,14–16] since the detection of CVDs at an early stage grants a higher rate of success on treatment and cure of a cardiac patient. With the technological advances seen in recent years there was a significant improvement in medical imaging and software assisted diagnostic solutions, making them more precise, efficient, and increasingly noninvasive. In cardiac health specifically, these improvements lead to the capability of analyzing three dimensional images of the whole heart and even to simulate reconstructive and invasive procedures based on computerized tomography (CT) and magnetic resonance imaging (MRI) data. From the cardiac diagnosis perspective, important advances also entail the ability to acquire these images over time, providing data that can be used to analyze cardiac function, e.g., to assess the left ventricle movement and contractility.

The sheer amount of data provided by the imaging technologies and their complexity, along with the need to increasingly move into a quantitative framework supporting diagnosis, makes computer assisted diagnosis (CAD) tools a requirement [9]. In this regard, a strong body of work exists regarding the research of more advanced diagnosis methods, addressing ways to improve the data that is made available to clinicians, e.g., summarizing and characterizing, in a quantitative manner (e.g., calcium scoring [8]) cardiac function. However, the path for their validation and deployment faces some challenges. First of all, cardiac CT exams entail a considerable volume of data (approx. 1.5 GB), resulting in processing and analysis methods that require a fair amount of computational resources and, second, the proposed approaches are not easily deployed to be tested by the clinicians due to the fact that the existing cardiac workstations (mostly proprietary) do not allow enough versatility for integrating these new solutions. These aspects work as barriers for a more thorough assessment and exploration by clinicians for several reasons: (a) the developers often need to create a base platform that provides the full set of basic features to manipulate and visualize the imaging data, aligned with the current standard clinical workflow, so that they can be used by the clinicians profiting from their familiarity and training in using other tools; (b) obtaining feedback from clinicians is possible, but made harder and less informing, given that these novel systems are not integrated in their workflow, which entails a limited amount of data considered for testing and less space to freely explore the available features in novel ways; (c) even if a new workstation is placed at the cardiac service, adding and updating the available features to keep up with development is troublesome, particularly if it entails doing it locally, each time; and (d) mobile platforms have gained space in the clinical setting, given their portability, e.g., to bring into consultation with a colleague or to check exams while with a patient – which can boost the discussion around new methods—, but the current setting for cardiac analysis is limited to workstations and, given the complexity of the data, is computationally intensive, not making it compatible with a mobile approach.

thesIn the scope of improving CAD tools for cardiac diagnosis based on CT imaging of the heart, our team is proposing novel ways to assess coronary

artery function through numerical simulations. In this context, and considering the challenges identified above, the work presented here starts by identifying major requirements to address them and explores the feasibility of a modular solution for clinical CT imaging processing that could serve the design and test of these novel methods. This should provide the grounds that both facilitate their deployment and exploration by the clinicians towards a more adequate setting to profit from their insights.

The remainder of this document presents the conceptualization of the required solution, in Sect. 2, resulting in the requirements that guide the design and development decisions. Then, Sect. 4 presents an overview of the current stage of the work describing a first proof of concept of our proposal built around the Open Health Image Foundation (OHIF) resources. Finally, some conclusions are presented, in Sect. 5, along with some routes for further work.

2 Background

In this section we start by briefly establishing the context for our work, particularly regarding the needs for the clinical side, and shortly overview common toolkits supporting the development of novel methods for image processing, analysis, and visualization. These help establish the overall scenario for the envisaged platform and define a first set of requirements guiding development.

