Master's thesis

2020

Sebastian Olafsson

Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Computer Science

NTNU
Norwegian University of Science and Technology

Sebastian Olafsson

Digital Twins for AI-based Medical Imaging

June 2020

NTNU
Norwegian University of Science and Technology

NTNU
Norwegian University of
Science and Technology

Digital Twins for AI-based Medical Imaging

Sebastian Olafsson

Informatics
Submission date: June 2020
Supervisor: Frank Lindseth

Norwegian University of Science and Technology
Department of Computer Science

Abstract

Artificial Intelligence (AI) in the medical field of radiology has advanced so far in the recent years that clinical application of AI assisted diagnosis has become a question of time, not of possibility. The common denominator found in research papers on radiology and AI is a challenge especially difficult for medical imaging: scarce availability of annotated data sets due to privacy and economic factors, as field specific expertise and significant time is required. Data sets are necessary for training AI models capable of performing operations such as medical image classification and segmentation. This thesis researches a solution for developing data sets by investigating a Digital Twin ecosystem where every citizen has an online twin in the cloud that medical images automatically upload to. Citizens will have the option to donate their data for the development of AI models, increasing the amount of images that can be used for data set generation. This thesis aims to investigate to what extent a Digital Twin ecosystem combined with modern enterprise software solutions such as the NVIDIA Clara suite can contribute to the generation of high quality annotated data sets. The ecosystem takes advantage of the sudden surge in images combined with new software tools essential for efficiently annotating them before they are used to train AI models, with the goal of ultimately removing the biggest bottleneck of AI and radiology.

Background theory for medical imaging and AI-based medical image analysis was conducted to gain a better understanding of the field. Use cases were mapped out for the most the actors that will be using the system such as citizens, patients, and radiologists. How the ecosystem will be used in a clinical radiology workflow was specified, along with a detailed overview of a Knowledge Generation Engine which is programmed to search Digital Twins for medical images, delegate them for annotation, and train AI models with the resulting data sets.

The results were obtained from an investigation on how NVIDIA Clara can be used, demonstrating the efficiency of labeling images using AI assisted annotation and training AI models on powerful supercomputers. The AI models produced were compared to pre-trained models from NVIDIA, showing similar performance. The results also propose a design of the Digital Twin ecosystem, along with pseudocode for three main components: the Digital Twin, application in the clinical setting, and automatic generation of new AI models. The conclusion suggests that the proposed ecosystem is technologically feasible but will require extensive resources, expertise, and more research before implementation. The possibilities and limitations on what should be done next were identified and discussed.

Sammendrag

Kunstig intelligens (AI) i det medisinske feltet radiologi har utviklet seg så raskt de siste årene at klinisk anvendelse av AI-assistert diagnose har blitt et spørsmål om når det skjer, og ikke om det er gjennomførbart. Fellesnevneren funnet i forskningsartikler om radiologi og AI er utfordringen spesielt knyttet til medisinsk avbildning: lav tilgjengelighet av annoterte datasett på grunn av personvern og økonomiske faktorer, ettersom feltspesifikk kompetanse og høyt tidsbruk er nødvendig. Datasett er nødvendig for å trene AI-modeller som er i stand til å utføre operasjoner som medisinsk bildeklassifisering og segmentering. Denne masteroppgaven forsker på en løsning for å utvikle datasett ved å undersøke et Digital tvilling-økosystem der hver innbygger har en nettbasert tvilling i nettskyen hvor medisinske bilder automatisk lastes opp til. Innbyggere vil ha muligheten til å donere dataene sine for utvikling av AI-modeller som øker totalmengden av bilder som kan brukes til generering av nye datasett. Denne oppgaven tar sikte på å undersøke i hvilken grad et Digital tvilling-økosystem kombinert med moderne bedriftsprogramvareløsninger som NVIDIA Clara-pakken kan bidra til generering av annoterte datasett av høy kvalitet. Økosystemet drar nytte av økningen i antall bilder tilgjengelig kombinert med nye programvareverktøy som er avgjørende for å effektivt kunne annotere dem før de brukes til å trene AI-modeller, med mål om å eventuelt fjerne den største flaskehalsen for AI og radiologi.

Bakgrunnsteori for medisinsk avbildning og AI-basert medisinsk bildeanalyse ble utført for å få en bedre forståelse av feltet. Bruksmønstre ble kartlagt for de fleste aktørene som vil bruke systemet, for eksempel innbyggere, pasienter og radiologer. Hvordan økosystemet vil bli brukt i en klinisk hverdag for radiologi ble gjennomgått, sammen med en detaljert oversikt over en kunnskapsgenerasjonsmotor som er programmert til å søke i digitale tvillinger etter medisinske bilder, delegere de videre for annotasjon og trene AI-modeller med de resulterende datasettene.

Resultatene i denne oppgaven er fra undersøkelsen av hvordan NVIDIA Clara kan brukes, og demonstrerer effektiviteten av å annotere bilder ved bruk av AI-assistert annotasjon og trening av AI-modeller på kraftige superdatamaskiner. AI-modellene som ble produsert ble sammenlignet med ferdigtrente modeller fra NVIDIA og viste lignende ytelse. Resultatene foreslår også en design av Digital Tvilling -økosystemet, sammen med pseudokode for tre hovedkomponenter: Digital Tvilling, anvendelse i kliniske omgivelser, og automatisk generering av nye AI-modeller. Konklusjonen antyder at det foreslåtte økosystemet er teknologisk gjennomførbart, men vil kreve omfattende ressurser, kompetanse og mer forskning før implementasjon. Mulighetene og begrensningene for hva som skal gjøres videre ble identifisert og diskutert.

Preface

This thesis was written as my master thesis for the Department of Computer Science (IDI) at the Norwegian University of Science and Technology (NTNU) over the course of the fall and spring semester of 2019 and 2020, respectively.

I would like to thank my supervisor, Frank Lindseth, for guidance and direction. His expertise within medical imaging was invaluable. Other awknowledgements include the NTNU High Performance Computing (HPC) Group, specifically Håkon Hukkelås, for guidance in taking maximum advantage of the supercomputers used to run experiments.

Sebastian Olafsson

Trondheim, July 12, 2020

Contents

1 Introduction and Problem Description **1**

1.1 Motivation 1

1.2 Goals and Research Questions 3

1.3 Contributions 3

1.4 Thesis Structure 4

2 Background Theory and Related Work **5**

2.1 Medical Imaging 5

2.1.1 X-ray 5

2.1.2 CT 6

2.1.3 MRI 7

2.1.4 Ultrasound 8

2.1.5 PACS 9

2.1.6 DICOM 10

2.2 AI-based Medical Image Analysis 10

2.2.1 Classification 11

2.2.2 Object Detection 11
2.2.3 Segmentation 11
2.3 Medical Imaging Data Sets 12
2.3.1 Medical Imaging Decathlon 12
2.3.2 CHAOS 13
2.4 NVIDIA Clara 13
2.4.1 NVIDIA DGX 13
2.4.2 Collaborative learning 14
2.4.3 Domain-optimized performance 14
2.4.4 Ease of integration 15
2.4.5 Model training pipeline 15
2.4.6 Spleen training example 15
2.5 Previous Work 17

3 Methodology **21**
3.1 Digital Twin Ecosystem 21
3.1.1 Key components 21
3.1.2 Ecosystem diagram 22
3.2 Stakeholders 23
3.2.1 Citizen 23
3.2.2 Patient 24
3.2.3 Radiologist 25
3.2.4 Hospital 26

3.2.5 Developer 26
3.3 Use Cases 27
3.3.1 Clinical Workflow 27
3.3.2 Knowledge Generation Engine 30
3.4 Tools 33
3.4.1 Data Set 33
3.4.2 Dice Score 33

4 Results **35**

4.1 Clara Run-Through 35
4.1.1 Spleen Segmentation 36
4.1.2 Annotating an image 36
4.1.3 Training a new model 44
4.1.4 Updating an existing model 46
4.1.5 Exporting the model for inference 46
4.1.6 Validation of the model 46
4.1.7 Using the model for inference 47
4.1.8 Video example 50
4.1.9 Combining Clara with the KGE 51
4.2 Digital Twin Design 52
4.2.1 General scope 52
4.2.2 Digital twin pseudocode 52
4.2.3 Knowledge Generation Engine pseudocode 53

4.2.4 Decision support pseudocode 55

4.2.5 Design requirements 56

4.2.6 Hospital requirements 58

4.2.7 Potential future use cases 59

5 Discussion **61**

5.1 General 61

5.2 Research Questions 62

5.2.1 Research question 1 62

5.2.2 Research question 2 63

5.2.3 Research question 3 64

5.3 Reflections 66

6 Conclusion and Future Work **69**

6.1 Conclusion 69

6.2 Future Work 70

6.2.1 Proof of concept product 70

6.2.2 Hospital deployment 70

6.2.3 Complete prototype and optimization 71

Bibliography **73**

List of Figures

2.1 CT Scanner 7

2.2 Proton Alignment in MRI 8

2.3 PACS flow 9

2.4 AI Techniques for Image analysis [1] 11

3.1 Ecosystem diagram 23

3.2 Clinical Production Flow Diagram 28

3.3 Knowledge Generation Engine Flow Diagram 30

4.1 Initial CT scan 37

4.2 Choosing the label 38

4.3 3D tools 39

4.4 Model selection 39

4.5 Segmentation result 40

4.6 Selecting extreme points 41

4.7 2D tools 42

4.8 Final annotation with CT scan 43
4.9 Final annotation by itself 43
4.10 Location of data sets 44
4.11 Training parameters 45
4.12 Correct label 47
4.13 Inference result for 100 (left) and 400 (right) epochs 48
4.14 Inference result for 1000 (left) and 2000 (right) epochs 49
4.15 Inference result (left) versus label (right) 50
4.16 Digital Twin Endpoints 57
4.17 Hospital Workflow 58

List of Tables

4.1 Validation results 46

Listings

4.1 Training parameters 45
4.2 Digital twin pseudocode 53
4.3 Knowledge Generation Engine pseudocode 54
4.4 Decision support pseudocode 56

Chapter 1

Introduction and Problem Description

The motivation for this thesis is to investigate the possibilities introduced by creating a personal medical Digital Twin that follows you from birth to death and beyond, keeping you informed on your health and storing all your health related data safely in the cloud.

This thesis will focus on the area of gathering specific medical info on an individual by exploring the possibilities for a specific field: Medical Imaging. One of the first medical fields expected to be revolutionized by AI is radiology. Automatic image classification and segmentation has shown to be a task computers can do with a high level of success, with the potential to increase the quantity and quality of diagnosis, ultimately saving many lives [2].

1.1 Motivation

The main motivation for this thesis is the introduction of new enterprise solutions like NVIDIA Clara that have the aim to assist some of the largest challenges of AI in radiography, as the process of transforming classification and segmentation techniques from research papers into a clinical setting has proven to be difficult.

Altman [3] outlined some of the challenges that have kept AI from entering medical imaging on a large scale:

1. Difficulty in developing methods that allow for integrating heterogeneous data sets. Most data sets are often biased data, or is built up of multiple independent data. These new methods need to allow for more flexibility, as well as being able to work with incomplete data.
2. The limitation of high quality annotated data sets. Experts within the field must spend countless hours to create and label these sets.
3. Poor performance on initial models. In the absence of large data sets, new methods have to support incorporating prior human knowledge to give a head start to the model which the system will later refine. Training a model from scratch using arbitrary parameters may produce poor results, so having a preconceived starting point will allow the model to achieve high performance quickly with less training data.
4. Social challenges like intellectual property, data provenance, regulatory, and economics have slowed many attempts at training medical AI models due to the fact that hospitals and patients have very strict laws for confidentiality.

The biggest challenge is undoubtedly the time and resources required to create high quality labeled data sets. Socioeconomic challenges are the main culprit for the scarcity. As annotating a single medical image normally takes four hours, there is very little incentive to give these away for free, especially when only medical professionals with high hourly pay are qualified to conduct the work. This naturally leads to hospitals and researchers not distributing their valuable data sets to the public.

In this thesis we will address the challenge of creating large labeled data sets by designing an ecosystem that would make this possible. NVIDIA Clara will be the candidate chosen as the third party solution. The ecosystem will be comprised of components including the Digital Twin and NVIDIA Clara. Such an ecosystem would lower costs and allow more actors to enter the AI radiology field, not just wealthy corporations and hospitals with extensive funding and resources.

By utilizing powerful supercomputers capable of training complex AI models in short periods of time, an envisioned result is an iterative ecosystem that continuously produces state-of-the-art AI models. Hospitals can employ these models to assist radiologists, and patients will have the option to donate their data to improve the models with their own medical images.

1.2 Goals and Research Questions

Goal *The main objective of this thesis is to explore the possibilities of utilizing third party solutions to facilitate an ecosystem that will enable data mining of medical images from Digital Twins, which will subsequently be aggregated into data sets with the purpose of training AI models to be deployed in hospitals and integrated into the radiology workflow.*

The following research questions were created to reach the goal.

RQ 1 *Is it feasible to use the NVIDIA Clara suite to easily and quickly annotate and train AI models for segmentation and classification of medical images?*

RQ 2 *How should a Digital Twin ecosystem be designed, and how can NVIDIA Clara be used in conjunction with this ecosystem to automate model training?*

RQ 3 *How can NVIDIA Clara combined with the Digital Twin concept be integrated into the radiology workflow, and how can it be useful to patients, hospitals, and researchers?*

1.3 Contributions

This thesis investigates the possibilities, limitations, and challenges of creating a Digital Twin ecosystem combined with NVIDIA Clara with the purpose of generating data sets used for training AI models. Two important use cases are described. The first use case details AI assisted decision support in a clinical setting, and the second shows how a Knowledge Generation Engine system can be created to continuously search for relevant medical images from Digital Twins and use these to train AI models. An investigation is conducted where NVIDIA Clara is used to annotate a real CT image of the spleen using AI assistant annotation. Clara is then used to train four different AI models on spleen segmentation. A design of the Digital Twin and its ecosystem is proposed, along with pseudocode on the main components.

1.4 Thesis Structure

The thesis is structured as follows:

1. **Introduction and Problem Description:** Introduces the thesis and the motivation behind it.
2. **Background Theory and Related Work:** Introduces the background for medical imaging, AI based medical image analysis, NVIDIA Clara, and previous work.
3. **Methodology:** Describes the Digital Twin ecosystem, identifies relevant stakeholders, and maps out use cases.
4. **Results:** Presents the results from the investigation conducted. Clara is presented first, following the design of a Digital Twin.
5. **Discussion:** An analysis of the results is presented to answer the research questions, followed by general reflections.
6. **Conclusion and Future Work:** An overall conclusion is detailed, and future work is discussed.

Chapter 2

Background Theory and Related Work

This chapter will go in depth into the background on this thesis, focusing on topics such as medical imaging, technological frameworks, and previous work.

2.1 Medical Imaging

Medical imaging is the process of producing visual representations of what is inside the body. These representations can be used for clinical analysis and medical intervention. More importantly for this thesis, the visual representations are used for analysing the state of organs and detecting abnormalities.

There are a number of ways to produce these visual representations as explained in the following subchapters. These visual representations, or medical imaging types, are often referred to as modalities.

2.1.1 X-ray

X-ray is the most common form of medical images, specializing in generating images of tissues and structures within the human body [4]. X-rays use electromagnetic radiation traveling through the body to generate an image, called a radiograph. The x-ray machine contains an x-ray source on one side and an x-ray detector on the other side, where the patient is located between these two points so the radiation can pass through the body. Bones and tissue absorb different amounts of x-rays, which makes it possible to produce an image based on the different absorption rates throughout the body. X-ray scans are often used for broken bones, cancer, blocked blood vessels, and infections.

Frequent exposure to ionizing radiation may be harmful to living tissue, but normal usage is considered safe for most people. X-ray machines are found in the majority of hospitals and scanning time is short, making x-rays the most common form of medical imaging.

2.1.2 CT

Computerized Tomography (CT) requires heavy computer processing, as it combines rotational x-ray images taken from different angles to produce cross-sectional images of blood vessels, soft tissues, and bones inside the body. [5] These cross-sectional images are referred to as slices and contain detailed information compared to normal x-rays. Slices are merged together to form a three-dimensional image used for diagnosis.

