Quality assurance framework for rapid automatic analysis deployment in medical imaging

Peltonen, Juha I.; Honkanen, Ari-Pekka; Kortesniemi, Mika

2023-12

Associazione Italiana di Fisica Medica

http://hdl.handle.net/10138/568417

Peltonen, J I, Honkanen, A-P & Kortesniemi, M 2023, 'Quality assurance framework for rapid automatic analysis deployment in medical imaging', Physica Medica, vol. 116, 103173. https://doi.org/10.1016/j.ejmp.2023.103173

Physica Medica 116 (2023) 103173

Contents lists available at ScienceDirect

Physica Medica

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

Quality assurance framework for rapid automatic analysis deployment in medical imaging

Juha I. Peltonen *, Ari-Pekka Honkanen, Mika Kortesniemi

HUS Medical Imaging Center, Helsinki University Hospital and University of Helsinki, Helsinki, Finland

ARTICLE INFO

Keywords:
Quality assurance
Medical imaging
Framework

ABSTRACT

Purpose: Automatic image analysis algorithms have an increasing role in clinical quality assurance (QA) in medical imaging. Although the implementation of QA calculation algorithms may be straightforward at the development level, actual deployment of a new method to clinical routine may require substantial additional effort from supporting services. We sought to develop a multimodal system that enables rapid implementation of new QA analysis methods in clinical practice.

Methods: The QA system was built using freely available open-source software libraries. The included features were results database, database interface, interactive user interface, e-mail error dispatcher, data processing backend, and DICOM server. An in-house database interface was built, providing the developers of analyses with simple access to the results database. An open-source DICOM server was used for image traffic and automatic initiation of modality-specific QA image analyses.

Results: The QA framework enabled rapid adaptation of new analysis methods to automatic image processing workflows. The system provided online data review via an easily accessible user interface. In case of deviations, the system supported simultaneous review of the results for the user and QA expert to trigger corrective actions. In particular, embedded error thresholds, trend analyses, and error-feedback channels were provided to facilitate continuous monitoring and to enable pre-emptive corrective actions.

Conclusion: An effective and novel QA framework incorporating easy adaptation and scalability to automated image analysis methods was developed. The framework provides an efficient and responsive web-based tool to manage the normal operation, trends, errors, and abnormalities in medical image quality.

1. Introduction

Medical imaging systems provide essential information that influence clinical decisions and patient care outcomes. Therefore, outputs from medical imaging systems must be monitored via strict quality assurance (QA) procedures for possible degradation in technical performance. The rapidly increasing quantity and complexity of medical imaging requires corresponding advances in QA processes [1].

In general, QA involves the acquisition of daily, weekly, and monthly test images that are employed to detect abnormalities and trends in the performance of devices. In addition to rapid intervention in case of acute malfunctions, this widescale continuous QA monitoring also enables pre-emptive maintenance without delays caused by manual QA analysis and monitoring process. These goals can be achieved with an automatic image processing pipeline that can rapidly convert images to quantitative monitoring results, provide further analysis, and send notifications when needed. This can release QA specialist (typically a medical physicist) resources from manual analysis work and provide more tools to follow the underlying effects objectively. However, while development of calculation algorithms is feasible with existing methods, clinical implementation of the QA process may pose challenges.

In medical imaging, the structure of image data and the accompanying image transfer protocol is well established through the continuously updating DICOM standard. However, DICOM data management still requires expertise in practical clinical settings. More specifically, effective implementation of the image processing pipeline requires seamless integration of image data receiving and queueing. Proper management of image traffic may require technical software solutions that are much more complex than implementation of the actual image analysis. Additionally, embedding of custom analysis codes is currently a rare feature in commercial picture archive systems (PACS). Fortunately, a few open-source solutions are available to handle medical

* Corresponding author at: HUS Medical Imaging Center, P.O. Box 340, FI-00029 Helsinki, Finland.
E-mail address: juha.peltonen@hus.fi (J.I. Peltonen).

https://doi.org/10.1016/j.ejmp.2023.103173
Received 2 May 2023; Received in revised form 29 September 2023; Accepted 16 November 2023
Available online 24 November 2023

image traffic on a small scale [2–4], thus providing a building block to be combined with QA-specific analysis codes in a more comprehensive QA framework.

