Performance of DICOM Data De-identification Process in a Single Board Computer

K Y E Aryanto¹, A A G Y Paramartha

Computer Science, Department of Informatics, Faculty of Engineering and Vocational, Universitas Pendidikan Ganesha, Singaraja, Bali, 81116, Indonesia

E-mail: ¹yota.ernanda@undiksha.ac.id

Abstract. This work focused on testing the Raspberry Pi device's performance with the MIRC CTP application, developed by RSNA. The application could de-identify medical image data. The medical images tested in this research were using the DICOM standard. The results have shown that the average data processing time for 100 DICOM files in CT modality was roughly 0.5 seconds per images. Meanwhile, the tests obtained an average of three images per seconds when more extensive data with average size of 132 KB were transferred, using 1000, 5000, and 10000 files, respectively. It was shown that the installation of the RSNA CTP in the Raspberry Pi device is not complicated. Furthermore, the transfer results demonstrated that the device could be used in the medical image data de-identification, specifically in image data sharing.

1. Introduction

Medical data sharing has become very common and widely conducted by various health institutions in recent years due to its benefit in advancing research or improving healthcare services [1]. In research, data sharing is recognized to be assuring scientific validity [2]. Meanwhile, sharing data is of great potential to improve public health by providing more objective evidence regarding patient safety and effectiveness [3].

Sharing such data raises awareness of what type of data or information could also be transferred and the medical data itself. In medical images, sensitive data can be embedded inside the images. The data could be in the form of text metadata in the file header or burnt in as pixels within the images. Thus, efforts to remove or modify such sensitive information should be made.

Removing original information that is considered sensitive, contained in the medical image data is known as data de-identification. It is mainly recognized in two types, namely anonymization and pseudonymization. The first one removes the intended data entirely, while the latter is done by replacing the targetted data with some codes generated by the related party. Both processes could deliver anonymity to the data so that the data owner's identification will be difficult or even impossible to do.

Many kinds of research had been done to provide attempts to de-identifying medical image data. Many applications were developed as well to supports the effort. One of the widely used application is the Medical Imaging Resource Center - Clinical Trial Processor (MIRC CTP) [4], developed by the Radiology Society of North America (RSNA) [5]. It is one of the applications that can de-identify metadata of the medical imaging file, in the format of Digital Imaging and Communication in Medicine (DICOM) standard [6], with a very high success rate [7] even when it was set in its default settings. The de-identification process was facilitated in a service that could be called through its pipeline.
configuration. MIRC CTP was explored in various research or projects, as a single processor application or bigger systems [8]–[11]. The application or the system was reported to be performed well when running under a personal computer or higher specification.

In recent years, computers in small sizes had been massively developed. One such device that has gained its popularity is Raspberry Pi [12]. It has been implemented in various medical fields research, resulting in a good performance of handling sophisticated processes. An assessment of the feasibility of running MIRC on the Raspberry Pi device was done by Pereira et al. [13]. However, the MIRC's performance in de-identifying DICOM data was not described since the work was mainly focused on delivering the requirements of a teaching file system. The single-board computer is practically cheap. The latest version, Raspberry Pi 4, are in the market for as low as US$ 35. It comes in the form of a credit card-sized computer [14]. However, the Raspberry Pi offered various features close to a standard personal computer's ability, and it is easy to use. Therefore, it should be able to process the data de-identification as well. In this work, an observation of the MIRC CTP application's ability installed in the Raspberry Pi device to perform the de-identification processes was conducted.

2. Materials and Method

2.1. MIRC CTP

The RSNA MIRC was published in the early 2000s as a software tool for transferring, processing, sending, and storing medical images for the radiology community. It was developed using Java code. The early version of MIRC required the installation of Apache Tomcat as the HTTP web server environment. In a later development, it was simplified, and therefore the requirement of Apache Tomcat is no longer remains. Furthermore, with the increasing acceptance and implementation of medical image data sharing, the MIRC was split into two separate applications: MIRC Teaching File System (TFS) and MIRC Clinical Trials Processor (CTP).

The MIRC CTP application is a standalone and open-source application developed by researchers and health institutions to exchange images and documents from clinical trial image data [15]. This application has several main features that are claimed to make it easier for users to process the exchanged files in clinical trials or research, including medical image data. Apart from being easy to install on computer devices, several processing features were developed to perform several parallel processes in each pipeline with different settings for each. Therefore, processing and data transfer can be carried out simultaneously through several available protocols [4]. Data transfer can be done using several protocols, including HTTP, secure HTTP, and DICOM.