CT scanners use x-ray technology placed on a rotational device called a gantry that moves around the patient while continuously shooting x-rays from one side of the machine through the body into an x-ray detector on the other side, producing measurements that are later combined into a two-dimensional image with the use of computer algorithms. The patient lies on a bed that moves through the CT scanner while the gantry continuously rotates around the body, as shown in Figure 2.1. Gathering many 2D slices allow computers to later produce 3D images for radiologist to diagnose.

The advantage of CT scans is low cost and quick scans for detailed 3D images. Dense structures like bone are easily seen with CT scans. The disadvantage of CT scans are the same as x-ray scans, as they produce ionizing radiation which has the potential to be harmful if frequently exposed over time by affecting living tissue.

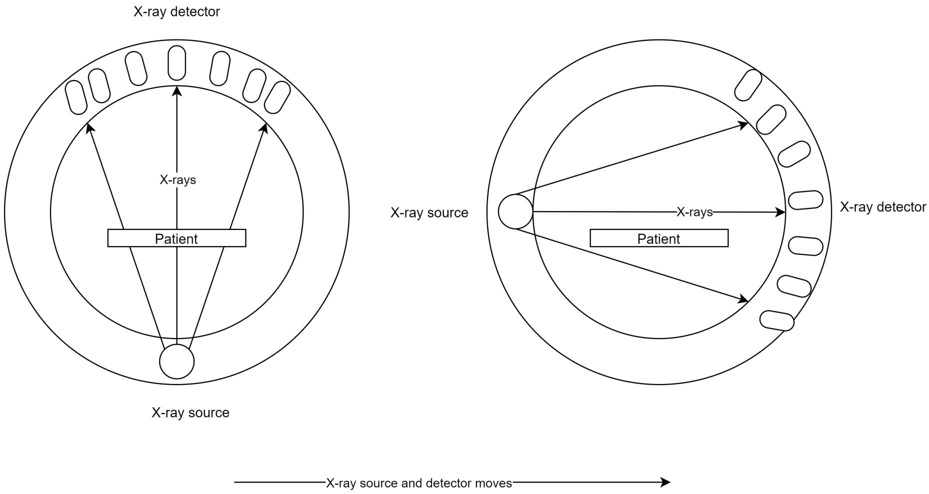

Figure 2.1: CT Scanner

2.1.3 MRI

Magnetic Resonance Imaging (MRI) is a technology used to scan and produce detailed three-dimensional visual representations of the interior of the body [6]. Patients are placed in a MRI machine containing large magnets and are told to lay still while the machine scans the body.

MRI machines force protons within the body to align with a strong magnetic field produced by the machine. The protons in the patient's body are stimulated by a radiofrequency current and consequently deter from the natural equilibrium as they attempt to fight against the force of the magnetic field. Figure 2.2 shows how the proton's direction alignment is natural on the left but becomes forced in a single direction by the magnetic field on the right. Turning off the radio frequency field makes the protons release energy as they realign with the magnetic field, and this energy change is measured by the MRI machine which produces the image by a computer. Different types of tissue in the body release the acquired energy at various rates, making it possible to differentiate cell types from one another.

The advantages of MRI compared to CT is that it does not produce ionizing radiation which can be harmful for humans under frequent exposure. This makes MRI more suitable for patients requiring frequent scanning, like cancer patients receiving regular diagnostics to track the status of a tumor. Soft tissue is bet-

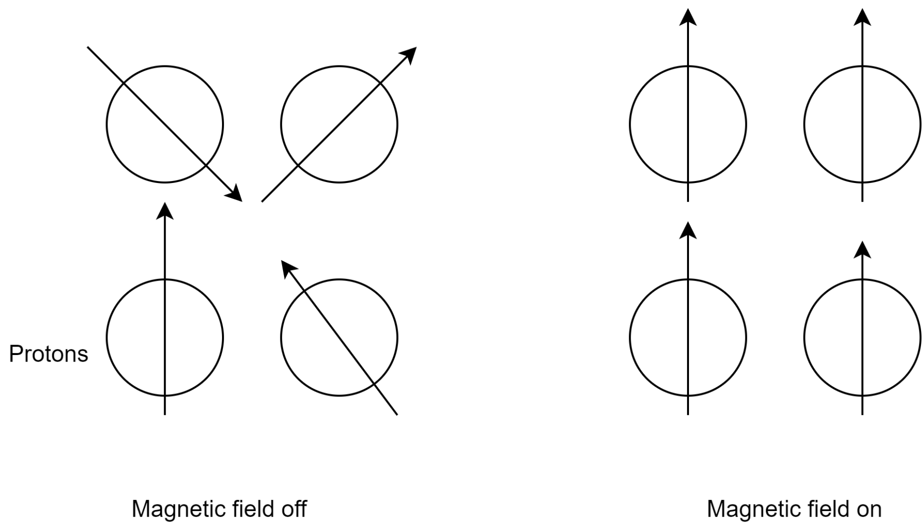

Figure 2.2: Proton Alignment in MRI

ter imaged by MRI, so muscles, ligaments, and tendons are represented more clearly and with higher resolution compared to CT, making MRI more suitable for shoulder and knee injuries.

The disadvantage of MRI is that the scanning machine is costly to purchase and operate, and some smaller hospitals cannot afford a scanner. Waiting times can therefore be long. Due to the magnetic fields created by the machine certain patients with iron implants are unable to undergo an MRI scan as the scanning machine is strong enough to pull metal out of the body. Scanning time is also lengthy, lasting from 20 to 90 minutes [7]. This can be troublesome for patients with claustrophobia who may be uncomfortable in such a machine over long periods of time.

2.1.4 Ultrasound

Ultrasound is another example of a noninvasive scanning technique to produce images of the body [8]. Diagnostic ultrasound uses probes called transducers that emit sound waves and detects the ultrasonic echoes being reflected back. As the transducer sends sound waves to the body, the waves are reflected back and electrical signals are generated which the ultrasound scanner measures. Different boundaries between tissues generate various electrical signals, such as the boundary between tissue and bone, making it possible to calculate the distance between boundaries to generate a 2D image of tissues and organs.

2.1.5 PACS

Picture Archiving and Communication System (PACS) was created to eliminate the use of physical films by enabling the transition to a digital environment by unifying how images are acquired, stored, transmitted, and displayed electronically [9].

The main advantage of PACS is improved efficiency that results from handling all data electronically instead of physically filing films in physical storage cabinets. Radiologists, patients, and the hospital save substantial amounts of time from the improved efficiency of digital communication. Once filed electronically, images become available at all times without the risk of being lost. Electronic storage allows multiple simultaneous viewing instances of the same image. Additional metadata becomes easier to store and query, such as the patient's name, hospital, date, clinician, and more. Electronic storage also allows for backups locally and remotely so that images are unlikely to go lost.

The main disadvantage of PACS is the upfront costs of installing and maintaining the system and the learning curve hospital staff have to go through to become familiar with the system.

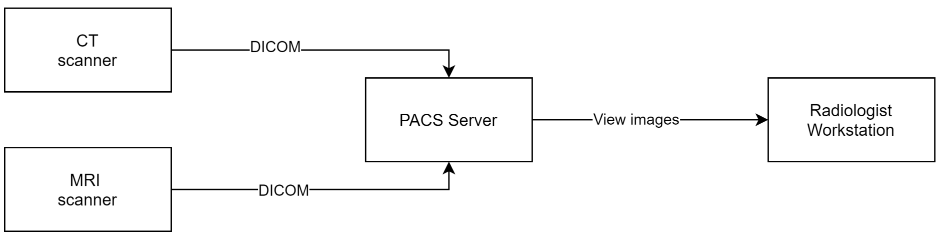

Figure 2.3: PACS flow

Figure 2.3 visualizes the flow from scanning an image which is transferred to the PACS server and then viewed by a radiologist at a workstation.

Images stored in the PACS system are in the DICOM format, further detailed in the following subsection. As the DICOM images are already tied to the patient, the PACS supports querying all images for a certain patient making it possible to program new functionality to export or modify all images. The modifications possible allow custom scripts to be integrated into the PACS system, laying the necessary technological foundations required for automatic image segmentation and classification.

2.1.6 DICOM

Digital Imaging and Communications in Medicine (DICOM) is an international standard used in most hospitals to handle medical images [10]. It is used to transmit, store, retrieve, print, process, and display imaging information. DICOM and medical imaging can be compared to JPEG and camera photos in the sense that it is a universally accepted format for the transfer of files. The scanning machines, computers, servers, and other technical equipment in hospitals all use the DICOM standard to communicate information.

DICOM aggregates relevant information into data sets such as the image and the patient ID, enabling a connection between the image and the patient at all times. The image can be single or multidimensional, supporting a wide range of modalities such as CT or MRI in either 2D, 3D, or even 4D. The protocols allow for the exchange of images, visualization, and presentation.

2.2 AI-based Medical Image Analysis

The introduction of artificial neural networks, also called deep learning, is currently advancing in many fields within industry and academia. Computers have gained the ability to recognize patterns in large data sets and eventually recognize patterns in unseen data. Medical image analysis is one of the fields that has seen breakthroughs from AI, as the tasks of classification, object detection, and segmentation can be performed by computers through new techniques and algorithms.

In general, classification aims to detect if something is in the image, object detection finds the location of an object in an image, and segmentation determines the individual pixels of an object in the image. Figure 2.4 gives a visual representation of the various techniques.

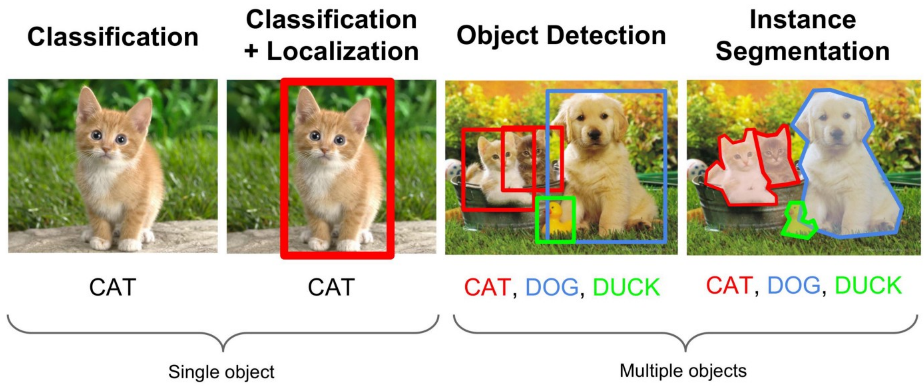

Figure 2.4: AI Techniques for Image analysis [1]

2.2.1 Classification

Image classification is one of the most common machine learning tasks found as performance has increased in recent years with the introduction of new techniques like deep neural networks. The goal of classification is to assign one or more labels to an image. An example in medical imaging if the classification of focal liver lesions on multi-phase CT images. As cancer in the liver is one of the leading causes of death, being able to use computers to classify focal liver lesions in CT images will increase the number of diagnoses being made in addition to their quality and accuracy [11].

2.2.2 Object Detection

Object detection is the process of identifying objects within an image. It is usually done by placing a bounding box around the object to determine the localization within the image. An example of object detection would be localizing where the liver or spine is within a CT image.

2.2.3 Segmentation

Segmentation is the process of identifying regions or boundaries within a 2D or 3D image. Segmentation of medical images is for example separating the lungs in an image, outlining only the lung. Having separated the organ of interest allows for precise measurement and simulations. The difference between segmentation and object detection on a technical level is that while object detection focuses on finding the location of the object in an image with a box, segmentation marks each pixel in the image with a label. For example, segmentation would label each pixel of the lung as the lung, and every other pixel would be labeled as not the lung.

Segmentation is often used to perform different operations on objects in a medical image, such as examining an anatomical structure, locating tumors or abnormalities, measuring the volume of tissue to document tumor growth, and determine radiation dose for patients who will be receiving radiation therapy [12]. This is done by locating regions of the image with homogeneous properties like texture, brightness, contrast, and color.

2.3 Medical Imaging Data Sets

The low availability of high quality data sets have made it difficult to perform research and develop AI models. Patient privacy and expensive labor costs contribute to the problem of aggregating large data sets, as thousands of images are necessary to train complex models. High quality public radiology data sets are hard to come by but do exist.

2.3.1 Medical Imaging Decathlon

Certain challenges and competitions within the field of AI provide data sets openly in hope of creating a new benchmark for testing new algorithms and models. The Medical Imaging Decathlon is an example of this and provides open source data sets that can be used by anyone, containing 2,633 3D images of various modalities, organs, and tasks from real-world clinical applications.

The challenge consists of 10 various data sets all having different organs and tasks, such as segmenting a cancerous tumor in the lung or classifying the tubular small structures next to the heterogenous tumor in the hepatic vessels. The large variation of organs and tasks combined with a high amount of images have led to this data set becoming common for researchers to use, as many other data sets focus on a single task and organ and contain substantially fewer images.

The goal of the competition is to train a general purpose machine learning algorithm that translates to unseen classification or segmentation tasks without the need of human interaction or task-specific predefined parameters [13]. This means the algorithm needs to adapt to any of the segmentation and classification tasks without an intervention of any kind. NVIDIA is one of the official partners for this challenge and use the supplied data sets for training the AI models provided in NVIDIA Clara.

2.3.2 CHAOS

Combined Healthy Abdominal Organ Segmentation (CHAOS) is another challenge focusing on the segmentation of abdominal organs such as the liver, kidneys, and spleen from CT and MRI data [14]. The motivation for this challenge was to produce AI models with high performance on extracting objects of interest from DICOM images. The challenge provided 20 training and 20 testing cases for CT images and the same amount for MRI images.

2.4 NVIDIA Clara

NVIDIA Clara is a healthcare application framework allowing AI-powered imaging and genomics [15]. This framework contains specialized full-stack GPU-accelerated libraries designed to run on NVIDIA hardware, such as the DGX-2 system located at NTNU. Claire is split into two parts: Clara Train and Clara Deploy. Clara Train focuses on the annotation and training process while Clara Deploy specializes in interfacing with existing hospital environments. Together with GPU-optimized software and simple to use SDKs, Clara enables real-time and scalable solutions that can be used to investigate a Digital Twin ecosystem related to medical imaging. This subchapter will first look at the key features of Clara and look at the technical details under the hood to gain an understanding on how the AI models are trained.

2.4.1 NVIDIA DGX

Being both a hardware and a software company, NVIDIA has the opportunity to create powerful enterprise solutions with software for customers with demanding hardware requirements. The DGX-2 is a purpose built workstation focusing on cutting-edge hardware specialized for AI tasks [16]. The system is comprised of 16 NVIDIA V100 Tensor Core GPUs delivering two petaFLOPS of performance. NVIDIA provides GPU optimized software designed to maximize performance from every GPU while simultaneously providing tools that lowers the learning curve to take full advantage of the system. NTNU has one DGX-2 system available for use, making it possible to experiment with training AI models that would normally take substantially longer using normal top-of-the-line GPU's.

2.4.2 Collaborative learning

Clara integrates two key techniques necessary for AI in the healthcare sector. The first technique is transfer learning, a technique that re-trains an existing pre-trained model. If a generalized pre-trained model is used as a baseline then it is possible to update this model on a given domain of medical images rather than having to start from scratch. This is useful in the beginning stages of developing a model where data sets may be scarce, as having access to a pre-trained model will spur the performance without the need of large data sets.

The second technique is federated learning, a technique where a global AI model is able to be trained securely by allowing different sites to collaborate, train, and contribute to the global model. This enables hospitals to train the model locally before the model weights are then uploaded to the main server and integrated into the global model. The global model is then distributed back into all the hospitals so all parties receive the latest version. This keeps sensitive info secure as it does not have to leave the local hospital in order to be used for training.

2.4.3 Domain-optimized performance

Clara includes a whole subset of features and techniques to achieve remarkable performance on training AI models for medical imaging, especially if used on NVIDIA'S own DGX platforms. This includes Horovod based multi-GPU scaling, Automatic Mixed Precision (AMP), and smart caching mechanism. It supports deterministic training, meaning Clara can guarantee reproducibility which is vital for testing. Multiple loss functions are supported, with new model architectures being added in the future as AI science advances.

2.4.4 Ease of integration

Integration with hospital equipment is key to the success of usability and adoption, and Clara has developed building blocks to build clinical workflows that interface with existing hospital equipment, such as the industry standard PACS system.