An efficient QA system should provide a sufficient amount of QA information in an easily comprehensible manner and the means to raise alerts when results are out of set tolerances. A clear and easy to use graphical interface is an integral component of such a system. Recently, many solutions have emerged that enable easier development of user interfaces following good practices in both usability and security. Namely, so-called frameworks allow seamless integration of data to a web-based interface, and well-defined style libraries allow usage and user-tailored views of data [5–7].

The development of a QA system requires considerable time and human resources. Open-source codes may improve resource use, as the same methods and algorithms can be shared by the entire professional community and different multidisciplinary shareholders. Further, open-source development may also provide a more robust platform where development-phase errors and bugs are collectively observed and corrected more efficiently than with singular team effort. Open-source solutions for implementing medical imaging analysis applications and workflows have been presented by a few authors [4,8–14].

The objective of this study was to develop a rapid automatic QA analysis system for medical imaging. At the technical level, all the required components supporting the core QA data analysis functionality were pursued. These included an image transfer node for routing the acquired QA data to the analysis; a developer interface for the QA database; a user interface to review analysed data and manipulate user-level QA system parameters; and a system to raise alerts. On a general level, the developed system was designed to be flexible in the required platform and connection methods, enabling user interaction through a web-based user interface.

2. Materials methods

The developed QA system consists of a DICOM-compatible image server handling the image traffic and queue for analysis, results database, database communication interface, user interface framework, and alert generation module. A simplified schema of the dataflow and the included modules is presented in Fig. 1. All components of the system were based on readily available open-source programs and frameworks, which were modified to match the typical medical imaging QA requirements. The key applications and libraries used are presented in Table 1. The components of the system are bundled as Docker containers [15], which allows them to be easily transferred to and run on different platforms, thus reducing the requirements of the host computer.

Table 1
Utilised open-source applications and libraries.

Feature	Used library or application	Reference
Image communications server	Orthanc	[3]
Database	PostgreSQL	[16]
Database interface	SQLAlchemy	[18]
Web framework	Django	[6]
User interface outlook	Bootstrap	[7]
Plotting	Bokeh	[19]
DICOM image viewer	Cornerstone	[4]

2.1. Image communications server

Orthanc [3] is an open-source DICOM server that was used to receive images and launch the QA data analyses. The server software is available as a Docker package, which allows wide and flexible customization by the user. The standard settings file enables the user to manipulate most of the typical medical image transfer-related parameters. An image viewer is embedded by default.

The most essential development feature of the DICOM server is the ability to embed Python code to interact within different phases of applied image traffic and an additional application programming interface (API) to retrieve and manipulate data in the server. These technical features allow the user to run in-house developed analysis code for the acquired QA data. The DICOM server also includes a queue system to automatically trigger analysis of image data in the receiving order to limit the load on the system and to provide continual online access of the QA results.

In our system, the analysis process was triggered when a study had not received new images in 10 s; the waiting period is freely adjustable in the settings of the DICOM server. When the analysis is triggered, the validity of the data is determined first. The methods for validity detection vary depending on the source data. However, typically in QA images, certain imaging parameters must match predetermined criteria. As an example case of CT QA images, the following image parameters were automatically checked for conformance with the criteria: the identity code of the study cannot be a valid national patient identification code and the slice number of the image must match the predefined device-specific setting.

In addition to the DICOM server, a file server was used to store analysed QA images. The server provided images for viewing in the user interface along with the accompanying analysis results. Additionally,

