FAKULTÄT
FÜR !NFORMATIK
Faculty of Informatics

Automated Visual Assessment of Osteoarthritis

Leveraging Interactive Visualisations in Osteoarthritis Diagnostics

BACHELORARBEIT

zur Erlangung des akademischen Grades

Bachelor of Science

im Rahmen des Studiums

Medizinische Informatik

eingereicht von

David Bauer
Matrikelnummer 01442385

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Ao.Univ.Prof. Dipl.-Ing. Dr.techn. Eduard Gröller
Mitwirkung: Alexander Krumböck

Wien, 17. Juli 2018

David Bauer | Eduard Gröller

FAKULTÄT
FÜR !NFORMATIK
Faculty of Informatics

Automated Visual Assessment of Osteoarthritis

Leveraging Interactive Visualisations in Osteoarthritis Diagnostics

BACHELOR'S THESIS

submitted in partial fulfillment of the requirements for the degree of

Bachelor of Science

in

Medical Informatics

by

David Bauer
Registration Number 01442385

to the Faculty of Informatics

at the TU Wien

Advisor: Ao.Univ.Prof. Dipl.-Ing. Dr.techn. Eduard Gröller
Assistance: Alexander Krumböck

Vienna, 17th July, 2018

David Bauer | Eduard Gröller

Erklärung zur Verfassung der Arbeit

David Bauer
Apoigerstraße 572/3/10, 3571, Gars am Kamp

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 17. Juli 2018

David Bauer

Danksagung

Mein herzlicher Dank gilt Richard Ljuhar und dem gesamten Team von ImageBiopsy Lab GmbH [Lju17]. Nur durch den einmaligen Teamgeist und die herausragende Zusammenarbeit dieser Truppe war ein Zustandekommen dieser Arbeit erst möglich. An dieser Stelle möchte ich auch Alexander Krumböck nennen, der mir im Rahmen meiner Arbeit besonders beiseite gestanden ist.
Ein besonderer Dank gebührt auch meinem Betreuer, Eduard Gröller, der mir beim Schreiben meiner Arbeit stets verständnisvoll und unterstützend entgegengekommen ist.

Die wichtigste Unterstützung kam mir jedoch von Freunden und Familie zu. Meine Eltern Susanne und Georg hatten stets ein offenes Ohr und ohne dem bedingungslosen Beistand meiner lieben Freunde - darunter besonders Lukas - wäre mir der rasche Abschluss dieses Studiums wohl nicht gelungen.

Danke.

Acknowledgements

My sincere gratitude goes to Richard Ljuhar and the whole team at ImageBiopsy Lab GmbH [Lju17]. Solely through the unprecedented team spirit and the outstanding cooperation of this troop was it possible for this paper to come about. In this regard I also want to mention Alexander Krumböck, who especially supported me in the course of my work.
I also want to express my gratitude towards my advisor Eduard Gröller, who has always been supportive and understanding in the process of writing this thesis.

Most importantly I want to thank my friends and family. My parents Susanne and Georg always had an open ear when I needed it and without the unconditional support of my dear friends - especially Lukas - I would not have been able to complete my degree in this swift manner.

Thank you.

Kurzfassung

Computergestützte Visualisierungen von Datenbeständen sind ein mächtiges Werkzeug um unübersichtliche Daten allgemein zugänglich zu machen. Auch eine relativ neue Sparte - Künstliche Intelligenz (KI) - bietet mannigfaltige Möglichkeiten semantischen Wert aus großen Datenmengen zu extrahieren. Sie ermöglicht es Sachverhalte in einer Art und Weise zu analysieren, die bisherigen Methoden in ihrer Spezifität und Genauigkeit teils weit voraus ist.
In der Medizin und im Speziellen in jenen Fachbereichen, die oftmals auf bildgebende Verfahren zurückzugreifen, erfreuen sich Visualisierungsverfahren steigender Beliebtheit. Unser Team bei ImageBiopsy Lab [Lju17] entwickelt und forscht im Bereich KI-gestützter Visualisierungen im medizinischen Bereich. In meiner Bachelorarbeit habe ich in diesem Team, aufbauend auf bestehenden Konzepten, ein System entwickelt, welches den Gelenksspalt von Röntgenbilder des Kniegelenks automatisiert vermisst und die Ergebnisse graphisch so aufbereitet, dass sie dem/der Benutzer/in als augmentiertes Originalbild dargestellt werden können. Dies geschieht in Form einer Maskierungsebene über dem zugrundeliegenden Knieröntgen. Die Messwerte beruhen auf den Parametern des Kellgren and Lawrence System (KLS) zur Klassifizierung von Osteoarthritis (OA).
Die vorgestellte Methode erlaubt es dem/der Nutzer/in auf den ersten Blick das Stadium und Tendenzen einer OA einzuschätzen und die errechneten Stützpunkte in Echtzeit am Röntgenbild anzupassen. Das System wurde in ein bestehendes, webbasiertes Framework aufgenommen und zeigt im Krankenhausbetrieb bereits sein Potential.

Abstract

Computer-aided visualisations are a powerful tool to make large datasets more accessible. Artificial intelligence (AI) also offers diverse ways in which to extract semantic values from large data stocks. It enables users to analyse records in ways that often exceed conventional methods in their specificity and accuracy.
Medicine - more specifically those specialisations requiring imaging methods - are in need of sophisticated visualisation techniques. Our team at ImageBiopsy Lab [Lju17] runs development and research in the field of AI aided visualisations in medicine. For my thesis I developed a system for measuring the joint space in x-rays of the knee, based on existing concepts. Results of the measurements are processed and presented to the user as an augmented picture. This is achieved by employing different layers of graphical overlays on top of the original image. All measurements are based on parameters of the Kellgren and Lawrence System (KLS) for classification of Osteoarthritis (OA).
The proposed method enables its users to asses the stage and tendency of OA in the knee at first glance as compared to conventional methods, which can be tedious and time-consuming. Calculated focus points in the mask layers can also be adjusted in real time to accommodate for statistical outliers. The system was incorporated into an existing web-based framework which already demonstrates its potential in a clinical environment.

Contents

CHAPTER 1

Introduction

1.1 Legal Disclaimer

This project was a cooperative endeavour between me and ImageBiopsy Lab GmbH for which contracts were signed accordingly. Due to this I am not eligible to disclose the full source code or configuration of the Image Analysis Server (IAS). However I was able to include selected algorithms and insights into the functionality and configuration of our service.

1.2 Background

The work presented in this paper is a multi-disciplinary approach which aims at enabling physicians to create and modifiy automated medical diagnosis. Therefore, in this chapter, I will clarify the setting of the project and describe some of the background needed to see the bigger picture.

1.2.1 Medical Background - What is Osteoarthritis (OA)?

Many cases of chronic knee pain are the result of a condition called Osteoarthritis (OA) [GG]. This condition is characterised by degenerative processes in joints causing the degradation of cartilage. This results in increased wear of the bone and ultimately in chronic pain and other inflammation-like symptoms [Neo12]. Occurrences of OA are not limited to the knee joint but can be found in a multitude of joints in the body. The most prominent regions include knees (Gonarthrosis), hips (Coxarthrosis) and the various joints of the hand [GG, KL57]. Since this work focusses on the analysis of knee pathologies, I will limit all further explanations to this area of the body.

Diagnosis of OA in the knee usually happens in multiple stages. First, the physician compiles data about the patient's background and previous conditions. This is followed by thorough physical examination. If, after these examinations, a case of OA is suspected, there are some medical imaging techniques to verify this preliminary diagnosis.
The most popular method is to record x-rays in two planes (usually front and side). Computer tomography (CT) is also an option, but rarely used unless specially indicated. Physicians will look at imaging results and measure the joint space between two bones. These measurements are then used to calculate different classification scores. One widely used score is the Kellgren and Lawrence System (KLS) [KL57] which incorporates not only joint space measurements but other factors like sclerosis of subchondral areas and osteophytes in the outer edges of the bone.
The results of these scores can be used to classify the given knee. There are other scores used in knee OA diagnostics like the Ahlbäck score [Ahl68], which yields comparable results and also uses a summative score ranging from 1 to 5. Ahlbäck classification is mainly based on joint space measurements as opposed to the multitude of parameters used in the KLS [Ahl68, PBS+97].

1.2.2 The Kellgren and Lawrence System

We have seen, that along with the Ahlbäck classification system [Ahl68], a popular score for classifying OA of the knee is the so-called Kellgren and Lawrence System (KLS) [KL57]. Results of evaluation are represented by a one-digit integer ranging from 0 to 4, with 0 being the best possible result while 4 indicates the most critical stage of OA.
Score grades are characterised as follows:

Grade 0:	No joint space narrowing (JSN) or reactive changes
Grade 1:	Doubtful JSN, possible osteophytic lipping
Grade 2:	Definite osteophytes, possible JSN
Grade 3:	Moderate osteophytes, definite JSN, some sclerosis, possible bone-end deformity
Grade 4:	Large osteophytes, marked JSN, severe sclerosis, definite bone ends deformity

Table 1.1: Definition of KLS score grades [KL57, SBBZ08]

In most cases, the basis of calculation is an anterior posterior (AP) x-ray of the knee. Using this image, the following data is collected [KSF16]:

- Space between the two joints is measured at different locations along the horizontal axis of the joint space
- Existence and level of expression of osteophytes is determined
- Existence and level of sclerosis in subchondral areas is determined

- Altered bone shape is evaluated

These variables are then used to determine the overarching score.