2.4.5 Model training pipeline

As NVIDIA Clara focuses on being user-friendly and does not require a deeper understanding of how everything works under the hood, it does not really matter how the models are trained to end users and is not featured on their website. There is another reason to this, as different tasks are trained using different methods. For example, chest x-ray disease pattern detection is trained differently from brain tumor segmentation. Clara has abstracted the different methods for the different tasks so that the user does not have to think about what lies under the hood, as the developers of Clara have chosen the current best performing methods. When using Clara to train a new model, the task and organ is all that has to be specified, and Clara trains the model with a predefined pipeline for the chosen task.

Clara allows the developers to change most of the components. For example, the data pipeline, model components, loss function, optimizer, metrics, and structure of training graph can all be changed. Clara's documentation provides model development guidelines and gives examples of what sections of code to change to make it compatible with the Clara Train API.

2.4.6 Spleen training example

The specifics on how models are trained for each task can be found in the documentation. Understanding what happens under the hood is not a necessary prerequisite for using Clara, but this can be useful to developers. As an example, we can look at how spleen segmentation is set up, as this is one of the few pre-trained models Clara has available.

The spleen model uses a training pipeline from the runnerup winner of the "Medical Segmentation Decathlon Challenge 2018". Xia et al. introduced this technique in the paper *3D Semi-Supervised Learning with Uncertainty-Aware Multi-View*

Co-Training. Authors from John Hopkins University and NVIDIA amongst others were part of this paper, and the end result achieved state-of-the-art performance on the Medical Segmentation Decathlon challenge, showing that Clara utilizes the best techniques available for training models. As some of the authors in this paper were from NVIDIA, users of NVIDIA Clara can be assured that optimal training techniques are used.

The premise of the paper was to create a semi-supervised algorithm to address the challenge of training models with unlabeled 3D data. The results were positive, and while using partially labeled data achieved about 4% gain compared to the previously best model, using fully labeled data yielded state-of-the-art performance, showing that the pipeline and techniques made for unlabeled data worked admirably when performing fully supervised training.

Co-training was the semi-supervised technique used in this paper. Co-training was first done with the aim of increasing performance of models with an abundance of unlabeled data with a small amount of labeled data [18]. This was done by augmenting labeled data sets through a partitioning technique. The experiment attempted to classify web pages by splitting the page into two views, the first view was words occurring on the web page and the second view was words occurring on the hyperlinks pointing to the web page. AI models can be trained to recognize either of these views, and these two models were then used to predict instances of the unlabeled data set, creating new labeled examples for the other model. The two distinct views have to be relatively compatible by having some correlation to each other, which was the case for the web pages in the paper, and successful results were found.

Applying the co-training technique to medical imaging required some changes. Having 3D data made it natural to have three views instead of two used in the original co-training paper. The views correlated to the coronal, sagittal, and axial views found in MRI scans. A requirement for co-training was having some level of compatibility by having correlation between the views, which the multi-planar views found in MRI scans naturally fulfill.

When segmenting 3D images it is common to augment 3D data, but this pipeline is initialized in 2D data instead. This was done to take advantage of pre-trained models that are publicly available, such as natural imaging tasks. The pre-trained models include weights that perform better compared to training a network initialized with random weights. These 2D models were then adopted to asymmetric kernels in 3D networks, a technique demonstrated by Liu et al.. Having models for every 2D image allowed the training algorithm to have biases for each 2D view, in turn giving the network more 3D information due to the complementary feature representations in all three views.

While the co-training multi-view paper used a network structure based on ResNet-18, Clara uses the AH-Net structure detailed in the paper by Liu et al. [19]. The motivation behind creating this structure was due to suboptimal performance in generalization when trying to use classic deep convolutional neural networks with 3D convolution kernels. The AH-Net architecture transfers shared convolutional features from 2D to 3D images, essentially exploiting knowledge found within 2D slices of the 3D images.

Clara combines the multi-view co-training pipeline with the AH-Net kernel for training spleen segmentation models, with the included training scripts abstracting the intricate details for developers.

2.5 Previous Work

As of writing, there are no studies on combining a Digital Twin ecosystem with a software solution to generate data sets. However, there are studies on individual components such as federated learning, transfer learning, and data set annotation. AI-based medical image analysis and its performance is well documented in countless studies, so this background study will instead focus on techniques that can improve the data set generation and training process.

Federated learning has been studied to find out if there is performance loss across the models developed in a centralized manner compared to a distributed manner. Li et al. produced a paper named *Privacy-Preserving Federated Brain Tumour Segmentation* [20] where they tested the two methods, using NVIDIA Clara for client-side local training. They concluded that a comparable segmentation performance on brain tumour segmentation was achieved without sharing clients' data, although twice the amount of epochs during training was required, as the decentralized model converged at 600 epochs compared to the centralized model's 300 epochs.

Optimal performance using federated learning is not exclusive to NVIDIA Clara, as Czeizler et al. produced a paper called *Using federated data sources and Varian Learning Portal framework to train a neural network model for automatic organ segmentation* [21] where similar results were achieved. This study focused on segmentation of the female pelvis organ, training two models in a centralized and decentralized manner, but used Varian Learning Portal (VLP) as the software solution. VLP is a distributed machine learning infrastructure comparable to

NVIDIA Clara, allowing training of AI models across hospitals without sharing private medical images. They concluded that the results were good, resulting in two models with similar performance levels, where one was trained in a federated manner and the other in a classic single location manner.

Transfer learning has become an important tool to give a head start when training AI models, especially in scenarios where obtaining data sets is a challenge. Shin et al. [22] investigated how effective transfer learning is when applied to the medical domain. A model pre-trained on natural image data sets was used as a base which was further fine-tuned to create models specializing in thoraco-abdominal lymph node (LN) detection and interstitial lung disease (ILD) classification. The pre-trained model was based on the ImageNet, a public image database with over 100,000 labeled images on various words or phrases. This pre-trained model offers high performance on natural image recognition, but is not adapted to the medical domain. The authors hypothesized that even though the model specializes in natural images, it can be fine-tuned to be effective for cross-modality imaging settings such as medical image recognition, as natural images contain similarities to CT and MRI images. The authors compared training the model with ImageNet as a base instead of randomly initialized parameters with trying to teach an adult to classify ILDs, as opposed to babies, meaning the model had a better starting point using ImageNet. The results indicated that transfer learning achieved consistently better results compared to training from scratch, suggesting cross-data set models are applicable to the medical image domain.

Transfer learning can also be relevant for solving the problem of the discrepancies between different scanners and different imaging protocols. Van Opbroek et al. produced a paper called *Transfer Learning Improves Supervised Image Segmentation Across Imaging Protocols* [23] investigating this issue, noting that while supervised learning techniques perform well on data that is exactly representative, even slight deviations in the target data will diminish the performance of the model. These deviations come from the differences in scanning equipment and which imaging protocols are used. The results compared performance of models trained with and without transfer learning. The model trained on data sets with variations in scanning properties and later fine-tuned with exact representative target data needed fewer labeled samples to reach the same performance compared to the model that was trained on exact scanning equipment. This supports the hypothesis that differences in scanning equipment does not need completely separate AI models, but rather a general model can be trained and fine-tuned to fit specific scanner properties with the help of transfer learning.

RIL-Contour is a medical imaging data set annotation tool focusing on using deep learning to accelerate annotation, noting that the largest barrier for development of creating AI models is the effort needed to curate these data sets [24]. The software supports fully automated deep learning methods, semi automated methods, and manual methods to annotate medical images. The proposed workflow is comprised of analysts annotating images, radiologists approving the annotations, and data scientists training deep learning models from the annotations. This methodology differs from Clara's workflow as more focus is placed mainly on the rapid collaboration between analysts, radiologists, and engineers. Clara offers similar functionality for annotation but focuses on a greater scope, encompassing the entire workflow from annotation to deployment.

Chapter 3

Methodology

This chapter will first detail a high level view of the Digital Twin ecosystem, define stakeholders, and model use cases.

3.1 Digital Twin Ecosystem

Three components create the backbone of the proposed ecosystem: all the individual Digital Twins, the Knowledge Generation Engine, and the Knowledge Bank.

3.1.1 Key components

Digital Twins

A Digital Twin is responsible for storing all medical data for a given citizen. Medical images will be stored in a patient's twin after scanning. Along with the ability to view medical records, citizens receive the option of donating data for research use. Consenting to donate grants the Knowledge Generation Engine access to their private data for research purposes and model development.

Knowledge Generation Engine

The Knowledge Generation Engine (KGE) is responsible for extracting data from Digital Twins and training AI models using said data. These generation pipelines have to be manually programmed for a specific purpose. For example, researchers may want to develop an AI model that can perform lung tumor segmentation. A pipeline can be programmed to extract lung tumor MRI scans from Digital Twins that have consented to donation. Given that these images have a corresponding label containing the correct segmentation of the lung tumor, the Knowledge Generation Engine collects a set amount of image and label pairs to produce a data set that can be used for training. The KGE initiates training using this data set with the help of Clara. The AI model produced undergoes a validation process to determine the accuracy of its inference performance. The KGE queries the knowledge bank to determine if the newly produced model received higher validation scores than the pre-existing model, updating the knowledge bank accordingly. The model is placed in a knowledge bank regardless of validation scores in the event where no pre-existing model exists.

Knowledge Bank

The knowledge bank is responsible for storing AI models after they have been produced. The knowledge bank is in the form of a digital register containing details on all AI models currently in the bank. For example, a chest x-ray classification model may have just been generated from the Knowledge Generation Engine and uploaded to the knowledge bank. The register would keep track of the task the model performs, validation score metrics, and side notes like which scanning equipment is the model suitable for. If a second chest x-ray classification model is uploaded and validation metrics show higher performance than the first model, the knowledge bank will update the register accordingly to reflect the currently best-performing model. Hospitals that have integrated the use of AI models for decision-making will routinely check with the knowledge bank to ensure the model being used is always up-to-date.

3.1.2 Ecosystem diagram

Figure 3.1 visualizes a high-level view of the ecosystem with three main components. The Knowledge Generation Engine requests data from Digital Twins, trains AI models, and exports them to the knowledge bank for Digital Twins to

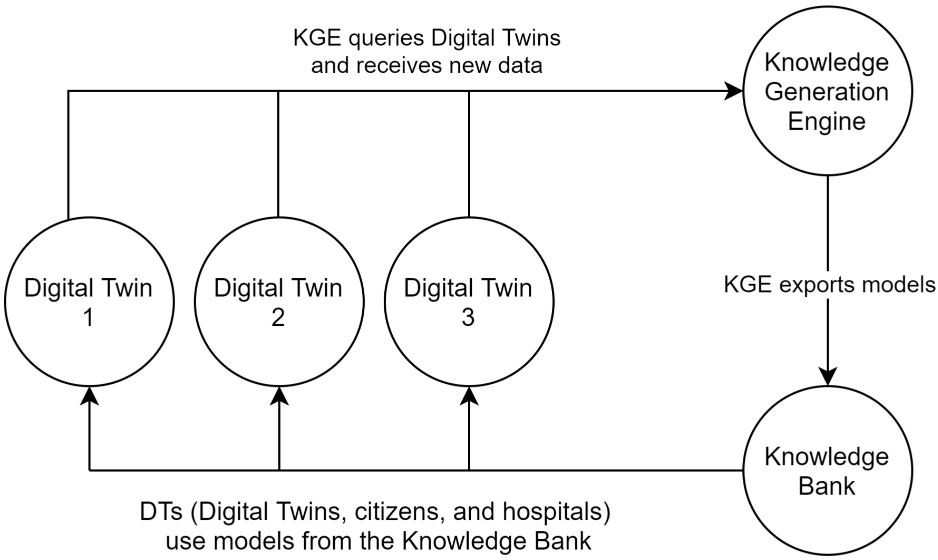

Figure 3.1: Ecosystem diagram

use.

3.2 Stakeholders

The ecosystem will be comprised of various stakeholders that regularly interact with the system. Identifying the stakeholders and investigating their needs is important to design a system that will be user-friendly and worthwhile.

3.2.1 Citizen

Every citizen will have a Digital Twin tied to them from before they are born until long after they are dead. They will have the peace of mind knowing that all their medical info is in one place and is forever accessible if they want it to be. Storing all types of data has many advantages, for example being able to utilize the data differently in the future. As new technology or new medical research is introduced, the previous data stored in your Digital Twin can automatically be fetched and analysed to monitor your health without the citizen having to initiate anything. Say blood pressure and heart rate has been tracked for many decades and stored in the Digital Twin. New research might lead to being able to predict a certain disease based on historical data of blood pressure and heart rate, and as the Digital Twin has accumulated this data over time it would be possible to do a retroactive prediction instead of having to wait for new data.

Citizens consequently have access to relevant medical info of their genetic history as data is collected on their biological family. If a person is predisposed to a medical condition due to genetics, the Digital Twin can keep track of symptoms for the disease and warn the citizen if they need to visit the doctor.

3.2.2 Patient

For the Digital Twin and medical imaging concept to have any chance of succeeding, there will have to be a large number of patients willing to donate their medical images for the purpose of training AI models. This will be an option turned off by default, but they can choose at any time to toggle it on or off. While turned on, their images can be used in data sets to train models. If a patient wishes to stop donating, their images will be removed from the data sets which prohibits them from being used for future training, but models that are already trained with their data will not be affected as it is not technologically possible to reverse the impact specific training data made on a model.

Patients will see benefits from the Digital Twin platform by making it easier and quicker for the hospital to diagnose them. Patients will not have to wait for a radiologist to perform a time consuming process of examining medical images if a computer can detect abnormalities within minutes. Quicker turnaround time from scan to diagnosis will make it possible for more patients to take scans, and increasing number of scans will naturally lead to more patients potentially finding diseases early on. All patients will have access to AI assisted diagnoses regardless if they themselves choose to donate their medical images or not.

Visiting different hospitals also benefits from having a complete digital archive of all medical data, as all data is stored in the twin and not in the hospital. This facilitates scenarios where a patient might take a medical scan at one hospital but wants a second opinion from somewhere else, as the new doctor with permission can directly look at the image data stored in Digital Twin instead of waiting for the previous hospital to transfer it.

3.2.3 Radiologist

While some believe AI will make radiologists redundant, the reality is that radiologists do many more tasks than simply diagnosing medical images. Searching for abnormalities in an image is a rather mundane task radiologists often have to do, and eliminating most of the searching will free up time for radiologists, allowing them to perform more value-added tasks like challenging diagnoses and interventional radiology. [25]. Radiology will be reshaped rather than replaced by AI.

After a medical image is taken of a patient, the image instantly transfers from the scanning machine through a pipeline where the AI model performs inference. Once inference is complete, the result is transferred back to the existing viewing software for the radiologist to view. The radiologist uses the inference result for AI based decision making instead of blindly trusting the result before making a diagnosis. For example, a patient could be taking an MRI scan because they have a suspected lung tumour. Searching for the tumour in a black and white MRI scan can be a tedious and time-consuming task for a radiologist to do, whereas the AI model can do it in a matter of minutes and display a 3D segmented image with the tumour outlined in color. Patients sometimes have to wait many days before the radiologist has time to analyse their images manually, so this solution would drastically speed up diagnosis and therefore reduce turnaround time.

Another important role for radiologists is to create data sets that will be used for training. They may periodically be delegated batches with medical images with missing labels from Digital Twins and will use NVIDIA Clara to assist them in the annotation process. Clara uses AI assisted annotation, meaning Clara helps by attempting to segment or predict the image first, leaving the radiologist with fewer steps in the annotation process such as only having to make minor adjustments in the event where the segmentation was slightly inaccurate. Annotation can also be done by other qualified professionals, such as medical PhD students. Batch annotation will be done outside of daily clinical workflow, but a real-time annotation process as medical images are taken can be implemented once Clara is fully integrated in hospitals. This real-time annotation would occur directly after a radiologist receives an inference result from an AI model. The radiologist analyses the result and determines if it is accurate enough in its current raw state to be used as a label for the image. If corrections need to be made the radiologist makes the necessary adjustments on the spot using Clara's AI assisted annotation. The real-time annotation process would occur continuously as images are taken, eliminating the need for images to be annotated at some point in the future and thereby reducing the workload for batch annotation.