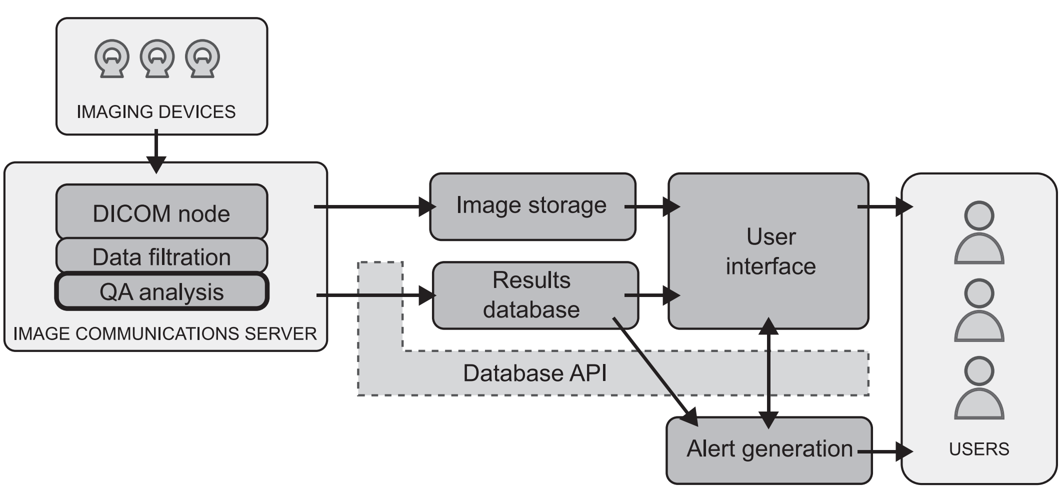

Fig. 1. A simplified schema of the modular design and data flow of the presented medical image QA analysis system. The system is built as a framework where a DICOM image server manages the image transfer and data queue for the actual QA analysis followed by updated results database. This further enables results review for the user, supported by error detection and alert functionalities.

the images in the file server could be used to update results if the actual analysis method is updated at some point.

2.2. Database

In the beginning of an image analysis project, the database solution used, and its structure often do not appear to be a major feature. It is likely that the analysis algorithm itself produces only a handful of results and source data are determined without accompanying meta-information. However, when the final algorithm is embedded in the healthcare provider-wide systems, the requirements of the data flow will increase substantially. The database must be able to handle multiple parallel read and write actions. In addition to the primary analysis results, the complementary information on how and where the source data were obtained must accompany the actual results. The solution for the large-scale database needs are the well-established relational databases. In our system, the PostgreSQL [16] database was chosen due to good documentation and its widely adopted position in the field.

The actual database structure is a key feature and largely defines how the entire system interacts with the data. In the database structure, all possible indications of use should be considered prospectively, since retrospective changes require system-level redesign and may become extremely difficult to apply. In our system, the design of the database was based on development sessions held with QA experts (medical physicists) responsible for different medical imaging modalities. In the sessions, the focus was on the typical needs in each medical imaging technique and the most relevant technical parameters needed in each application. As a result, a database structure with two interconnected branches was developed. The first branch was designed to reflect the imaging device and organizational information to group the devices accordingly. The second branch was dedicated to QA results to include information on performed measurements, accompanying analyses settings, and obtained results. The connection between each QA result to a certain imaging device and its location (across different radiology departments) over the device lifecycle allows for advanced solutions, such as determination of device-specific error boundaries and site-specific browsing of the results.

2.3. Database interface

While a well-defined database structure is essential for successful presentation of data, it may become too complicated for an analysis developer to follow. Thus, a communication interface was developed that provides an easy and well-defined connection to the database. The interface was designed to receive new data as a Pandas dataframe [17]. This includes a few mandatory fields to identify the data source and the actual measurement result fields. The interface itself handles recording of the results to the database under the correct location and device combination and updating the organization hierarchy if necessary. On the other hand, the communication interface can be used to interrogate the system to retrieve, for example, a result and corresponding error thresholds from a certain imaging device. The necessary database interaction functions were developed in-house, and actual communication was performed based on a SQLAlchemy [18] library.

2.4. User interface

The user interface design is often an overlooked feature in medical imaging analysis development. However, if these analysis methods are deployed in daily clinical use, it is essential to make the system approachable for all user groups independent of their understanding of the underlying technical methods. In the developed system, the well-established Django [6] framework was used to connect the user interface to the database and generate responsive web pages for interaction. An interactive web-based user interface is also a convenient way to achieve maximum cross-platform compatibility, as no additional installations beyond a normal web browser are required on client computers. An example of a user interface for computed tomography QA is presented in Fig. 2.