2.1 Clinical Context

As computed tomography imaging evolved, two imaging protocols emerged for cardiac diagnosis: Coronary computed tomography angiography (CCTA) and Myocardial CT Perfusion (CTP). Coronary computed tomography angiography images the coronary arteries and allows evaluating the presence and severity of cardiac artery disease (CAD) by allowing the inspection to detect if plaque buildup has narrowed the coronary arteries. CCTA is performed by injecting contrast in the blood vessels and the image can be acquired throughout the cardiac cycle [1,4,7] using an acquisition technique that trigger scans during a specific interval of the cardiac cycle which provides higher-quality scans with no pulsation artifacts (ECG gated Angiography). Although CCTA can rule out the presence of obstructive CAD with a high degree of certainty, the morphologic information provided by this technique is still not sufficient to determine the downstream functional consequences of a given coronary lesion. The technological development in CT lead to the possibility to perform stress myocardial perfusion to evaluate coronary artery disease, providing not only anatomical information, but also covering the functionality of heart's stenosis [2,17]. Myocardial perfusion is based on the distribution of the iodinated contrast injected in the patient, during its first pass in the myocardium. The hypoattenuating areas containing less contrast material will determine myocardial perfusion defects [18]. The goal of Perfusion CT is to reconstruct and display the myocardium of the left ventricle as a volumetric perfusion map by enabling quantitative measures

of myocardial perfusion as attenuation of this region over time [6]. A number of requirements have to be fulfilled for these techniques to work as intended, after the acquisition of a CT exam, the image dataset needs to be reconstructed, processed and prepared for the following diagnosis process. In CCTA multiple reconstruction parameters will dictate the quality of the reconstructed axial image. For post-processing CCTA uses different methods as: curved multiplanar reformats, shaded-surface display (SSD), maximum intensity projection (MIP), and volume rendering [12].

With any of these types of exams there are a few steps and tools that compose a common workflow of the clinicians while exploring them to gather information supporting the diagnosis, as depicted in the diagram of Fig. 1. Data is acquired from the CT exam and is automatically sent to the PACS server, which is connected to a viewer where a clinician can browse the patients and associated studies. By selecting one study, the viewer will load the CT image from the PACS server. Then, the clinician adjusts the viewing configuration to position the different viewing planes to obtain the best possible orientation to examine, for example, the left ventricle. To improve viewing, the way the pixels are displayed can be adjusted to enhance ranges of intensity. At this point, the clinician can take measures of particular structures of interest or apply any processing to extract notable information, e.g., computing the amount of blood inside the left ventricle, over time.

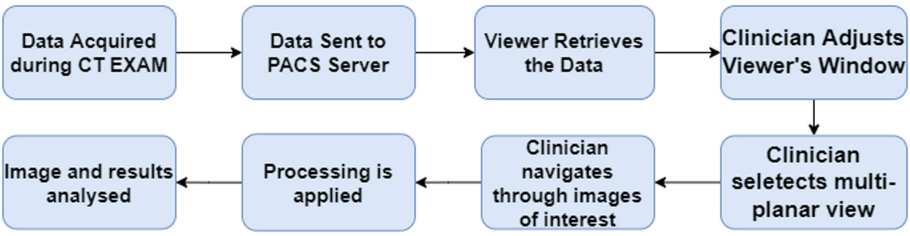

Fig. 1. Overall steps taken by a clinician to access and examine a cardiac CT exam and gather information to support diagnosis.

By analyzing the overall depiction of the clinician's workflow, when working with the cardiac images, it is relevant to note that there are several features to adjust the viewing for the intended diagnosis goals and image type that need to be present. These are important because they allow setting optimal viewing conditions for the anatomy of interest and according to the clinician's preference for interpreting the data. Being able to set particular conditions for visualization of the anatomy is paramount to harness the clinician's experience and reach views that are comparable with, e.g., other cases, and abide, for instance, to recommended practice. Therefore, to bring novel processing and analysis methods into the clinician's workflow, it is vital to ensure all the steps that precede their application and allow proper visualization and analysis of the outcomes.

While the existing workstations, at the clinical setting, provide these tools (e.g., syngo.CT[1]), they are mostly proprietary and integration of novel processing and analysis methods is not possible or in a very limited manner. Additionally, mobile devices, such as tablets, have gained prominence, in the clinical setting, and being able to provide at least a simplified version of the workflow in these devices, e.g., to support bedside consultation can expand its usage scope. Therefore, having a platform that could ensure the required workflow for different platforms and devices, and an easy integration and evolution of novel processing and analysis methods is a key resource to advance research in this topic.