3.2.4 Hospital

Smaller remote hospitals will use the exact same models that larger hospitals do for performing inference on medical images, making every hospital uniform in the quality and consistency that patients can receive around the country. This will ensure that every citizen will have access to the same treatment expertise no matter what city they are in. The differentiating factor between a hospital having excellent or mediocre radiologists will not be how fast or accurately they can diagnose an image, but will place more focus on the entire process and how well they communicate with patients.

3.2.5 Developer

The developers are responsible for designing an automatic training pipeline that continuously trains AI models. This pipeline will generate new knowledge by having access to large and high quality data sets that developers can experiment with to obtain best case results. If a new model training architecture is released achieving higher performance than current architectures, developers will use Clara to train updated models using the data sets at their disposal and switch out the old models. NVIDIA's engineers will most likely integrate the new architecture into Clara themselves, and once this integration is done, developers can simply run the training scripts again to produce new classification and segmentation models. Additionally they are able to alter the parameters of training, such as the loss function or optimizer that allows them to experiment to choose the most optimized parameters for the data sets on hand.

Besides designing the training pipeline, developers are needed to facilitate integration with existing hospital solutions. Developers have access to easy-to-use APIs from Clara to interface with existing hospital solutions like PACS servers, making integration possible for most hospitals around the world. Hospitals using PACS already have a pipeline to retrieve images digitally from the scanning machines and place them into storage, so developers would use Clara's APIs to add additional steps in this pipeline to retrieve the image, convert it to a format compatible with Clara, run inference on the image, and convert the result back into a PACS compatible image. This image can be viewed in existing software on workstations that radiologists already use to diagnose medical images.

3.3 Use Cases

Defining use cases is important to further understand the domain and stakeholders in a proposed system, so the following sub chapters will explain the use case of the most important actors.

The overall use case can be explained in a very simple form. Patients go to the hospital and medical images are produced. The images are uploaded to their Digital Twin where the patient has the option to donate their data. Donated images will be annotated by medical professionals, producing large sets of labeled medical images. These images will be used to either create or improve AI models which will be deployed in hospitals. Radiologists then use these models for decision support in diagnostics and treatment planning.

3.3.1 Clinical Workflow

Figure 3.2 details the use case of how NVIDIA Clara and a Digital Twin system can be used in conjunction with each other in a production environment for clinical workers, such as radiographers and radiologists.

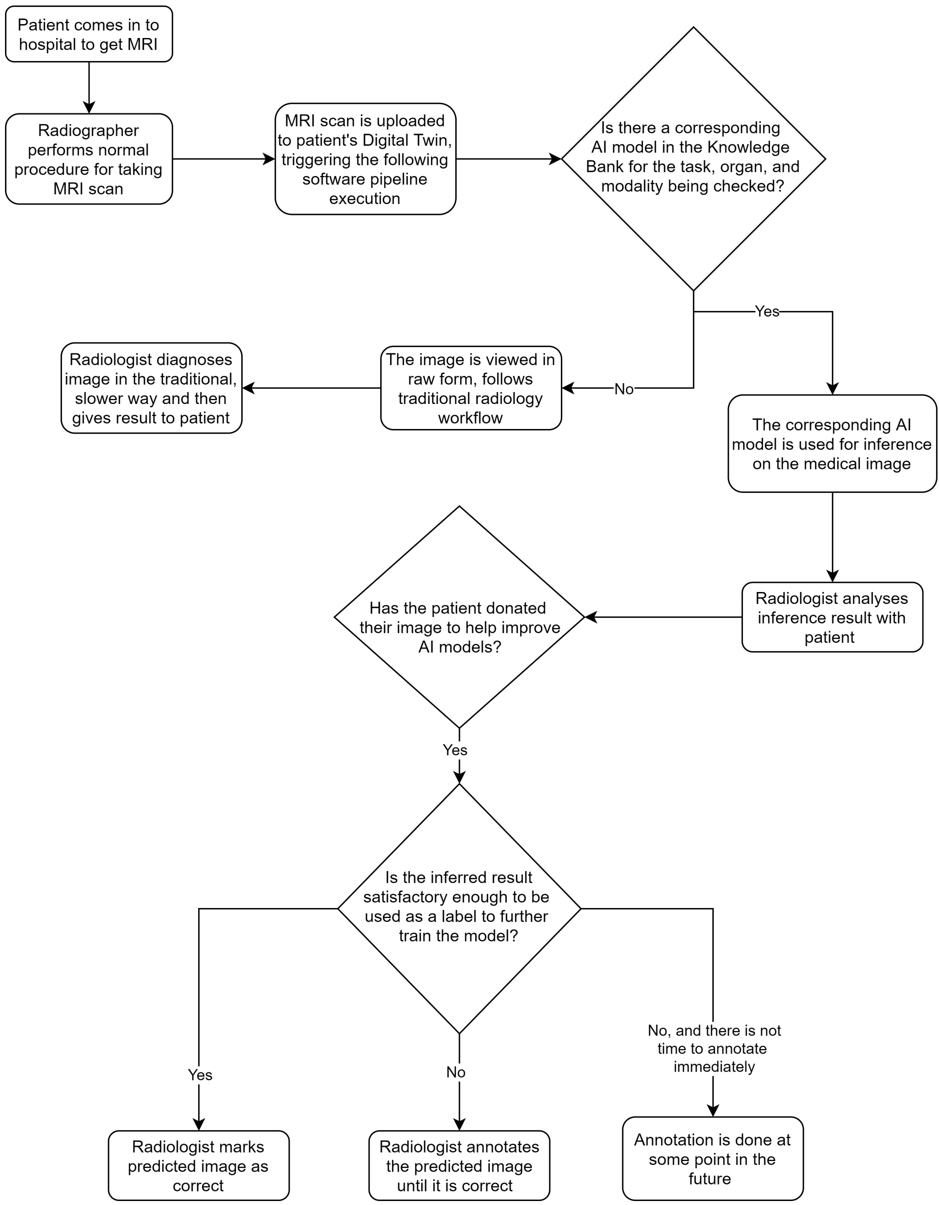

Figure 3.2: Clinical Production Flow Diagram

To begin, say a patient takes an MRI scan at a hospital to check for any lung tumors. The radiographer will perform the same procedure of acquiring the scan as is done normally, where the scan is taken and uploaded to the PACS system in the DICOM format. There is now integrated software in the PACS to automatically upload the DICOM scans to the Digital Twin ecosystem where it will be tied to the patient's Digital Twin, which will be permanently accessible by the patient or relevant medical personnel like the patient's physician.

Once uploaded to the Digital Twin, the system starts a software execution sequence which first checks if there is a corresponding AI model which can be used for inference. This is checked automatically as scans are uploaded with the help of metadata describing the task, modality, and what organ is being examined. The current supported tasks are disease detection, localization, segmentation, and classification. A lung tumor MRI AI model is necessary to diagnose the patient's MRI scan in this case.

If a corresponding model exists, the Digital Twin ecosystem will perform inference on the MRI scan using integrations from the Clara Deploy framework. Clara Deploy will receive the DICOM image from the Digital Twin, convert it into the required image format for running the inference algorithms, and then convert the inferred result back to DICOM so it can be viewed in any existing hospital viewer. Essentially, Clara Deploy will be a "black box" solution within the Digital Twin ecosystem that clinicians will never have to interact with as the pipeline is automatic.

As the AI model and NVIDIA hardware are very optimized for the computational work necessary for the inference task, results will generally be available in under two minutes. The patient and radiologist can therefore see the result immediately on existing workstation viewers and discuss the outcome. If the patient has chosen to donate their data, the radiologist can quickly determine if the inferred result is accurate enough to be used as a label that can be further used for training AI models. If the label needs adjustment, the radiologist can either quickly annotate the image correctly or leave it unlabeled, in which case the Knowledge Generation Engine would delegate the annotation task to other professionals at a later date.

The MRI scan and the inferred result will also be available in the patient's Digital Twin, opening up the possibilities for better communication with patients. A physician could quickly pull up previous diagnoses during a consultation, or a surgeon could review the medical images with a patient before an operation. Having readily access to previous data will make it easier for a patient's previous history to follow them throughout their lives.

3.3.2 Knowledge Generation Engine

Figure 3.3 details the use case of how the Knowledge Generation Engine (KGE) will be used for traversing through Digital Twins and extract relevant images which will be used to train new and existing AI models.

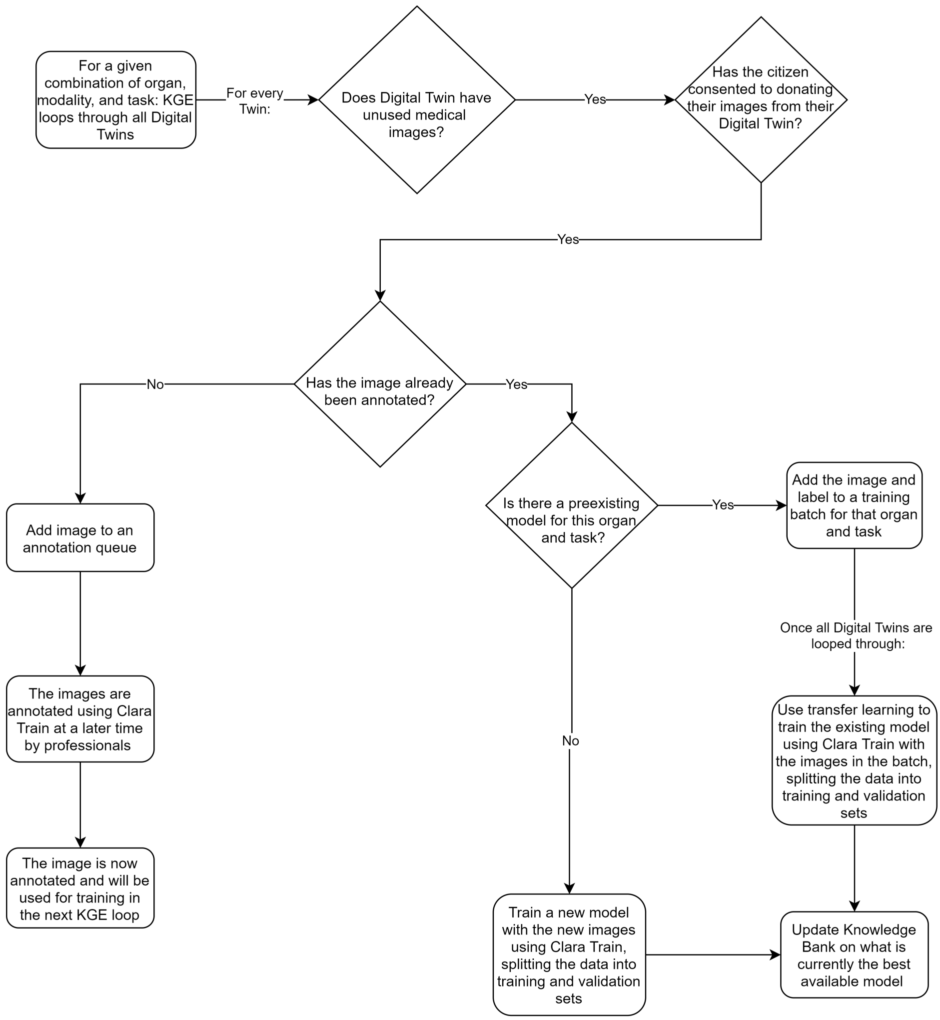

Figure 3.3: Knowledge Generation Engine Flow Diagram

General
Prerequisites for the KGE include a functioning Digital Twin ecosystem infrastructure, where medical images are automatically uploaded to a patient's Digital Twin after a scan. This will over time accumulate into a large database of Digital Twins with associated medical images, along with other medical data. The KGE will be instructed to scan for new images at a set time interval, for example once a week.

The KGE can attempt to train models before a large database of images has been accumulated from real patients. This is useful in the beginning stages of the Digital Twin ecosystem as it will take many months before a substantial amount of images has been collected. Models can still be trained on public data sets that come with annotated data. There may also be medical images collected by hospitals that are available for public use, these may even be labeled. Aggregating all possible medical images from sources outside of the Digital Twin ecosystem and annotating these will allow a head start for the KGE to begin training models. This will enable AI assisted annotation in a shorter timeframe compared to waiting for Digital Twins to accumulate enough images.

For a given combination of organ, modality, and task, the KGE loops through Twins to find relevant unused images. A KGE loop example would be trying to find images to train a new or existing model on spleen CT segmentation. Images will only go through the training pipeline if the citizen has consented to donating their medical images to research, which includes improving the AI models. The image will continue in the training process if it is unused and consent has been granted.

The next step is to check if the image is labeled. There will be two ways of annotating images: annotating at the hospital immediately after the scan by the radiologist who is diagnosing, or annotating by qualified professionals during batch annotation at a later time with the help of an annotation queue. The only way for an image to enter the training process is if a corresponding label is included.

If the image is unlabeled
Creating labeled data sets quickly is the highlighted feature of Clara, as annotation can be done in minutes instead of hours. When creating the initial data sets, large batches of images will have to be annotated by radiologists. These batches can also be referred to as annotation queues, which can be further split up into organ and task specific annotation queues for different combinations, such as a queue for brain tumor MRI segmentation or a queue for chest X-ray classification. As of writing, Clara supports the tasks of classification or segmentation on

various organs.

Consider how the KGE searches for new images and creates these queues. The KGE executes loops looking for specific combinations of tasks, modalities, and organs. Searching and aggregating image results on specific combinations makes it possible to delegate different queues to different professionals so they only go through a queue they have expertise on. Deciding which combinations are the KGE searches for has to be a manual decision, as certain combinations of a task and organ perform better than others with today's best performing AI architectures, but this may change over the years. In practice, this means that certain task and organ models with low performance due to technical limitations are not yet suitable for a clinical setting, or resources are better spent elsewhere. If a new combination suddenly becomes feasible due to technological advancements, the manual decision to search for relevant images will be made and the KGE will include those images on future loops.

Once the loop is complete and all annotation queues are created, medical professionals will annotate the images using Clara Train to create associated labels. When complete, the labels are uploaded back to the Digital Twin, creating a new data set. The image will now be found and used for training the next time KGE executes.

If the image is labeled
Given an annotated image combined with the consent of donation by the patient, the image can continue in the pipeline to be used for training. Clara Train includes simple scripts to train models with optimized algorithms fitted for NVIDIA hardware, providing efficient computations and reduced time consumption. All Clara Train requires is the annotated data sets and settings for which parameters should be used under training, like loss function and learning rate. The data set will be split into training and validation sets, for example 80% for training and 20% for validation, but these values are modifiable.

The KGE will eventually have collections of annotated data sets that are ready to be used for training. For each set it checks to see if there is an existing model for this organ, modality, and task combination. If there is a match, the KGE initiates transfer learning to improve the existing model. If there is no existing model, a new one will be created.

Once a new model is trained or an existing model is updated, the KGE runs validation tests on the models. This is a check to evaluate how the accuracy of the new model compares to the old, in case the new model does not yield better results. The KGE updates the Knowledge Bank on which model is currently the best performing model for any combination of organ, mortality, and task.

3.4 Tools

Two tools that will be used for the results were chosen based on the research conducted. The first is the data set, and the second is a validation metric used to measure performance of AI models.

3.4.1 Data Set

The data set that will be used for experimentation is the Medical Segmentation Decathlon due to its extensive collection of various images across multiple modalities, tasks, and organs. In addition, this is the data set NVIDIA used to produce their pre-trained models in Clara, ensuring compatibility between the data set and software.

3.4.2 Dice Score

Validating the performance of AI models is important when trying to compare which model performs best and how well the model performs in general. The Dice score, also called the Sorenson-Dice coefficient, is a common metric used to measure performance of image segmentation. The Dice score is a measurement of how similar objects are, such as the similarities between two segmentations. A numerical value can be calculated between the similarity of the predicted segmentation and the label segmentation, also called the ground truth. The Dice score compares the overlap of the two segmentations divided by the total size.

$$DiceScore = \frac{2 \cdot TP}{2 \cdot TP + FP + FN} \tag{3.1}$$

The equation for the Dice score above shows how the calculation is made. TP is true positives, the total number of pixels with the same value in both segmentations. FP is false positives, the number of pixels which appear in one segmentation but not in the other. FN is false negatives, which is pixels that should have had a certain value but did not.