The Django framework used allows for extensive interaction with the database. Basically, all data input, retrieval, and manipulation can be performed with standard features of the framework. A subset of these features can be provided for a user through a graphical user interface. Typically, tools for error threshold updating, imaging device naming, datapoint commenting, and image data download are relevant features for end users. User groups with different levels of feature access are also enabled by the framework.

A graphical representation of the results was implemented using the Bokeh library [19]. The library allows for a seamless integration with the Django environment and can also support limited interactive features, such as datapoint selection with a mouse.

The appearance of the user interface webpages was defined through the Bootstrap library [7]. The same library is widely used in many well-known web services and is familiar for the typical user, thus increasing approachability.

Although presentation of medical DICOM images on a web page is not a widely supported option, this is one of the key elements in medical imaging user interfaces. Only a few open-source libraries are readily available [4,11,20]. In our system, Cornerstone [4] was used to present images interactively. The user can see relevant images with the standard zoom, drag, and windowing tools to manipulate image size, position, and contrast/brightness.

2.5. Alert generation

The alert generation in the system utilized the developed database interface. The interface provided tools to interrogate the latest results for each device with corresponding error thresholds. This information was used by a scheduled script to generate a list of new error incidences that can be sent to the respective QA specialists to trigger corrective actions as required. Optimally, the alert generation would be connected to an existing task managing system with an open interface. In our current solution, a list of new errors was sent daily by an automatically generated e-mail.

3. Results and discussion

The development of an automatic analysis pipeline for QA may provide a fundamental improvement to the efficiency of the QA process in medical imaging and the corresponding medical physicist daily routine and role as a QA supervisor. It can replace manual analyses and shift the QA expert's focus towards inspecting detected findings and solving problems on an actual physical and technical level. This allows the medical physicist to focus their professional effort on more effective utilization of their knowledge, skills, and competence. This aspect is even more important when considering the increasing need for medical physicist involvement in diagnostic and interventional radiology [21].

Automatic analysis also has other fundamental strengths regarding robustness, objectivity, consistency, and reproducibility of the analysis results. An essential benefit is to increase the sensitivity of the analysis due to increased repeatability of the analysis process. With demonstrative data visualization, the same results can be simultaneously shared for all users involved in imaging. Shared data allows for effective discussion on the detected issues. However, even the best analysis tools can be hampered by limited usability and data input, which are not the focus of the methodology development itself. In the presented system, we have addressed this problem by implementing all required supporting features around the actual QA analysis algorithm.

Overall, a QA analysis service application, such as the one developed in this study, provides an accessible, rapid, and well-defined deployment of image analysis in the practical clinical workflow. In contrast to research-oriented medical imaging departments, where development