2.2 Tools for Medical Image Processing and Visualization

There is a great volume of tools and packages commonly used by researchers to develop novel methods to visualize, process, and analyze medical images for a wide range of applications [10]. There are powerful tools such like OpenCV[2] and ITK[3] that offer a range of features for image manipulation, processing and analysis. Other tools offer useful solutions for image visualization such as VTK, Cornerstone, SCIRun and Volview. Additionally there are more complex tools such as MevisLab[4] or even Matlab that provide a mixture of features from simple image manipulation up to powerful processing solutions for data and information visualization. This reflects different architectures, approaches, and languages that are quite suitable for fast prototyping of novel methods, but this heterogeneity creates a challenge on how to integrate solutions by there tools into the same platform. This is one additional reason for our aim for modularity as explained in the next sections.

3 Conceptual Vision and Requirements

The end goal for the development of this software is to support not only current imaging-based diagnosis methodologies but, most importantly, to integrate the features that support the test of new ones. In our envisaged scenario (see Fig. 2 for an illustrative diagram), researchers are developing new tools to support diagnosis, at their lab, and using a small set of anonymous imaging data for testing. When the methods reach a stage where a number of features is already usable, the researchers decide that they require the methods to be tested and validated by clinicians to inform further development. At this time, the researchers make the method available in the platform.

At the cardiology service, a clinician loads a recent CCTA exam from the PACS server and uses the newly deployed method to analyze the data, e.g., to extract the coronary arteries. This can be performed both at a workstation or while using a tablet, e.g., to check some detail about a particular patient when discussing the case with a colleague.

[1] syngo.CT.
[2] OpenCV: https://opencv.org/.
[3] ITK: https://itk.org/.
[4] MevisLab: https://www.mevislab.de/.

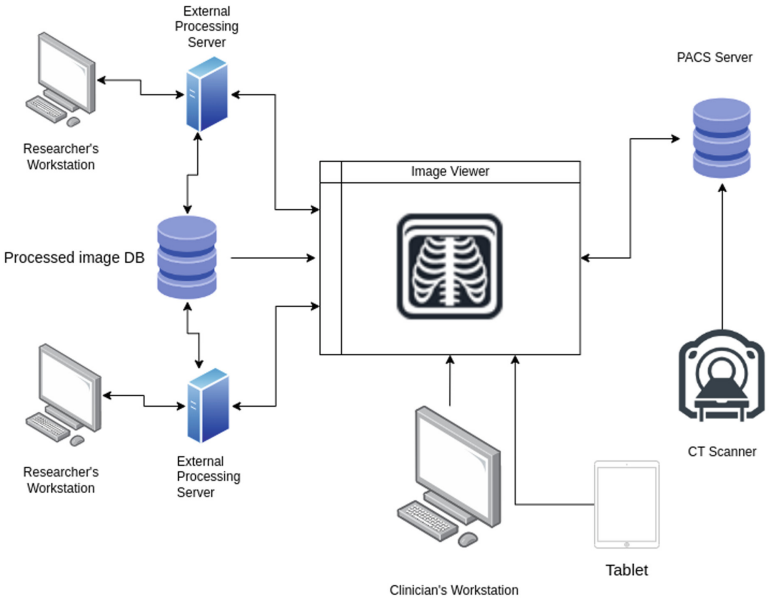

Fig. 2. Diagram depicting the overall scenario envisaged for supporting a faster translation from research to the clinical office supporting a faster cycle for exploration and test of novel processing, analysis, and visualization methods.

3.1 Requirements

To support the vision described above and depicted in Fig. 2, it is important that the proposed platform provides the clinician with a set of tools that are common to the cardiac workflow (see Table 1). It needs to include: (a) access to a DICOM Image Archive to retrieve patient's data and images; and (b) a user interface that will combine a number of essential features, e.g., Image Visualizer (the canvas itself) , window level adjustment, viewer layout customization (where images can be displayed in different quadrants of the canvas, aimed to display different image planes). Overall, it is important to note that the researchers will profit from all the requirements identified for the clinician side, since it will also provide them with ways of testing their methods inserted into the clinicians' workflow. Additionally, from the researchers' perspective, the platform needs to enable using different tools for developing novel methods and fast ways to deploy and update these methods.