Chapter 4

Results

This chapter will first show a practical run-through of how Clara will be used to train AI models from medical images, and will then showcase a proposed design for a Digital Twin ecosystem.

4.1 Clara Run-Through

This section details the results obtained during the investigation of how Clara Train can be used to annotate images, train a new model, use transfer learning to update models, export models for inference use, and finally perform an inference test using a model.

Clara is split into two SDKs, Clara Train and Clara Deploy. The first focuses on creating data sets and models by assisting in annotation and training, while the second focuses on deployment in clinical settings, such as integrating with PACS and creating custom pipelines. As this thesis focuses on the annotation and training process, Clara Train will be used for the investigation.

4.1.1 Spleen Segmentation

To demonstrate available functionality, we will be training a model to perform volumetric 3D segmentation of the spleen from CT images. The spleen was chosen due to preexisting annotation and segmentation models for this task found in the NVIDIA NGC catalog. The images and labels are retrieved from the open-sourced Medical Decathlon Challenge.

Images along with explanations will be presented in the following subsections, but a video showing the annotation and training process is additionally provided in Chapter 4.1.8.

4.1.2 Annotating an image

Annotating images is most time consuming step and requires the most attention. NVIDIA has partnered with two open-sourced imaging viewers: The Medical Imaging Interaction Toolkit (MITK) and 3D Slicer. Both of these programs include plugins that connect the viewer to Clara's AI Assisted Annotation Server (AIAA). The added functionality allows the viewer to send the image to the server and receive a result for annotation and segmentation.

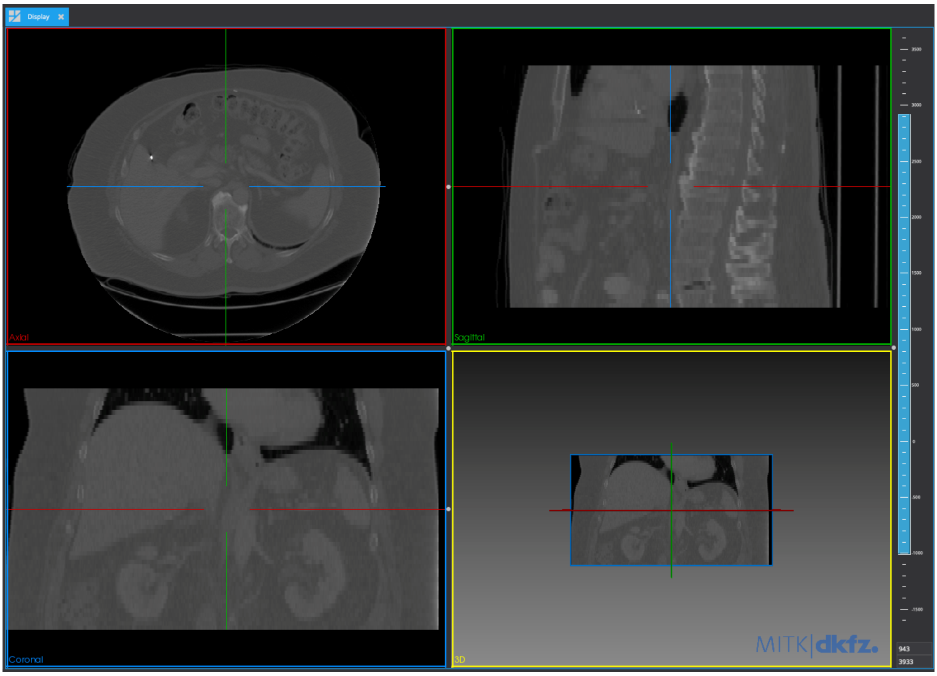

Figure 4.1: Initial CT scan

To begin annotating, we open an image in the MITK viewer. This figure shows a CT image including the spleen. The goal is to segment the spleen as a volumetric 3D object.

The red box is the axial view, the green box of the sagittal view, and the blue box is the coronal view. The yellow box is a combination of all these three views which shows the CT image in 3D. As this is a grayscale image it can be difficult to find where the spleen is. Normally, radiologists have to look for organs and tumors using these types of images in grayscale, with use of an AI model the relevant part of the image can be found immediately.

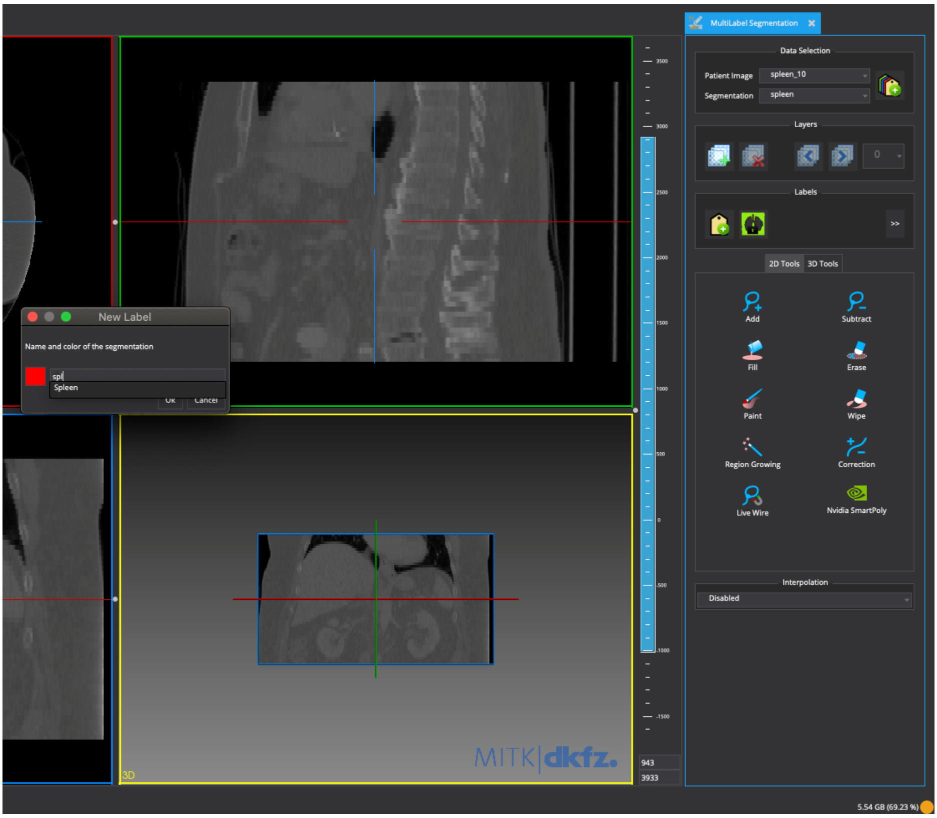

Figure 4.2: Choosing the label

After MITK is configured with a connection to the annotation server, we can create a new label for segmentation. We select the spleen as the organ to segment. If we were working with a different organ, this is where it would be selected.

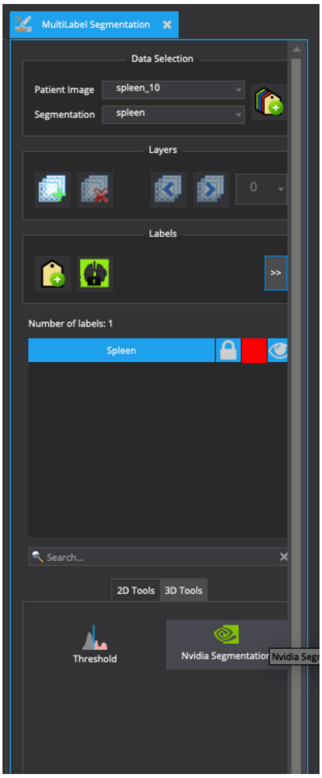

Figure 4.3: 3D tools

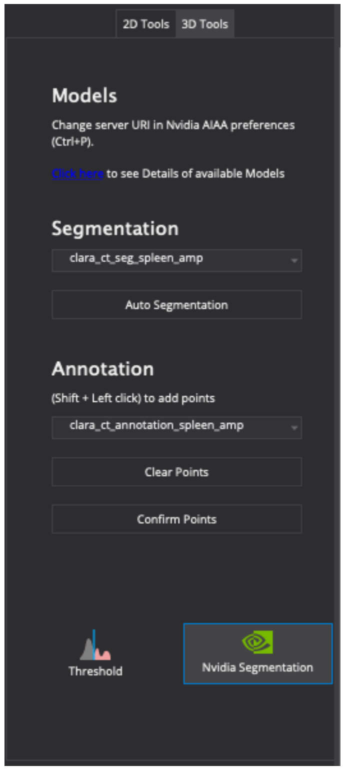

Figure 4.4: Model selection

Selecting a compatible organ enables the use of tools such as NVIDIA segmentation as seen in Figure 4.3. There are both 2D and 3D tools, where segmentation can be found under 3D tools. This is part of an extra plug-in module integrated to support AI annotation by MITK viewer.

In the settings for 3D segmentation tools seen in Figure 4.4, there will be a segmentation and annotation model if the relevant models have been preconfigured. These models have to be downloaded and configured in the annotation server before they can be used. For this example, we are using NVIDIA's pre-trained models that are available for public use. These models can be used as a baseline model, meaning they can be further trained with new data so developers do not have to start from scratch. First we perform an auto segmentation to receive an initial result. This sends the unlabeled data to the annotation server and triggers an auto segmentation.

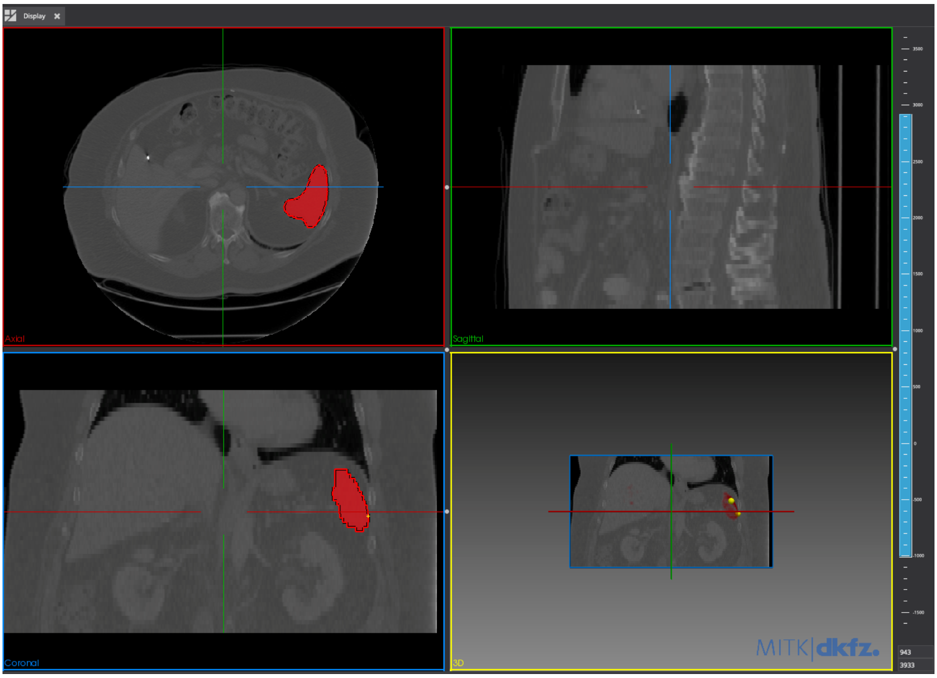

Figure 4.5: Segmentation result

This is the result of an auto segmentation. The benefit of performing an initial segmentation is to make it quicker for the annotator to select extreme points that will be used for the annotation model. The input required for the annotation model is a CT image combined with extreme points that locates the outer boundaries of the spleen, and the initial segmentation helps find these faster than if the radiologist has to find the extreme points of the spleen manually from a grayscale image.

The red outline shows the location of the spleen. The next step will be selecting the extreme points.

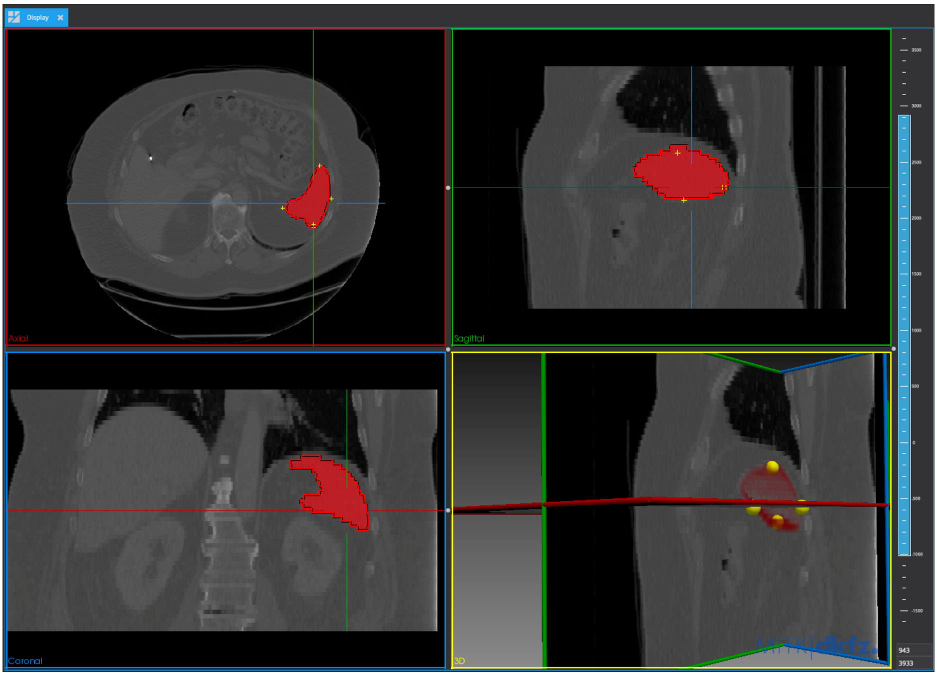

Figure 4.6: Selecting extreme points

The yellow crosshairs indicate the manually chosen extreme points, which can also be seen as the yellow bubbles on the lower right-hand 3D image. Note that these extreme points are not in the correct location as that would require expertise from a radiologist, these locations only serve as an example.

From here we execute the annotation model, this step sends the image along with the extreme points to the annotation server. This results in a more accurate annotation than the initial auto segmentation could do by itself. If the result is still not satisfactory, we can manually adjust each 2D slice as will now be demonstrated.

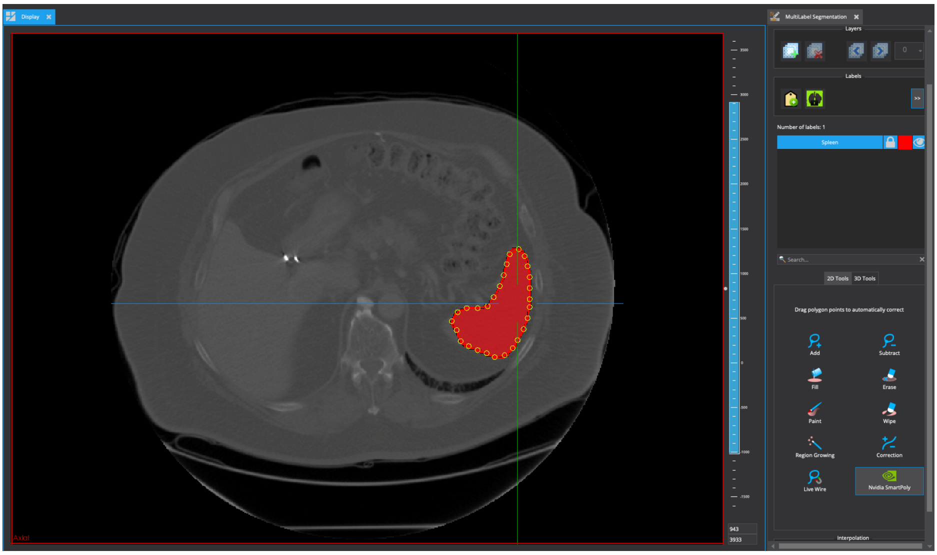

Figure 4.7: 2D tools

If the annotation was not satisfactory and needs to be corrected, this can be adjusted using NVIDIA'S SmartPoly 2D tools. Dragging the yellow circles to fit the spleen accurately fine-tunes the annotation.