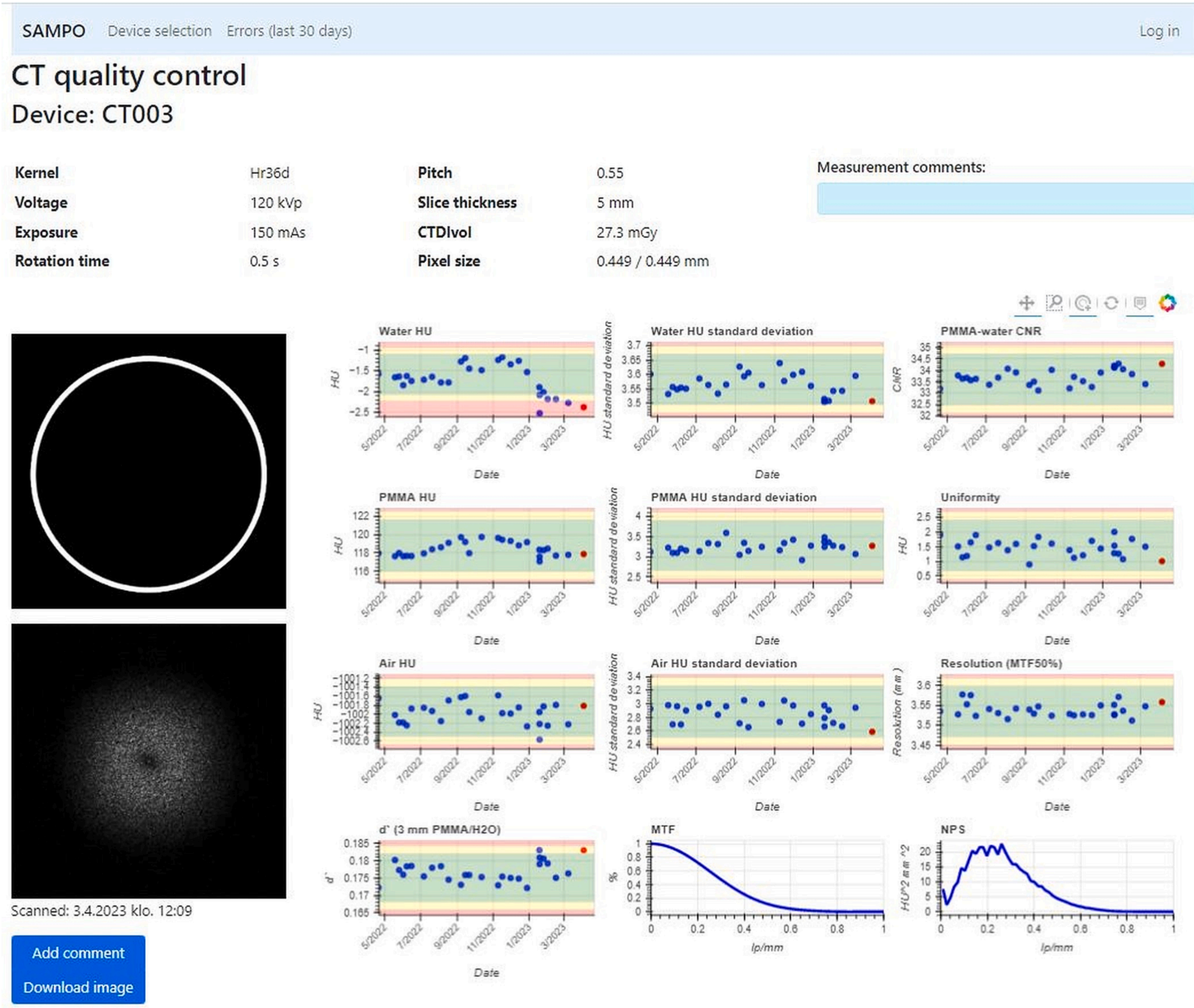

Fig. 2. The user interface used in computed tomography quality assurance. The user can choose a datapoint by clicking on any of the timeseries. The respective source image is presented in a viewport on the upper left along with 2D noise power spectrum (NPS) on the lower left. In our case, the analysed QA parameters included various basic image-quality parameters (CT numbers of basic materials, noise, contrast, uniformity, and resolution values) and expand to task-specific image quality implemented as detectability index (non-prewhitening model observer based on 3-mm round nodule model with PMMA/water contrast corresponding to soft-tissue lesion, applying corresponding radially averaged MTF and NPS curves).

work for new analyses is typically performed by graduate students or other temporary research workers as a specific project, the developed system allows such projects to focus on actual analysis methods and their implementation without an effort to encode auxiliary software features. This is an essential feature, since insufficient compatibility of the image analysis process to the clinical workflow may become a major problem that prevents the effective use of the analysis application.

The presented system is also versatile and can be applied to any medical imaging modality that provides digital images. Arguably all current manually applied image-quality measurements can be automated with modern tools. However, the applicability of different imaging modalities varies. For example, computed tomography data are already almost platform independent and do not require major pre-processing effort in the analysis pipeline. On the other hand, ultrasound images carry many platform-dependent features that require additional developer effort to achieve standardized results.

The guideline for system development in this project was to use well-established and public frameworks and libraries for the image analysis process. The availability of such open-source solutions has increased dramatically in the last decade. While software development originally required comprehensive understanding of the underlying hardware, modern tools have distanced typical developers further from underlying mechanisms. Currently, what was previously known as programming is increasingly superseded by effectively combining existing application modules to build more comprehensive systems and functions. Thus, while actual programming is still valid when optimizing calculation-intensive algorithms, supporting services can be effectively out-sourced. Although these applications are suitable for general use to provide appropriate services, they are rarely developed specifically for medical imaging. However, they are flexible and well tested platforms, making them highly valuable also for medical imaging implementations.

In the development and design phase of any practical system, the user perspective should prioritized to the practical experience and knowledge of the users regarding the preferred functionality in a daily workflow. The project for the system presented here was initiated by multiple key user planning meetings. In these meetings, the required features and functionalities were defined extending to the required data structures. While this method was successful, the possibility to use additional professional-level usability expertise from the start could have been beneficial. Such an expert could encourage key users to formulate their desired software requirements, which would likely increase development speed.

The technical solutions behind the image analyses are typically beyond the knowledge base of a typical heterogeneous user group. Thus, the design of the user interface must simplify complex algorithms to a form where the user can extract essential information with a single glance. For example, corrective action required for specific equipment based on a recent measurement result must be clearly indicated. Due to the variability of the relevant types of monitoring data, the applied user interface must provide of broad range of visualisations and reporting formats. Additionally, interactive features are valuable practical features to provide the user with the possibility to select, compare, and troubleshoot the QA results.

While it is possible to develop a universal user interface to present all results in a standardized format, this is often not desired as there are differences in how the data should be presented between medical imaging modalities. For example, sometimes only the time series is a subject of interest, while in other cases, a single datapoint requires more precise inspection. Thus, instead of a universal but rigid solution it is more useful to use a more flexible modular approach, where the initial deployment can be achieved rapidly and allows further application-specific fine-tuning to achieve optimal usability and relevant results.

In the developed system, the interaction between the user interface and the database is not synchronized. Therefore, the results presented on each web page are uploaded once and are not updated before the complete web page is uploaded again. Usage is restricted to limited interactive features by shaping the loaded data through a sub-application on the web page. As a more demanding alternative, it would also be possible to develop a web page completely synchronized with the database and provide more interface features online. However, this would require the use of a more complex backend framework and advanced skillset regarding the development and maintenance of the system.

We have presented the developed system in the context of medical QA image analysis. However, image data is not the only relevant data format in medical or even radiology QA. Due to compatibility or applicability, it may be preferable to present data in spreadsheet format. Furthermore, the data can be related to a mechanical inspection of an imaging device or measured with an additional device. In these cases, it may be applicable to incorporate extra data inputs for database insertion. In addition to the provided database interface, these inputs often require additional data parsing and a conversion layer attached to the presented system.

There are also use cases where there could be a manual data manipulation stage in addition to automatically generated results, such as manually inspecting the region of interest delineations. Additionally, there may be a need to insert subjective quality grading in addition to automatic results. In such an application, images could be sent through an automatic analysis pipeline while a separate image review is provided to obtain subjective impressions on image quality with a predefined grading scale.

The system presented here requires user expertise on medical image transfer and structure. The image routing from the scanner to the analysis server often requires specific connection node settings and separately configured auto routing if PACS is used as an image relay. In most cases, QA data analysis and image processing uses DICOM header information. While this information is generally provided in standardized format, new scanner software versions may require additional DICOM data validation and adaptation to reach compatibility with the applied analysis. Thus, the maintenance events and related updates of the system must be acknowledged.

While the initial introduction of an automated system can replace manual analyses, the digitalization of the process also enables more ambitious possibilities. The increasing amount of data enables more data-centric solutions where e.g. statistical analysis combined with artificial intelligence may allow for a more multidimensional approach to detect abnormal behaviour of imaging equipment or a specific technique. Additionally, technical QA data can be combined with other data types, such as clinical patient dose information, to gain a deeper understanding of the detected QA results and their relationship to clinical effects.

The availability of effective and free development tools is likely to drive a rapid increase in projects similar to those presented here and by other authors [4,13,14]. Cross-institutional cooperation use may face certain restrictions at least partly driven by medical records security requirements and European General Data Protection Regulation (GDPR). However, development of dedicated open-source medical image analysis tools would benefit considerably from cooperation in the form of multi-organizational joint effort instead of multiple small projects with similar aims. A culture of open collaboration, code sharing, and system development should be strongly encouraged.

4. Conclusions

We have developed a system that provides services for automated medical QA image analysis. The system presented here permits rapid deployment of new analysis methods, supports efficient communication on results achieved at a database level, and increases the sensitivity and effectiveness of the entire QA process. The system can be deployed independent of platform and is based on open-source software.

Funding

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] International Atomic Energy Agency. Handbook of Basic Quality Control Tests for Diagnostic Radiology. IAEA Human Health Series No. 47; 2023.
[2] Costa C, Ferreira C, Bastião L, Ribeiro L, Silva A, Oliveira JL. Dicoogle-an open source peer-to-peer PACS. J Digit Imaging 2011;24:848–56.
[3] Jodogne S. The Orthanc ecosystem for medical imaging. J Digit Imaging 2018;31:341–52.
[4] Ziegler E, Urban T, Brown D, Petts J, Pieper SD, Lewis R, et al. Open health imaging foundation viewer: an extensible open-source framework for building web-based imaging applications to support cancer research. JCO Clinical Cancer Informatics 2020;4:336–45.
[5] Grinberg M. Flask web development: developing web applications with python. O'Reilly Media, Inc; 2018.
[6] Django, https://djangoproject.com [accessed 28 April 2023].
[7] Bootstrap, http://getbootstrap.com [accessed 28 April 2023].
[8] Ratib O, Rosset A. Open-source software in medical imaging: development of OsiriX. Int J Comput Assist Radiol Surg 2006;1:187–96.
[9] Fouard C, Deram A, Keraval Y, Promayon E. CamiTK: a modular framework integrating visualization, image processing and biomechanical modeling. Soft Tissue Biomech Model Comput Assisted Surg 2012,:323–54.
[10] Urban T, Ziegler E, Lewis R, Hafey C, Sadow C, Van den Abbeele AD, et al. LesionTracker: extensible open-source zero-footprint web viewer for cancer imaging research and clinical trials. Cancer Res 2017:e119–22.
[11] Mantri M, Taran S, Sunder G. DICOM integration libraries for medical image interoperability: a technical review. IEEE Rev Biomed Eng 2020;15:247–59.
[12] Sohn JH, Chillakuru YR, Lee S, Lee AY, Kelil T, Hess CP, et al. An open-source, vender agnostic hardware and software pipeline for integration of artificial intelligence in radiology workflow. J Digit Imaging 2020;33:1041–6.
[13] Pemmaraju R, Minahan R, Wang E, Schadl K, Daldrup-Link H, Habte F. Web-based application for biomedical image registry, analysis, and translation (BiRAT). Tomography 2022;8:1453–62.

[14] Chen TT, Sun YC, Chu WC, Lien CY. BlueLight: an open source DICOM viewer using low-cost computation algorithm implemented with javascript using advanced medical imaging visualization. J Digit Imaging 2022;36:753–63.
[15] Merkel D. Docker: lightweight linux containers for consistent development and deployment. Linux j 2014;239(2):2.
[16] PostgreSQL, https://www.postgresql.org/ [accessed 28 April 2023].
[17] McKinney W. Data structures for statistical computing in python. InProceedings of the 9th Python in Science Conference 2010; 445:51-6).
[18] Bayer M. SQLAlchemy. In: Brown A and Wilson G, editors. The Architecture of Open Source Applications Volume II: Structure, Scale, and a Few More Fearless Hacks; 2012.
[19] Bokeh: Python library for interactive visualization. URL http://www.bokeh.pydata.org [accessed 28 April 2023].
[20] Jodogne S. Rendering Medical Images using WebAssembly. InBIOIMAGING 2022: 43–51.
[21] International Organization for Medical Physics. Medical Physics Staffing Needs in Diagnostic Imaging and Radionuclide Therapy: An Activity Based Approach. IAEA human health reports No. 15; 2018.