Since one major goal for the platform is to be extensible and dynamic, having a modular approach is paramount since each component can be developed independently. This is a major concern, because it grants a higher consistency to the application, since changing or updating one feature will not modify the system as a whole. If a new processing methodology is needed or was developed and it needs to be tested, it can be easily added to the features by creating a new service or modifying an existing one without major compromises with the whole

Table 1. Overview of main requirements considering the perspective of the researcher and clinician for the envisaged system.

Researcher-side	Clinician-side
Fast Image Data Retrieval and Sending	DICOM data Retrieval
Agile deployment of new processing functions	Multi-planar View
Abstraction from the viewer complexity	Viewer Layout customization
Technological independence from the remaining components	Measurement tools
	Window levels adjustment
	Annotation tools
	Study and Series Browser and Lister
	DICOM Metadata access

project. It was also deemed important to prioritize open-source solutions when choosing an approach. Within the scientific research context, it is important that there is transparency with the process of result acquisition – as opposed to a black box solutions –, since it grants the possibility to critically analyze the results and reproduce them, something that is hampered when the source code is not available [5]. Furthermore, modularity should also consider an approach that enables their remote execution, e.g., in a remote server, instead of needing to run locally. This is important to enable using the system in devices with less computational resources, e.g., tablets, by offloading the processing operations.

4 Design and Development

Based on the modularity aspect required it was decided to explore a Micro-Kernel or Micro-Service Software architecture, where each component is independent of each other and inter-operate through an API or having a core system where each component acts as a plugin. The data storage service should work independently of the viewer, and more complex processing operations should be hosted in external services, removing any limitation in the language and framework that the image viewer might impose, making it possible for someone to develop new processing units without needing to know how the core system works, creating an abstraction layer from this service to the core system. Based on the requirements previously listed our system will consist of a PACS Server, a DICOM Image Viewer and one or more image processing services.

We decided on starting with the Image Viewer, since it will be the central component of the platform. It is the viewer which loads images from the PACS, provides access to the basic imaging manipulation tools, and where the most complex imaging processing features will be triggered. In this regard, we endured some research on current existing open-source image viewers.

4.1 Selection of Image Viewer Core Framework

Based on the conceptualized approach and requirements, the literature was surveyed regarding existing solutions that could serve as the core framework for our work. While a few exist, e.g., PostDicom[5], Horos[6], MANGO[7], and DICOM Web Viewer[8], two deserved particular attention for their considerable range of features and were tested more thoroughly.

Weasis. The first open source project that was taken in consideration was Weasis [13]. It is a standalone and web-based DICOM viewer with a modular architecture based on OSGI (Java). It is multi-languague and has a flexible integration with various imaging information systems as PACS, RIS (Radiological Information System), PHR (Personal Health Record) and HIS (Hospital Information System). This viewer allows high-rendering with OpenCV library. Its architecture is based on plugins which are organized in Plugin Type bundles and are in the form of Maven Archetypes. However, since this project has little code-level documentation and is organized as a Java Enterprise project it was found to have a steep learning curve to start developing for it, making it an unrealistic solution for the time constraints of this project.

OHIF. The second open source viewer that was considered was OHIF [3] an open source, web-based, medical imaging platform running as a node.js application. Originally build for radiology it is designed to load large volumes of data in a short period of time, retrieving metadata and pixel data ahead of time. It uses cornerstone.js, a Javascript library for visualization of medical images in web browsers that support the HTML5 for rendering, decoding and annotation of medical images. Works out-of-the-box with Image Archives that support DICOMWeb (DICOM Standard for web-based medical imaging using RESTful services). Provides a plugin framework for creating task-based workflow modes which can re-use the core functionalities. This web-based software includes an extensible user interface (UI) containing various components available in a reusable component library built with React.js and Tailwind CSS. OHIF is currently at its third version OHIF-v3 with which is the current stable version, however since version 3 still lacks a multi-dimensional image handler, it was decided to move to the previous version 2 which includes certain frameworks not yet implemented in version 3. We have found this solution more adequate for the needs of this project, it has an extensive documentation, and several guidelines on the development process. It is a fully web-based application, with an active open-source development community with new features and extensions in sight to be released, and built to be extendable, adaptable and customizable for different workflows and scenarios. With that in mind, OHIF would be our