MIRC CTP was developed based on the Java platform so that this application can be installed under a variety of supporting operating systems such as Microsoft Windows and Linux. Thus, this application should also run under an operating system that is a derivative of such an extensive operating system such as Raspbian [12] or Windows 10 IoT Core [16] as an operating system installed on a Raspberry Pi device. Furthermore, MIRC CTP is an open-source application so that it can be easily modified and integrated according to institutional user needs.

MIRC CTP, apart from being used to transfer data, is also used to process medical image data using the DICOM standard, which consists of a header and a scanned image itself. There is metadata in the DICOM file header containing all data related to medical images in the file, including patient data, patient families, related medical personnel, health institutions, time and image capture devices, and others. This part is an essential concern in the data sharing process because applicable regulations, such as the GDPR in the European Union (EU) [17] and HIPAA in the United States (US) [18], have indeed obliged data collectors to maintain the confidentiality of the data and do not share it with other parties.

One of the MIRC CTP application features is de-identification, a process to remove or encode patient data in medical files so that they do not refer directly to the patient concerned. In this case, the data contained in the tags in the DICOM file's header are deleted or modified in such away. Various functions have been provided to do this, such as deleting, modifying, saving, encoding, and so forth.
2.2. DICOM Standard

DICOM is one of the standards used globally for medical images and other related information [6]. This standard was developed to support the interoperability of various systems that generally involve retrieving, storing, displaying, transmitting, processing, or receiving medical image data. This standard itself is widely used because of several main benefits such as free standards that are not specific to a particular vendor, making it easier for medical images to be stored and operated in or exchanged between systems, and their development is still being actively carried out to meet technological developments and medical imaging needs.

The DICOM standard medical image file structure consists of two main elements, namely the header and the image data set. The header section contains data or information relating to the file itself. Meanwhile, the image data set part contains several data elements related to the stored image. These data elements include various metadata containing important data or information related to the image contained in them. The data or information can be in the form of data associated with the identity of patients, institutions, medical personnel, studies, series, modalities, image formats, pixel attributes, and others.

There are two types of elements or attributes, in a DICOM file, including standard data elements and private data elements. These elements can generally be written as two groups of numbers (xxxx, yyyy) where xxxx and yyyy are in hexadecimal number format, respectively. The first four numbers (xxxx) are generally referred to as the group number, while the rest is the element number. Standard data elements will have even group numbers, while private data elements will have odd group numbers.

Images stored using the DICOM standard can come from a variety of modalities. There are more than 50 types of modalities that can be stored in the DICOM standard, which include Computed Radiography (CR), Computed Tomography (CT), Magnetic Resonance (MR), Positron emission tomography (PET / PT), X-Ray Angiography (XA), and Ultrasound (US). The resulting file size varies considerably between each of these modalities. CT, MR, and the US scanned files are on average in the range of 100 to 600 kilobytes (KB), while images of modalities such as Mammography (MG) and CR can reach file sizes in the range of 27 to 30 megabytes (MB) per image [19].

This standard has been widely implemented in radiology, which allows the process of digitizing workflows to replace scans with final physical outputs. Furthermore, all the modalities used can be retrieved, stored, and distributed in a single system that utilizes existing network technologies, including internet technology. DICOM has made it possible to develop applications that connect all lines and update both doctors' and patients' communication process.

2.3. Method

This study focuses on the data de-identification process carried out using the MIRC CTP application, installed on the Raspberry Pi 4 device. The data used is medical image data using the DICOM standard. Data is public data obtained through the Kaggle.com website. There are no restrictions regarding file size; however, the size used is in the range of 132-512 KiloBytes (KB). This research was conducted in 2 stages, namely: (1) installation of the MIRC CTP application on the Raspberry Pi 4 device, followed by setting up the device and testing to ensure that the "anonymizer" feature can function properly; and (2) Testing the "anonymizer" feature in de-identifying several medical image data. The numbers used in this work are 1000, 5000, and 10000 medical image files, respectively. Observations were made of the time required to carry out the de-identification process, the de-identification process's success, as seen from the absence of quarantined files. The de-identification scheme used is to change the DICOM file header tag's value, which is "de-identified".

Installation was done by first looking at the needs of the CTP application itself. After the module or library of the application has been successfully installed, the CTP application is installed into the device according to the operating system's installation mode. In this study, the Ubuntu operating system was used in this work. The tuning is done by adding a processing feature known as an "anonymizer". The processes conducted in this work can be seen in figure 1.
3. Results and Discussion
The implementation of this study follows the method described in the previous section. The method implemented in this research is divided into 4 (four) steps, including: (1) problem analysis and installation, (2) setting up limited scope of devices, applications, and infrastructure, (3) data de-identification trials, and (4) analysis of results.