1.3 The Problem

We have seen that OA in the knee is a major cause for chronic pain and inflammation. Progression of OA can be classified using different systems [KL57, Ahl68] of which the KLS is a popular choice among physicians. Calculating this score requires the evaluator to obtain measurements for every knee to be analysed. This can be a rather time-consuming endeavour considering the number of patients the average medical doctor has to treat every day. The team at ImageBiopsy Lab [Lju17] developed a novel solution to this problem by automating the process of measurement retrieval and score calculation. This method is based on traditional computer vision and image processing techniques as well as modern machine learning approaches to obtain necessary data. Until now, the only way to use the software was by going through an elaborate process of installing and deploying necessary modules, which limited accessibility.

1.4 Contribution

Basing my work on existing paradigms developed by ImageBiopsy Lab [Lju17], a colleague and I developed a new web-based system providing the aforementioned functionality of measuring and calculating the KLS. The product was given the name Image Analysis Server (IAS). This solution is augmented by mask overlays on top of the original x-rays, allowing for real-time manipulation of measurements and score properties. Realising these overlays constituted the main part of my contribution to the solution, which is why I will focus on these aspects in this paper.

These additions enable physicians to adjust the score to their liking and correct values within the margin of error of the automated calculation.
Since this project is web-based, analysis of knee x-rays can be undertaken by any physician using merely a web-browser without having to go through the strenuous setup procedure.

By the time of writing this paper the operation of a test instance of this service has been approved by the management of the "Landesklinikum Horn". There are already ongoing negotiations with the "Landesklinikum Holding" about deploying the solution in hospitals all around Lower Austria.

1.5 Organisation of the Project

The implementation of Image Analysis Server (IAS) was a joint effort between me and my colleague Lukas Maximilian Masopust. Other people involved were my personal supervisor Alexander Krumböck as well as the technical supervisor Zsolt Bertalam. For the deployment of IAS we had ongoing correspondence with the representative of VISUAPPS GmbH Armin Kanitsar [Kan].

As for my part of the project, I was responsible for planning and developing the modules that would take previously calculated analysis data and further process it for visual representation in the scalable vector graphics (SVG) format. I also developed the servlet interface of IAS and all related command functionality like dataset initialisation, download of PDF reports, etc. (for details see Sections 3.1 and 4.1).

All requirements for the first working version of IAS were defined in accordance with Armin Kanitsar and our project team. Internally we organised the project using a simplified management approach roughly based on the widely used SCRUM methods.

1.6 Outline

The following topics will be presented in this paper:

- Discussion of the state-of-the-art of automated medical analysis and visualisations comparable to those used in this work.
- Overview of the methods and tools used to achieve the results described in 1.4.
- Detailed description of my suggested approach
- Conclusion and critical reflection concerning my work and achieved outcomes
- Discussion about current state and future work to be done in the field

CHAPTER 2

State of the Art

Over the past years, and therefore some time after machine learning had gained momentum in informatics, the field of medicine has adopted these new approaches to solve problems in its own area. AI, in this regeard, lends itself mostly to identification and classification problems and is especially suited for analysing and processing images obtained by various medical imaging techniques.

Since my project is formally part of a larger cluster of automated medical applications, we will look at the state-of-the-art of already established solutions in this field. There are however, to this date and to the best of my knowledge, no comparable solutions to the one discussed in this paper.

2.1 Image Segmentation

Analysis of images is still considered the gold standard in many areas of medical diagnostics. Popular techniques include magnetic resonance imaging (MRI), x-rays, PET scans or microscopy of histologic specimen. This holds especially true for the field of oncology, as we will see below.

Due to tremendous advancements of neural networks (NN) - a subgroup of AI - in past years the power to reliably analyse, segment and classify digital images has since surpassed the capability of human evaluators in many cases [JJZ+17]. Tasks tackled with these methods mostly revolve around segmentation and classification of images. Segmentation is the extraction of relevant parts of an image that are to be analysed further. Classification describes the process of categorising images based on some selected features. These categories are often defined as different types of pathologies or yes/no decision problems.

This is why many research teams have applied sophisticated machine learning techniques to medical problems. El-Dahshan et al. have looked at several methods of employing machine learning for image segmentation and classification in brain MRI scanning while still retaining computational viability in production systems. They used two different kinds of NNs to accomplish both segmentation and classification of malign tissue alterations. For region of interest (ROI) extraction they used a derivative of the pulse coupled neural network (PCNN), the FPCCN, "F" standing for feedback, indicating that output data is used again as input to modulate overall calculations. The PCNN is considered to be a very powerful type of neural network for image recognition and segmentation tasks and is inspired by the mammalian visual cortex [EDMRS14]. A back-propagation neural network (BPNN) was used to classify images as normal or abnormal based on previously selected features.

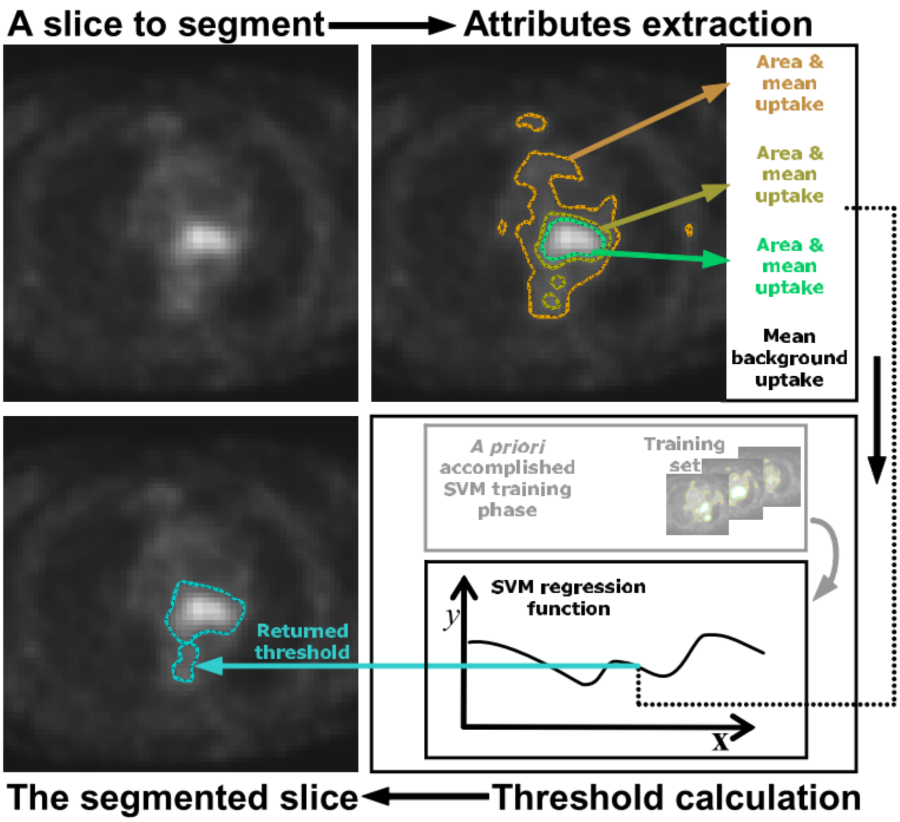


Figure 2.1: Usage of SVM to determine optimal ROI thresholds [KSQ+09]

Kerhet et al [KSQ+09] have used machine learning to find optimal ROI segmentation thresholds in positron emission tomography (PET) scans of the lung. The refined threshold can then be used to extract more accurate ROIs and moreover gain better analysis results (Figure 2.1).

2.2 Image Classification

As previous work points out [EDMRS14], further steps after image segmentation usually include classification of suspected pathologies. In oncology these classifications are mostly done in a binary fashion, discriminating between benign and malign tumours - in other words determining if there are tissue alterations or not. Another approach are so-called multi-class classifications, which are used to discern multiple types of cancer.

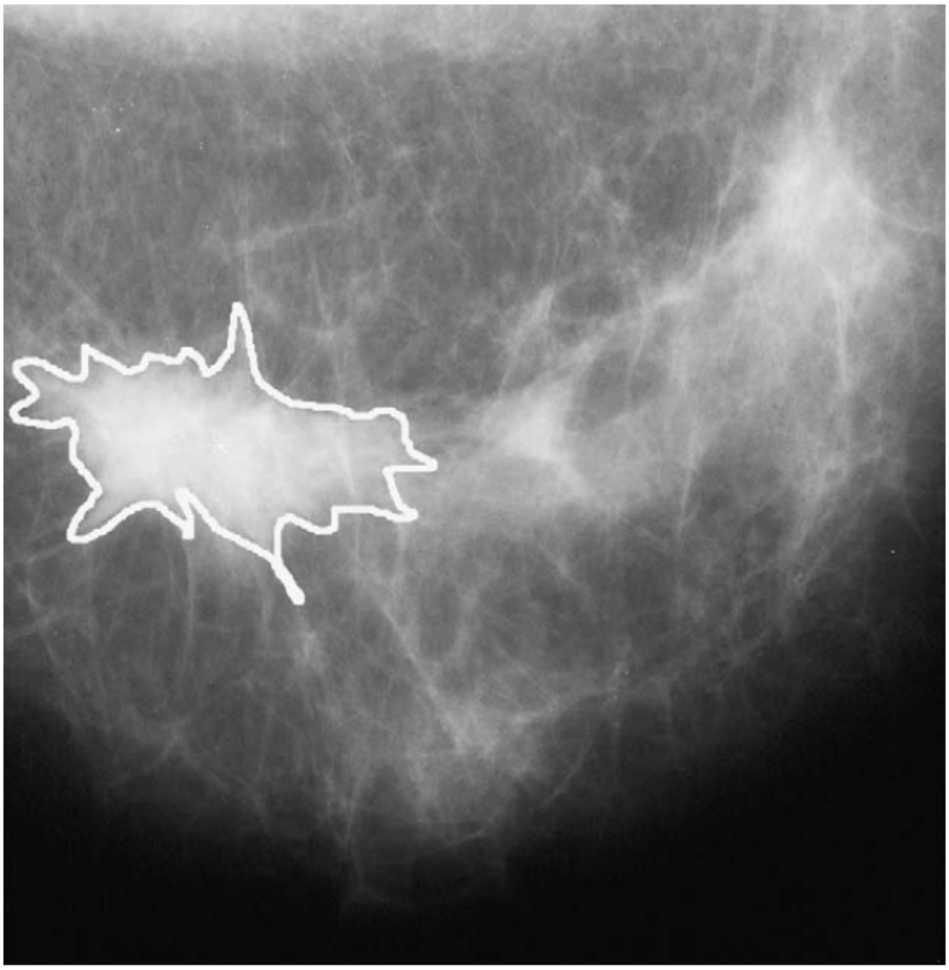


Figure 2.2: Pre-segemented mammographic scan [MGD+06]

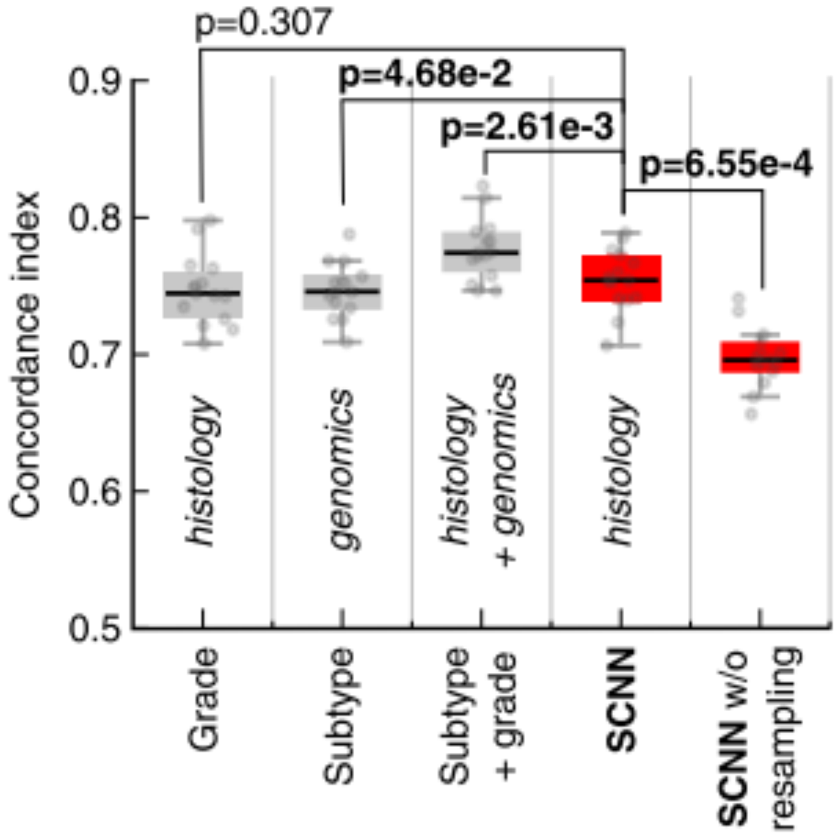


Figure 2.3: AI accuracy compared to traditional methods [MYA+18]

These classification problems have also been subject to AI research. Mammographic scans were processed using support vector machines (SVM), k nearest neighbors (kNNs) and multi-layer perceptrons (among others). Previously segmented images where analysed and classified based on selected texture features within selected ROIs (Figure 2.2) [MGD+06]. The described methods were applied respectively to the top 10 most expressive image features within relevant areas.

Another team applied AI-backed classification methods to histologic images of brain tumour tissue [MYA+18]. The specimen where analysed using the previously described binary mode as well as using multi-class classification systems.

Classification results reached levels comparable to ratings given by experts employing traditional diagnostic tools like manual histology or genetic analysis (Figure 2.3). Since machine learning can be employed quite efficiently when used for image analysis, results of these classifications where enough to process them further and create intuitive heatmap-like visualisations of infiltrated areas in the tissue (Figure 2.4).

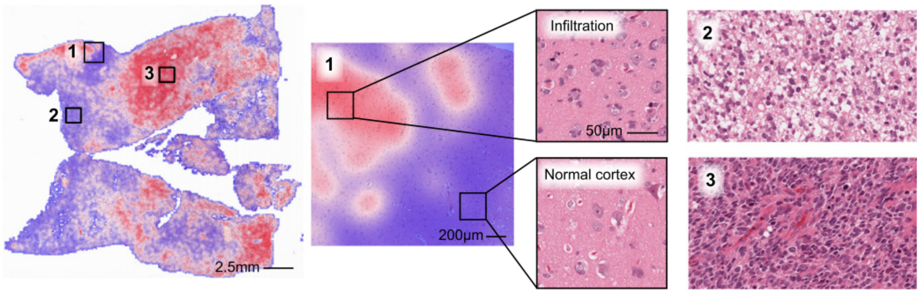


Figure 2.4: Malignancy heatmap-overlay in histological findings [MYA+18]

CHAPTER 3

Methodology

In this chapter I will discuss technologies and methods used to develop our final product, the Image Analysis Server (IAS).

3.1 Technical Environment

One of the earliest goals we set for IAS was that it should be modular and easily accessible from anywhere. This means that, in a best-case scenario, the service will be installed on a single server which then serves a multitude of hospital clusters and physicians offices. Keeping in mind that the software should also be deployable in decentralised hospital environments, the design had to rely on well-established technologies supported by a wide variety of systems.

Given the nature of IAS as non-invasive support software, it has to be certified by a qualified entity as a Class I Medical Product according to the EU Medical Device Regulation (MDR) act [Eur17]. This further limited the choice of technologies.

3.1.1 Language and Frameworks

The constraints listed above led to the decision of using Java as the main programming language. This was supported by the fact that Java has already been validated for the MDR by ImageBiopsy Lab. Since certification is a rather complicated and time-consuming process, many hospitals and medical institutions rely on legacy systems that are already validated. This was the basis for choosing a Java Servlet structure as the underlying architectural base.

Java is well-supported by all major operating systems and the servlet structure is conveniently built into the Java Enterprise Edition (Java EE). As for the version, JDK 8u171 and 8u172 were both used in the course of development.

3.1.2 Basis of Development

Prior to project initiation there was already a basic prototype implementation. This precursor was very limited in its functionality since it merely rerouted HTTP requests to existing routines running separately from this dispatcher, which also only provided minimal functionality.

For my work the relevant parts of this prototype were reused. These mainly consisted of a list of supported HTTP requests as well as the basic command parsing functionality, which abstracted HTTP POST and GET requests into one generic type of request. Further plans for architecture and functionality were also worked out with my advisor prior to development (see 4.1).

3.1.3 Server Infrastructure

Servlets are compiled to packaged **.war* files. Not unlike the more commonly known **.jar* file type, they are slightly altered versions of ZIP archives. These packages are deployed on so-called Application Servers, which run them as separate instances serving incoming requests. Established brands include Tomcat (Apache Software Foundation) and Wildfly/JBoss (JBoss, Red Hat). ImageBiopsy Lab, having previously worked with Application Servers, suggested using the Wildfly server, which is also used by our first external deployment partners - the "Landeskliniken-Holding" in Lower Austria. Wildfly version 10.1.0 was chosen for all development stages. Test servers themselves ran the 64bit version of Ubuntu 16.04.

3.1.4 Software Clients

To test IAS we used the radiology web frontend "Orthoweb", which was kindly provided to ImageBiopsy Lab by VISUAPPS GmbH [Kan].

3.1.5 Scalability

IAS was designed to be robust and resource-conserving, while retaining accessibility to multiple clients simultaneously. It therefore was implemented using a lightweight infrastructure that only saves basic image and metadata submitted by clients and recalculates necessary output upon request. Tight session handling with timeouts of around 30 minutes (adjustable) serve as a good compromise of usability - considering the time a physician typically spends diagnosing one image - and economical resource management.