[5] PostDICOM: https://www.postdicom.com/.
[6] Horos: https://horosproject.org/.
[7] Mango: https://ric.uthscsa.edu/mango/.
[8] Dicom Web Viewer: https://ivmartel.github.io/dwv/.

viewer component that will then be extended with the features required to integrate with external processing and analysis methods. In light of our choice, it is important to understand how OHIF is structured. To this effect, a synthesis of its main architectural aspects is covered in what follows.

4.2 OHIF Architecture

The OHIF project [3] is maintained as a mono-repo, mainly divided in two directories, with /platform containing the business logic, component, and application libraries that, combined, make this medical viewer and the/extensions directory containing different packages that can be registered in the core of the application at the/platform directory by an Extension Manager module. The business logic consists of pre-packaged features that are common to different medical image Viewers needs as Hotkeys, DICOM Web requests, hanging protocols, managing study's measurements and metadata and many more while being decoupled from any view library and rendering logic. It uses React as the front-end framework but it can be used with any other front-end frameworks. The extension layer serves as the ohif instance specification where the needed functionalities for specific uses, ui components and new behaviors are located. This layer allows to wrap and integrate functionalities of 3rd party dependencies in a reusable way, to change how application data is mapped and transformed, to display a consistent/cohesive UI and to inject custom components to override build-in components. Several extensions are already maintained by the Viewer itself i.e.. Cornerstone.js[9], a JavaScript library for displaying medical images in modern web browsers and VTK.js, another browser-based medical image framework which supports 3D Images and advanced renderings like multi-planar reconstruction.

4.3 Overall System Architecture

Figure 3 depicts the overall components of the architecture devised for the platform. At its core, it has the OHIF viewer. Since OHIF has a flexible policy with which data archive to choose to inter-operate with. Any data archive can be integrated with the viewer. As a PACS solution, we adopt the dcm4chee-arc-lite DICOM archive running in a Docker container connected to the node.js application of the viewer. For managing the integration of external processing methods from custom image processing servers, represented in Fig. 3, and without loss of generality, by a Flask[10] server, two OHIF extensions are added to the OHIF viewer for the communication with the server. Extensions in OHIF are the considered "plug-ins" in this project's context. The *Data Selector and Sender* extension module filters data from the retrieved DICOM data, transforms it in a way to be sent to the image processing server with a POST Request (via XmlHttprRequest). The *DisplaySet Builder* module is responsible for taking the

[9] Cornerstone.js: https://www.cornerstonejs.org/.
[10] Flask: https://flask.palletsprojects.com/en/2.2.x/.