3.1. System Installation
The MIRC CTP application can be downloaded via the website http://mirc.rsna.org. The installation process requires a GUI-based display, where accessing the downloaded files will result in the process of extracting files to the CTP folder. Due to the focus of the process that does not involve many features of the operating system itself, the Ubuntu Server operating system is used in this study. Thus, the extraction process can be carried out on another machine with a GUI-based window display and then transferred to the Raspberry Pi device.

This application is based on the Java programming language. Therefore, it requires the Java running environment (JRE) package on the operating system used. JRE can be provided by installing the appropriate application package. In running the application for the first time, a problem was encountered regarding the Java Imageio library version stored in the imageio folder. However, the library is only used if there is the processing of the image portion of the DICOM file, while in this work, it solely focuses on the header file. Thus, the library can be removed, and after that, the application can run as usual. As for running the application, the file that must be executed is “Runner.jar” with the appropriate command. The system then can be controlled through a website provided by the MIRC CTP itself.

3.2. System Settings
All of the pipelines and containing services should be determined before the application is executed. The setting can be configured following the format stated on the documentation page of the MIRC CTP. In this study, only a de-identification service was required. Therefore, the only setup that should be prepared is where the "DicomAnonymizer" was selected.

There are two ways of configuring the setup of the MIRC CTP. First, the configuration can be done using the CTP DICOM Anonymizer setup format, as mentioned in the documentation page of MIRC CTP. The step should be applied to the "config.xml" file. The example of a simple setup used in this work can be seen in figure 2.

The other way, and the more straightforward way of setting up the required services in MIRC CTP, is through the MIRC CTP launcher program’s configuration tab. However, the application required a GUI to implement this option, which is not supported by Ubuntu Server by default. The "trick" is to set it up using another machine with a GUI-based window and transfer the XML file to the Raspberry Pi 4.
device's installation folder. The steps of configuring pipelines and services using the GUI-based launcher of the MIRC CTP are shown in figure 3.

3.3. De-identification Process

After the required settings were all configured, 100 for functional testing, followed by 1000, 5000, and 10000 data was conducted. The transfer of 100 data was done using a set of DICOM data with the size of each image was approximately 170 KB. From the test, it was found that the overall process required to de-identify 100 DICOM files were nearly 1 MegaByte per second (MBps) or approximately 7.84 Megabits per second (Mbps). It was equal to 2 images per second was obtained, as seen in table 1.

| Feature            | Size   | Unit       |
|--------------------|--------|------------|
| Total Data         | 100    | Images     |
| Average image size | 512    | KB         |
| Total image size   | 51.2   | MB         |
| Total Time         | 51     | Seconds    |
| Average            | 7.84   | Mbps       |

The results shown above were roughly half of the transfer process's speed obtained in work by Aryanto et al. [20]. The work was using 25,000 images through a complete process of receiving,
processing, and storing images. However, the work was done using an intranet scheme, which means the higher specification of devices and networks were used.

The processing of three datasets was done using images that were available freely on the internet. The DICOM file's average size was 132 KB. The results showed that an average of four images per second was processed through the provided service. There was a slight drop in processing the DICOM images when more images were transferred. The results of the data processing of the three datasets are shown in the table 2.

| Feature                      | Test #1 | Test #2 | Test #3 |
|------------------------------|---------|---------|---------|
| Total Data (images)          | 1000    | 5000    | 10000   |
| Average of image size (KB)   | 132     | 132     | 132     |
| Total image size (KB)        | 135,141,662 | 676,816,698 | 1,353,633,396 |
| Total Time (seconds)         | 238     | 1357    | 3038    |
| Average (Images/seconds)     | 4       | 3       | 3       |

The table 2 shows that the amount of data between test #1, test #2, and test #3 did not significantly differ. The results indicate consistency in data processing using one modality type, as seen from the file sizes. The results were comparable in the number of images processed per second compared to the previous other work. However, the difference in image size or modalities may influence the processing speed as well.

The decrease of the average processing time may be caused by the amount of queue processing in the pipeline that was also significantly more than the test of 100 images data. More data to process means that more resources were required. Therefore, with the considerable amount of data in the queue list, it is expected to have a more significant drop of time necessary for the data processing.

4. Conclusion
A data processing regarding the de-identification of DICOM image data were conducted. The system consisted of a Raspberry Pi 4 device implemented with MIRC CTP to perform the intended task. Although the process of more data means consumed more time, the differences were not significantly high. The result shows that the Raspberry Pi can deliver de-identification strategies of DICOM images data and be used in DICOM image data sharing with the advantage of its portability.

Acknowledgments
The authors would like to thank the Study Program of Computer Science, the Department of Informatics, Universitas Pendidikan Ganesha, to conduct this research. The authors would also like to thank Ann for her editorial help in preparing the manuscript.

Funding
This work was supported by Universitas Pendidikan Ganesha under the DIPA BLU funding, grant number SP DIPA – 023.17.2.677530/2020 Revision 1 (March 16, 2020).

References
[1] Lo B 2015 Sharing clinical trial data: maximizing benefits, minimizing risk JAMA vol 313 no 8 pp 793–4 DOI: 10.1001/jama.2015.292
[2] Freymann J B, Kirby J S, Perry J H, Clunie D A and Jaffe C C 2012 Image data sharing for biomedical research—meeting HIPAA requirements for de-identification J. Digit. Imaging vol 25 no 1 pp 14–24 DOI: 10.1007/s10278-011-9422-x
[3] Lo B, Coetzee T and Demets D 2015 Committee on Strategies for Responsible Sharing of Clinical Trial Data. Board on Health Sciences Policy, Institute of Medicine. Sharing clinical trial data: Maximizing Benefits, Minimizing Risk (Washington: National Academies Press)

[4] RSNA 2019 MIRC CTP Retrieved on 15 October 2020 from https://mircwiki.rsna.org/index.php?title=MIRC_CTP

[5] RSNA 2020 Radiological Society of North America Retrieved on 15 October 2020 from https://www.rsna.org/

[6] National Electrical Manufacturers Association (NEMA) 2020 DICOM Standard Retrieved on 15 October 2020 from https://www.dicomstandard.org/

[7] Aryanto K Y E, Oudkerk M and van Ooijen P M A 2015 Free DICOM de-identification tools in clinical research: functioning and safety of patient privacy Eur. Radiol. vol 25 no 12 pp 3685–95 DOI: 10.1007/s00330-015-3794-0

[8] Clark K et al 2013 The Cancer Imaging Archive (TCIA): maintaining and operating a public information repository J. Digit. Imaging vol 26 no 6 pp 1045–57

[9] Erickson B J, Fajnwaks P, Langer S G and Perry J 2014 Multisite image data collection and management using the RSNA image sharing network Transl. Oncol. vol 7 no 1 pp 36–9 DOI: 10.1593/tlo.13799

[10] Chatterjee A R et al 2017 Image sharing in radiology—a primer Acad. Radiol. vol 24 no 3 pp 286–94 DOI: 10.1016/j.acra.2016.12.002

[11] Caturla S I et al 2018 Integrating information in radiology: Teaching File System (TFS) implementation in a third-level hospital using MIRC ECR 2018

[12] Raspberry Pi Foundation 2020 Raspbian Retrieved on 27 October 2020 from https://www.raspbian.org/

[13] Pereira A, Atri M, Rogalla P, Huynh T and M E O'Malley 2015 Assessment of feasibility of running RSNA’s MIRC on a raspberry pi: a cost-effective solution for teaching files in radiology Int. J. Comput. Assist. Radiol. Surg. vol 10 no 11 pp 1793–801 DOI: 10.1007/s11548-014-1143-4

[14] Raspberry Pi Foundation 2020 Raspberry Pi 4 Model B – Raspberry Pi Retrieved on 15 October 2020 from https://www.raspberrypi.org/products/raspberry-pi-4-model-b/

[15] RSNA 2020 Imaging research tools Retrieved on 15 October 2020 from https://www.rsna.org/research/imaging-research-tools

[16] Microsoft 2020 Windows for Internet of Things Retrieved on 15 October 2020 from https://developer.microsoft.com/en-us/windows/iot/

[17] EC Directorate-General for Communication 2018 Data Protection in the EU Retrieved on 15 October 2020 from https://ec.europa.eu/info/law/law-topic/data-protection/data-protection-eu_en

[18] US Department of Health & Human Services 2020 Health Information Privacy Retrieved on 15 October 2020 from https://www.hhs.gov/hipaa/index.html

[19] Dandu R 2008 Storage media for computers in radiology Indian J. Radiol. Imaging vol 18 no 4 p 287 DOI: 10.4103/0971-3026.43838

[20] Aryanto K Y E, Broekema A, Oudkerk M and van Ooijen P M A 2012 Implementation of an anonymization tool for clinical trials using a clinical trial processor integrated with an existing trial patient data information system Eur. Radiol. vol 22 no 1 pp 144–51 DOI: 10.1007/s00330-011-2235-y