The figure shows correcting the slice in the axial view, but the annotation can be corrected in the sagittal and coronal views as well.

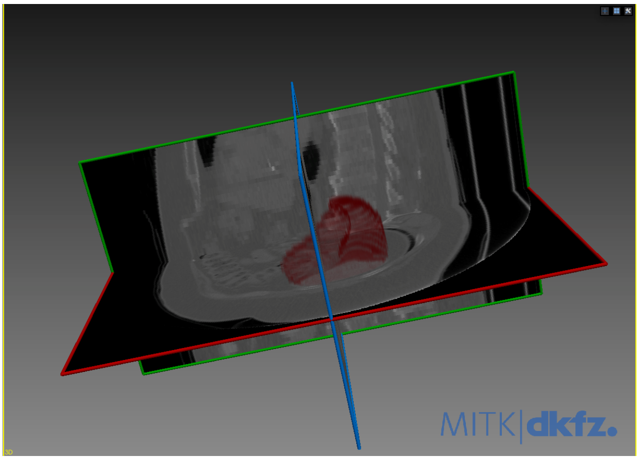

Figure 4.8: Final annotation with CT scan

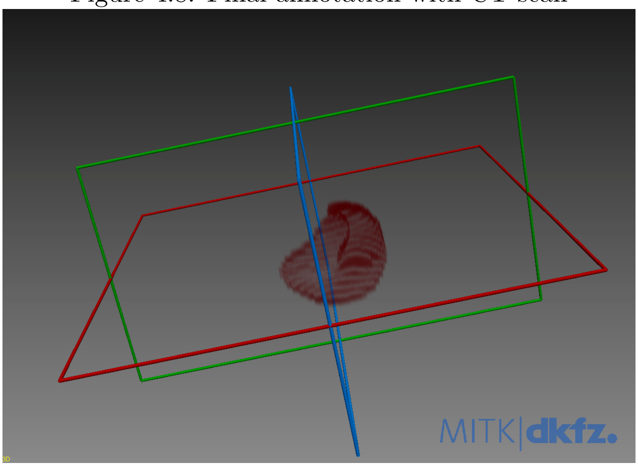

Figure 4.9: Final annotation by itself

The end result in Figure 4.8 is a volumetric 3D annotation of the spleen, as shown by the red surface model. This has been corrected using 2D tools. The label, along with the original CT image, will be used for training.

Disabling the view of the CT image leaves only the spleen visible in Figure 4.9. Segmenting the spleen, finding extreme points, and making adjustments in 2D tools is a quick process that can often be done in minutes.

4.1.3 Training a new model

After generating one or more labeled image as shown in the previous subsection, these can be used to train a completely new model using Clara's built-in script *train.sh*.

To use the training script, all that is required is a JSON file that gives the location of the training and validation sets. These sets are composed of an image and a corresponding label image. For this example we will be using four training images and two validation images from the Medical Decathlon Challenge.


datalist.json

```
{
  "training": [
    {
      "image" : "imagesTr/spleen_2.nii.gz",
      "label" : "labelsTr/spleen_2.nii.gz"
    },
    {
      "image" : "imagesTr/spleen_10.nii.gz",
      "label" : "labelsTr/spleen_10.nii.gz"
    },
    {
      "image" : "imagesTr/spleen_16.nii.gz",
      "label" : "labelsTr/spleen_16.nii.gz"
    },      {
      "image" : "imagesTr/spleen_38.nii.gz",
      "label" : "labelsTr/spleen_38.nii.gz"
    }
  ],
  "validation": [
    {
      "image" : "imagesTr/spleen_12.nii.gz",
      "label" : "labelsTr/spleen_12.nii.gz"
    },
    {
      "image" : "imagesTr/spleen_60.nii.gz",
      "label" : "labelsTr/spleen_60.nii.gz"
    }
  ]
}
```

Figure 4.10: Location of data sets

From here, training parameters are adjusted before executing training.

```bash
#!/usr/bin/env bash

my_dir="$( cd "$(dirname "$0")" >/dev/null 2>&1 ; pwd -P )"
. $my_dir/set_env.sh

echo "MMAR_ROOT set to $MMAR_ROOT"
additional_options="$*"

# Data list containing all data
CONFIG_FILE=config/config_train.json
ENVIRONMENT_FILE=config/environment.json

python3 -u -m nvmidl.apps.train \
    -m $MMAR_ROOT \
    -c $CONFIG_FILE \
    -e $ENVIRONMENT_FILE \
    --set \
    epochs=10 \
    learning_rate=0.001 \
    num_training_epoch_per_valid=1 \
    multi_gpu=false \
    ${additional_options}
```

Figure 4.11: Training parameters

Training is a time-consuming process, so for this example we start by only performing 10 epochs. This means going through the entire training data 10 times. Throughout training, there will be checkpoints where the current model is saved. Performing validation on these checkpoints will give an indication of the model's performance at the checkpoints model during training. Often, the best performing model will not be the most trained model, but rather one of the checkpoint models. Each checkpoint model is compared with the previous best performing checkpoint model, and if the checkpoint model performs better than the previous one, the old model will be overwritten with the new one. At the end of training, two models will be stored: the best checkpoint model and the final model.

```
epochs=10 \
num_training_epoch_per_valid=1 \
```

Listing 4.1: Training parameters

Listing 4.1 shows parameters for the number of epochs and how often to perform a validation check, respectively. Training runs through the data set 10 times, performing a validation check between each time. To generate more data and compare results with different parameters, we will also train models going through the data set 400 times, 1000 times, and finally 2000 times. The pre-trained model from NVIDIA is trained with 2000 epochs which can be used for comparison.

4.1.4 Updating an existing model

Once a model is generated, it is often desirable to update this existing model with new training data instead of training a new one from scratch. In this example we updated the model we just created, but NVIDIA provides pre-trained models online for many segmentation and classification tasks which can be used as a starting point for anyone. This is called fine-tuning the model, and executed with the *train_finetune.sh* script which has the same parameters as *train.sh* in Figure 4.11, but is initialized by loading the previous existing model and its weights and continues training on said model.

4.1.5 Exporting the model for inference

After a model completes training it can be exported. As previously mentioned, training outputs both the checkpoint model and the final model. These have to be exported to be in the correct format which can be used for inference. Executing *export.sh* chooses the best performing model which will then be exported into a frozen graph. The frozen graph contains only the model and its weights, instead of all the metadata that is saved along with the checkpoint models. A frozen graph cannot be further trained, it is only meant to be used for inference.

4.1.6 Validation of the model

Validation metrics can be generated from running validation on the validation data set. This computes the average Dice score, a metric that gauges the similarity between the original label and the inference result.

	Mean	**Median**	**Max**	**Min**	**90 percent**	**STD**
100 epochs	0.163	0.163	0.181	0.145	0.149	0.018
400 epochs	0.363	0.363	0.476	0.250	0.272	0.113
1000 epochs	0.936	0.936	0.940	0.933	0.934	0.003
2000 epochs	0.948	0.948	0.964	0.932	0.935	0.016
Pre-Trained	0.971	0.971	0.976	0.965	0.966	0.006

Table 4.1: Validation results

Table 4.1 shows the results of experiments conducted with different training parameters. Training for 100, 400, 1000, and 2000 epochs took approximately 20 minutes, 90 minutes, four hours, and eight hours, respectively. NVIDIA's pre-trained model was also trained with 2000 epochs, which received similar results to the one trained here for 2000 epochs.

4.1.7 Using the model for inference

Testing the model for inference is possible before it has been deployed to the production software which will then be integrated into a clinical setting . This can be done by simulating an inference result in a development environment. This saves time by eliminating the need to export the model into the production environment every time there is a need to test the generated model.

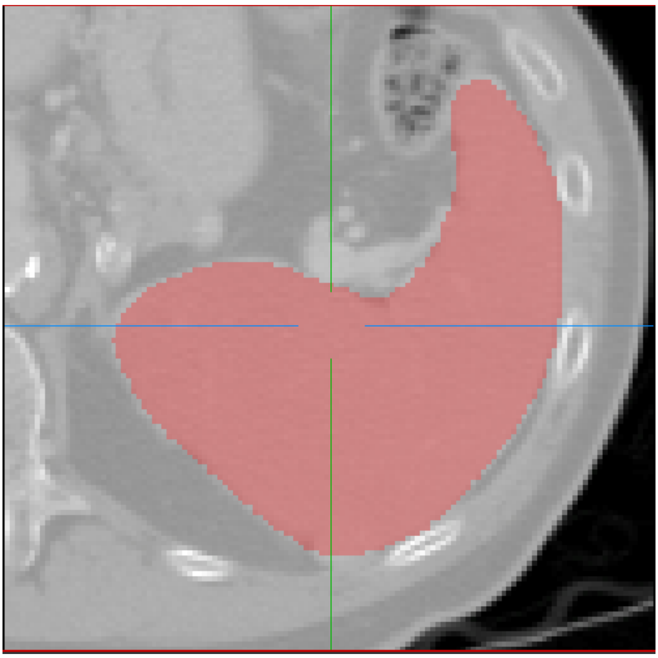

Figure 4.12: Correct label

Figure 4.12 shows the correct label for the inference example that we are working with. The chosen image is a spleen segmentation that the model has never seen while training, simulating a realistic example the model has to be able to infer. The red outline shows the spleen in the axial view on top of the CT image.

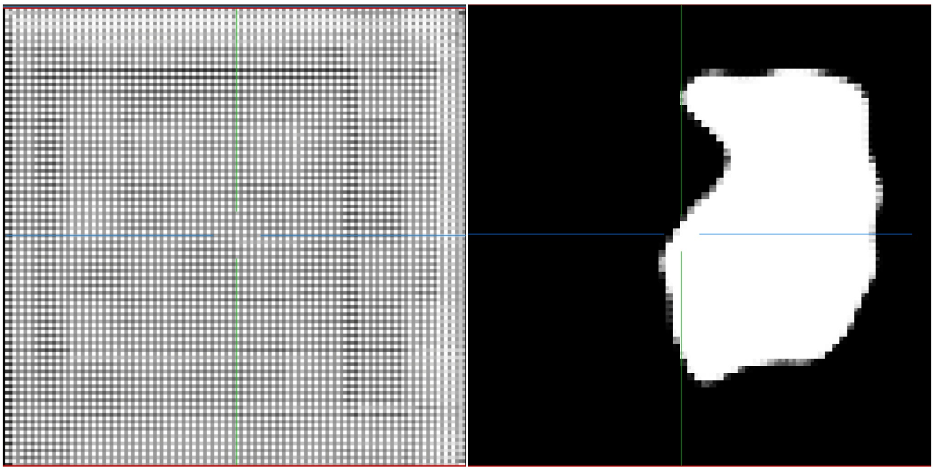

Figure 4.13: Inference result for 100 (left) and 400 (right) epochs

The inference result from the model trained for 100 epochs did not result in an acceptable segmentation. It failed to recognize even a general outline, and instead produced checkered boxes of black-and-white. This was not a surprise as the validation metrics for this model were poor, with a mean of 0.163.

While the model trained for 400 epochs succeeded in providing a general outline, it still failed to segment the correct shape of the spleen. The borders of the spleen are also of low resolution, showing that the segmentation model is struggling with identifying the outlines.

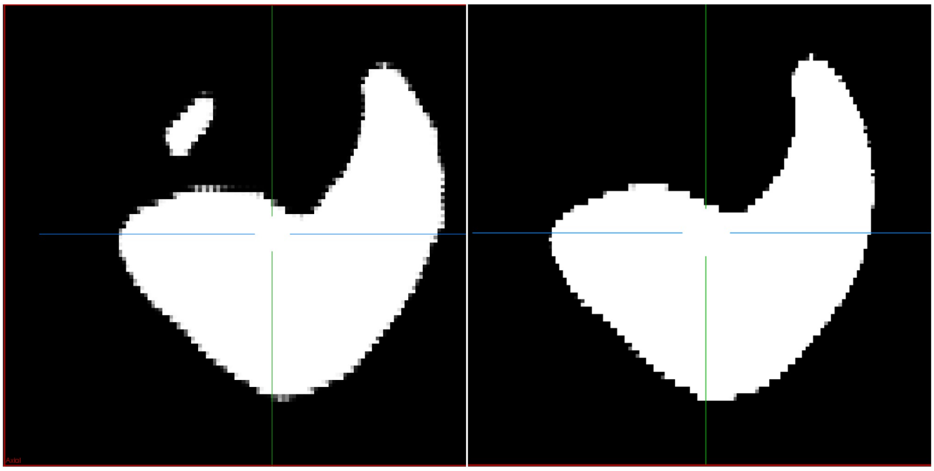

Figure 4.14: Inference result for 1000 (left) and 2000 (right) epochs

Both inference results received after training models with a high number of epochs gave satisfactory results.

The 1000 epoch model succeeded in finding the general outline of the spleen, but wrongfully segmented an incorrect part of the image that it thought was the spleen, as shown by the floating shape in the top left corner.

The 2000 epoch model did not produce this error, in addition it succeeded in providing a higher resolution segmentation around the edges.

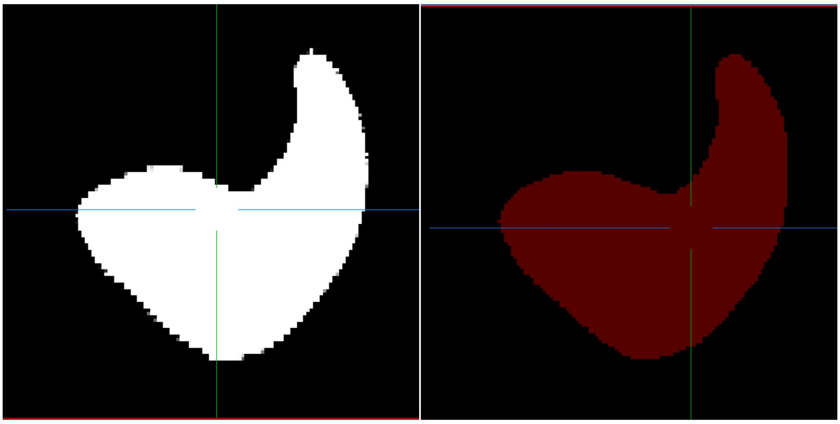

Figure 4.15: Inference result (left) versus label (right)

Comparing the result of the 2000 epoch model with the original label shows a very positive result. The Dice score validation metric was computed to 0.964, a very satisfactory result.

Although only four training images were used, a model that only took eight hours to train was able to segment an unseen spleen image with very high accuracy.

4.1.8 Video example

A video detailing the annotation and training process can be found here:

https://vimeo.com/428781417

The video is a reiteration of the images and contains no new information, but may be easier to follow along.

4.1.9 Combining Clara with the KGE

A discussion around combining Clara with the KGE is appropriate after getting hands-on experience with the annotation and training technology.

Integrating training with the KGE is a straightforward process due to the flexibility Clara provides related to training. After annotated data sets are generated from the KGE and it is time to either train new or update existing models, all the KGE has to facilitate is creating the JSON file with data set names which points to the images that will be used for this training session. For example, if 24 new annotated images were created for brain tumor segmentation, a script would be run by the KGE that compiled all 24 images and all 24 labels into the format seen in Figure 4.11. Not all 24 images would be used for training, as common practice includes splitting the data set into training and validation partitions. 18 images could be used for training while the remaining six could be used for validation. This ratio of training and validation is entirely up to the developer, an 80:20 split is a common guideline. The ratio could also be dependent on the total amount of training images, as having only two images would make it efficient to use both of them for training, so a mathematical equation to determine a reasonable ratio could be developed to split the data set into appropriate sizes depending on the total amount of images.

After the script is run and the JSON file pointing to the location of all the current images is generated, the KGE can initiate training without having to change any other settings. The number of epochs typically remains constant between training sessions but this parameter can be changed if desired.

Once training is complete, the KGE can execute the remaining scripts to export and validate the new model. The KGE can run the validation script to compare the validation metric results of the newly created model with the previous existing one. If the validation results of the new model are lower than the previous, the already existing model will still be marked as the best performing model in the AI model register responsible for keeping track of the best available model at any given time. If the new model provides better results, the exported model along with its validation scores will be uploaded to the servers while simultaneously updating the register pointing to the best available model.

Performing an inference result during the KGE training cycle is not a necessary step, but is still useful for developers who may wish to experiment with adjusting parameters of training to optimize the creation of the best possible models.

If an entire DGX-2 cluster was dedicated to training medical imaging AI models, this would enable 16 different models to be train at the same time, as the default cluster comprises of 16 graphical processing units each. Clara includes the option to train using two GPU's simultaneously, speeding up total training time. Given a scenario where multiple GPU's are not scheduled for training over a certain period of time, the KGE could perform a check to determine if training using two GPU's is possible.

Concluding the training section, it is apparent that Clara with its easy to use scripts is capable of seamlessly integrating with the KGE.

4.2 Digital Twin Design

This section will provide pseudocode for some of the main components of the system followed by a high level description of the Digital Twin design and how it can integrate with hospitals and Clara.

4.2.1 General scope

The goal for the Digital Twin is to keep a record of all medical data for every citizen. As technology advances and and it is possible to gather more health metrics on an individual, the data stored in a Digital Twin will continuously be expanding. For example, the rise of wearables introduces the possibility of gathering health data during every second of the day. In the future these devices will improve in being able to gather more types of data while increasing the quality of the data possible to record.

For medical imaging to be integrated in the Digital Twin solution, functionality for storing these new types of data will have to be implemented. Investigating Clara resulted in a clear set of possibilities and limitations that the Digital Twin will have to consider.

4.2.2 Digital twin pseudocode

The Digital Twin pseudocode contains the classes and methods related to the Digital Twin and medical images. Each instance of a Digital Twin can be tied to any number of medical images, whether that be zero or hundreds.

```
# Digital Twin class
Class DigitalTwin:
    name
    height
    weight
    birth_date
    family_members
    donation_status
    images

# methods:
# returns all instances of the MedicalImage class for a given twin
get_medical_images(DigitalTwin)

# medical image class
Class MedicalImage:
    image_file
    label_file
    task_type
    organ
    modality
    image_dimensions
    equipment_info
    donation_status
    notes

# methods:
# methods to add or remove an image tied to a DigitalTwin
add_image(DigitalTwin(MedicalImage))
remove_image(DigitalTwin(MedicalImage))
```

Listing 4.2: Digital twin pseudocode

4.2.3 Knowledge Generation Engine pseudocode

The Knowledge Generation Engine pseudocode details how the KGE extracts information from Digital Twins to build data sets and train AI models.

```
# begin by making Clara functions available
import Clara SDK as clara

# this function checks the model register and
# returns either None or the current best model
get_current_best_model(task)

# data variables
training_dataset
annotation_queue
JSON_file


# this function generates a training data set and
# annotation queue while taking a given task
# as input, such as spleen segmentation
function generate_knowledge(task):
    # loop through all Digital Twins
    for twin in DigitalTwins:
        # search twins consenting to donation
        if twin.donation_status:
            # create a list of images
            # that have yet to be donated
            unused_images
            for image in get_medical_images(twin):
                if donation_status == false:
                    append image to unused_images
                    # set donations status to true
                    donation_status = true
            for image in unused_images:
                # image is added to training set if label exists
                if image.task_info = task and image.label_file:
                    append image to training_dataset
                # image will be added to annotation queue
                # if label does not exist
                if image.task_info = task and not image.label_file:
                    append image to annotation_queue


# this function generates the JSON file
```

```
# locating the paths of the images, as required by Clara
function generate_JSON_file(training_dataset):
    # split data set into a training and validation set
    training_set = 80% of dataset
    validation_set = 20% of dataset
    append "training" header to JSON_file
    for pair in training_set:
        append image.image_file to JSON_file
        append image.label_file to JSON_file
    append "validation" header to JSON_file
    for pair in validation_set:
        append image.image_file to JSON_file
        append image.label_file to JSON_file


# this function executes the Clara training scripts
function train(JSON_file, task):
    # if an AI model already exists for this
    # task it will be updated with train_finetune.sh
    if task exists in model_register:
        previous_model = get_current_best_model(task)
        clara.train_finetune.sh(previous_model, JSON_file)
    # a new model will be trained using train.sh
    # if there is no pre-existing model
    else:
        clara.train.sh(JSON_file, task)
```

Listing 4.3: Knowledge Generation Engine pseudocode

4.2.4 Decision support pseudocode

The decision support pseudocode contains the function that will be implemented into the PACS pipeline. After a medical image is taken in the hospital, it is sent to the PACS server for storage. This pipeline is adjustable and automatic inference is therefore possible to integrate. The decision support system takes the medical image as input and checks the register of AI models to localize a model that supports inference of the task, such as spleen segmentation. If a compatible model is found in the register, Clara runs inference on the image using said model. The output is an inferred result of the task which is also stored in the PACS server. A radiologist views the results with the intention of using it as decision support for a diagnosis while simultaneously manually checking correctness of the model.

```
# begin by making Clara functions available
import Clara SDK as clara

# this function checks the model register and
# returns either None or the current best model
get_current_best_model(task)

# this function will be attempted to run on
# every medical image taken
function infer_image(image):
    task = image.task_type
    model = get_current_best_model(task)
    if model == None:
        raise error: Missing model for this task
    # uses Clara to run inference using the
    # appropriate model
    return clara.infer(model, image)
```

Listing 4.4: Decision support pseudocode

4.2.5 Design requirements

A Digital Twin will in essence be a data entry in a database where the data entry is connected to a citizen, with each citizen connected to different types of medical data. From before a citizens birth, a database entry will be created for them and already begin storing medical data.

An appropriate solution for realizing the technical requirements is creating a cloud server that hosts the database of Digital Twins. This server will contain APIs that support uploading and downloading of data to relevant actors. Figure 4.16 outlines these actors that would interact with the server, hosting endpoints which the PACS system can send (POST) images to, and endpoints the Knowledge Generation Engine can request (GET) data from.

Medical images vary in file sizes depending on modality and resolution, ranging

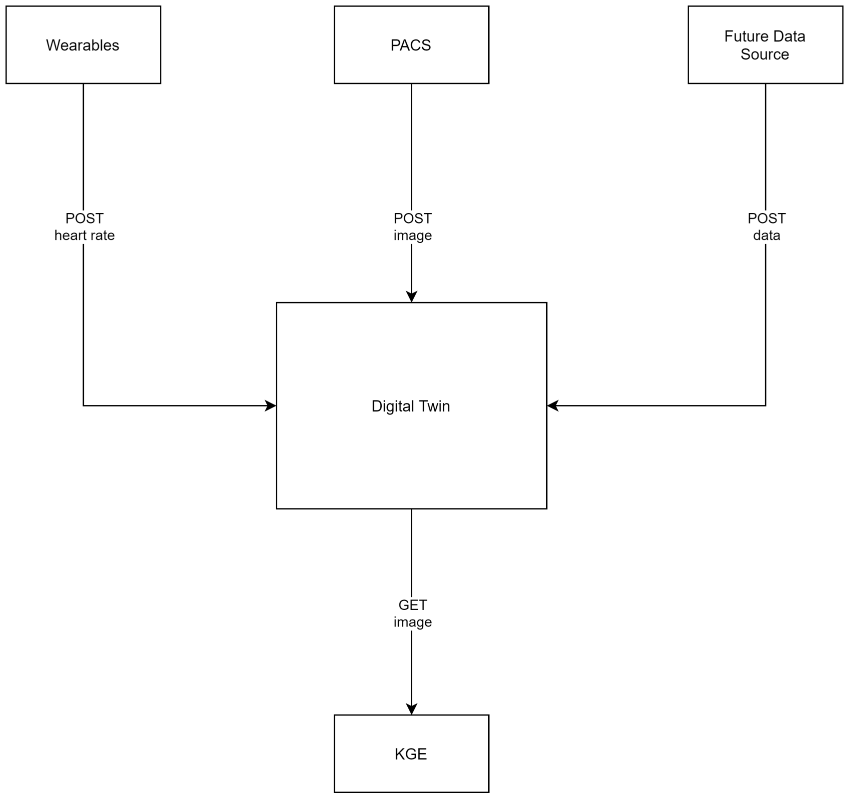

Figure 4.16: Digital Twin Endpoints

from 100 kB to 30 MB for the modalities researched for this thesis. Other modalities such as real-time 3D used for ultrasound are larger in file size. The label to these images are under a megabyte, less than a fraction of the original image. The frequency of scans normal patients undergo every year is low, so storage requirements related to images is unlikely to be an issue given that the patient does not undergo an abnormal amount of scans.

Other potential digital metrics such as heart rate, blood pressure, and lab results from the hospital all rely on text and numbers. Images propose new challenges, like supporting viewing functionality and image exports from a Digital Twin viewing dashboard. Dental records is another example of images that eventually could be stored in a Digital Twin, so image functionality is inevitably necessary for a fully fledged Digital Twin.

4.2.6 Hospital requirements

Public hospitals have to integrate the new technical requirements necessary to support Digital Twins. Currently, hospitals differ in their IT solutions, but for a consistent unified system to be possible throughout the country they would all have to revise areas of their IT systems.

To ensure consistency, a potential direction to take is to hire Digital Twin project leaders to organize an investigation that outlines the current systems of every hospital that will use Digital Twins. This would provide the insight needed to design and overhaul the technological solutions that will serve as a foundation for integrating Digital Twin functionality. The project leaders could either develop a guideline of requirements every hospital would have to implement themselves or they could assemble teams to travel around the country and be responsible for facilitating the technological integration.

For medical imaging, introducing an automatic storage step in the process of taking medical images is the most important feature to implement in hospitals to realize data mining functionality. For every patient undergoing a scan, an automatic integration between the scanning equipment and the Digital Twin is key. Figure 4.17 presents the envisioned workflow from taking an image to having a result for the radiologist to view.

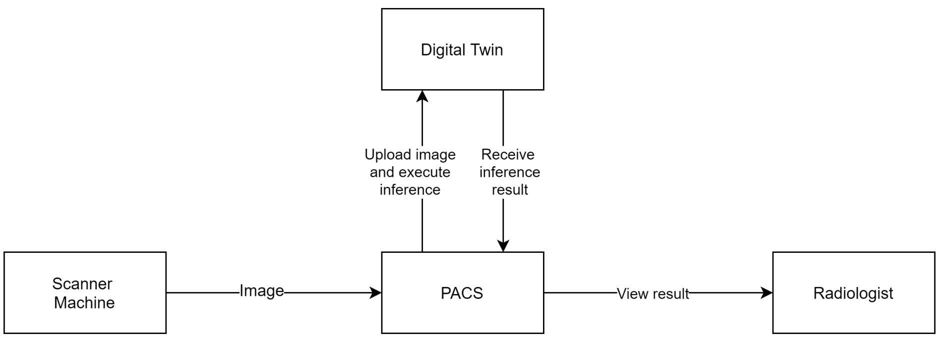

Figure 4.17: Hospital Workflow

The radiologist or radiographer ensures the scanning machine schedule is tied to the patient's Digital Twin. After a scan is completed it enters the PACS system and automatic uploading is possible after this step. The PACS is programmable to support the task of uploading a medical image to a patient's Digital Twin. Uploading will be done by posting the scan and the task to the patient's Digital

Twin API endpoint. This triggers automatic segmentation or classification within the Digital Twin platform if there is an AI model for the given task and the PACS system receives a result for the radiologist to view. Hospitals will have the necessary credentials to gain permission to upload and download from the Digital Twin server to ensure security.

After an initial development of the software necessary to facilitate the functionality required for automatic uploading is complete and the software has been correctly implemented in all hospitals, minimal maintenance is required for sufficient upkeep. All medical images onward are automatically uploaded to a patient's Digital Twin.

4.2.7 Potential future use cases

As every Digital Twin will store as much medical data as possible, the collection of large amounts of data could potentially be used for new use cases in the future. Documenting blood pressure every day for multiple years may not be of significance today, but future research may introduce new methods of diagnosing or preventing diseases based on the collection of medical data we already have access to. This new research could create possibilities where every Digital Twin could have the potential to use data stored over many years and run the relevant data through the new research method and receive immediate results, all without having to visit the hospital. This example shows that even if storing all medical data as of today may be redundant, there is still potential for everything to be used in the future.

Medical research will also face new possibilities with the introduction of abundant digital health data. A citizen can be given the option to donate all their medical data to research, and receive notifications for what their data was used for and even if they contributed to saving other lives. Researchers gain the possibility to conduct a higher number of studies, as data collection from test subjects all over the country is now a possibility. Data scientists can research trends and discover many new findings that have the potential to increase national health. Studies can select between location, gender, age, and many other filters that can show correlation which may lead to new findings.

Chapter 5

Discussion

This chapter will discuss findings from the results and reflect on implications these may have on future development of a Digital Twin ecosystem. The research questions are discussed followed by additional reflections on potential challenges.

5.1 General

The findings from the results made it possible to determine the feasibility of a Digital Twin ecosystem. The investigation made it clear that Clara is ready to be used for annotated data set generation, and the Digital Twin design gave insight into what components need to be in place for the ecosystem to function. The short time needed to train the models coupled with the high performance obtained made it apparent that generating large annotated data sets is possible for a wider audience than initially anticipated, based on the usability of the software and performance of the models generated. The Digital Twin design and pseudocode for the various components were useful to get a better understanding of the scope required to eventually develop such a system, as it contributes to the overall feasibility by outlining how the scope of the individual components are reasonable to develop without immense resources.

5.2 Research Questions

The research questions are now able to be discussed in detail after conducting the investigation.

5.2.1 Research question 1

Is it feasible to employ the NVIDIA Clara suite to easily and quickly annotate and train AI models for segmentation and classification of medical images?

Annotation using Clara is the most important feature needed for data set creation, and from the investigation conducted it became apparent that NVIDIA was able to succeed in relevant areas like integration tools, technical capability, and usability. The integration between the open sourced image viewers and the annotation server function seamlessly, only requiring a simple set up and never causing errors during use. The added user interface functionality in the viewer was simple and concise, containing only the bare necessities instead of unnecessary confusion. The simplicity of the process lends a hand to the reasonably manageable learning curve for annotation. Sending images to and from the annotation server is still a minor drawback, as segmentation and annotation computations in addition to sending the image over a network may take up to two minutes, leaving the annotator waiting. Fine tuning tools were intuitive to use and decrease the amount of work necessary to change the annotation, as only small changes need to be made from an initial starting point. From a technical and usability perspective, annotation is feasible on both small and larger collections.

A discussion on feasibility needs to assess include a wider range of factors in addition to the technical aspect, most importantly the time required for training. The time required to train an acceptable model using only four images took eight hours, while normally a model would be trained with substantially more images and consequently require more training time. The Medical Decathlon Challenge trained their best model using over 50 images, which is a realistic number of images that could be generated on a weekly basis. Mapping out the total number of models desirable to clinically use along with historic data on how many medical images are donated per week are metrics necessary to estimate the time feasibility. A brute force solution is simply to purchase additional DGX-2 clusters, but this may not be necessary. Top-performing models of today use a relatively small amount of images for training, therefore it may be beneficial for model performance to not only train with a small number of images for every model update, but by extension also training for a shorter amount of time. Training time also changes dynamically, the initial model will require more time compared to incremental model updates. Training models is feasible from a technical perspective, but unforeseen real world implementation factors will decide if training continuously over time without generating a backlog is possible.

Not all combinations of tasks and organs are ready for the production environment as of today. While spleen segmentation is a test proven to be successful, a select few task and organ combinations still have some technological improvements required before they can be used as a reliable decision-making tool. This limitation should not be a hindrance to begin the development of data sets and clinical integration, as research for improved model architectures continuously happens and will likely improve to the extent where clinical integration for all tasks and organs is likely to become possible.

Questions surrounding usability challenges were also answered with findings that support the feasibility of Clara. The professional annotators are first and foremost radiologists, not AI developers. The annotation process needs to focus on ease-of-use and a reasonable learning curve for it to be feasible for these doctors. Using the two most popular medical imaging viewing software instead of developing their own viewer is an advantage for usability. Annotators will have familiarity with navigating the software and it will eliminate the need to learn a new user interface.