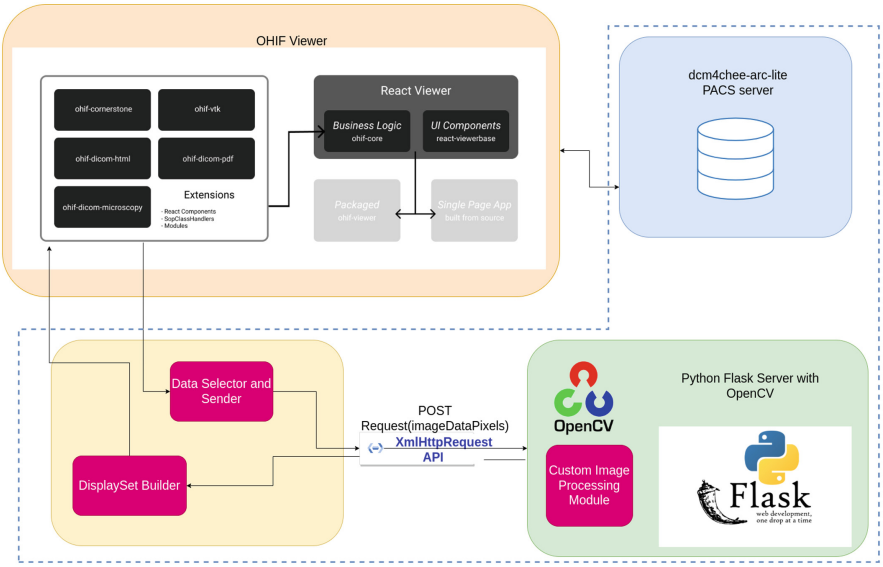

Fig. 3. Overall System Architecture. Adopting OHIF at its core, several components are considered to extend functionality towards the desired requirements. The components inside the dashed region result from the proposals in this work.

processing outcome of the external method and making it available for inspection, on the viewer. Those two modules are triggered by a new button added in the Viewer's toolbar.

4.4 Proof-of-Concept Instantiation

To test the current stage of the proof of concept we adopted a simple use case entailing opening a CT exam from a PACS server, applying a Gaussian smoothing using a method deployed in the cloud and visualizing the outcomes of this processing back in the image viewer. This would allow demonstrating the feasibility of our proposal by instantiating it for a particular purpose. In what follows we provide a brief description of the concrete roles played by each of the contributed modules in this use case scenario.

PACS Server. As said before, Dcm4chee Dicom was the chosen DICOM archive. It was easily integrated with OHIF since there is already a DICOM Query retrieval component which was configured to point to the new image archive. From that step on, it was then possible to browse patients and their studies stored at the PACS server.

Data Selector and Sender retrieves the pixel data of the current image by using the image loader provided by the cornerstone.js extension CornerstoneWadoImageLoader, which loads an image object using its imageID URL scheme. At its current stage of development, OHIF supports reading 3D exams, but an iteration through the slices of the CT exam is needed to load the full volume and populate a three-dimensional array with the data loaded by *cornerstone*. Having a three-dimensional structure with the pixel data of the volume, this module serializes it to a JSON object and sends it to the processing server via XmlHttpRequest.

DisplaySet Builder is responsible for the deserialization of data received from the processing server and loading the data to a three-dimensional structure which will feed a new ImageSet, the object responsible for organizing images in its respective series, listing them in the study list, and making them available for visualization in the image viewer.

Image Processing Method. As depicted in the architecture diagram, in Fig. 3, we adopted a Flask Server and developed a small Python script that uses openCV to apply a Gaussian smoothing filter. The server receives a POST request sent by the Data Selector and Sender module, loads the JSON object and consumes a *numpy* array with the data which will then be processed by a Gaussian filter implemented in OpenCV. After the processing is done, the processed array is serialized and sent back to the viewer as the POST response.

Overall Viewer Interface. Figure 4 shows different screenshots of the user interface from the listing of exams available at the PACS server, the overall disposition of elements on screen, multiplanar views and, in the last row, a sample image slice before and after the application of Gaussian smoothing – by integration with our custom module—to illustrate the successful implementation of the processing pipeline.