3.2 Medical Considerations

In medical informatics there are certain, widely accepted standards concerning transfer and storage of clinical data [DAB+01, DIC, ICD, LOI]. For medical imaging data, like that used by IAS, the most widely used format is the so-called *DICOM* standard (digital imaging and communications in medicine) [DIC]. These files are specified to contain direct payload like images, videos or analysis data. Along with these main data come specific metadata blocks, which contain information about the patient, the practitioner, the device used to record the data and many more.

IAS uses the *DICOM* standard to load images, store metadata and generate reports in *DICOM* and *PDF* formats. To achieve this I utilised version 3.3.7 of a library called "dcm4chee" [DCM] which was specifically designed to process the *DICOM* format in Java. All *DICOM* image data shown in this paper originates from the multicenter Osteoarthritis (OA) study [SNG+13].

3.3 Evaluation

One major concern when developing IAS was runtime performance. This becomes especially important when serving multiple clients from one system. Since there was already a prototype prior to my work, we felt it suitable to use this as a reference point for performance benchmarks.

Performance plays a role in the initialisation phase but is especially important when using the *modifyPoint* and *getSVG* commands (see Table 4.2), since most clients will send these requests in real-time. Considering this, the best-case outcome would be an effective frame rate of 10-30 frames per second.

For our evaluation measurements, separate requests were sent to the server. Each command was first executed in an initial-run scenario, effectively measuring caching and calculation times. We tested this initialisation state for each implementation, each time restarting the whole service (n = 10). For applicable requests - those being *getSVG* and *modifyPoint* (see Table 4.2) - successive measurements were also executed. Each of these commands was sent multiple times, measuring response times separately in milliseconds (n = 30).

To make measurements from different runs and systems comparable, all time records t were converted to their respective frame rate f (where applicable): $f = \frac{1000}{t}$
Here f denotes the respective frame rate whereas t represents the measured time in milliseconds. The value 1000 is used a time reference value (also in milliseconds) to calculate frames per second.

Statistical evaluation was done using the standard t-test. Response times were compared using a significance threshold of 5 %.

As for the technical details of evaluation we used a variety of different tools. To send requests to the service we used a tool called "Postman" [Pos] which allowed us to construct fine-tuned commands, especially tailored for our server infrastructure. This application was also used to measure effective response times.

Measurements were all conducted within our company network, which is a standard Gigabit LAN infrastructure. The server running both IAS and the previously mentioned prototype - hereafter called EXT - had the following specifications:

- AMD FX(tm)-6100 Six-Core Processor running at 3.30 GHz
- 16.0 GB of working memory
- Wildfly 10.1.0.Final
- Microsoft Windows 7
- Ubuntu Linux 16.04

Due to technical constraints, EXT was running under Windows, while IAS was set up on Ubuntu. Both were installed within Wildfly.

CHAPTER 4

Suggested Solution

This chapter documents the development process and functionality of IAS. As stated previously (see Section 1.1) the source code and detailed architecture are not fully disclosed in this chapter.

4.1 Functionality and Architecture

The software architecture consists of different modules which can be categorised accordingly:

Request Processing	Incoming requests are handled by the servlet class
Request Handling	This module routes requests to the according modules
Numerical Calculations	Data is preprocessed and calculated using external modules
Mask Generation	Overlay SVG masks are constructed according to calculation data
Report Generation	Reports are constructed based on calculation data
Focus Point Modification	Modifications are reflected locally and re-calculated by the calculation module

Table 4.1: Modules of IAS

Numerical calculations are done by external modules, that process a given image or parts of it. This is done using machine learning techniques and traditional computer vision algorithms.

To better illustrate the core functionalities of IAS, a typical use-flow is laid out in this section. First, the client contacts the server with an *init* command, sending the *DICOM* to be analysed along with the request in a byte stream. This step is required to start a new session, if none is already established. If the first command sent by a client is not the *init* command, the server will refuse communication.

Initialisation loads the image into memory and triggers calculations. These calculations detect two landmarks on the image which are the outermost parts of the head of the tibial bone (Figure 4.1). Using these landmarks, or focus points (FPs), other FPs are calculated around the joint space area. These FPs constitute the supporting mesh for constructing and adjusting the mask overlay. Additionally, measurements concerning joint space width and area between the bones are also calculated by the external numeric calculation units.

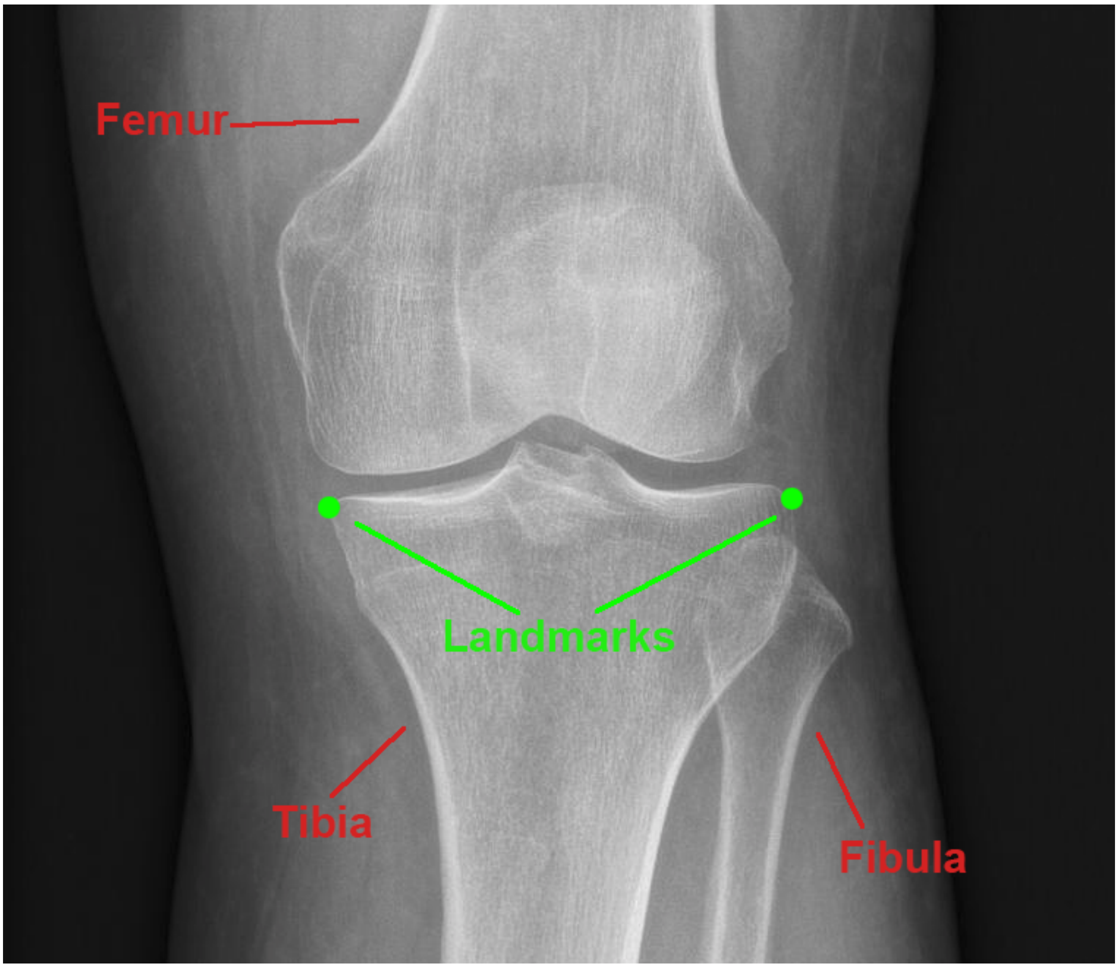


Figure 4.1: Raw x-ray overview with landmarks marked in green

Because all relevant calculations are executed upon initialisation, all following commands can be sent in arbitrary order. Calling *init* again will simply reset the current session and re-initialise all data with the newly transmitted image.

A detailed list of supported commands can be found in Table 4.2. Figure 4.2 shows the whole process schematically.

Command	Input	Output	Description
init	*DICOM* byte stream	none	Initialises the service and calculates measurements
getPoints	none	JSON Array of FPs	Returns a list of all calculated FPs
modifyPoint	pt, x, y	SVG mask overlay	Moves FP (pt) to x and y
getSVG	none	SVG mask overlay	Constructs the SVG mask overlay
getSR	none	Report in *DICOM* format	Generates a structured report (SR)
report	none	Report in *PDF* format	Generates a PDF report

Table 4.2: Specifications of HTTP servlet requests

4.2 Algorithms

IAS contains some key functionalities that are essential to the workings of the overall product. Here I will list the most important algorithms used in our project.

4.2.1 Nonorthogonal Image Cropping

The algorithm takes the whole image data as a one-dimensional array and the desired crop area as arguments and performs cropping by utilising efficient array operations. Rectangles (see input for Algorithm 4.1) are collections of three points, marking the top left, top right and bottom right corner of the rectangle in world coordinates. Most x-ray images are not aligned perfectly to one axis. This is why this algorithm is capable of performing nonorthogonal, rotational crops.

To understand the need for this algorithm, performance has to be thought of first. When calculating landmarks, the whole image data is sent to external modules. This is necessary because the whole image is scanned to detect the position of the knee joint.

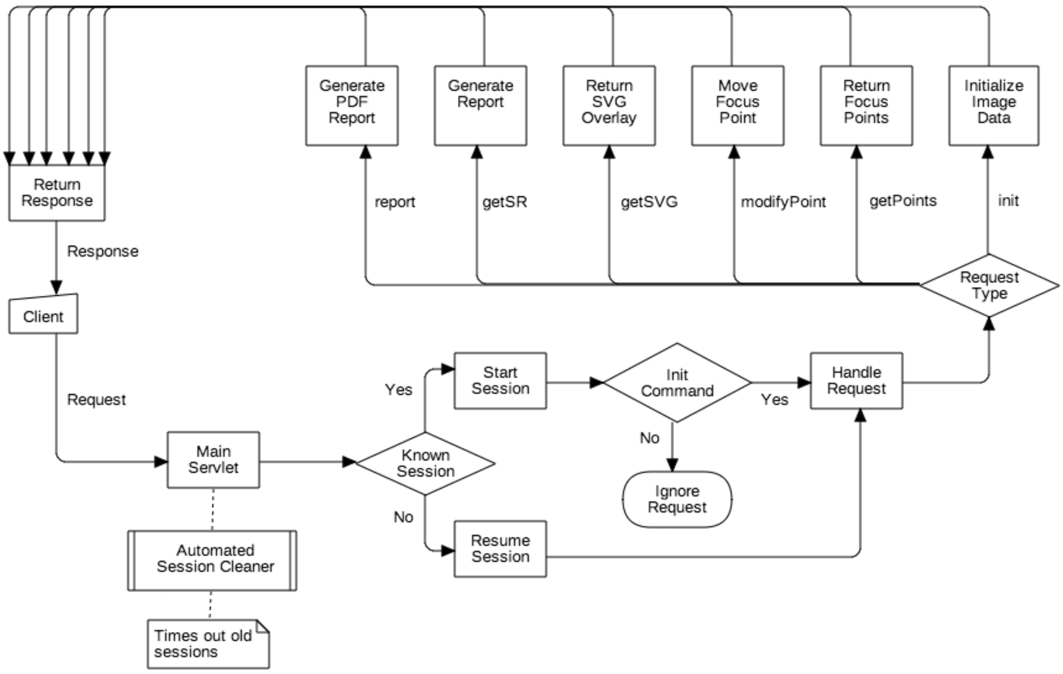


Figure 4.2: Schematic overview of the server architecture. Incoming requests are assigned to their respective sessions, which take according actions (Bottom). Supported request types include initialisation of the service, retrieval of calculated focus points, modification of focus points, retrieval of calculated mask overlays in SVG format and generation of reports in *DICOM* or *PDF* format (Top Right). All commands return their results back to the main servlet entry point which in turn responds to the client (Top Left).

Other calculations like FPs in the joint area or joint space width do not need to use the full image (although this would be possible). Images are rather big and it would not be feasible to send the whole picture for all minor calculations. That is why this algorithm is used for all smaller calculations to extract relevant image areas before sending them to external modules.

Cropping is not done pixel perfect, because it does not have to be and because the proposed approach is more efficient. Pixel interpolation is also omitted for the same reasons (Figure 4.3).

Algorithm 4.1: Nonorthogonal Image Cropping

input : Image to be cropped (*image*), Cropping Area (*cropArea*)
output : Cropped image (*croppedImage*)

Calculate step size for upper edge based on rotation angle as $ueStepX$, $ueStepY$;
Calculate step size for lower edge based on rotation angle as $leStepX$, $leStepY$;
for $y \leftarrow 0$ **to** $cropArea.height$ **do**
    $tempX \leftarrow cropArea.upperLeft.x + ceil(leStepX * y)$;
    `/*` $leStepY$ `has to be subtracted due to orientation of SVG coordinate system */`
    $tempY \leftarrow cropArea.upperLeft.y - floor(leStepY * y)$;
    for $x \leftarrow 0$ **to** $cropArea.width$ **do**
        $croppedImage[y * cropArea.width + x] \leftarrow image[floor(tempY) * image.width + ceil(tempX)]$;
        $tempX \leftarrow tempX + ueStepX$;
        `/*` $ueStepY$ `has to be subtracted due to orientation of SVG coordinate system */`
        $tempY \leftarrow tempY - ueStepY$;
    end
end

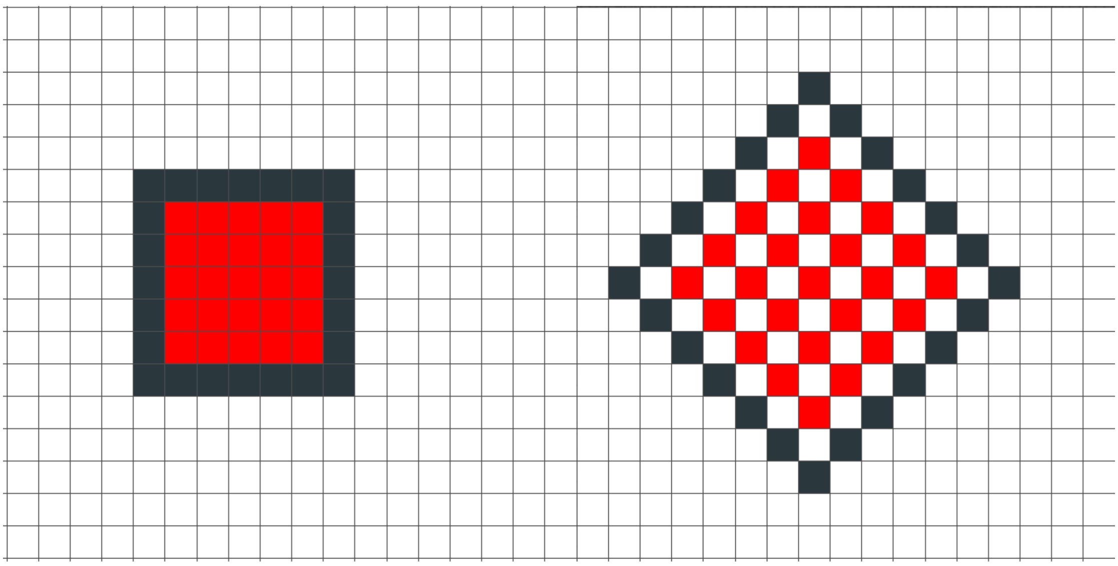


Figure 4.3: Cropping accuracy of the algorithm. If the cropping area is perfectly aligned with one axis all pixels are considered in the crop (Left). In cases where the cropping area is tilted, some pixels are left out for the sake of array size consistency and algorithm speed (Right).

4.2.2 Relative Landmark Transformations

When working with mask overlays we needed to find an easy and uncluttered way to use the values obtained from the external calculation modules to generate respective SVG data that visually represents these values. The SVG standard supports the application of matrix transformations to single elements and whole groups of objects [DLC+06]. This feature lent itself well for our specific use case.

To achieve the desired outcome we first constructed a basic SVG mask shape manually, which in turn was to be used as a template. This shape contains all the information the final mask overlay does but in it all points are untransformed and lie within a normalised coordinate system. In this system all coordinates start at 0 and can reach a maximum absolute value of 100 (Figure 4.4 (a)). This value range was chosen to easily construct masks in terms of percentage measurements. Since it is untranformed raw data, this basic SVG can be easily reused for all possible mask overlay constellations.

After loading the template, the appropriate transformation matrix has to be applied to the SVG data to obtain the final mask overlay data. To calculate the transformation needed in each case respectively, an algorithm (Algorithm 4.2) was constructed to calculate translation, rotation and scale given two normalised points and their transformed counterparts. In simpler terms, the algorithm can be viewed as a way to compare two corresponding lines based on their transformation. It was used to compare the untransformed values, taken from the SVG template, with those of the external calculation modules. The latter values are given in absolute image coordinates, so they can be seen as the transformed version of the untransformed template data. An example of the resulting mask after applying transformations can be seen in Figure 4.4 (b).

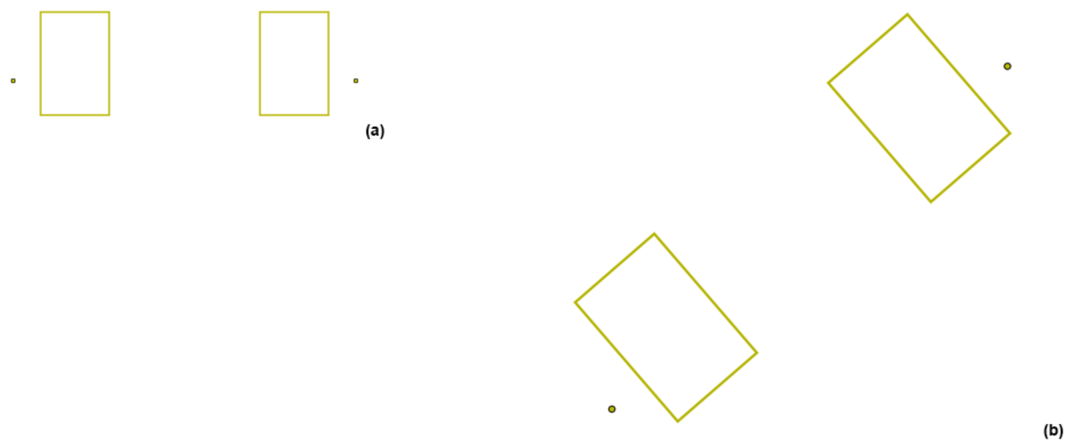


Figure 4.4: Landmark transformations. (a) Original, normalised SVG. (b) Final, transformed SVG.

In this way, we were able to conveniently extract matrix properties for each new calculation which were then simply applied to the root element of the SVG object. This in turn transformed the whole mask overlay to fit the raw x-ray exactly.

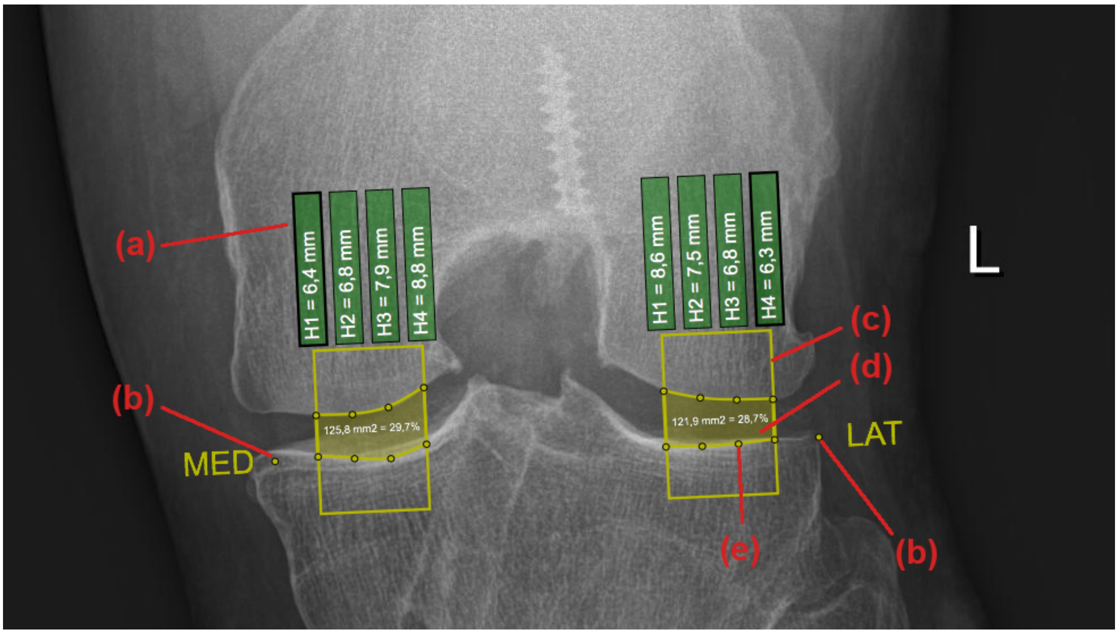
Algorithm 4.2: Landmark Transformation

```
input   : Normalised points (p1n, p2n), Transformed points (p1t, p2t)
output  : Matrix representing the transformation (t)
Create temporary transformation objects (tNorm, tTrans);
lengthNorm ← p1n.distanceTo(p2n);
lengthTrans ← p1t.distanceTo(p2t);
if lengthNorm ≠ 0 then
    scale ← lengthTrans/lengthNorm;
else
    scale ← 1;
end
dxNorm ← −p1n.x;
dyNorm ← −p1n.y;
tNorm.translate(dxNorm, dyNorm);
p2n ← tNorm.apply(p2n);
dxTrans ← −p1t.x;
dyTrans ← −p1t.y;
tTrans.translate(dxTrans, dyTrans);
p2t ← tTrans.apply(p2t);
angle ← p2n.angle(p2t);
crossProduct ← p2n.cross(p2t);
if crossProduct ≠ 0 then
    direction ← crossProduct/ abs(crossProduct);
else
    direction ← 0;
end
angle ← angle ∗ direction;
return t.translate(dxNorm,
  dyNorm).rotate(angle).scale(scale).translate(−dxTrans, −dyTrans);
```

4.3 Mask Overlays

The core features of IAS are its visualisation capabilities. Up to this point I discussed the underlying functionalities for handling requests, calculations and algorithmic data manipulations. We have already seen some commands (*getSVG* and *modifyPoint*) that return SVG graphics to the client. These are in turn placed on top of the x-ray as a mask overlay by the client viewer. Figure 4.5 gives a detailed overview of the different parts of the SVG mask.

Figure 4.5: SVG mask superimposed on top of original x-ray. (a) Joint space measurement between upper and lower FPs (see (e)) in mm. Colours correspond to the contribution to the KLS score ranging from green to red. (b) Tibia landmarks detected by external calculation module. Laterality on the x-ray is also detected and denoted by MED = medial and LAT = lateral. (c) Joint space measurement box. This area is used for cropping (see Algorithm 4.1) to reduce calculation complexity. (d) Joint space area between the femoral and tibial bone. The calculated area is displayed in mm^2 and the percentage ratio of the area as compared to the full joint space measurement box (see (c)). (e) Single joint space FP framing the joint space area.



4.3.1 Systematisation

The mask itself is structurally divided into a *mask* and an *overlay* component:

- **Mask** is the basic structure needed to guide calculations and positioning routines. This includes the landmark FPs seen in Figure 4.5 (b) and joint space measurement boxes (c).

- **Overlays** are all parts which are constructed from dynamic, ever changing properties generated by the external calculation modules. These are marked (a), (d) and (e) in Figure 4.5.

Architecturally masks are arranged in a flat hierarchy:

- **KneeMask** containing the landmarks (Figure 4.5 (b))

  – **JointSpaceMask (Left)** containing left-hand sections (a), (c), (d) and (e) of (Figure 4.5)

  – **JointSpaceMask (Right)** containing right-hand sections (a), (c), (d) and (e) of (Figure 4.5)

Each element of the hierarchy is responsible for generating its respective *mask* and *overlay* components.

4.3.2 Characteristics of Mask Overlays

Mask overlays directly reflect calculations. To emphasise these diagnostic outcomes visually, IAS uses color codings (Figures 4.6 and 4.7) offering users the possibility to assess the joint's general condition at first glance.

In addition, physicians using IAS have the possibility to adjust the score manually by dragging any FP to a new position. Owing to the hierarchical approach chosen for development only areas that are affected by changes need to be re-evaluated by the service.

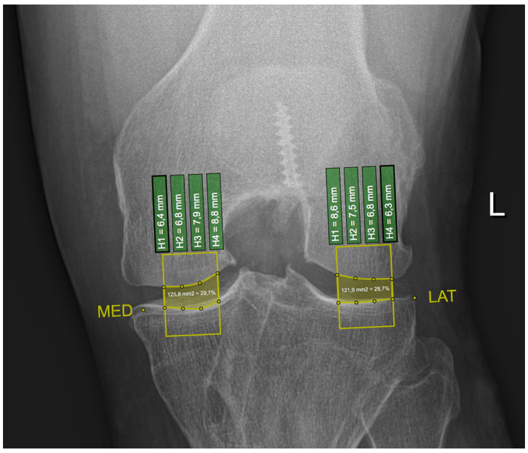


Figure 4.6: Visual assessment of a healthy knee joint with KLS grade 0. Green labels indicate healthy conditions.

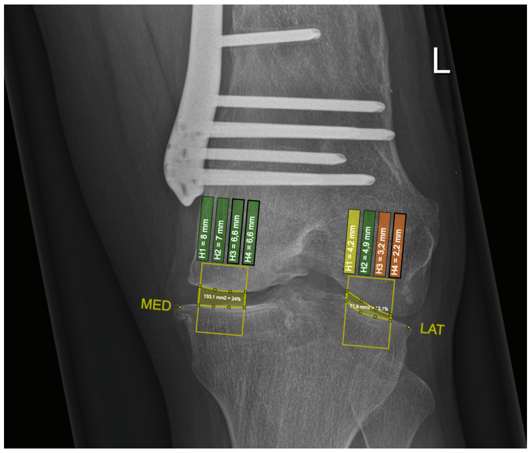


Figure 4.7: Visual assessment of a pathological knee joint with KLS grade 3. Yellow and orange (and potentially red) labels indicate joint impairments.

4.4 Evaluation of Results

A fully functional version of IAS was tested using the evaluation procedure described in 3.3. Results were compared against equivalent measurements conducted on the prototype implementation (EXT).

Results of initial-run performance are shown in Figures 4.8, 4.9 and 4.10. Response time measurements of initial and successive tests can be found in Figures 4.11, 4.12 and 4.13 for *getSVG* and *modifyPoint* respectively.

All measurements of IAS commands were compared to their equivalent request in the prototype implementation. Some comparisons were omitted due to missing data. This is because the EXT implementation does not support all functionalities that IAS does. Values were compared using a standard t-test. Results are listed in Table 4.3 for initial runs and Table 4.4 for successive requests.

The data shows a significant performance increase in all cases. Commands which are utilised for real-time mask manipulations show the highest gains, further underlining IAS's potential in scenarios where live feedback is essential for user productivity.

	getPoints	*getSVG*	*modifyPoint* (joint space)	*modifyPoint* (landmark)
Mean response time (ms) IAS	17.8	26.4	87.9	212.2
Mean response time (ms) EXT	900.3	39.6	531.6	475.6
p-value	0.03469	0.04248	0.0001475	0.001092

Table 4.3: Statistical evaluation of initial-run response times (ms) using a standard t-test

	getSVG	*modifyPoint* (joint space)	*modifyPoint* (landmark)
Mean response time (ms) IAS	38.0	84.5	105.1
Effective frame rate (fps) IAS	26.3	11.8	9.5
Mean response time (ms) EXT	44.3	459.0	451.0
Effective frame rate (fps) EXT	22.6	2.2	2.2
p-value	0.3027	$3.44e-12$	$< 2.2e-16$

Table 4.4: Statistical evaluation of successive response times (ms) using a standard t-test. Effective frame rates are based on response time measurements.

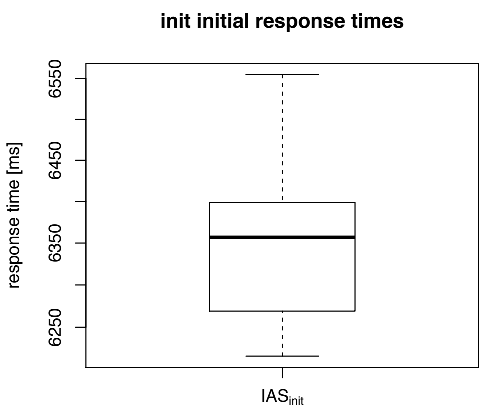


Figure 4.8: Performance of *init* command. Measurements for EXT are unavailable due to technical incompatibilities when initiating the service manually via Postman [Pos].

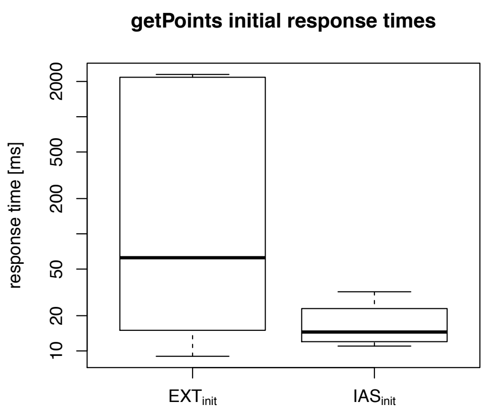


Figure 4.9: Performance of *getPoints* command. IAS shows more reliable response times with less spread.

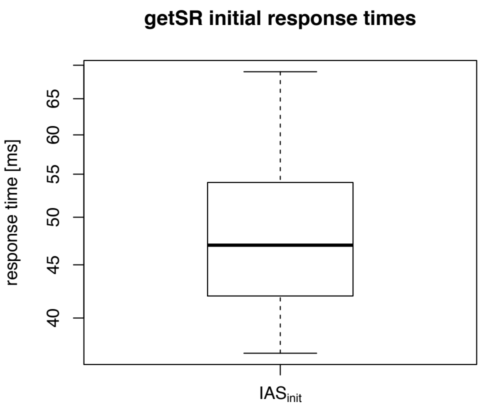


Figure 4.10: Performance of *getSR* command. Measurements for EXT are unavailable because this feature was not supported by the prototype.

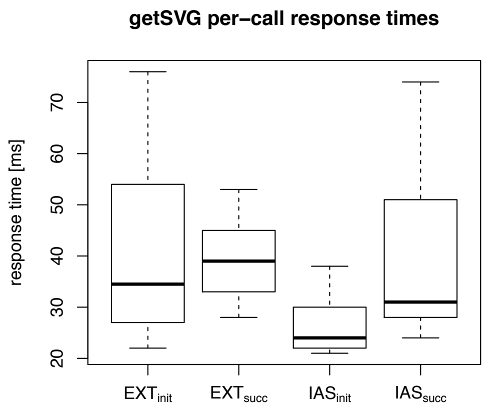


Figure 4.11: Performance of *getSVG* command. Both implementations show comparable response times. This is due to cached SVG data in both variants. The deviation is most likely caused by network fluctuations.

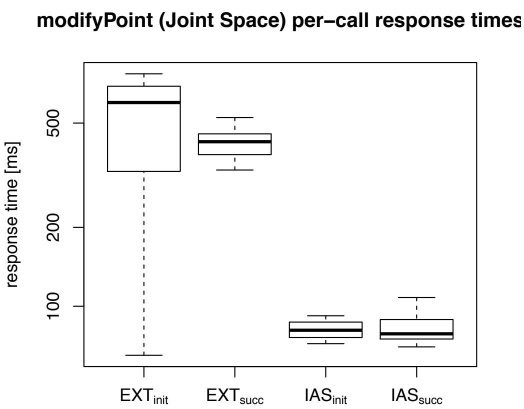


Figure 4.12: Performance of *modifyPoint* command applied to joint space FPs. This command especially shows the significant work load reduction provided by IAS, since *modifyPoint* triggers re-calculation every time and can therefore not be cached.

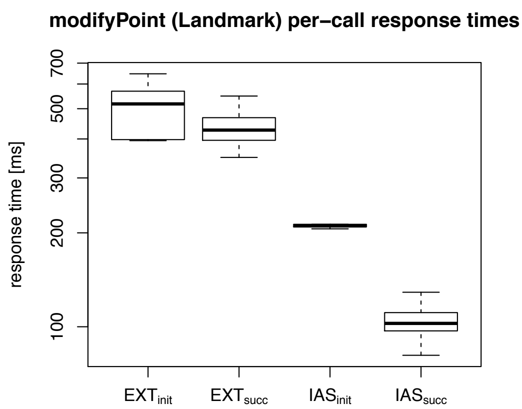


Figure 4.13: Performance of *modifyPoint* command applied to landmark FPs. Landmark modifications take slightly longer than joint space movements because they trigger and include those same calculations in addition to their own.

CHAPTER 5

Discussion

5.1 Summary

We have seen the potential of IAS as a modular service offering its capabilities to clients via a simple request-based interface. Data from this communication interface can be integrated into interactive client software as we have discussed in previous sections (see 3.1.4 and Figures 4.5, 4.6 and 4.7). We are confident that this solution has the potential to be widely used in hospitals and doctors' offices with minimal maintenance.

Although not yet optimal, it provides many useful features to automate OA analysis and offer data in a modular, reusable fashion. IAS is subject to continuous improvements and is now being developed by many employees of ImageBiopsy Lab.

5.2 Related Topics

In this paper we discussed the parts of IAS which are closely connected to its modular visualisation capabilities. There is, of course, more to this project than just mask handling, which will be briefly discussed here.

Using the mask interface is the way users interact with IAS, but there are also times it is necessary to export all data related to a certain analysis. We have developed modules to do this by collecting data from all modules of the service and aggregating them into a single report in XML format.

There is also a whole side-project concerned with providing convenient ways to access the external calculation modules mentioned throughout this paper. It allows for easy and modular exchange of units which makes version updates or methodology changes in these modules easy to implement.

5.3 Performance

Results presented in Section 4.4 show promising values for successive run performance. All measurements were significantly faster than the prototype and show worst-case mean frame rates of 9.5 with top values ranging up to 22.6 frames per second. Some of these values are even within the threshold of continuous visual motion perception [RMG00] which makes for a smooth, real-time user experience.

These rates are more than suitable for most use-cases covered by IAS. However we are working on code optimisations to further increase effective frame rates, with a goal of around 30 frames per second in mind. To this end we have already conducted tests with more powerful production servers and plan to deploy the application to easily scalable Amazon Web Service instances using Docker containers.

5.4 White Spots

The first version of IAS is completed and stable. There are however still many enhancements that we are currently working on.

There is evidence that textural analysis can yield beneficial results for more accurate OA diagnostics [JJV+17, WPS+12, KFW+09]. This could be realised in our future work to strengthen predictive capabilities of IAS.

Looking beyond KLS, there are many unexplored opportunities to expand IAS using other classification systems with respectively different mask interfaces. Our team at ImageBiopsy Lab has also started looking into ways of analysing different joints, like the hip or palmar region using the same modular approach that this project offers.

CHAPTER 6

Conclusion

In this paper we have seen how machine learning can be applied to diagnostic problems in the field of medicine and different approaches of visualising this data have been presented. We discussed the potential of automated Osteoarthritis (OA) evaluation of the knee joint and looked at our proposed solution to this task.

6.1 Image Analysis Server (IAS)

Functionalities and architectural structures of the solution were explained, giving insight into the systematics behind automated measurements using raw x-ray images. Further we looked at our proposed visualisation technique for the measurement output from the external calculation modules. This was realised by hierarchically ordered mask modules, which in turn generate SVG data for individual use by the client (see Section 4.3). Potentials for client interactions have been tested with the interactive web frontend "Orthoweb" which enables users to modify FPs and generate reports (see Subsection 3.1.4).

6.2 Future Work

The discussion points to white spots and open topics concerning the future of IAS. There are some key aspects we will focus on in the near future of development. They will be listed here.

6.2.1 Texture Analysis

One of the more immediate tasks that will follow up on this work is the integration of bone texture analysis. This will be done by cropping the upper area of the tibial bone, on which texture algorithms will heuristically calculate bone density and other parameters concerning the condition of the bone itself.

Technically this new method will use the same hierarchical sub-mask approach as the joint space modules did in this work. New texture mask modules will be integrated into the existing mask hierarchy and will be responsible for data handling and calculations related to bone texture analysis.

6.2.2 Deployment in Lower Austria

At the time of writing, IAS is in its test phase at the "Landesklinikum Horn". We hope to be able to deploy the product in all of Lower Austria by the end of summer 2018.

6.2.3 EnvoyAI Integration

EnvoyAI [Env] is a platform for companies to host their medical analysis software using dedicated Docker containers. We are currently working on porting IAS to this new format. Basic functionality of our software is already integrated into EnvoyAI.

6.2.4 Outlook

In the future we plan to make IAS even more modular by breaking up its architecture into separate containers. These will be designed in a way that they are interchangeable and easy to configure. Clients could then for example choose to analyse their images using the local filesystem and the HL7 standard in one case and easily switch to remote analysis using traditional XML encoded data in another case.

To summarise this work, it can be said that the IAS has been proven to open up many interesting opportunities for automated medical analysis. There is yet great potential to further drive its development.

List of Figures

List of Tables

List of Algorithms

Acronyms

AI artificial intelligence. xiii, 5, 8, 31

AP anterior posterior. 2

CT computer tomography. 2

FP focus point. 16–18, 22, 23, 26, 29, 31

IAS Image Analysis Server. xvi, 1, 3, 4, 11–15, 17, 22–30, 33

JSN joint space narrowing. 2

KI Künstliche Intelligenz. xi

KLS Kellgren and Lawrence System. xi, xiii, 2, 3, 22, 24, 28, 31, 33

kNN k nearest neighbor. 8

MRI magnetic resonance imaging. 5, 6

NN neural network. 5, 6

OA Osteoarthritis. xi, xiii, 1–3, 13, 27–29

PET positron emission tomography. 7

ROI region of interest. 6–8

SVG scalable vector graphics. 4, 15, 17–21, 31

SVM support vector machine. 8

Bibliography

[Ahl68] S. Ahlbäck. Osteoarthrosis of the knee. A radiographic investigation. *Acta radiologica: diagnosis*, pages Suppl 277:7–72, 1968.

[DAB+01] R. H. Dolin, L. Alschuler, C. Beebe, P. V. Biron, S. L. Boyer, D. Essin, E. Kimber, T. Lincoln, and J. E. Mattison. The HL7 Clinical Document Architecture. *Journal of the American Medical Informatics Association : JAMIA*, 8(6):552–69, 2001.

[DCM] DICOM Java Library, https://www.dcm4che.org, accessed on 2018-06-05.

[DIC] DICOM Standard, https://www.dicomstandard.org/current/, accessed on 2018-06-05.

[DLC+06] P. Dengler, C. Lilley, C. McCormack, D. Schepers, J. Ferraiolo, and J. Fujisawa. Scalable Vector Graphics (SVG) 1.1 (Second Edition). *World Wide Web Consortium (W3C)*, 2006.

[EDMRS14] E. S. A. El-Dahshan, H. M. Mohsen, K. Revett, and A. B. M. Salem. Computer-aided diagnosis of human brain tumor through MRI: a survey and a new algorithm. *Expert Systems with Applications*, 41(11):5526–5545, sep 2014.

[Env] EnvoyAI, https://www.envoyai.com, accessed on 2018-06-05.

[Eur17] European Commission. REGULATION (EU) 2017/745 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 5 April 2017 on medical devices, amending Directive 2001/83/EC, Regulation (EC) No 178/2002 and Regulation (EC) No 1223/2009 and repealing Council Directives 90/385/EEC and 93/42/EE. *Official Journal of the European Union*, 60:2–175, 2017.

[GG] S. R. Goldring and M. B. Goldring. Clinical aspects, pathology and pathophysiology of osteoarthritis. *Journal of musculoskeletal & neuronal interactions*, 6(4):376–378.

[ICD] ICD-10 Version:2016, http://apps.who.int/classifications/icd10/browse/2016/en, accessed on 2018-06-05.

[JJV+17] T. Janvier, R. Jennane, A. Valery, K. Harrar, M. Delplanque, C. Lelong, D. Loeuille, H. Toumi, and E. Lespessailles. Subchondral tibial bone texture analysis predicts knee osteoarthritis progression: data from the osteoarthritis initiative: tibial bone texture & knee OA progression. *Osteoarthritis and Cartilage*, 25(2):259–266, feb 2017.

[JJZ+17] F. Jiang, Y. Jiang, H. Zhi, Y. Dong, H. Li, S. Ma, Yi. Wang, Q. Dong, H. Shen, and Yo. Wang. Artificial intelligence in healthcare: past, present and future. *Stroke and Vascular Neurology*, 2(4):230–243, dec 2017.

[Kan] A. Kanitsar. VISUAPPS GmbH, http://www.visuapps.com, accessed on 2018-06-05.

[KFW+09] V. B. Kraus, S. Feng, S. Wang, S. White, M. Ainslie, A. Brett, A. Holmes, and H. C. Charles. Trabecular morphometry by fractal signature analysis is a novel marker of osteoarthritis progression. *Arthritis & Rheumatism*, 60(12):3711–3722, dec 2009.

[KL57] J. H. Kellgren and J. S. Lawrence. Radiological assessment of osteo-arthrosis. *Annals of the rheumatic diseases*, 16(4):494–502, dec 1957.

[KSF16] M. D. Kohn, A. A. Sassoon, and N. D. Fernando. Classifications in brief: Kellgren-Lawrence classification of osteoarthritis. *Clinical orthopaedics and related research*, 474(8):1886–93, aug 2016.

[KSQ+09] A. Kerhet, C. Small, H. Quon, T. Riauka, R. Greiner, A. McEwan, and W. Roa. Segmentation of lung tumours in positron emission tomography scans: a machine learning approach. pages 146–155. Springer, Berlin, Heidelberg, 2009.

[Lju17] R. Ljuhar. Image Biopsy Lab - IB Lab GmbH, https://imagebiopsylab.com/, accessed on 2018-06-05, 2017.

[LOI] LOINC - Medical identifcation system, https://loinc.org/, accessed on 2018-06-05.

[MGD+06] M. E. Mavroforakis, H. V. Georgiou, N. Dimitropoulos, D. Cavouras, and S. Theodoridis. Mammographic masses characterization based on localized texture and dataset fractal analysis using linear, neural and support vector machine classifiers. *Artificial Intelligence in Medicine*, 37(2):145–162, jun 2006.

[MYA+18] P. Mobadersany, S. Yousefi, M. Amgad, D. A. Gutman, J. S. Barnholtz-Sloan, J. E. Velázquez Vega, D. J. Brat, and L. A. D. Cooper. Predicting cancer outcomes from histology and genomics using convolutional networks. *Proceedings of the National Academy of Sciences*, 115(13):E2970–E2979, mar 2018.

[Neo12] T. Neogi. Clinical significance of bone changes in osteoarthritis. *Therapeutic advances in musculoskeletal disease*, 4(4):259–267, aug 2012.

[PBS+97] I. F. Petersson, T. Boegård, T. Saxne, A. J. Silman, and B. Svensson. Radiographic osteoarthritis of the knee classified by the Ahlbäck and Kellgren & Lawrence systems for the tibiofemoral joint in people aged 35-54 years with chronic knee pain. *Annals of the rheumatic diseases*, 56(8):493–6, aug 1997.

[Pos] Postman - API Development Environment, https://www.getpostman.com, accessed on 2018-06-05.

[RMG00] P. Read, M. P. Meyer, and Gamma Group. *Restoration of motion picture film*. Butterworth-Heinemann, 2000.

[SBBZ08] D. Schiphof, M. Boers, and S. M. A. Bierma-Zeinstra. Differences in descriptions of Kellgren and Lawrence grades of knee osteoarthritis. *Annals of the Rheumatic Diseases*, 67(7):1034–1036, jul 2008.

[SNG+13] N. A. Segal, M. C. Nevitt, K. D. Gross, J. Hietpas, N. A. Glass, C. E. Lewis, and J. C. Torner. The multicenter osteoarthritis study: opportunities for rehabilitation research. *PM&R*, 5(8):647–654, aug 2013.

[WPS+12] T. Woloszynski, P. Podsiadlo, G. W. Stachowiak, M. Kurzynski, L. S. Lohmander, and M. Englund. Prediction of progression of radiographic knee osteoarthritis using tibial trabecular bone texture. *Arthritis & Rheumatism*, 64(3):688–695, mar 2012.