Investigating the use of Clara for annotation resulted in insights into how efficient AI assisted annotation can be with the use of these powerful third-party solutions, suggesting the software is ready for enterprise application. Creating large data sets with this powerful tool has proven to be a very feasible task to complete, even for users who are not experienced within the field of AI. Abstracting the intricate details of training AI models while simultaneously retaining the option to adjust important parameters makes Clara a feasible tool for even novice developers, lowering the technical and financial threshold for potential clients to develop AI models that are ready for clinical integration today. The largest bottleneck of enterprise AI is effectively eliminated, where only hospitals with an exceptional amount of resources and expert data scientists are able to create the large, high-quality data sets necessary for sufficient training of AI models.

5.2.2 Research question 2

How should a Digital Twin be designed, and how can NVIDIA Clara be used in *conjunction with the twins to automate model training?*

The pseudocode developed in the results made it apparent that looping through Digital Twins is bound to be a frequent event, further confirming the necessity of an efficient database design to allow search optimization.

Through the investigation it was apparent that Clara supports a wide configuration of integration possibilities with any first party or third-party solution from the openness of the API, from annotation to training to deployment. This is the biggest selling point of Clara and will enable the system to be compatible in an extensive range of environments.

Creating internal workflows using the KGE and the Digital Twins to create data sets and utilizing the Clara functionality is not only possible but a realistic achievement in a medium-term time frame.

The main challenge lies in physically integrating the new data storage solution of Digital Twins in hospitals and transitioning from current systems.

5.2.3 Research question 3

How can NVIDIA Clara combined with the Digital Twin concept be integrated into the radiology workflow, and how can it be useful to patients, hospitals, and researchers?

The radiology workflow and surrounding actors will inevitably see changes from automated diagnoses. After AI model integration is complete, time spent manually examining images on a day-to-day basis will see a substantial reduction from radiologists. The time recovered can be spent doing more difficult tasks and allow for more patients. Patients enjoy a shorter waiting time and quicker diagnosis, in addition to a universal online medical archive that simplifies the process of accessing their data such as medical images. Getting a second opinion from a separate hospital will be streamlined, giving them access to remotely stored patient's data opposed to manually requesting images from the first hospital. Researchers benefit from a larger pool of data available for research, in terms of both the amount of data and the number of patients choosing to donate.

Hospitals are introduced with new benefits but also new challenges. Smaller hospitals may notice AI inference to lower the burden of having a smaller radiology staff, exploiting the same level of high-quality expertise that larger hospitals have. Because of differences in hospitals, inconsistencies can cause problems with Clara and Digital Twin integration.

Differences in technology solutions from hospital to hospital may become a challenge if different equipment is used to take images. Most hospitals use PACS to store medical images, so while the software additions of adding a step in the PACS pipeline to automatically infer a result for all images is universally possible, differences in scanning equipment could cause a challenge. CT and MRI scanners differ slightly between manufacturers, which may not be a problem for a radiologist manually diagnosing an image, but a computer expecting consistent image formats in terms of resolution and image properties may struggle from even minuscule differences. In practice, this may result in incompatibility between the AI models and the different scanners from various manufacturers. If the AI model cannot adapt and create a one-size-fits-all solution for all scanning machines, then machine specific AI models will have to be created. This challenge is very feasible to overcome, as all training images come with metadata containing the details of how the image was created, such as the exact type of scanner used and technical details of the image. If necessary, the KGE can be programmed to divide training images into manufacturer specific data sets, and multiple machine specific AI models would be trained for a certain task and organ combination instead of a single model. Determining if machine differences need to be taken into consideration is unknown and requires extensive testing across all potential hospitals that aim to integrate AI inference in their workflow, but the solution of creating machine specific models is already a realistic workaround.

The intent is to initially only use the AI models only for decision based diagnosis, not a complete diagnosis by itself. Doctors still need to personally diagnose images themselves, using the inference results as technological help. A potential risk that needs to be overlooked is the case where doctors eventually begin to trust the AI models to the point where they themselves pay less attention to making sure that the AI diagnosis gave the correct results. Even though the models could achieve an accuracy of over 95%, this could lead to a false sense of trust in the AI models from doctors who may become too accustomed to agreeing with the AI diagnosis. As machines having the option to affect a human's life is a very intricate and potentially controversial practice, a scenario where a patient is misdiagnosed due to a false prediction from the AI model and a careless radiologist not catching the mistake would have the potential to dissuade public opinion on the use of AI-based decision-making for medical imaging. If citizens lose trust in the system, hospitals may be forced to restrict or remove the use of the AI models, reverting back to manual diagnosing.

5.3 Reflections

Reflecting on the results and research questions sheds light on challenges that may affect the proposed ecosystem. The challenges related to a successful implementation of a Digital Twin ecosystem and taking advantage of NVIDIA Clara lies in a number of places.

The entire workflow integration and management of tasks will be the bottleneck. A streamlined solution for continuously annotating images will require extensive planning and coordination. As impressive results can be achieved with a low number of training images, a cost-effective solution would be to first outsource the annotation task to professionals who will generate large batches of initial training data, and introduce live annotation of the inferred AI results in a clinical setting at a later time. While continuous live annotation after every inferred AI results by every radiologist at every hospital is an efficient long-term solution to generating the largest medical imaging data sets ever seen, in the short term simply generating smaller data sets by professionals annotating in a batch oriented workflow will suffice in creating ready to deploy AI models.

Another factor that needs to be taken into consideration is how long NVIDIA will be supporting Clara. While NVIDIA is very invested into powering the future of medical imaging through its AI solutions, there is always a possibility that things do not go according to the vision they have today. NVIDIA faces a risk of bankruptcy like any other company. Competitors may introduce similar solutions that outperform Clara, and if NVIDIA fails to compete they will lose customers, inevitably forcing them to abandon their product. Currently, NVIDIA is at the forefront of graphical processing units and researching their potential integration into medical imaging so this risk is currently low. If the case of a competitor outperforming the performance of Clara occurs, it is likely that the competitor will facilitate the transition to their own product, and any similar products are forced to adapt to existing technological solutions in hospitals, such as PACS, so that a transition to a new product may not be as costly or difficult.

Successful and high-performing AI models are contingent on being able to train on high quality, correctly annotated data sets. Given the scenario where a professional quickly skims through the annotation process, there may be incorrect labels that the AI models will use for training. While a low number of mislabeled images will not noticeably change the performance of a model, a consistent stream of poorly labeled images will result in the AI model consistently inferring incorrect results. The professional annotators that are given the task of creating the initial data sets will naturally understand the importance of accurate annotation, but over many years and with the introduction of live annotation by radiologists on a day-to-day basis, there is a likelihood of deteriorating annotation quality when this process becomes routine, creating the human challenge resulting in reducing the mindfulness of the professionals conducting the annotation practice. Routinely checking the quality of data set generation can monitor this risk and introduce measures that will take place if the annotation process needs quality improvements. This challenge is also minimized when the KGE checks newly trained models to determine if they are better than the existing ones, so given a scenario where there is a batch of low-quality annotated data sets that deteriorates the performance of an AI model after training, this new version would simply not be deployed.

The initial planning and development of the end system will take the longest time and require the most effort. Long-term planning will need to be conducted for this solution to be successfully integrated. Usability often tends to be the reason for a solution to either be successful or its downfall. If the entire workflow and usefulness is not satisfactory from a usability perspective, hospitals and doctors will not be inclined to either integrate the system at all or could eventually abandon the use of AI if usability becomes a hindrance rather than a tool. Performing an extensive investigation into how the day-to-day applications would realistically be used is therefore an essential process where all potential actors of every use case will have to thoroughly document requirements and potential challenges that could occur. While the technology is here for successful AI based decision-making, it will not be useful if the human factor is not carefully considered from a usability perspective.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

Radiology will inevitably go through fundamental changes in the coming years as AI advances and becomes commonplace in modern hospitals. Tasking computers with performing mundane time-consuming operations such as segmentation will free up valuable time for radiologists while simultaneously allowing more diagnosis to be made. This increase will lead to earlier diagnosis and an increase in chance of survival for many patients. This thesis made progress in investigating potential solutions and identifying their limitations for solving the biggest challenge of medical imaging AI: how to create an autonomous system purposed with generating high quality annotated data sets used for training AI models by utilizing medical data donated from Digital Twins.

In this project, NVIDIA Clara was used to demonstrate how medical images can be annotated using powerful supercomputers in a fraction of the time previously needed with the use of AI assisted annotation. Annotated images were then used to train AI models which was later tested for performance, achieving satisfactory results. A Digital Twin design was produced detailing requirements on a technical level as well as identifying potential possibilities and limitations concerning hospitals and other future use cases. Pseudocode and flow diagrams were created for applications in the clinical setting and the Knowledge Generation Engine. Future challenges and future possibilities that the Digital Twin ecosystem would need to consider were mapped out, such as the biggest obstacles for adoption to happen and future research possibilities.

6.2 Future Work

Improvements and more research can be made in many areas on the path to creating a Digital Twin ecosystem.

6.2.1 Proof of concept product

A proof of concept product can be created by realizing the pseudocode components in the results section. A basic Digital Twin platform that is ready to receive and store medical images combined with sample data would be a sufficient starting point, this would allow enough test data to create realistic scenarios for the Knowledge Generation Engine to work with. These two components can be used as a starting point to test an automatic loop designed to look for medical images in Digital Twins, create annotated data sets, and use those sets to train AI models. This proof of concept can be developed as a standalone solution without any hospital integrations as the AI models are created independent of clinical settings.

6.2.2 Hospital deployment

Hospital deployment is likely to be the task which needs the most time and requires most resources, as discussed previously. Successful hospital integration depends on extensive research on the equipment currently used in hospitals today to get a better understanding of how Clara needs to interface with the current technology. Clara Deploy would be the technology framework used to interface with the PACS systems. The work would consist of visiting hospitals, mapping out current solutions, becoming familiar with Clara Deploy, and eventually creating a proof of concept product. This proof of concept would attempt to automatically upload images to the Digital Twin platform from the scanning machines, run inference on the images, and display the result on existing workstations in the hospital.

6.2.3 Complete prototype and optimization

Once a proof of concept has been created and hospital deployment has been researched, a complete prototype can be put together from the individual components. A full loop would allow optimization to begin as well, such as implementing routines for when annotation should occur and who should be responsible, how often AI models should be trained or updated, and how well they perform in real life. Reaching this stage is important to begin more research into the effectiveness of the entire solution, like tracking AI model performance metrics over time. As the Digital Twin and AI ecosystem is dependent on citizens donating their images, long-term research on effectiveness of AI models in radiology is critical to persuade more citizens into donating.

Bibliography

[1] Review of Deep Learning Algorithms for Object Detection. URL https://medium.com/zylapp/review-of-deep-learning-algorithms-for-object-detection-c1f3d437b852.

[2] Ahmed Hosny, Chintan Parmar, John Quackenbush, Lawrence H Schwartz, and Hugo JWL Aerts. Artificial intelligence in radiology. *Nature Reviews Cancer*, 18(8):500–510, 2018.

[3] RB Altman. Artificial intelligence (AI) systems for interpreting complex medical datasets. *Clinical Pharmacology & Therapeutics*, 101(5):585–586, March 2017. doi: 10.1002/cpt.650. URL https://doi.org/10.1002/cpt.650.

[4] X-rays. URL https://www.nibib.nih.gov/science-education/science-topics/x-rays.

[5] Computed Tomography (CT). URL https://www.nibib.nih.gov/science-education/science-topics/computed-tomography-ct.

[6] Magnetic Resonance Imaging (MRI), . URL https://www.nibib.nih.gov/science-education/science-topics/magnetic-resonance-imaging-mri.

[7] MRI (Magnetic Resonance Imaging) — FDA, . URL https://www.fda.gov/radiation-emitting-products/medical-imaging/mri-magnetic-resonance-imaging.

[8] Ultrasound. URL https://www.nibib.nih.gov/science-education/science-topics/ultrasound.

[9] Nicola H Strickland. PACS (picture archiving and communication systems): filmless radiology. Technical report.

[10] DICOM Standard. URL https://www.dicomstandard.org/.

[11] Weibin Wang, Dong Liang, Qingqing Chen, Yutaro Iwamoto, Xian Hua Han, Qiaowei Zhang, Hongjie Hu, Lanfen Lin, and Yen Wei Chen. Medical Image Classification Using Deep Learning. In *Intelligent Systems Reference Library*, volume 171, pages 33–51. Springer, 2020. doi: 10.1007/978-3-030-32606-7_3. URL https://link.springer.com/chapter/10.1007/978-3-030-32606-7{_}3.

[12] Neeraj Sharma, Amit K. Ray, K. K. Shukla, Shiru Sharma, Satyajit Pradhan, Arvind Srivastva, and Lalit Aggarwal. Automated medical image segmentation techniques, jan 2010. ISSN 09716203. URL /pmc/articles/PMC2825001/?report=abstracthttps://www.ncbi.nlm.nih.gov/pmc/articles/PMC2825001/.

[13] Medical Segmentation Decathlon. URL http://medicaldecathlon.com/.

[14] Description - CHAOS - Grand Challenge. URL https://chaos.grand-challenge.org/.

[15] NVIDIA Clara — NVIDIA Developer. URL https://developer.nvidia.com/clara.

[16] DGX Systems : Built for the Unique Demands of AI — NVIDIA. URL https://www.nvidia.com/en-us/data-center/dgx-systems/.

[17] Yingda Xia, Fengze Liu, Dong Yang, Jinzheng Cai, Lequan Yu, Zhuotun Zhu, Daguang Xu, Alan Yuille, and Holger Roth. 3d semi-supervised learning with uncertainty-aware multi-view co-training. In *The IEEE Winter Conference on Applications of Computer Vision*, pages 3646–3655, 2020.

[18] Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*, pages 92–100, 1998.

[19] Siqi Liu, Daguang Xu, S Kevin Zhou, Olivier Pauly, Sasa Grbic, Thomas Mertelmeier, Julia Wicklein, Anna Jerebko, Weidong Cai, and Dorin Comaniciu. 3d anisotropic hybrid network: Transferring convolutional features from 2d images to 3d anisotropic volumes. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 851–858. Springer, 2018.

[20] Wenqi Li, Fausto Milletarì, Daguang Xu, Nicola Rieke, Jonny Hancox, Wentao Zhu, Maximilian Baust, Yan Cheng, Sébastien Ourselin, M Jorge Cardoso, et al. Privacy-preserving federated brain tumour segmentation. In

International Workshop on Machine Learning in Medical Imaging, pages 133–141. Springer, 2019.

[21] Elena Czeizler, Wolfgang Wiessler, Thorben Koester, Mikko Hakala, Shahab Basiri, Petr Jordan, and Esa Kuusela. Using federated data sources and varian learning portal framework to train a neural network model for automatic organ segmentation. *Physica Medica*, 72:39–45, Apr 2020. ISSN 11201797. doi: 10.1016/j.ejmp.2020.03.011.

[22] Hoo Chang Shin, Holger R. Roth, Mingchen Gao, Le Lu, Ziyue Xu, Isabella Nogues, Jianhua Yao, Daniel Mollura, and Ronald M. Summers. Deep Convolutional Neural Networks for Computer-Aided Detection: CNN Architectures, Dataset Characteristics and Transfer Learning. *IEEE Transactions on Medical Imaging*, 35(5):1285–1298, may 2016. ISSN 1558254X. doi: 10.1109/TMI.2016.2528162.

[23] Annegreet Van Opbroek, M Arfan Ikram, Meike W Vernooij, and Marleen De Bruijne. Transfer learning improves supervised image segmentation across imaging protocols. *IEEE transactions on medical imaging*, 34(5): 1018–1030, 2014.

[24] Kenneth A. Philbrick, Alexander D. Weston, Zeynettin Akkus, Timothy L. Kline, Panagiotis Korfiatis, Tomas Sakinis, Petro Kostandy, Arunnit Boonrod, Atefeh Zeinoddini, Naoki Takahashi, and Bradley J. Erickson. RIL-Contour: a Medical Imaging Dataset Annotation Tool for and with Deep Learning. *Journal of Digital Imaging*, 32(4):571–581, aug 2019. ISSN 1618727X. doi: 10.1007/s10278-019-00232-0. URL `https://doi.org/10.1007/s10278-019-00232-0`.

[25] Filippo Pesapane, Marina Codari, and Francesco Sardanelli. Artificial intelligence in medical imaging: threat or opportunity? radiologists again at the forefront of innovation in medicine. *European radiology experimental*, 2(1): 35, 2018.