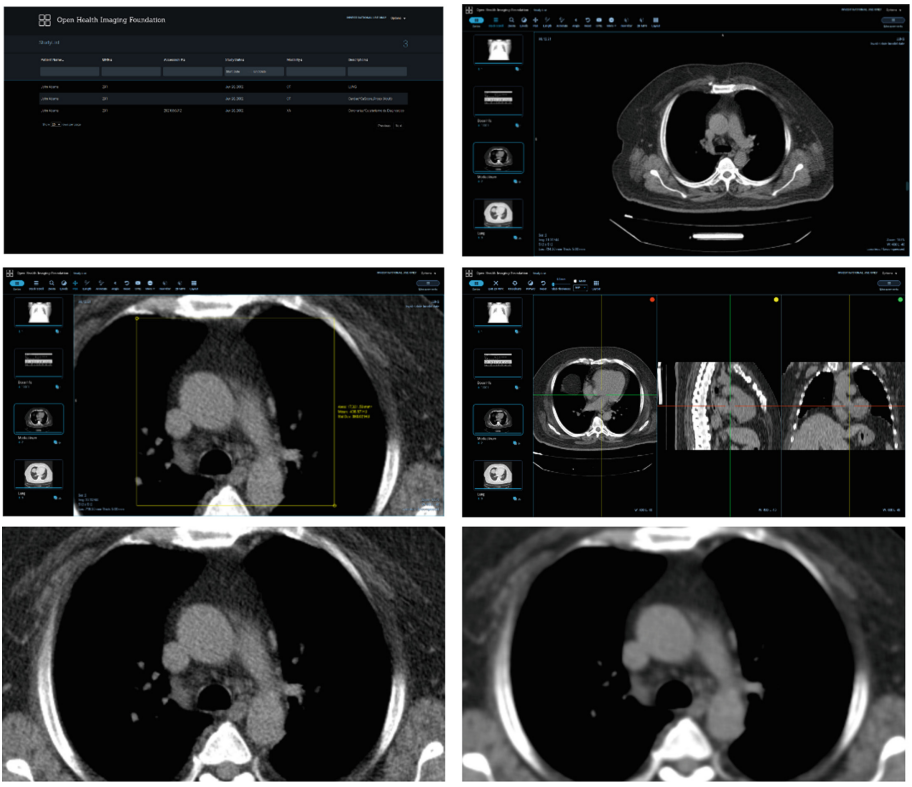

Fig. 4. Representative examples of the current state of the graphical user interface showing, from left to right, and top to bottom: (a) list of exams retrieved from the PACS server; (b) exam slice visualization; (c) region of interest selection; (d) orthogonal planes view; and (e) and (f) a detailed view of an exam before and after being smoothed through the implemented modules.

5 Conclusions

In light of challenges identified for improving how research on novel methods for cardiac analysis from CT can be deployed for evaluation and testing, in clinical environments, we identify a set of requirements that should contribute to address these challenges. The work presented here demonstrates the suitability of our proposal to support the application and development of novel diagnosis techniques for Cardiac CT, offering a modular solution for clinical analysis and at the same time being a shared environment between the researcher and the clinician. To this effect, we show that it is possible to profit on the OHIF resources and develop custom modules to integrate processing features into the viewer. In this regard, the researcher has a layer of abstraction from the core system not limiting the technologies and frameworks used for development and, as so, skipping completely any stage of adaption or learning curve of the system that

would be involved. From the clinician perspective, while our proof-of-concept can still evolve, a basic set of common tools is already showcased and should provide the grounds for addressing their common workflow while dealing with cardiac CT. The devised approach can be accessed through any device or platform and its modular nature allows offloading the computationally intensive processing to the cloud, enabling the use of smaller devices, such as tablets.

The current stage of our proof-of-concept opens routes for several advances in two main fronts. The chosen core framework, the OHIF viewer, is in constant development with many new features on the horizon to be released, such as segmentation support, DICOM image upload to PACS and MPR view with the integration of Cornerstone3D.js. Since our approach involved proposing extensions to the OHIF framework, and no core code has been touched, we can easily profit from its evolution.

Additionally, this proof-of-concept opens routes for further exploring the development and integration of new functionalities, the most prominent being: (a) the integration of (semi-)automated segmentation of the left ventricle, a central anatomical landmark for the processing and analysis of these exams; (b) a multi-profile feature so that different clinicians can access a varying set of functionalities of the viewer depending, for instance, on the device (e.g., less features on a tablet); and (c) deal with processing outputs that are not an image volume (as in our use case), e.g., a polygonal mesh generated from an image.

Acknowledgements. This research is supported by National Funds, through FCT, in the scope of project CAD-FACTS (PTDC/EMD-EMD/0980/2020) and by IEETA - Institute of Electronics and Informatics Engineering of Aveiro Research Unit funding (UIDB/00127/2020).

References

1. Chang, H.J., et al.: Selective referral using CCTA versus direct referral for individuals referred to invasive coronary angiography for suspected CAD: a randomized, controlled, open-label trial. JACC: Cardiovasc. Imaging **12**(7 Part 2), 1303–1312 (2019)
2. Danad, I., Szymonifka, J., Schulman-Marcus, J., Min, J.K.: Static and dynamic assessment of myocardial perfusion by computed tomography. Eur. Heart J. - Cardiovasc. Imaging **17**(8), 836–844 (2016). https://doi.org/10.1093/ehjci/jew044
3. Erik Z., et al.: Open health imaging foundation viewer: an extensible open-source framework for building web-based imaging applications to support cancer research. JCO Clin. Cancer Inf. **4**, 336-345. https://doi.org/10.1200/CCI.19.00131. https://github.com/OHIF/Viewers
4. Hoffmann, U., Ferencik, M., Cury, R.C., Pena, A.J.: Coronary CT angiography
5. Ince, D.C., Hatton, L., Graham-Cumming, J.: The case for open computer programs. Nature **482**(7386), 485–488 (2012)
6. Nieman, K., Balla, S.: Dynamic CT myocardial perfusion imaging
7. Marano, R., et al.: CCTA in the diagnosis of coronary artery disease. Radiol. Med. (Torino) **125**(11), 1102–1113 (2020)
8. Mu, D., et al.: Calcium scoring at coronary CT angiography using deep learning. Radiology **302**(2), 309–316 (2022)

9. Noack, P., Jang, K.H., Moore, J.A., Goldberg, R., Poon, M., et al.: Computer-aided analysis of 64-and 320-slice coronary computed tomography angiography: a comparison with expert human interpretation. Int. J. Cardiovasc. Imaging **34**(9), 1473–1483 (2018)
10. Nolden, M., et al.: The medical imaging interaction toolkit: challenges and advances. Int. J. Comput. Assist. Radiol. Surg. **8**(4), 607–620 (2013)
11. Nowbar, A.N., Gitto, M., Howard, J.P., Francis, D.P., Al-Lamee, R.: Mortality from ischemic heart disease: analysis of data from the world health organization and coronary artery disease risk factors from NCD risk factor collaboration. Circ.: Cardiovasc. Qual. Outcomes **12**(6), e005375 (2019)
12. Rankin, S.: CT angiography. Eur. Radiol. **9**(2), 297–310 (1999)
13. Roduit, N.: Weasis medical viewer : Weasis documentation (2021). https://nroduit.github.io/en/
14. Roth, G.A., Mensah, G.A., et al.: Global burden of cardiovascular diseases and risk factors, 1990–2013;2019. J. Am. Coll. Cardiol. **76**(25), 2982–3021 (2020). https://doi.org/10.1016/j.jacc.2020.11.010
15. Sun, W., Zhang, P., Wang, Z., Li, D.: Prediction of cardiovascular diseases based on machine learning. ASP Trans. Internet Things **1**(1), 30–35 (2021)
16. Tzoulaki, I., Elliott, P., Kontis, V., Ezzati, M.: Worldwide exposures to cardiovascular risk factors and associated health effects: current knowledge and data gaps. Circulation **133**(23), 2314–2333 (2016)
17. Van Assen, M., et al.: Prognostic value of CT myocardial perfusion imaging and CT-derived fractional flow reserve for major adverse cardiac events in patients with coronary artery disease. J. Cardiovasc. Comput. Tomogr. **13**(3), 26–33 (2019)
18. Varga-Szemes, A., Meinel, F.G., Cecco, C.N.D., Fuller, S.R., Bayer, R.R., Schoepf, U.J.: CT myocardial perfusion imaging. Am. J. Roentgenol. **204**(3), 487–497 (2015). https://doi.org/10.2214/ajr.14.13546
19. World-Health-Organization: Cardiovascular diseases (CVDs) (2021). www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds)