TU WIEN Informatics

Prognose einer schnell fortschreitenden Knie-Osteoarthritis mithilfe eines Convolutional Neural Network

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieurin

im Rahmen des Studiums

Biomedical Engineering

eingereicht von

B.Sc. Magdalena Vogel
Matrikelnummer 11927170

an der Fakultät für Informatik
der Technischen Universität Wien
Betreuung: Ast. Prof. Dr. Renata Raidou

Wien, 1. September 2022

Magdalena Vogel | Renata Raidou

TU WIEN Informatics

Prediction Of Accelerated Knee Osteoarthritis Using a Convolutional Neural Network

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieurin

in

Biomedical Engineering

by

B.Sc. Magdalena Vogel
Registration Number 11927170

to the Faculty of Informatics

at the TU Wien

Advisor: Ast. Prof. Dr. Renata Raidou

Vienna, 1st September, 2022

Magdalena Vogel

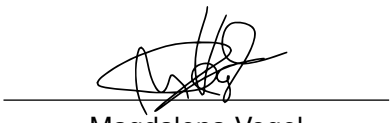
Renata Raidou

Erklärung zur Verfassung der Arbeit

B.Sc. Magdalena Vogel

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, 1. September 2022



Magdalena Vogel

Danksagung

Zuallererst möchte ich meinem Betreuer, Zsolt Bertalan, von ImageBiopsyLab herzlich danken, der mir in jeder Situation geholfen und mich unterstützt hat. Ich habe in meinen Monaten bei IBLab sehr viel gelernt, was ich vorallem dir zu verdanken habe. Auch vielen Dank an meine Betreuerin an der TU Wien, Renata Raidou. Danke für die organisatorische Begleitung meiner Masterarbeit und die Geduld für viele Fragen.

Kurzfassung

Osteoarthrose (OA) ist weltweit die häufigste degenerative Gelenk Krankheit, wobei ein Endstadium innerhalb von 10 bis 15 Jahren erreicht wird. In 3.4% der Fälle tritt jedoch eine sich schnell entwickelnde KOA auf (AKOA), bei sich die Entwicklung bis zum Endstadium auf 1 bis 4 Jahre reduziert und häufig auf ein künstliches Kniegelenk hinausläuft. Eine frühzeitige Diagnose von AKOA könnte mehr Zeit verschaffen, um alternative, weniger invasive, medizinische Behandlungen zu finden. Außerdem würde die Forschung im Bezug auf krankheits modifizierender Medikamente beschleunigt werden, indem AKOA Patienten als Testgruppe einfacher rekrutiert werden können. Bisher ist die Vorhersage von AKOA anhand eines Röntgenbildes nicht möglich, da keine wesentlichen optischen Unterschiede KOA von AKOA im frühen Stadium zu erkennen sind. Da Neuronale Netzwerke in der Lage sind Strukturen auf Bildern zu erkennen, welche für das menschliche Auge nicht ersichtlich sind, werde ich in dieser Arbeit ein Convolutional Neural Network (CNN) verwenden als Klassifizierer verwenden, um AKOA zu prognostizieren.

Die mir zur Verfügung stehenden Daten extrahierte ich aus den Datensätzen drei verschiedener Studien. Als Input für das Netzwerk dienten neben dem Röntgenbild, numerische Informationen über Body Mass Index (BMI), Alter, Geschlecht, Western Ontario and McMaster Universities Arthritis Index (WOMAC) score, symptome in der Hüfte, Zuführung von Knieарthrose Medikamenten und die Kellgren-Lawrence (KL)-grade. AKOA wurde zuerst mit $> 10\%$ Gelenkspalt Verringerung (JSN) und später mit $> 20\%$ Gelenkspalt Verringerung innerhalb von mindestens 2 Jahren definiert. Beide Definitionen verwendete ich, um die Vorhersagekraft des Netzwerks zu optimieren. Mit numerischen Daten trainierte ich ein Extreme Gradient Boosting (XGBoost) Modell unter Einbezug aller numerischer Features und dem Osteoarthritis Research Society International (OARSI) score von Sklerose und Osteophytose. Hierbei erziehlte das Netzwerk eine AUC (Fläche unter der Receiver Operating Characteristic (ROC) Kurve) von 0.6616 (20% JSN/ 2 Jahre). Um die Bilddaten zu integrieren, nutze ich ein CNN, dessen Architektur auf einem Residual Network (ResNet) 50 basiert. Das CNN mit rein Bilddaten als Input, klassifizierte mit einer AUC von 56.26% (10% JSN/ 2 Jahre). Nach dem Hinzufügen der wichtigsten numerischen Daten (Geschlecht, BMI, kontralaterale KOA, KL-grade) als Input, erreichte ich eine AUC von 68.78% (20% JSN/ 2 Jahre).

Diese Ergebnisse zeigen, dass es möglich ist eine Risikoeinschätzung über die Entwicklung

von AKOA mithilfe eines Röntgenbildes und der numerischen Daten von Geschlecht, BMI, KL-grade und der Information über vorliegende KOA zu machen. Da bisher keine anderen verlässlichen Hilfsmittel und Methoden zur Verfügung stehen, haben Neuronale Netzwerke großes Potenzial dies zu ermöglichen.

Abstract

Osteoarthritis (OA) is a slowly degenerative joint disease, with cartilage loss as one of the most characteristic symptoms accompanied by pain and functional disability. The knee region is the most affected area. 22.9% of the worldwide population over the age of 40 were affected in 2020 by Knee Osteoarthritis (KOA). Besides normal KOA, which develops over multiple years, the accelerated form of KOA(AKOA) develops between 1 and 4 years and is accompanied by increased pain and movement restrictions as well as a higher chance of obtaining a knee replacement. The development of AKOA is not yet predictable on the basis of a single X-ray image because there is no obvious optical difference between the baseline X-ray of KOA and AKOA. Since Convolutional Neural Networks (CNN) are able to identify image structures, a human eye can not see, I want to realise an early diagnosis of AKOA by using a Convolutional Neural Network (CNN) as a classifier between slow- and fast-progressing KOA.

For this purpose, I used the data from three different studies, including knee X-ray, Body Mass Index (BMI), age, gender, Western Ontario and McMaster Universities Arthritis Index (WOMAC) scores, hip symptoms, knee medication injection and Kellgren-Lawrence (KL)-grade, as input for binary classification models. I defined AKOA once with Joint Space Narrowing (JSN) $>$ 10%/ 2 years and once with JSN $>$ 20%/ 2 years and performed different experiments in order to find the best method to predict AKOA. I trained the numeric data only on an Extreme Gradient Boosting (XGBoost) model. Here I achieved the highest performance of an Area Under the Curve (AUC) of 0.6616) when including the Osteoarthritis Research Society International (OARSI) score of sclerosis and osteophytosis to the numeric input data (20% JSN/ 2 years). To use image data only and the combination of both I created different CNN models, whose architecture is based on a Residual Network (ResNet) 50 model provided by ImageBiopsyLab. The CNN model, which I trained only with image data, yielded an AUC of 56.26% (10% JSN/ 2 years). Using the image data complemented with the most important numeric features (gender, BMI, contralateral KOA, KL-grade) as input, I achieved an AUC of 68.78% (20% JSN/ 2 years). Comparable results, but obtained with other class definitions than in this work, were higher and yielded AUCs of around 0.8.

These results show that it is possible to make a risk assessment about the development of AKOA using the baseline X-ray image, gender, BMI, the KL-grad and the information about contralateral KOA. Until now, radiologists are not capable of predicting fast-

progressing KOA. Hence, these networks have a great potential to be used as AKOA prediction tools.

Contents

CHAPTER 1

Introduction

1.1 Motivation and Problem Definition

Osteoarthritis is a degenerative joint disease, with loss of cartilage as one of the most characteristic symptoms [1]. Accompanied by a lot of pain and movement restrictions, especially for knee or hip osteoarthritis, the most effective treatments so far are pain medication and joint replacement. Besides being a radical medical intervention, joint replacement is very expensive and is considered to be a high socioeconomic burden, due to high costs for health insurance [2].

The most affected area of the joint disease Osteoarthritis (OA) is the knee region [3]. In 2020 22.9% of the worldwide population over the age of 40 were affected by knee OA (KOA) [4]. Usually, KOA progresses slowly over several years and with current possibilities, diagnosis happens proportionally late [5]. Among all KOA patients, 3.4% reach the advanced stage of disease already within 4 years, and some even less than 12 months measured from the time where no radiographic symptoms were visible.

The fast-progressing course is known as accelerated KOA (AKOA) [6]. The exact definition of AKOA varies between different studies. Here, fast-progressing KOA is defined using joint space narrowing (JSN) per year. It is proven that this specific form of KOA leads to an increased likelihood of frequent knee pain, as well as to even higher movement restrictions. Thus, the quality of life is significantly reduced compared to regular OA [7]. Additionally, one out of seven of these patients receives a knee replacement in a median time of 2.3 years [8]. Early detection of AKOA would bring more time to determine alternative and less radical medical interventions [8, 9].

Concerning drug development, the prediction of disease would also bring a clear advantage. The long and so far unpredictable progression of KOA, makes it almost impossible to realise a disease modification relating to the intake of drugs. Hence, if it would be possible to detect fast progressors in the early stage, this group of patients could represent a

more homogenous test group resulting in faster results [10]. With the currently available knowledge, the chance to develop AKOA can be roughly estimated including typical risk factors like Body Mass Index (BMI) and age [6]. Still, too little is known about the development, etiology and exact prediction factors of AKOA [11]. It is not yet possible to make a precise classification between slow- and fast-progressing KOA patients based on a single radiograph. Hence, a method is missing which takes, in addition to the most relevant demographic and clinical factors, a knee X-ray image into account to classify between the slow and fast progression of KOA.

1.2 Aim of the Work

The analysis of radiographs, i.e., defining the severity of KOA performed by a human, is highly affected by subjectivity. This makes the use of neural networks very attractive in order to achieve more uniform and accurate decisions. Some approaches exist, also based on Convolutional Neural Network (CNN) models, which predict AKOA among slow and non-progressing patients. More clinically relevant, however, because of a higher number of available images, is the case of predicting only between slow- and fast-progressing KOA patients. Therefore, this thesis aims to develop a new strategy based on a neural network, which can differentiate between slow- and fast-progressing KOA at an early stage. This could allow an earlier start of disease treatment, which could result in more effective and less radical medical interventions [11]. The accurate detection of fast progressors would also facilitate the selection of patients to test disease-modifying drugs [10].

The aim of this work is to use next to the X-ray image data demographic and clinical characteristics of the patient (numeric data) to answer the research question **"Is it possible to classify KOA progression into fast and slow progression (defined by JSN per year) using Convolutional Neural Networks?"**. The first task is to figure out how to define the classes, slow and fast progressors, and which demographic and clinical factors are important to consider. The next goal will be to implement a model, which will be able to consider a baseline X-ray image in combination with the most relevant clinical and demographic factors to classify between slow- and fast-progressing KOA. Even if most of the existing studies identified AKOA among slow- and non-progressing patients, the exclusion of non-progressors could better represent the cohort of radiographs taken in a hospital. To accomplish the challenge, defined in the research question, I want to figure out the optimal architecture and model parameters, the most relevant numeric factors and the better threshold of the class definition to be able to predict AKOA.

1.3 Methodological Approach

The first step consists of a literature review that focuses on the different factors influencing the progression of KOA. In addition to these findings, I consider medical expert opinions to make a pre-selection of the most common criteria leading to AKOA and to define the

exclusion criteria for the composition of the sub-cohort of patients for this work. The data I use consists of three different datasets with the aspect of having as diverse data as possible. I compile different batches of data, using different class definitions. The first definition of at least 10% JSN per two years was previously defined by Image Biopsy Lab(IBLab), according to previous literature and medical experts' opinion. As a second, a more extreme threshold, I use 20% JSN per two years, in order to increase the difference between slow and fast progressors. I use these definitions further to label the data by classes 0 and 1, corresponding to slow- and fast-progressing KOA, respectively. I analyse the data of all three datasets, concerning possible correlations between AKOA and some selected numeric factors, as well as to detect possible dataset-specific characteristics, which could influence the quality of training.

As a next step, I train an Extreme Gradient Boosting (XGBoost) model with the numeric data, in order to create a reference and to solve the task of evaluating the importance and the power of only the numeric features for the classification task. To add the X-ray data as input, I create different CNN models. For some training runs, I use a pre-trained model, provided by IBLab, and apply the method of transfer learning to use the knowledge of this pre-existing model. Other training runs I perform from scratch. All models are based on the architecture of RetinaNet. In order to optimise the accuracy of classifying between slow- and fast-progressing KOA, I train and evaluate five different CNN models, with different data batches. For evaluation I use the area under the Receiver Operating Characteristic curve (AUC-ROC) and Confusion Matrices. In addition, I carry out experiments using two datasets as training set and evaluating with the third one (e.g., training with Osteoarthritis Initiative (OAI) and Cohort Hip and Cohort Knee Study (CHECK) and testing on Multicenter Osteoarthritis Study (MOST). This shows the ability of a model to perform on images from a, for the model, unknown dataset. The contribution of this work is a neural network, which classifies between fast- and slow-progressing KOA patients using basic demographic and clinical information next to the X-ray image of the patient. The definition of AKOA in this work is easy and hence less prone to error compared to the definition based on the KL-grade. Besides, it is clinically more relevant to exclude non-progressing KOA patients. Since there does not exist such an X-ray-based method to identify AKOA yet, this new diagnostic method would be a reliable, simple, and fast method, which only requires easily accessible data.

1.4 Structure

The thesis is structured into four big parts. The first chapter encompasses the introduction of the topic together with the problem definition and the short preamble describing the development of the model. For the second chapter, a clinical background regarding KOA and its current ways of analysis is provided in order to facilitate the problem's comprehension. Moreover, the technicalities concerning a CNN will be clarified in the chapter "Technical Background". Subsequently, related work regarding the prediction of AKOA will be presented. The next chapter will provide the full approach, from the literature review and the data selection to the implementation of the models and

experiments. This is followed by the sixth chapter, which lays down the results and outcomes to be further evaluated and discussed. The final chapter encompasses the conclusions made, as well as the appropriate next steps for further development of this topic.

CHAPTER 2

Clinical Background

In this chapter, the clinical background of OA and its progression and definition will be described. Worldwide, OA is the most spread joint disease [2]. In the age group, between 40 and 50 years, people already have an increased risk of developing OA [12]. Almost every person who reaches the age of 80 is affected by OA, resulting in even higher numbers of OA patients with an increasingly ageing population [12, 13]. Despite diagnosing OA on radiographs, symptoms may not be present though [12]. Therefore, the number of unrecorded cases of OA could be even higher.

OA is a degenerative joint disease, which can occur in all parts of the body. The most common area is the knee joint [3]. All parts of the knee (medial, lateral and patellofemoral joint), as well as the surrounding tissue [12], can be affected [14]. The medial side of the joint is more often affected by OA than the lateral side [15]. Commonly the first and most characteristic symptom of KOA is the loss of cartilage [8]. As seen in Figure 2.1, the articular cartilage is one of the main components of the knee joint. A healthy knee shows an equilibrium of degenerative and regenerative enzymes to keep the cartilage maintained [17]. Among KOA patients an increased load of degenerative enzymes can be observed, which leads to a reduction of cartilage, ending up in JSN [17]. Next to the volume loss of cartilage, the morphology of the subchondral and periarticular bone changes as well [18]. As a result of early bone reformation bone mass increases locally and osteophytes are formed [18] at the anterior and posterior endings of the bone, as seen in Figure 2.2. These formations can then imply the increased joint stiffness among AKOA patients [19]. Sclerosis, which manifests as thickening of the subchondral bone [20], is considered to be another reinforcing factor that increases joint stiffness [18]. Consequently, stiffness of the subchondral bone rises, which increases peak dynamic forces thus resulting in even more bone and cartilage damage [18]. These adjustments in joints can imply inflammation, swelling, stiffness, and increased pain hence reducing the patient's physical functionality [10, 14].

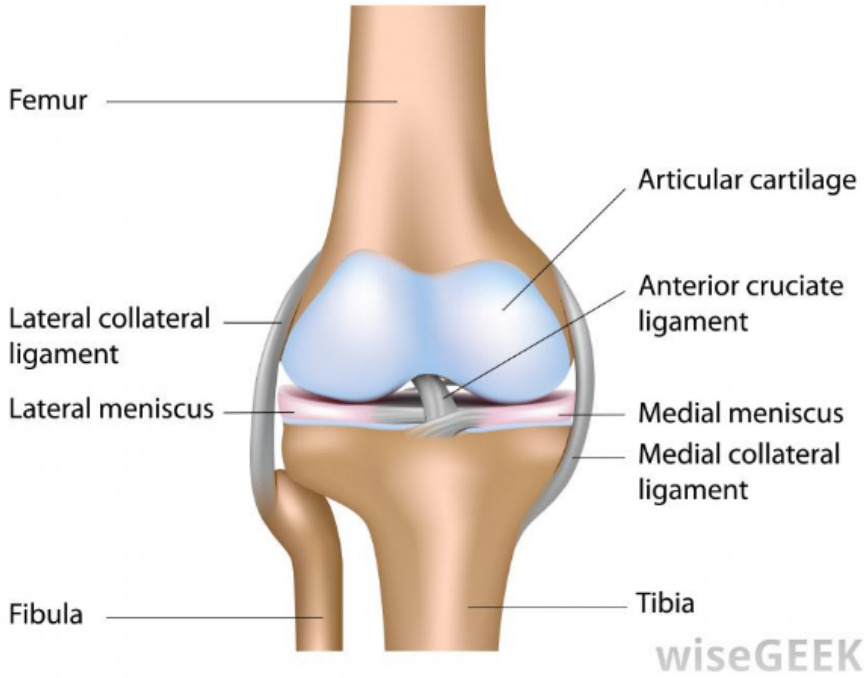


Figure 2.1: Anterior view of a human knee [16].

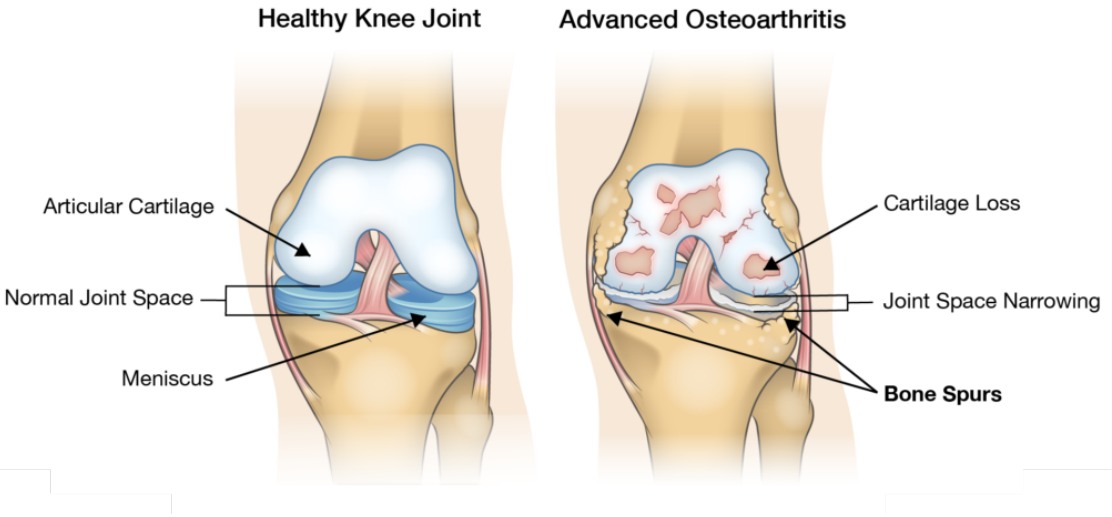


Figure 2.2: Anterior view on a healthy knee joint (left) and a knee joint with advanced OA with visible joint space narrowing, cartilage damage and bone spurs [21].

For each patient, the development, course, and symptoms of disease can vary [17]. This heterogeneity of osteoarthritis for each patient makes it difficult to predict progression

[17, 22]. In usual cases, KOA develops over 10 to 15 years [14], but around 3.4% of all OA patients experience a fast progression OA. Meaning within 4 years, sometimes even less than 12 months, the advanced stage of disease is reached [6]. It is proven that this specific form of OA leads to greater pain, even prior to radiologic diagnosis [23], an increased likelihood of frequent knee pain, as well as to higher movement restrictions [7, 15]. In spite of KOA being such a common and painful disease, there is still little knowledge about concrete causes, development and treatment of knee osteoarthritis [11]. This culminates in the inability of differentiating the patients between slow and fast progressors.

Disease influencing factors can be split into modifiable and non-modifiable ones [14]. Several studies showed that non-modifiable factors, for instance, advanced age, being female, having former knee surgeries, knee injury or having a preloaded family history, can increase the risk of developing AKOA [6, 9, 13, 14, 24]. The most common modifiable factor is obesity corresponding to a high BMI [14] and also to hard physical work [17]. The BMI is calculated by dividing the weight by the squared height of the patient. Depending on these results the patients can then be classified into underweight, normal weight and overweight, corresponding to lower than 18.5 kg/m^2, between 18.5 kg/m^2 and 24.9 kg/m^2 and greater than 24.9 kg/m^2, respectively [25]. A high BMI and physical work increase the wear and tear in cartilage and meniscus, which may lead to meniscal pathology and influence initiation of disease [17]. This was also proved by Harkey et al. [7]. They showed evidence of medial meniscal extrusions as being a characteristic risk of developing AKOA even one year prior to disease onset [7]. On baseline images meniscal pathology, for instance, meniscal tears, degenerative ligaments, and effusion-synovitis have been identified as radiographic symptoms of AKOA and represent an increased risk of being a fast progressor [7].

Moreover, the geometry of the knee joint can influence the development of AKOA. Knee malalignments influence the biomechanics of the whole joint and therefore lead to changes in the coronal tibial slope. The slope can be measured by connecting the medial and lateral upper part of the tibia plateau, as seen in Figure 2.3. A greater slope is primarily associated with the incidence of Joint Space Width (JSW) reduction of the medial part compared to the lateral part of the knee; besides medial joint space narrowing (JSN) occurs more often. The risk of developing AKOA increases 15% with every degree of increased tibial slope [15]. What makes this study even more interesting, is the fact that the slope just correlates to the AKOA but not to KOA [15]. Hence, from this study, it can be concluded that measuring the medial joint space narrowing per year makes sense in order to define AKOA. Furthermore, it shows X-rays to be promising to use as a prediction method for AKOA.

Another problem is the lack of disease-modifying treatment methods [10]. Since the current most effective treatment consists of behavioural changes, such as body weight reduction or decreasing peak load on joints by avoiding hard physical work [11, 14]. Medical treatments include pain medication, steroidal injections or anti-inflammatory drugs, which have in common to only treat the symptoms, but not the causes [2, 22].

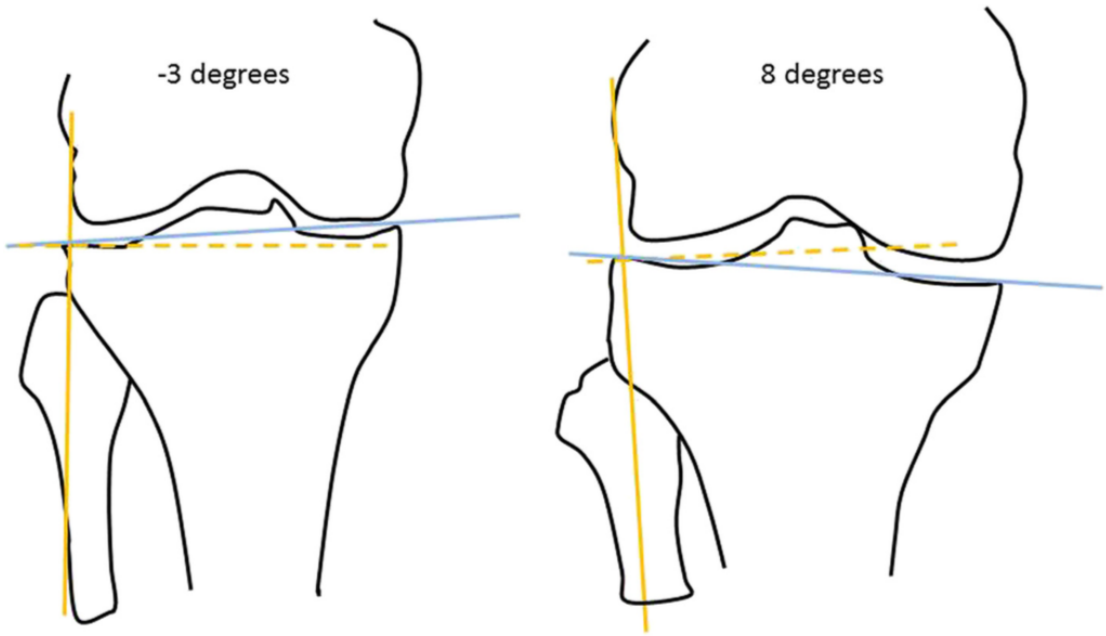


Figure 2.3: Frontal view of two knees with -3 degree (left) and 8 degree (right) tibial slope. Yellow lines: vertical and horizontal reference lines. Blue lines: connection between left and right corner of the tibia to measure tibial slope [15].

Despite these medical treatments, most of the patients receive knee replacements, causing additional pain and costs [2, 22]. Especially among fast progressors, surgery for an implant occurs in the median time of 2.3 years, which translates into less time for searching for less radical methods [8].

Besides, the research in disease-modifying drugs is hampered by the unpredictability and heterogeneity of the course of KOA [10]. Due to the long course for testing drugs on regular KOA patients, the observation period would be quite long. The long and so far unpredictable progression of KOA poses a big challenge to realise a disease modification, which relates to the intake of drugs. Hence, with early detection of fast progressors, a more homogeneous group of AKOA patients can be selected to test new drugs. This could lead on one hand to faster outcomes and on the other hand to a higher quality of results [10].

2.1 Grading and Scoring Systems for Knee Osteoarthritis

To be able to classify different progression of OA, there exist multiple grading scales. The most popular one is the Kellgren-Lawrence (KL) grading system with grades from 0 (no OA) to 4 as seen in the following table. The system integrates information of JSN, osteophytes, sclerosis and deformation of bone [14]. What complicates the usage of the KL-grading system is the fact of defining an OA grade for the whole knee, disregarding

KL	JSN	Osteophytes	Sclerosis	Bone deformation
0 = None OA	no	no	no	no
1 = doubtful OA	doubtful	possible lipping	no	no
2 = minimal OA	possible	definite	no	no
3 = moderate OA	definite	moderat multiple	yes	possible
4 = severe OA	marked	large	severe	definite

Table 2.1: Description of the Kellgren-Lawrence System, splitted in JSN, osteophytes, sclerosis and bone deformation [26].

the lateral and medial side of the joint [27]. A scoring system, which allows grading the lateral and medial part of the knee individually is the Osteoarthritis Research Society International (OARSI) score. It suggests a grading from 0 to 5 for the individual issues JSN, osteophytes, and sclerosis and hence represents a more specific scoring system. This results in a better assessment of the heterogeneous development of AKOA [27]. Since both of the scoring systems are affected by subjectivity, using neural networks could compensate for this disadvantage.

To describe the symptomatic situation of KOA, the Western Ontario and McMaster Universities Arthritis Index (WOMAC) score was invented [28]. The scoring system is split into three sections: pain, stiffness and physical function, each containing different self-answered questions. Every question can be answered with a score between 0 and 4, with 0 as none and 4 as extreme. Pain is assessed during sitting, lying, in bed, upright standing and walking. Knee stiffness is observed after the first walk after a break and towards the end of the day. The physical function consists of the following 17 questions: "using stairs, rising from sitting, standing, bending, walking, getting in/ out of a car, shopping, putting on/ taking off socks, rising from bed, lying in bed, getting in/ out of bath, sitting, getting on/ off toilet, heavy domestic duties, light domestic duties" [28]. These answers result in WOMAC pain, WOMAC stiffness and WOMAC disability score, scaled from 0-20, 0-8 and 0-68 respectively. All individual measurements can then be included in the summed up WOMAC total score [28].

2.2 Definitions of Accelerated Knee Osteoarthritis

Different definitions are used to classify KOA patients into no progressors, slow and fast progressors. The most common definition makes use of the above-mentioned KL-grading system. No changes of the KL-grade within 48 months define non-progressing KOA. An increase of at least 1 KL-grade is classified as progressing KOA. Among these, patients with KL-grade lower than 1 reaching a KL-grade larger than 2 within 4 years, are defined as fast progressors; the remaining ones as slow progressors [6, 7, 15, 23]. As well the loss of cartilage volume is used as a measuring factor of AKOA. The volume loss is observed over a time of two years. Knees with a global reduction of more than 13-15% are classified as fast progressors. Slow progressors show a cartilage volume loss of less than 2% [13, 29].

Considering now the reduction of the JSW, AKOA can also be defined by an absolute number of JSN per time. Numbers between 0.25 mm and 1.05 mm per year can be found in literature [30, 31, 32]. But since the initial JSW and the patient's anatomy would also be important to consider, absolute numbers can not help this. Using a percentage of JSN as a class definition, I will take this aspect into account. Besides, JSN is connected to both of the previous class definitions. The minimal JSN correlates to the cartilage volume loss, according to Eckstein et al. [33] and JSN is also used for the definition of the KL-grade as explained in Section 2.1. Hence, using the JSN to define AKOA could be a promising and simple compromise.

2.3 Summary

Regarding the background of AKOA and the importance of an early diagnosis the following facts should be kept in mind. The most characteristic symptom of KOA is the loss of cartilage, which further leads to a local change of bone mass and consequently to increased joint stiffness, inflammation and pain. These symptoms are reinforced for AKOA, of which 3.4% of all KOA patients are affected. Here, the end-state can be reached between 1 and 4 years and result often in artificial knee replacements. Risk factors for fast progressors are high age, high BMI, being female, and knee geometry. The latter is only correlated to AKOA but not to KOA, which is promising for a prediction tool based on X-ray images. AKOA can be defined by an increase of the KL-grade, change of cartilage volume, and reduction of the JSW. It would be an important achievement to predict AKOA since an early diagnosis could lower the number of knee replacements and reduce pain and costs.

CHAPTER 3

Technical Background

In this chapter, I delineate the current state of the art, concerning the technical methods I use in this work to build the classification model. Technical background information about CNNs, the different layers and functions, the XGBoost model, which is used to train on numeric data, and transfer learning will be given. The latter is applied to use the knowledge of pre-trained models to improve the performance of the new CNN model.

3.1 Convolutional Neural Network

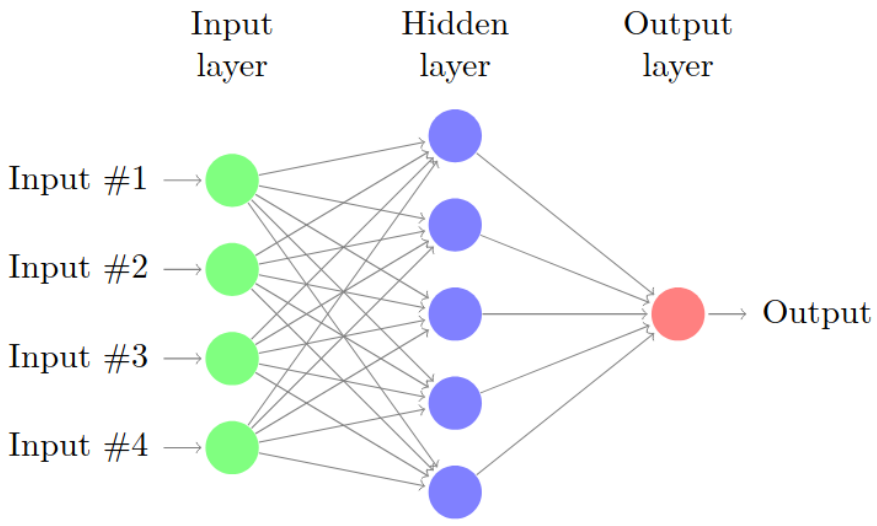


Figure 3.1: Simple structure of a neural network with green dots as input layer nodes, blue dots as hidden layer nodes and the red dot as output node[34].

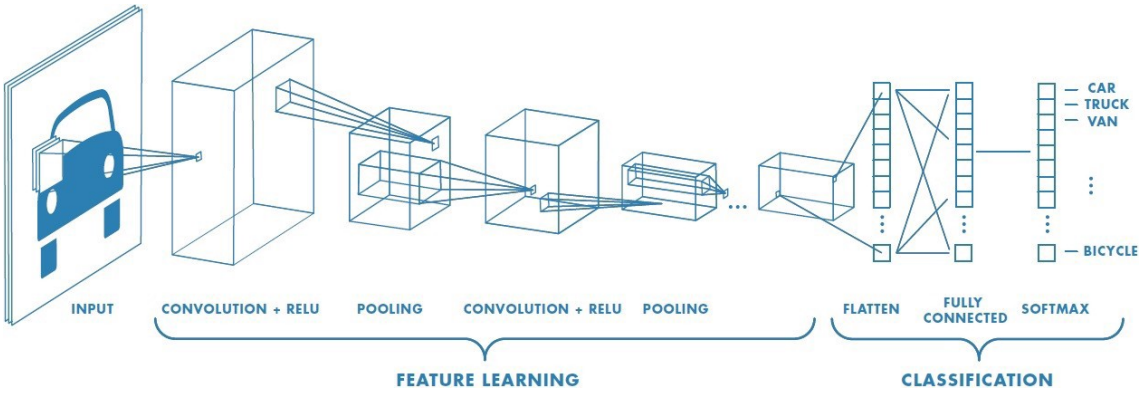


Figure 3.2: Structure of a Convolutional Neural Network. From left to right: input, feature learning, classification. Feature learning contains convolutional layers with ReLU as activation function and pooling layers (Section 3.1.1 and 3.1.2). Classification contains flatten, dense and softmax layers (Section 3.1.3) [35].

The smallest building block of a neural network is a single neuron. The principal structure, which is inspired by the human brain [35], can be seen in Figure 3.3. Each neuron has a function $f()$, which depends on one or more input variables. In this example, these variables are x and y and the function subsequently $f(x, y)$. The output is given as z. The process of adding input and calculating the output is called forward propagation, which can be seen as "Forward Pass" in Figure 3.3. Backpropagation or "Backward Pass" corresponds to the reverse calculation [34]. Multiple neurons together built a so-called layer. An input layer, a hidden layer, and an output layer together build up a neural network, which can be seen in Figure 3.1 [34]. In this work the input layer refers to the image input and the output layer is represented by the two output nodes of slow- and fast-progressing KOA.

A special form of a neural network is a CNN, which reduces dimensionality without losing important information [35]. Multiple arrays can be processed, which allows CNN to detect patterns on images [36], such as, for this work important, characteristic structures of AKOA on X-ray images. This is used to recognize multiple objects and differentiate images from one another [35]. A CNN can learn these characteristics of the input and improve its classification ability by forward and backward propagation [34, 35]. With forward propagation an output is calculated, which then can be compared to the ground truth, the actual value [34]. Concerning this work, the prediction output for a class 1 (fast progressor) image could be for instance 0.7, which is compared to the ground truth of 1. This difference is calculated with the loss function [34, 37]. For individual tasks and applications, there exist different kinds of loss functions. One common loss function is cross-entropy. The categorical cross-entropy would be used for more than two classes, binary cross-entropy for binary classification problems, as required in this work [38]. Mathematically it is described as in the following equation (Equation 3.1), where $\hat{y}$ represents the output from the model and y the ground truth [37]:

$$Loss = -\frac{1}{outputsize} \sum_{i=1}^{outputsize} y_i \cdot log\hat{y}_i + (1 - y_i) \cdot log(1 - \hat{y}_i) \tag{3.1}$$

This loss is then sent backwards to update the weights of the neurons (backward propagation) [34]. In order to find the optimised weights, the minimum of the loss function has to be identified by using a descent gradient method [39]. A special form of it is called Adaptive Moment Estimation (ADAM) [40], which I will also use for training my models. In previous works, the ADAM optimizer is suggested to deliver faster and better results than the other optimization methods, like Adadelta or AdaGrad [40, 41]. This is also influenced by the fact that the learning rate, which defines the step size for finding the minimum, is adapted for every single parameter in every step [37, 39].

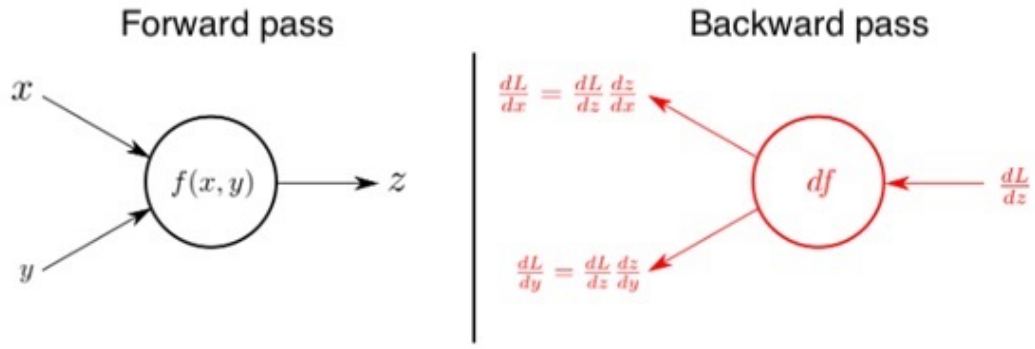


Figure 3.3: Image of a single neuron with x and y as input variables and z as output. Left: forward propagation, right: backpropagation. [42]

To assess the model's performance, a metric function is added at the end. There exist different types, depending on the desired evaluation of the model, such as accuracy metrics, regression metrics or classification metrics. Classes of the classification metrics are for instance the Area Under the Curve (AUC), precision or recall class. Since my model is a classification model and in most of the studies similar to my work the AUC class is used, I will also use this metric function to make the results comparable.

Before starting the training of a neural network, the whole data has to be split into training, tuning and test set. Usually, the largest amount of data will be used for training. This is the data the network uses during its learning process as described above. The data of the tuning set the model uses for validation during training and to optimise the hyperparameters of the model. After training and tuning are accomplished, the model can be evaluated by using the test set. Testing for instance different models with the same test set allows a good comparison of these [43].

The architecture of CNNs can be seen roughly in Figure 3.2. The input is followed by the feature extraction part, which are the hidden layers, consisting of convolutional

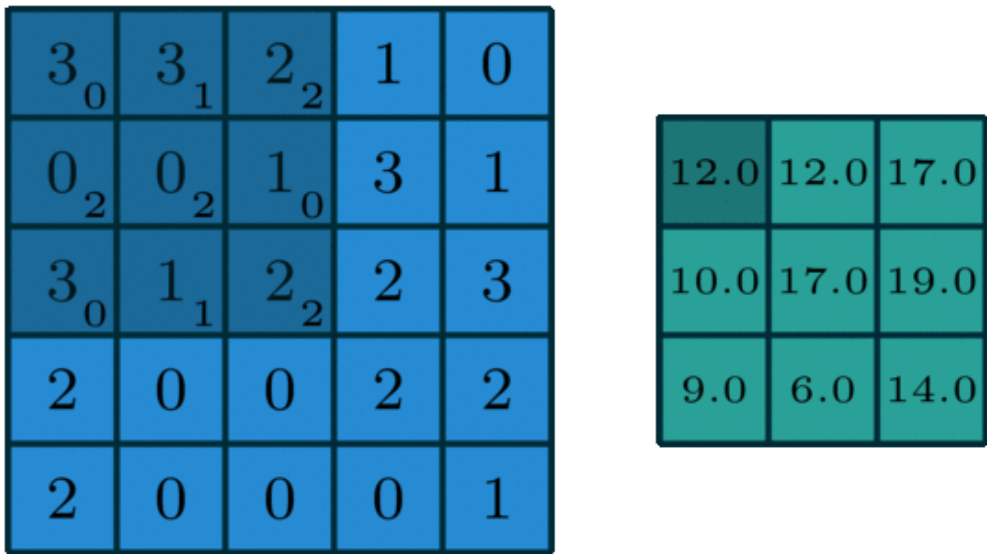


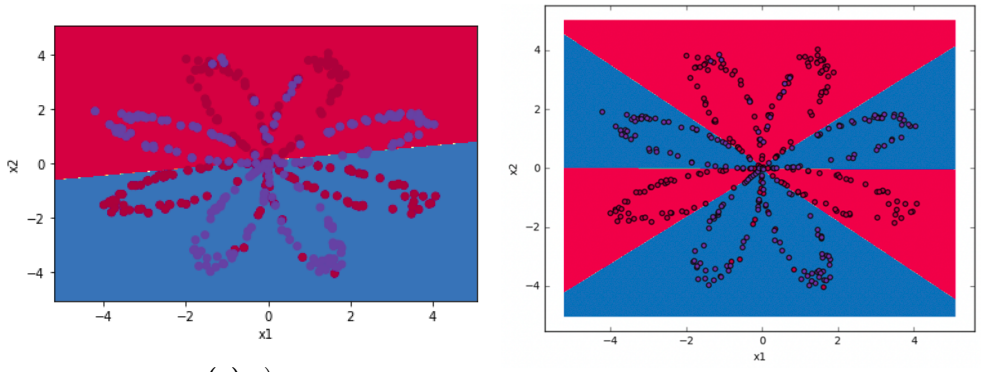
Figure 3.4: Schematic principle of a standard convolution. Light blue matrix: input matrix, dark blue matrix: kernel (small numbers represent the weights of the kernel), green matrix: convolution output matrix containing summed values [44].

layers with activation function and pooling layers. The classification part is assembled of flattening, fully-connected layers and softmax function [35, 36]. The function of each layer, which also forms the model I use, is described in the following sections.

3.1.1 Convolutional Layers

The main components of a CNN are convolutional layers. Here the first step of dimensionality reduction happens. These layers convert a larger matrix into a smaller matrix with features still located at roughly the same position, as seen in Figure 3.4 [44]. The left square of the image acts as the input matrix, in which the dark blue part represents the kernel. A kernel, which usually has a smaller size than the image, scans the input matrix in order to summarise the covered values [35]. The small numbers in the bottom right corners of the kernel squares are the weights of the kernel matrix [44], which are then multiplied by the respective area of the input matrix [35]. These results are subsequently mapped on a feature map [45]. In this way, the size of the output matrix decreases, but depth increases [35]. For example, when having a coloured image with all three RGB channels as input, one independent kernel with individual weights is used per layer [35, 46]. This will provide a three-channelled output [46]. Hence the convolutional layer is used as an extractor of high-level features, which are meant to be rough characteristics of images, like shapes and edges [35]. The more convolutional layers added in a row, the more abstract features can be learned by the network [47].

Convolutional layers also make use of activation functions. There are linear and non-linear activation functions. In general, these functions convert the output into values between



(a) a)

(b) b)

Figure 3.5: Example of logistic regression using in a) a linear function and in b) a non-linear function for classifying data points into class red and class blue [48].

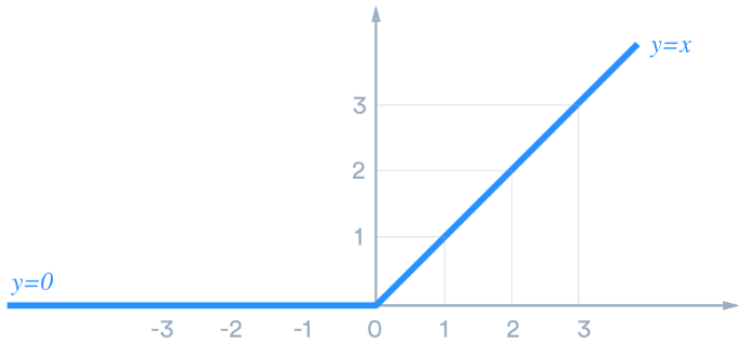


Figure 3.6: Rectified Linear Unit (ReLU) function. $x < 0 : y = 0, x > 0 : y = x$ [50].

-1 and 1 or 0 and 1 (depending on the function), which enables the network to make for instance a binary decision [49], as required in this work. However, this classification can not always happen linearly. Images, for example, are most of the time very non-linear [48]. An example can be seen in Figure 3.5, where the two colours of data points can not just be separated by a straight line. Hence a non-linear function as in Figure 3.5b has to be added and is usually located directly behind a convolutional layer [46] in order to make non-linear decisions and learn faster [48]. The most common non-linear activation functions are the sigmoid, hyperbolic tangent (tanh) and the Rectified Linear Unit (ReLU) function [46, 48]. The mathematical description of the ReLU function is: $y = max(x, 0)$. As seen in Figure 3.6, the function can only get activated with positive input values. It remains zero with negative input. Hence, only parts of the input get processed, which also decreases processing time and costs. Another advantage of ReLU function is the constantly increasing slope, which implies no saturation with large input values. This avoids the vanishing gradient [50]. The negative aspect of this function would be the so-called "dying ReLU". Once the function has reached 0 due to only negative inputs, it is very unlikely to be activated again. A solution for this would be

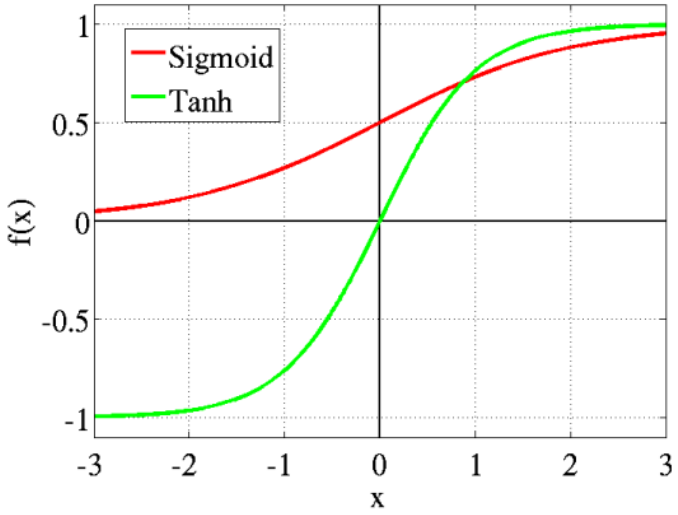


Figure 3.7: Red curve: sigmoid activation function, green curve: tanh activation function [49].

the Leaky ReLU. Here, a very small slope of $0.01x$ is added for negative inputs. This compensates for the "dying ReLU" and calculates even faster due to more balanced results [50]. The sigmoid and tanh functions are plotted in Figure 3.7. Sigmoid (red graph) is mathematically described as seen in Equation 3.2, where ρ defines the slope, which is constantly positive [51]:

$$S(x) = \frac{1}{1 + e^{-\rho x}} \tag{3.2}$$

Due to values between 0 and 1, this function is often used for the prediction of probabilities [49]. The green function in Figure 3.7 is the tanh function. It has the same shape as the sigmoid activation, but the output values reach from -1 to 1. Mathematically it is described in Equation 3.3 [49]:

$$S(x) = \frac{2}{1 + e^{-2x}} - 1 \tag{3.3}$$

At this point as well a so-called dropout rate can be added to a convolutional layer, which is another regularisation method of a CNN. The rate implies in which frequency some random input variables are set to 0 during each training step. With this, the weights have to be adapted to a smaller number of features which improves the updating of the weights and minimises overfitting [52].

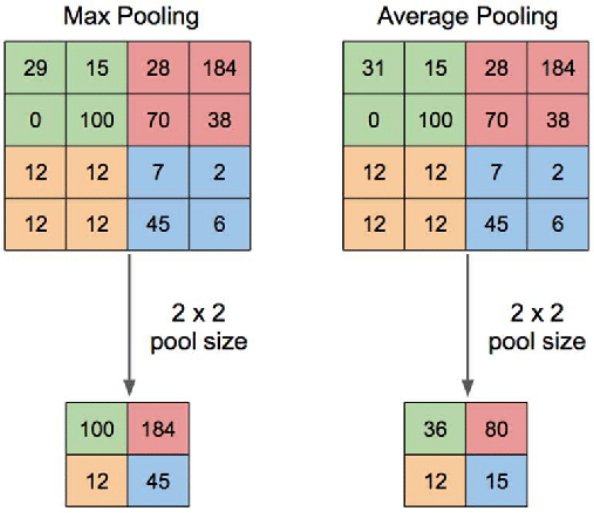


Figure 3.8: Left: max pooling: maximum value out of the four green cells of the large matrix is mapped in the green cell of the small matrix and so on. Right: average pooling: average value of all green cells of the large matrix is mapped in the green cell of the small matrix and so on. [54].

3.1.2 Pooling Layers

Pooling layers represent the second part of dimensionality reduction of the input matrix [35] and filter out the most informative features [53]. This significantly lowers GPU memory. These layers also provide the CNN with the ability to detect an object, even if the object's location differs on the image [47]. The principle of pooling is shown in Figure 3.8. In this example, the pooling size is defined as 2x2. Hence, pixels of this size are summed up, using respective functions [46, 54]. The most common ones are max pooling and average pooling. As seen in Figure 3.8, Max pooling always takes the highest value, whereas the output of average pooling is the average of the values covered by the pooling area size [46]. In addition to the produced noise reduction, however, pooling can lead to loss of background or foreground texture information [55].

3.1.3 Fully-Connected Layers

Fully-connected layers or dense layers are the last layers added to a CNN. Since the output of the pooling layer is a matrix, it has to be flattened into a vector to serve as input for the dense layer. With the help of these layers, the neural network is able to learn non-linear correlations of features [35]. As the name implies, fully-connected layers connect all input and output nodes [46] and classify the images into classes [53]. To end up finally with a probability distribution as output, a softmax function $\sigma_i(z)$ is used, which transforms the input to values between 0 and 1 (see Equation 3.4). Summed up, all outputs result in 1. The lower term converts all values between 0 and 1 [56].

$$\sigma_i(z) = \frac{e^{z_i}}{\sum_{j=1}^{n} e^{z_j}} \tag{3.4}$$

where the input vector is represented by z and its values by z_i [56].

3.2 Transfer Learning

Transfer learning is a method to make use of the knowledge of a pre-trained network. The knowledge of a source model is used for the new model, the target model (Figure 3.9) to save time and minimise resources. In this work I will also take advantage of this method, using the information of a pre-trained model by IBLab to train my desired target model. Without this method, which means training a model from scratch, the initial parameters would be set randomly. To tune these, the network would have to perform several steps of forward and backward propagation. These steps can be minimised by using transfer learning. Instead of random initial weights, the weights of the pre-trained network are used here [53]. A simple approach to the workflow to design new CNN models can be seen in Figure 3.10.

Since image detection is mainly based on recognising simple structures, shapes and edges, the images, which were used for the training of the source model, do not have to relate strongly with the new dataset [53, 57]. The pre-trained model was usually trained with a large number of images resulting in feature vectors [53]. These vectors can subsequently be used as a basis for the new model in the form of initial weights. The architecture of the new model is based on the architecture of the pre-trained model and one or several of the last dense layers are replaced by new fully-connected layers [58]. Regardless of the pre-trained model, the number of output nodes is set in the last added layer [59]. The new model uses the pre-trained features to predict the new images [59]. The training with the new images can then be started. In order to keep the initial weights from the pre-trained model, these old layers are frozen while the newly added layers are trained on the new images. To achieve high performance and reduce overfitting, the new model is trained with a low learning rate. The fine-tuning happens by also setting parts of the source model as trainable and training these with a low learning rate [57, 58, 59]. Another advantage of transfer learning besides reduced training time is a smaller required amount of data to train a neural network, compared to training from scratch [59].

3.3 Residual Network (ResNet50)

For the implementation of my model, I will use the pre-trained model, which architecture is based on the . Therefore, I will give a short overview of the model's architecture and its properties.

The ResNet is a residual network consisting of fully-connected layers, of convolutional and max pooling layers with a total number of 50 layers [61]. In Figure 3.12 the whole

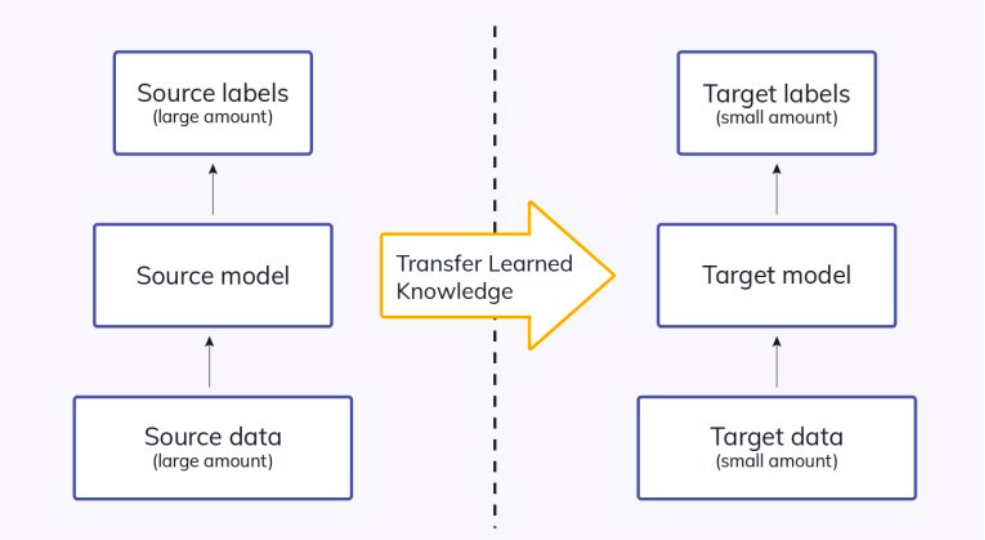


Figure 3.9: The idea of transfer learning. The source model on the left side represents the pre-trained model. The target model on the right side represents the new model. [59]

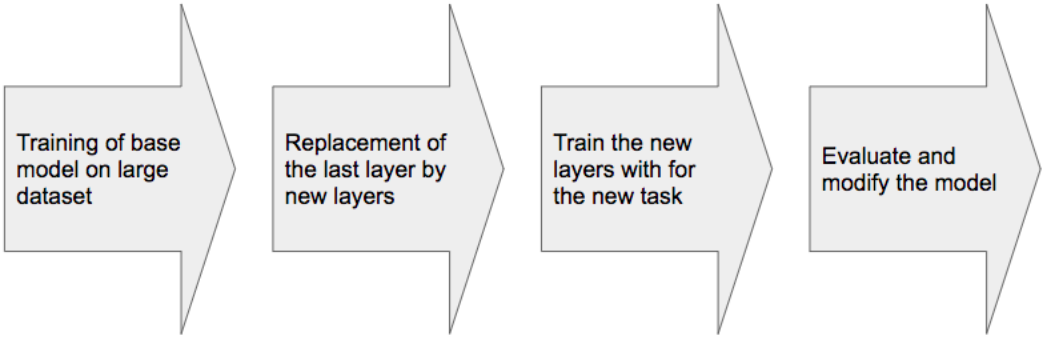


Figure 3.10: Description of the workflow for the creation of a new CNN models using transfer learning.

architecture of a ResNet including only 34 layers is described [60]. A general problem with very deep networks is their complexity. The deeper a network, the more complex training gets [62]. Due to the high number of layers, the backpropagation gradient disperses, which leads to a high training error. A solution for this would be residual networks [62]. Residual learning blocks, which are the characteristic parts of the ResNet, include skip connections, which enable the network to skip one or more layers (Figure 3.11) [60, 61]. These blocks consist of two ways to proceed. One is the mapping $F(x)$ of the residual function and the other is the unchanged mapping x of the input, also called identity mapping [60, 62]. $F(x)$ and x are summed up and passed to a ReLU function [62]. These blocks are then connected in series (as seen in Figure 3.12), where the input of the first layer in a convolutional block is connected to the output of the last convolutional layer [60, 63]. The solid lines in Figure 3.12 represent the same dimensionality of the in- and

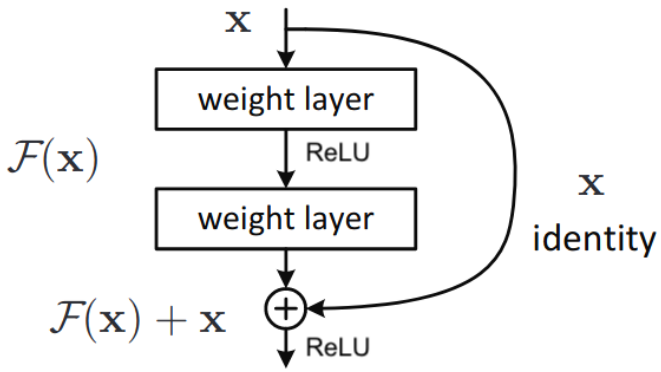


Figure 3.11: Structure of a scip connection with ReLU. x: input, $F(x)$: output of weight layer [60].

output. This corresponds to Equation 3.5.

$$y = F(x, \{W_i\}) + x \tag{3.5}$$

Dotted lines imply an increase of dimensionality, in which case Equation 5.2 is relevant. Here, dimensions are adapted for the identity mapping with W_s [60].

$$y = F(x, \{W_i\}) + W_s x \tag{3.6}$$

3.4 Extreme Gradient Boosting Machine Model

I will also test the ability of the numeric (demographic) data alone to predict AKOA. For this purpose, I will use an XGBoost Machine model. The XGBoost model is built on the gradient boosting framework and can be utilised for classification tasks [64]. Generally, gradient boosting performs with high accuracy, while showing high computational speed [65]. My task of classification was e.g., performed in less than 10 minutes. This is because boosting models show little complexity, which also prevents overfitting [64]. Overfitting means that a model would consider variances of the residuals to build the function of the model. This can be seen in Figure 3.13, where the green line represents the overfitted model, whereas the black graph corresponds to a well fitted model [66].

The structure of the XGBoost model starts with training one single model, based on a decision tree. Depending on the performance of the classification, these weights are adapted and used for a new decision tree, which is added afterwards [64, 65]. This proceeds until the performance of the model stagnates [65]. Finally, these decision trees are merged, resulting in an even more accurate model [64]. The special form of gradient boosting models, the XGBoost model, proceeds the same way but exhibits a loss function based on the Taylor expansion [67]. Due to the normalisation of this function, the

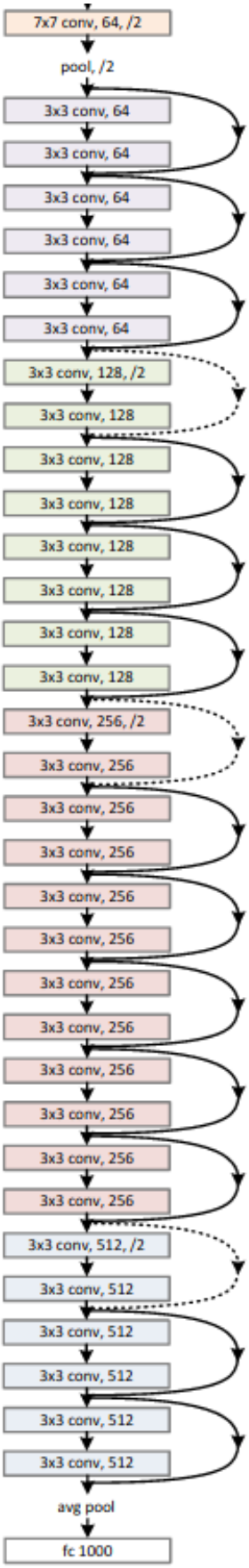


Figure 3.12: Architecture of a ResNet50 [60].

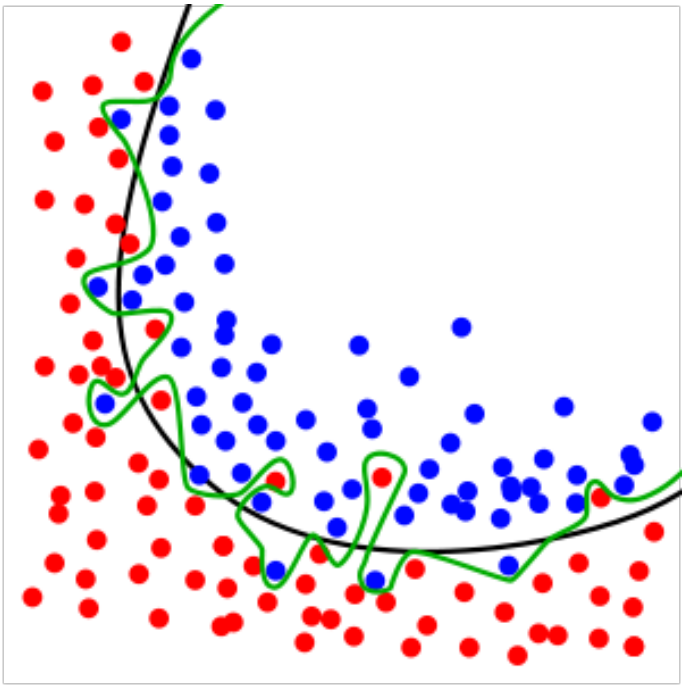


Figure 3.13: Example of a statistical model function. The black line corresponds to a normalised model function. The green line represents an overfitted model, which includes all variances of the residuals to build the function of the model [66].

variances of these models can be reduced [64]. Another advantage of this model is the ability to learn how to handle missing data best [67, 68]. Concerning my application case, some patients do not provide information for all factors I include. Hence, the handling of missing data is an important feature here.

3.5 Area Under the Receiver Operating Characteristic curve (AUC - ROC)

The ROC can be used as a method, which I also use in this work, to evaluate the performance of a classification model. The predictions of a classification model are compared to the ground truth and assessed for their correctness. It is considered a true positive (TP) prediction, if the predicted class from the model matches the true label, the ground truth. The TP rate, also named sensitivity, is then the chance of the model to predict a true input as true. It is considered a false positive (FP) classification if the model predicts a false input as true. The FP rate, 100 minus the specificity, is the chance of the model to make an FP classification. Specificity is defined as the true negative rate, which is the chance of the model to classify a false input as false. The ROC curve describes the TP rate plotted over the FP rate (the blue curve in Figure 3.14). The area underneath (AUC) can be calculated and then used as the evaluation value of a

classification model. The closer the blue curve passes the upper left corner, the higher the AUC and the better the performance of the model. The sensitivity and specificity in the upper left corner would be 1 (100%). The dashed black line represents random guessing with an AUC of 0.5.

Depending on the classification thresholds of the model, a single point on the ROC curve refers to a specific sensitivity and specificity [69, 70]. These can either be taken from the diagram or can be calculated with the number of TP, FP, true negative (TN) and false negative (FN) observations as seen in Figure 3.7 and 3.8. When setting a classification threshold close to 1, which means that the model classifies an input only with high certainty as true, the sensitivity is high whereas the specificity decreases. Depending on the threshold which output between 0 and 1 corresponds to the classes, the sensitivity and specificity can be adapted depending on the desired usage of the model. This evaluation method can be applied for the classification of KOA patients. When having a closer look at patients, which are classified as fast progressors, the more important task is to detect as many fast progressors as possible. Therefore, the classification threshold can be set close to 1.

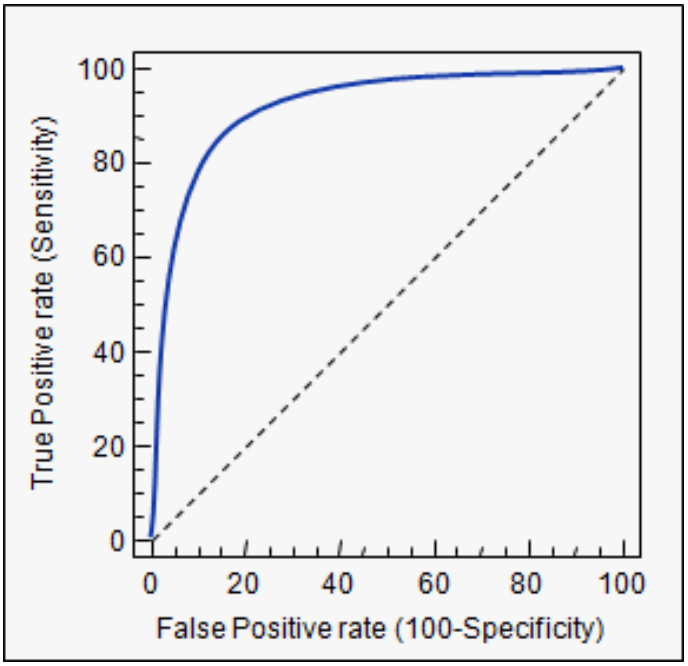


Figure 3.14: Example of the Receiver Operating Characteristics (ROC) curve. The true positive-rate is plotted against the false positive-rate [70].

$$TP\ rate = \frac{TP}{TP + FN} \tag{3.7}$$

$$TN\ rate = \frac{TN}{TN + FP} \tag{3.8}$$

where TP is the number of true positive classifications, TN is the number of true negative, FP is the number of false positives and FN is the number of false negative classifications the model makes.

CHAPTER 4

Related Work

In this chapter, I delineate previous work concerning the prediction of AKOA using deep learning. Due to the high incidence of KOA, a lot of studies have been conducted regarding the prediction of disease progression. I present several previous studies, where statistical and machine learning models were used to find correlations between image and numerical information and the progression of KOA.

4.1 Feature Extraction for AKOA Prediction

There are several studies dealing with the correlation of risk factors and fast-progressing KOA. Due to the different ways to define AKOA, the studies can not exactly be compared. In most papers accelerated AKOA is defined by the change of KL-grade, as explained in Section 2.2. The correlation of several risk factors and single characteristics to AKOA were examined.

4.1.1 Demographic and Clinical Criteria

Several studies described with statistical regression and classification models the higher risk of developing AKOA for patients aged older than around 63 years and younger people with obesity [6, 13, 23]. AKOA was defined by the change of KL-grade [6, 23] and by the percentage of cartilage loss [13], as explained in Section 2.2. The opinions concerning the WOMAC score, which describes pain, physical function, and stiffness, differ between several studies. In the study [23] difficulties when lying down, pain during straightening the leg and during walking came frequently apparent among AKOA patients. Widera et al. also confirmed a certain correlation between the WOMAC score and AKOA [71]. No significant correlation between the WOMAC score and AKOA could be found in the study of Raynault et al., except a slight trend of a higher WOMAC score at baseline for fast progressors [13]. Neither is the score significantly connected to the cartilage volume,

which is a symptom of AKOA [29]. In the study of Raynauld et al. they suggest, next to the previously named criteria, also a higher risk of developing AKOA among females [29].

4.1.2 Radiographic Structures

Besides, radiographic structures also play an important role in predicting AKOA. The goal of the study from Harkey et al. was to find radiographic structures and characteristics even before the onset of AKOA, to be able to classify between non-progressors and fast progressors. OAI data at baseline visit and two years later was evaluated, where AKOA was also defined by the change of KL-grade. Different logistic regression models approved the following symptoms to be predictors for AKOA: degenerative cruciate ligaments, meniscal pathology on lateral and medial side, effusion-synovitis volume and infrapatellar fat pad signal intensity alteration [7]. 73% of fast progressors versus 19% of slow progressors (defined by cartilage volume loss) showed meniscus extrusion and tear [13].

More, a greater tibial slope can be associated with fast-progressing KOA, whereas slow progressors did not correlate to this. Experiments also showed the relation between malalignments, which imply a greater tibial slope, and progression of KOA [15]. These results go along with the study of Driban et al., where previous knee injuries are linked to AKOA [24].

4.2 AKOA Prediction Models Using Numeric Data

Using now the evaluated risk factors and characteristics, prediction models based on demographic, clinical, and radiographic data were developed. The performance of these models is described with the AUC, which is detailed in Section 5.2.1. The highest score of AUC would be 1 and random guessing would be 0.5 [5].

Jamshidi et al. built different feature selection models to predict cartilage volume loss, KL-grade, and JSN using data from about 4800 patients of the OAI study [5]. All these factors are related to the progression of KOA. Subsequently, they used these selected features as input for different binary classifiers. The Gradient Boosting Machine (GBM) achieved the highest AUC of 0.7 predicting cartilage volume loss. The best performance predicting JSN of an AUC of 0.95 was by using a Multi-Layer Perceptron [5]. Halilaj et al. [22] obtained good results to predict JSN. They used knee symptoms, intake of medication, general, nutritional, and mental health, information about walking and upper length muscle strength, X-ray analysis, and malalignments of the knee as features for prediction models. In addition to the baseline information, data from the following year was also taken into consideration here. They classified three groups of JSN with an AUC of 0.86. In this study, they also took data from the OAI study [22].

4.3 AKOA Prediction Models Using Numeric Data and Image Data

As proven in several studies [13, 72] many radiographic characteristics are linked to AKOA. Hence chances are high that a CNN is able to use these specific features or find even new characteristics on X-ray images to predict a fast progressive KOA. On the left side of Figure 4.1 a slow-progressing KOA X-ray and on the right side a fast-progressing KOA X-ray image can be seen. The upper images were taken at the baseline visit. The lower ones 48 months later. Since it is very difficult to differentiate between slow- and fast-progressing KOA by looking with the naked eye at the baseline images, a CNN can be used for this. Previous studies already obtained promising results.

In the study [11], Tiulpin et al. defined AKOA according to the KL-grade. To automate this process they use a residual network for the determination of the KL-grade and the OARSI-grade. They achieved an AUC of 0.98 using the OAI data for training and the MOST dataset for testing [27]. KOA progression was then predicted [11]. In contrast to the approach in this work, Tiulpin et al. separated KOA depending on the KL-grade change into three groups: no progressors, who experience no KL-grade change, slow progressors with a KL-grade change after 5 years, and fast progressors with any KL-grade change within 5 years. They did not consider patients with an KL-grade from 0 to 1 as fast progressors. To end up with a binary classification they merged the group of no and slow progressors into one single class.

Data was taken from the MOST (about 4000 knees) and OAI (about 5000 knees) study. As numeric clinical information, they selected the age, sex, BMI, surgical and injury history of the knee, KL-grade defined by a radiologist and the WOMAC total score into account. As in my work, the numeric and image data was trained separately and subsequently in combination. Using a GBM the training of the clinical data achieved an AUC of 0.76. To train exclusively with the image data, they implemented a CNN. Here the image served as the input for a CNN with two classification branches: prediction of KOA progression and classification of the KL-grade. As in my thesis, they used a pre-trained backbone (se-resnext50-32 xd) as a feature extractor. Training this CNN with the baseline image data achieved the same performance of an AUC of 0.76 as for the GBM. In order to include the numeric and the image data, the two branched output of the CNN is added in addition to the numeric data as input in the GBM model. The structure of the model can be seen in Figure 4.4. This model was able to classify with an AUC of 0.8 between non-progressors and progressors [11].

With a similar deep learning approach, the model of Guan et al. to predict KOA progression, performed with an AUC of 0.863 even better. The structure can be seen in Figure 4.5. Here, they defined progressing patients with at least 0.7 *mm* medial JSN per two years, which is, compared to [11], more similar to the definition I am using in this thesis. The first part of the model is divided into two separate CNNs. One network to extract the Region of Interest (ROI) of the input X-ray image and the other one to extract its relevant features. They merged these outputs with the extracted features of

Figure 4.1: Examples from OAI study of slow progressor (left) and fast progressor (right) [73]

the conventional clinical risk factors (such as age, BMI or tibiofemoral angle) in a new vector, which then represented the input of a fully-connected network. Data was used from the OAI study, including about 4500 unilateral knee images. [32]. These results show the high load of information an image contains to predict the progression of KOA.

The difference between both studies [11, 32], compared to my approach, are the classes between which the model is classifying. The previously presented results from Tiulpin et al. and Guan et al. differentiated between the two classes of non-progressors and progressors, whereas I classify only progressors between fast- and slow-progressing patients. The classes are clarified in Figure 4.2 and 4.3. Even if the approach I am taking is more

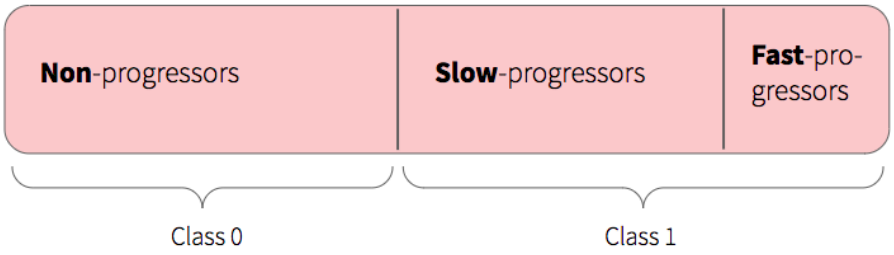


Figure 4.2: Definition of class 0 and class 1 according to the study of Tiulpin et al. [11].

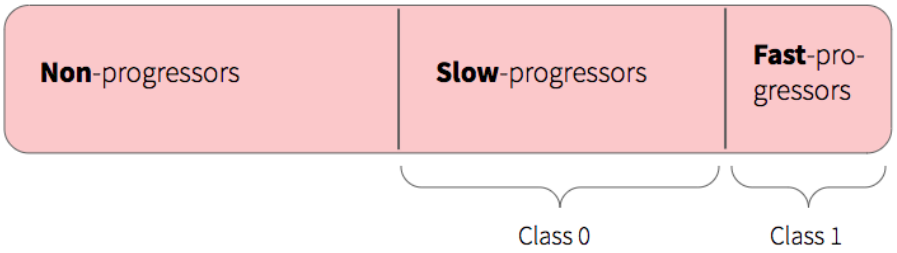


Figure 4.3: Definition of class 0 and class 1 according to this work.

challenging, it is more relevant for the situation radiologists are confronted with. The question if a person is a fast-progressing KOA patient comes up more often if they have symptoms and are already diagnosed with KOA.

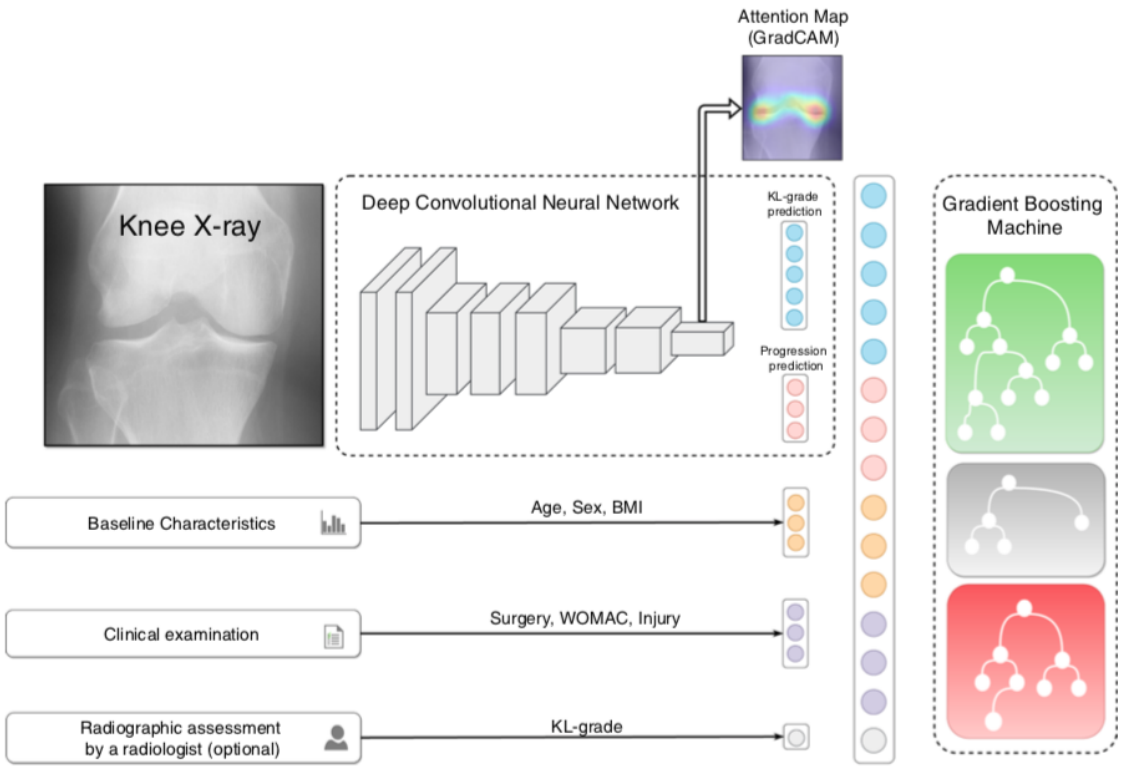


Figure 4.4: Structure of the classifier of Tiulpin et al. [11]. X-ray image as input for the deep CNN. Baseline characteristics, clinical examination and radiographic assessment and output of the CNN used as input for the GBM to classify into fast, slow and no progressors.

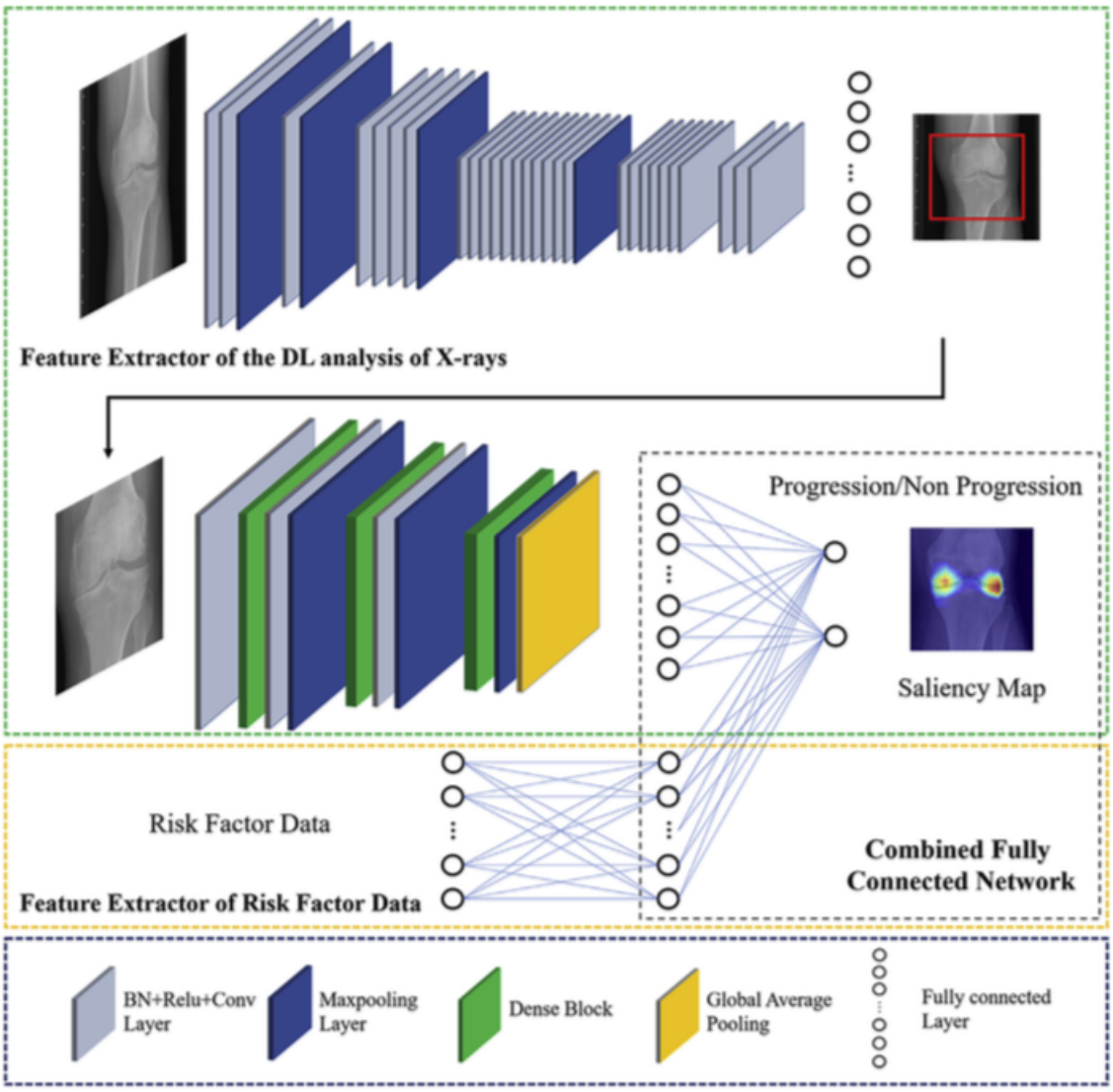


Figure 4.5: Model structure of Guan et al. [32]. The green dashed square borders the two CNNs, which extract the Region of Interest (ROI) and the most important features of the knee X-ray. This output is combined with the numeric data vector in a Combined Fully Connected Network to predict progression of JSN.

CHAPTER 5

Materials and Methods

This chapter deals with the preparation work and the procedure of developing and testing different models to predict AKOA among progressing KOA patients. First, I describe the data and its origin in detail. After the definition of both classes the models shall predict, I outline the pre-processing of the entire data, including numeric and image data. A description of the subcohort follows. Subsequently, I explain the implementation, training and validation procedure of the XGBoost model and the CNN models.

5.1 Data

The final model in this work should be able to classify between slow- and fast-progressing KOA on images, coming from different hospitals and X-ray machines. If a neural network is trained on image data derived from one single dataset (same technical advice, same specific perspective used), characteristics contained only in this dataset can result in misleading correlations to the classes [74]. Therefore, it is important to use different datasets to increase the robustness of the model. IBLab provided me with three different datasets. Since several studies have already used this data [11, 15, 23, 24], the results of this work can be compared reasonably with other studies. In the following, I will expose the three datasets and the subgroup, which I defined for this thesis. To finalise the compilation of the data, I will label the data with its ground truth and will subsequently normalise the numeric data and pre-process the image data.

5.1.1 Data Description

Since the data is the basis of training a neural network, the quality and amount of images have a big influence on the performance of the model. The sum of the three different data sets provides me with a variety of patients showing different baseline symptoms of KOA. The data contains image data and a high amount of clinical and demographic

information about the patients from several years. All studies are population studies and no cohort studies, which means people without any KOA symptoms are also included.

Data from Osteoarthritis Initiative (OAI)

OAI is a public-private initiative funded by the National Institute of Ageing [75]. Their goal is to build a source of image and non-image data to enhance research in the progression, diagnosis, and treatment of osteoarthritis [75]. OAI contains data of around 4800 women and men, aged between 45 – 75. Participants are included, if they suffer from frequent knee pain and/or knee stiffness or show other characteristic symptoms leading to an increased risk of developing KOA. This results in two groups of patients: one group diagnosed with KOA, which will be likely to develop a progressing KOA and the other group, the incident cohort, which exhibits risk factors for the initiation of KOA [73]. All patients were followed up for over ten years. Anterior-posterior X-ray images of the knee were taken at baseline (i.e., year 0) and 1, 2, 3, 4, 6, and 8 years after. The questionnaires about demographic, clinical and health status were filled out yearly [75].

Data from the Multicenter Osteoarthritis Study (MOST)

The MOST study was also funded by the National Institute of Ageing. This study should create a source of data for research about modifiable risk factors and for the ways they influence the development of KOA. This dataset includes around 3000 participants, aged between 50 and 79, with either increased risk of developing KOA, due to risk factors, or with the presence of symptomatic or radiographic pre-symptoms. All participants were followed over seven years. After the enrollment, visits were carried out after 15, 30, 60, 72 and 84 months. Every visit included a clinical evaluation about for instance health, physical function and pain, in addition to X-ray imaging [76].

Data from Cohort Hip and Cohort Knee Study (CHECK)

The CHECK study was performed in the Netherlands to evaluate hip and knee OA in its development, progression and mechanisms. The data of around 1000 patients was collected over 10 years including X-ray images and questionnaires about medical health, physical history, and biochemical and demographic information. Participants with more severe OA visited yearly for examination. The ones exhibiting fewer symptoms were examined only 2, 5, 8 and 10 years after enrollment. The CHECK Cohort was aged between 45 and 65 years [77]. Compared to OAI, patients here showed fewer radiographic symptoms, but higher pain scores at baseline [78].

The data from these three datasets available to me, also contain incomplete observations. Because of the ability of the XGBoost models to handle missing data, I train with the incomplete datasets. For the CNN models, which are not able to train with missing data, I eliminated these observations.

5.1.2 Numeric Data Selection

As not all clinical information, which is contained in the three studies, is relevant for predicting AKOA, I had to consider the most significant ones. As mentioned in section 4.1 several studies and medical experts suggested the following factors to correlate with AKOA and hence potentially play an important role in the prediction of AKOA. Besides, age, gender and BMI are easily accessible information and hence contain nearly no missing values in the data. The treatment of arthritis in the knee also plays a role in the prediction, since it is connected to reduced knee symptoms. Medical experts from IBLab also considered hip symptoms as being an indicator for painful AKOA, due to a change in posture to avoid pain, which leads to hip symptoms. I also considered the WOMAC score to be interesting to include in my work. Different studies suggest different roles of experienced pain, stiffness and physical functionality in terms of developing AKOA, which I showed in Section 4.1.1. Hence, I want to examine this correlation as well. Since Metcalfe et al. confirm a high chance of developing KOA when showing a contralateral KOA [79], medical experts suggest finding possibly a similar pattern for AKOA. Therefore, I also include the contralateral KL-grade. I also consider the KL-grade of the index knee, as it is part of the definition of AKOA. I excluded the criteria which were proven to have no significant correlation with AKOA, e.g., blood pressure [9] and biomarkers from urine [73]. I have also excluded factors, such as degenerative cruciate ligaments or pathological meniscus [7], which are related to AKOA but for which no data are available in my datasets. In the following, I list all numeric criteria I include in my work.

- Age
- Gender
- BMI
- Knee injection (medication for arthritis)
- Hip symptoms
- WOMAC disability score
- WOMAC stiffness score
- WOMAC pain score
- KL-grade
- Contralateral KOA (KL-grade >= 3 at other knee)

Except for the KL-grade, I took all information out of the OAI, MOST and CHECK dataset. To obtain the KL-grade of the examined and the contralateral knee I use the **IB Lab KOALA™** (Knee Osteoarthritis Labelling Assistant) software from IBLab

[80]. The software can determine the medial and lateral KL-grade on the basis of an X-ray image, calculating the OARSI grade of JSN, sclerosis and osteophytosis [80, 81]. An additional displayed value of the KOALA output is the JSW, where the minimal distance is used [80]. This minimal JSW I further use for calculating the ground truth. The output of KOALA is pictured in Figure 5.1.

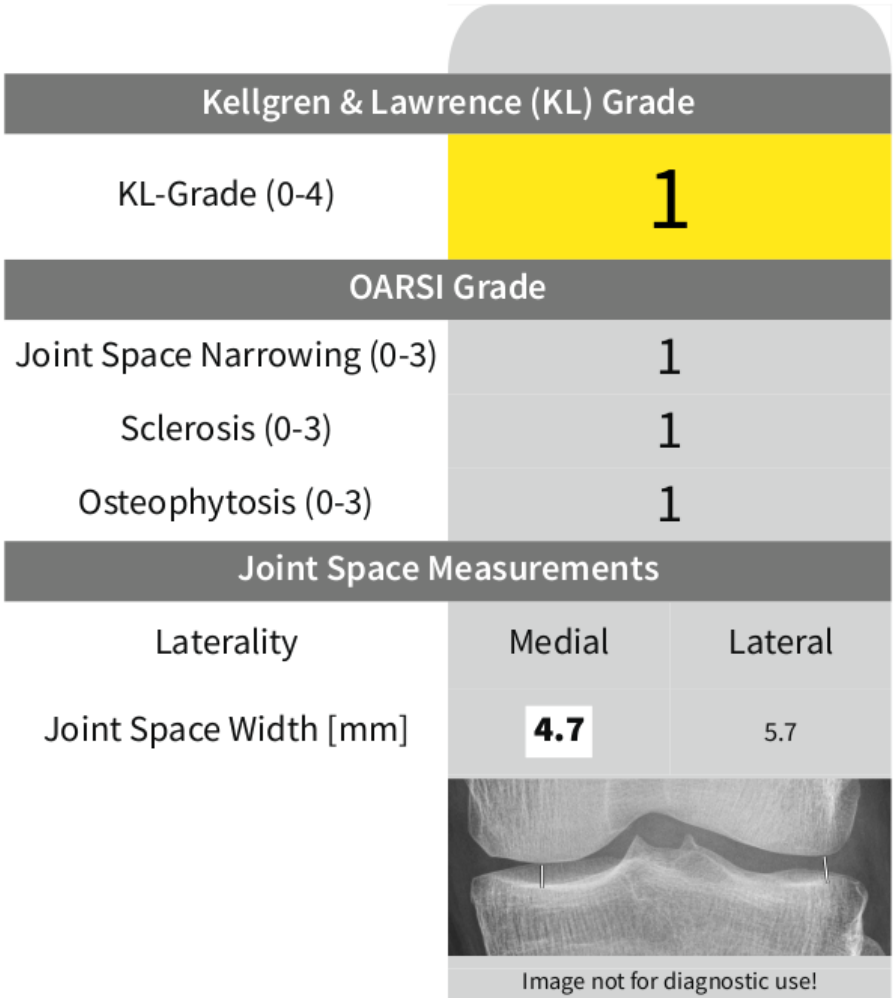


Figure 5.1: Visual output report of the IB Lab KOALA™ software by IBLab. Automated definition of the OARSI grade of JSW, sclerosis and osteophytosis to define the KL-grade. The minimal JSW is calculated for the medial and lateral compartments.

The KL-grade can take the values of 0, 1, 2, 3 and 4. If patients are currently treated with medication for arthritis by injections, the variable knee injection would be 1. If not, the value turns to 0. The criteria of hip symptoms is answered subjectively by the patient with yes (1) or no (0). A detailed description of the WOMAC score about physical function (disability), stiffness and pain can be found in Section 2.1. These variables can

take values of the respective scoring system. The BMI results from the patient's weight divided by the squared height, which results in kg/m^2. For male patients, the variable gender corresponds to 0 and for female patients to 1.

5.1.3 Class Definition and Calculation

The performance of a classification model highly depends on the definition of its classes. The following section treats the two class definitions, which I will use to calculate the true labels of the input data. Annotating images with the true label, the ground truth, creates the foundation of a classification model. Giving the network the image in combination with its true label poses the learning process of a CNN. After the training, the network is ideally able to predict the right label using only the input data. The goal of this work is to predict fast-progressing KOA patients defining AKOA with JSN per year.

The advantage of the class definition using JSN is its simplicity. The JSW of a knee is easy to measure manually or automatically on a simple X-ray image because no bone structure has to be considered. The measurement of the JSW can be seen clearly on the X-ray image at the bottom of Figure 5.1. In contrast to this, the class definition using the KL-grade requires specific medical knowledge or good software to identify the KL-grade. To define the KL-grade of a knee on an X-ray, the OARSI score of sclerosis and osteophytosis has to be established by involving the bone structure. JSW, however, is correlated to cartilage volume loss [30, 33] and the KL-grade and is, due to its easy calculation, a simple and consistent approach defining AKOA.

Referencing previous studies, which used absolute numbers of JSN to define AKOA (see Section 2.2), IBLab defined fast progressors with 10% of JSN per two years [30, 31, 32]. Since using this class definition, the signal-to-noise ratio was too little, I increased the threshold to 20% JSN per two years. By doing this, class 1 will be more "extreme" and the classification can be expected to be clearer and could increase the classification accuracy. I will train all models using both ways to define AKOA. Both definitions are described in detail. The class separation is also illustrated in Figure 4.3.

Class definition of 10% JSN per 2 years

- **Class 0: "Slow progressors":** Patients with less than 10% of joint space width reduction per two years.
- **Class 1: "Fast progressors":** Patients with at least 10% of joint space width reduction per two years.

Further, this ground truth will be abbreviated with "10 % JSN".

Class definition of 20% JSN per 2 years

- **Class 0: "Slow progressors":** Patients with less than 20% of joint space width reduction per two years.

- **Class 1: "Fast progressors":** Patients with at least 20% of joint space width reduction per two years.

From this point on, this will be abbreviated with "20 % JSN".

Calculation

This section deals with the calculation of the ground truth. Since I defined AKOA with a certain threshold of JSN per at least two years, I calculate the JSW reduction in this period of time. These values I use to label each input image with the true label. Hence, I extracted all images, to which an image in the following 1 or 2 years does exist, to calculate the JSN per at least two years.

Because the visits of the CHECK and the OAI study took place every 12 months, the number of visits corresponds exactly to the number of years:

- baseline: V00
- 12 months later: V01
- 24 months later: V02
- ...

Visits of the MOST study differ by intervals of 15 months. Since my class definition is based on total numbers, the visit intervals also have to be total number intervals. In order to make the studies comparable without losing too much data, I adapted the visit numbers of the MOST study and accept the time interval uncertainty of six months for V02 compared to the other studies.

- baseline: V00
- 15 months later: V01
- 30 months later: V02
- 60 months later: V05
- 84 months later: V07

In the following, I define these two images (visit number interval of at least two) as one "image pair". One pair consists of image A and image B, which correspond to year x and year x+1 or x+2, respectively. Because my dataset sources from a longitudinal study over at least seven years, one single patient can provide several "image pairs". The following example should clarify this:

For example, a patient provides X-ray images from baseline (V00), year 1 (V01), year 2 (V02), year 6 (V06) and year 8 (V08). I can now calculate the JSN for images, for which an image at least 2 years later is available. In this case, the JSN can be calculated for image V00 (image A), using image V01 (image B). The same applies to image V01 (image A) using image V02 (image B) and image V06 (image A) using image V08 (image B). These are 3 image pairs resulting in 3 images, for which I can calculate the true JSN per at least 2 years. The procedure is depicted in Figure 5.2.

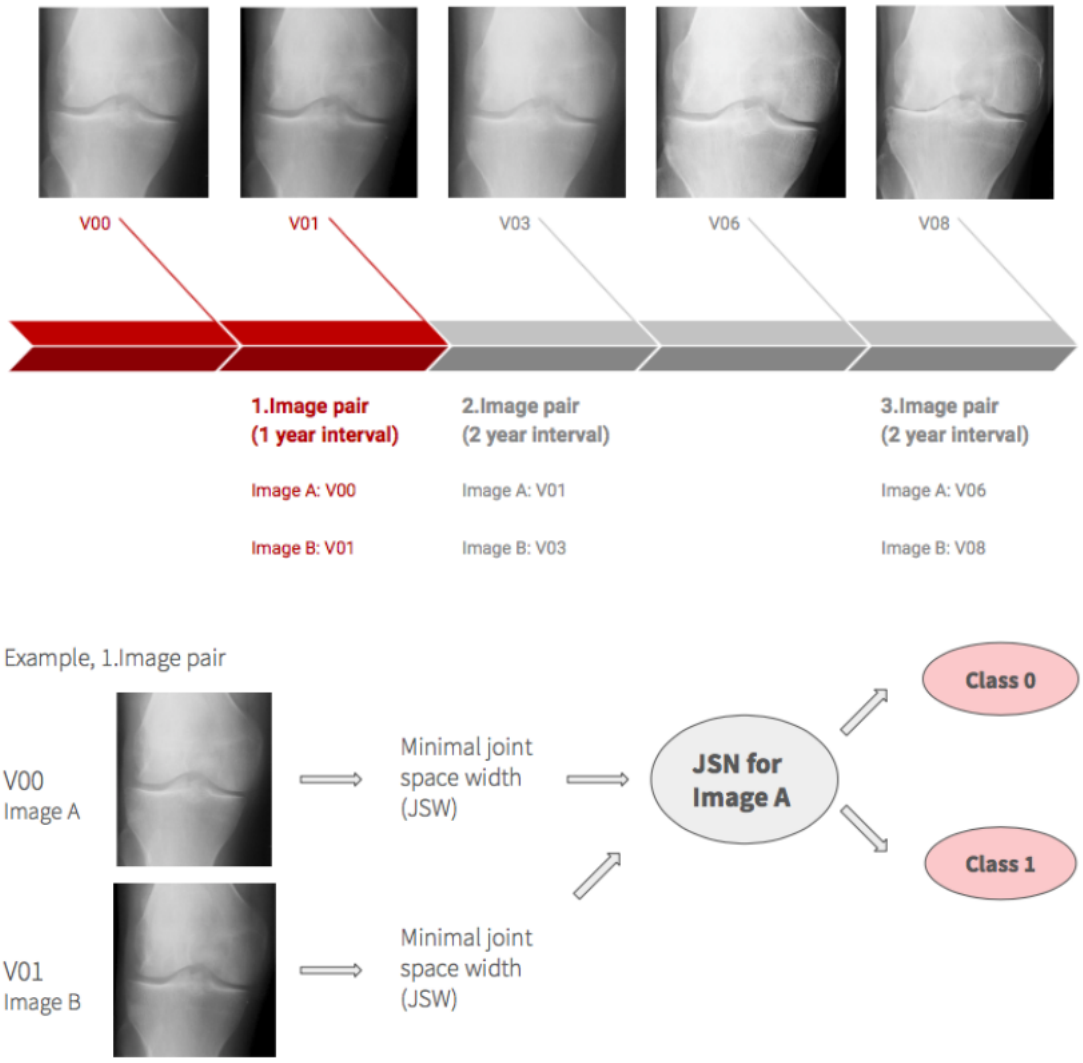


Figure 5.2: Outline of the processing of the image labelling. All X-ray images derive from the same right knee from year 0 to year 8. For the 1st image pair, the X-ray from VISIT 0 refers to image A and from VISIT 1 to image B. These images are used to calculate the JSN and with this the label for Image A. For the 2nd image pair, the X-ray from VISIT 1 refers to image A and from VISIT 3 to image B and so on. All images are sourced from the OAI database.

To now calculate the reduction of the JSW of image A in the following one or two years, the JSW of image A and image B is required. Here I use the minimal JSW, which I calculate automatically with the KOALA software (Section 5.1.2). This provides me with the lateral and medial minimal JSW of image A and image B and hence with the absolute value of JSN for one image pair. I can now use this value for image A to decide if this knee X-ray belongs to a fast- or slow-progressing patient.

In the flowchart, seen in Figure 5.4, I point out the pipeline of classifying images into fast and slow progressors. The chart images the procedure for the medial side. The same procedure applies to the lateral side. As mentioned before, I use the minimal JSW of image A ($JSWA$) and image B ($JSWB$) to calculate the difference of JSW (D) as in the following equation (5.1):

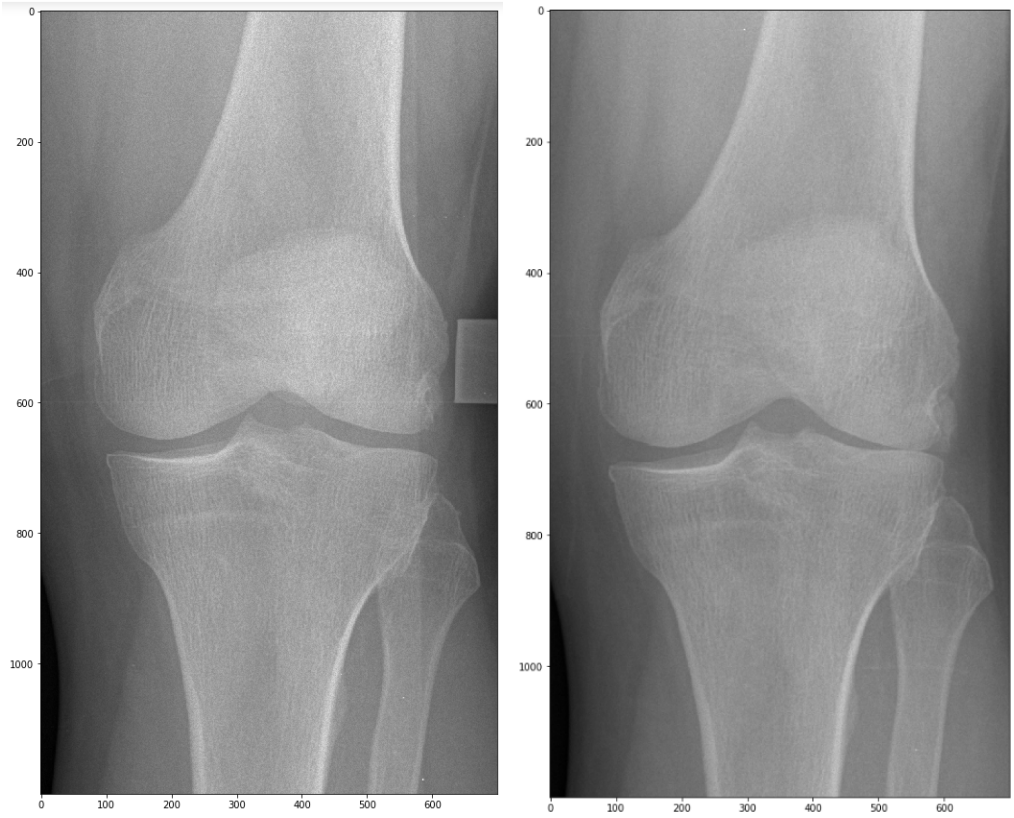
$$D = JSWA - JSWB \tag{5.1}$$

I only consider image pairs with a positive difference D, which means a JSW reduction of image A in the following 1 or 2 years. A negative difference D would imply an increase of JSW. If this applies to the medial and lateral side, I exclude this image. Developers of the KOALA software advised me to choose a margin of 0.4 mm for the measurement uncertainty. I consider a negative difference D of maximal 0.4 mm as possible measurement uncertainty and hence as 0 increase ($D = 0$), which prevents these cases from being excluded. There exist cases, where instead of both only one side of the knee, lateral or medial, shows an increase in JSW. In this case, the increase of JSW implies a skewed knee, as imaged in Figure 5.3. Here the medial side shows an increase in JSW, whereas the lateral side shows a decrease. These image pairs I also include in my class calculations.

To now distinguish between class 0 and class 1, which correspond to slow and fast progressors, respectively, I compare the value of D with the threshold. Once I calculate the ground truth with the threshold of 10% and once with 20% of JSN. If D is below the threshold, I consider this side of the image as class 0. Above the threshold, I specify it as class 1. The classification of the whole knee results from merging the medial and lateral classes. If at least one of both sides is labelled as class 1, I defined the total knee as being a fast progressor. If both sides are class 0 or one side 0 and the other Not a Number (NaN), I consider the total knee to be a slow progressor. Hence a skewed knee with a NaN value on one side will end up with a valid class for the total knee.

5.1.4 Exclusion Criteria

This work will cover two different kinds of exclusion criteria. Since the goal of this thesis is to differentiate between slow and fast progressors I considered only a subcohort of the patients, which show progressing KOA. Therefore, I excluded all image pairs with a remaining KL-grade of 0 for image A and image B, because KL 0 implies no KOA and hence no progressor. I also sort out the image pairs with remaining doubtful KL-grade, which are also not considered to be progressors. Apart from these, I also eliminate the



(a) Visit 0 **(b)** Visit 1

Figure 5.3: Example of a left skewed knee from the OAI dataset. a) Baseline visit knee X-ray with KL-grade 2. b) Knee X-ray of 1 year later with KL-grade 4. Increase of the medial JSW of more than 0.4 *mm* and decrease of the lateral JSW.

patients showing KL 4 at baseline, as being clinically irrelevant [11]. Further, these exclusion criteria I name "Ex014". A summary of the exclusion criteria for this batch of data can be seen in Table 5.1.

For comparison, I also created one batch of data where I also included the non-progressors of KOA patients ("Ex4"). Here I only excluded image pairs exhibiting a KL-grade of 4 on image A, due to clinical irrelevance (see Table 5.2). Due to a higher number of images and a higher image entropy between the classes I expect an increased performance of the models while applying the "Ex4" exclusion.

All class calculations and exclusion criteria are based on the output of the KOALA software. Hence I checked them for correctness. As a basis for the landmarks, which then serve as reference points for calculations, KOALA segments the tibia and femur on the knee X-ray. An incorrect segmentation means consequently wrong JSW measurements. Therefore, I inspected all KOALA outputs manually and eliminated all images with inaccurate segmentations. In Figure 5.5 the segmented compartments of the lateral and

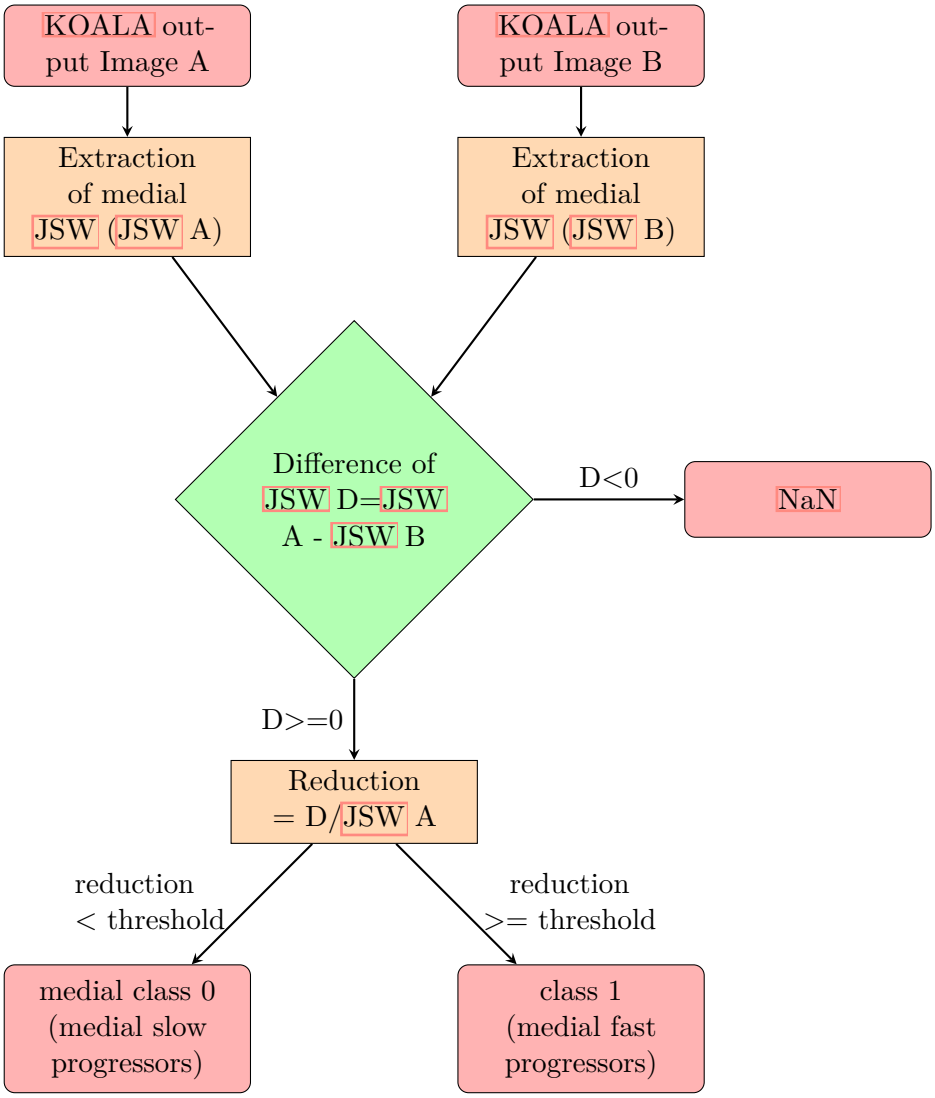


Figure 5.4: Process of calculating the ground truth. This will be repeated for the values 10% and 20% as the threshold. Class 0 corresponds to slow progressors and class 1 to fast progressors. The KOALA output XML contains the lateral and medial minimal JSW of the knee. This flowchart can be transferred exactly to the lateral side.

medial part of the left and the right knee can be seen, respectively. I flipped the right medial and the left lateral sections of the knee. An example of an excluded image can be seen in Figure 5.6. I filtered out 139 images this way.

Since patients with knee implants will not show OA anymore, I excluded these images by plotting the whole image without segmentation. The remaining number of images for each batch of data I will present in chapter 6.

Visit x		Visit x+1 or x+2
KL 0	→	KL 0
KL 1	→	KL 1
KL 4	→	KL 4
implant	→	implant

Table 5.1: Round 1: exclusion criteria for slow and fast progressors

Visit x		Visit x+1 or x+2
KL 4	→	KL 4
implant	→	implant

Table 5.2: Round 2: exclusion criteria for no fast progressors and fast progressors

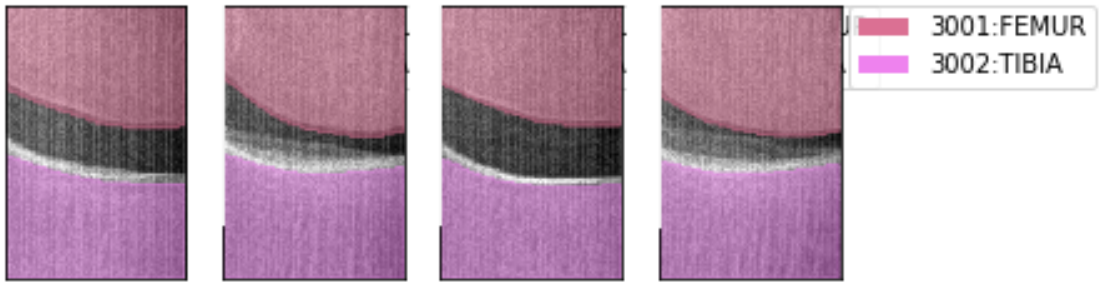


Figure 5.5: Correct segmentation (produced by KOALA) on bilateral knee X-ray image (image from the OAI study).

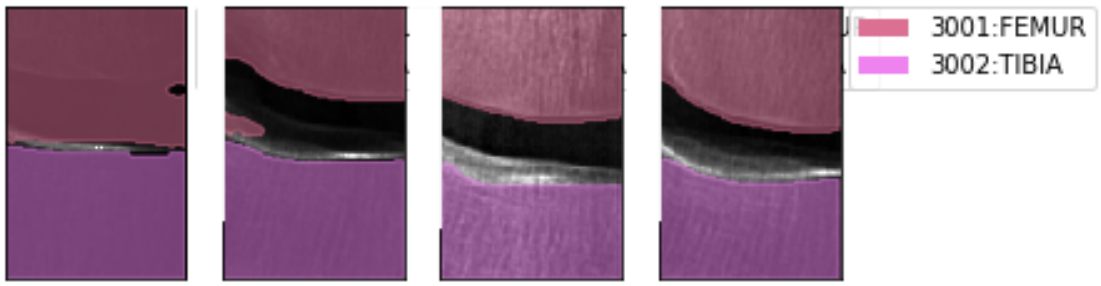


Figure 5.6: Incorrect segmentation (produced by KOALA) on bilateral knee X-ray image (image from the OAI study).

5.1.5 Pre-Processing

This section covers the pre-processing of the image and the numeric data. Pre-processed data will provide a more uniform input and consequently, the performance of the classification model can be improved.

Pre-Processing of the Image Data

Even if MRI images may contain more information since they consist of three-dimensional data, the availability of X-rays among patients with no or little symptoms is higher [13]. Therefore, the input image data in this work will only consist of knee X-ray images. The pre-processing of these images is the golden standard of deep learning. Since the images originate from different X-ray machines from different hospitals, the light intensity, brightness or contrast can vary a lot between the images. To avoid wrong biases due to the origin of the image, all inputs should be standardised.

First, I cropped all images to the same size to diminish unnecessary information and focus on the knee region. A medical expert defined the region of interest for defining AKOA as in Figure 5.7. Using the length X, which corresponds to the tibia plateau (Figure 5.7), the length of this image refers to 1.6 multiplied by X and the height to 2.0 multiplied by X, measured from the centre of all KOALA landmarks, which correspond to the middle of the knee. This cropping length is then applied to all input images. The obtained knee area is expected to contain all information required to predict the progression of disease. Since the difference between left and right knee shall not be considered by the network and hence to avoid a wrong bias, I flipped all left knee images to the right side, resulting in only right knee images, as seen in figure 5.7.

Further pre-processing I performed using the OpenCV2 library (https://opencv.org/releases/). In the next step, I resize all images to the same amount of pixels, since the architecture of my model requires the same size of all input images plus the network learns faster, the smaller the image [82]. As a compromise between reducing training time and not losing information, I resize the images to a width of 512 and a height of 1024 pixels. Here I use cv2.INTER_NEAREST as an interpolation method, where the adjacent pixel is used for interpolation. Taking only one single pixel into account for interpolation makes this method a very fast one [83]. As a next step, I applied normalisation in order to convert the mean of all pixel values to 0 and the standard deviation to 1. This process minimises the amount of non-zero gradient, which decreases the learning time of the model [84, 85]. All X-ray images consist of an amount of different pixel intensity values. This range can be distributed very differently for every image. The distribution of intensity values of the Dicom images, which I used in this work, are plotted in Figure 5.8. The x-axis represents the pixel intensity value and the y-axis represents the number of pixels of all images. To use only a certain range of values, the best range for 16-bit images has to be figured out. Therefore, I plotted all means, which can be seen in Figure 5.9. I eliminated the outliers with values higher than 20,000 by setting the lowest pixel intensity values to 0 and the maximum value to 20,000. This narrows the range of pixel values, which then increases the contrast in the range, where most intensity values of all images exist [86].

Thereupon, I applied a blurring method in order to reduce noise in the image, make edges more prominent and reduce the high-frequency noise. According to previous work, this pre-processing improves the performance by differentiating between real edges and

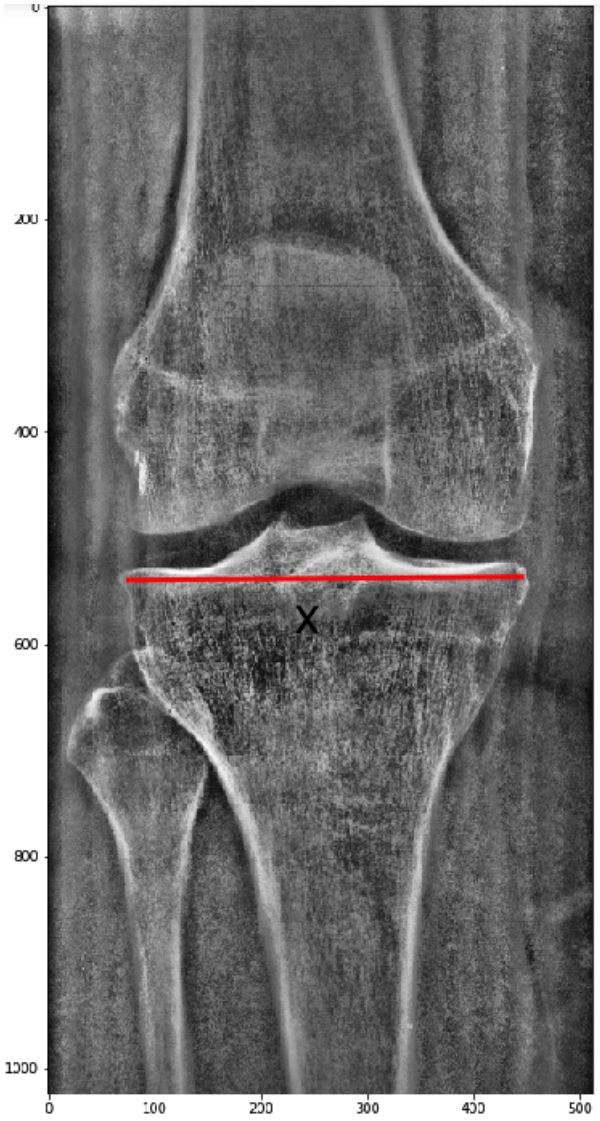


Figure 5.7: Example of a cropped unilateral knee X-ray image (image from the OAI study). X is the length of the tibia plateau.

noise-induced edges [87, 88]. As in other studies, I used Gaussian Blurring as a blurring method [11, 27, 89], which acts as a low-pass filter [90]. This filter calculates for every pixel the weighted average intensity including the surrounding pixels [91]. Contrary to other methods, like average blurring, the central pixel is weighted the highest [87, 92]. Like this, false edges resulting from noise can be eliminated [93]. The number of pixels taken into account is defined by the kernel size. The larger, the more blurred. I defined the kernel size as 3 x 3, which was also used for X-ray pre-processing in the study of Nguyen et al. [89] and achieved the highest performance during image quality measurements [91]. I defined the parameter of the function accounting for the standard

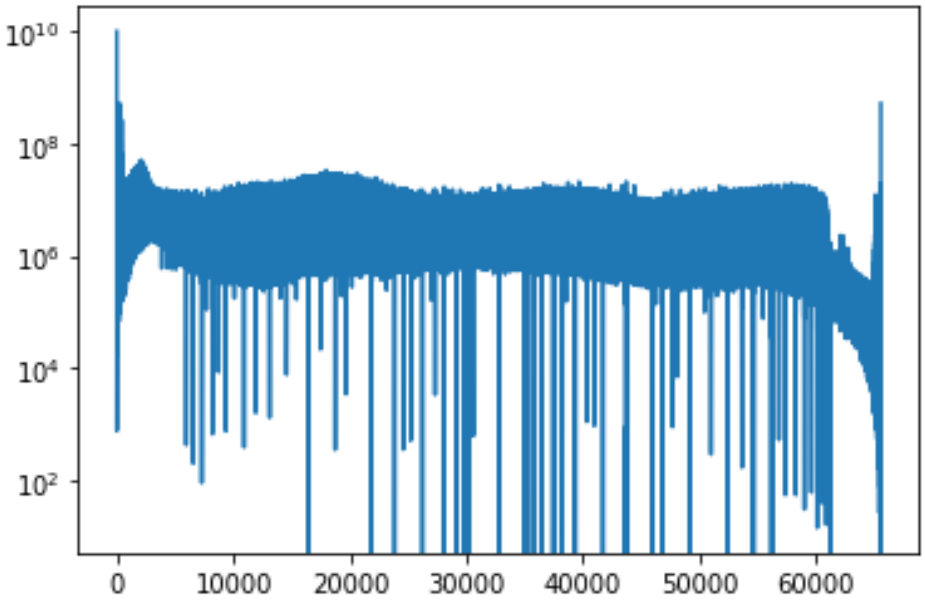


Figure 5.8: Pixel intensity histogram of all images of used data. X-axis: pixel intensity value, y-axis: number of pixels.

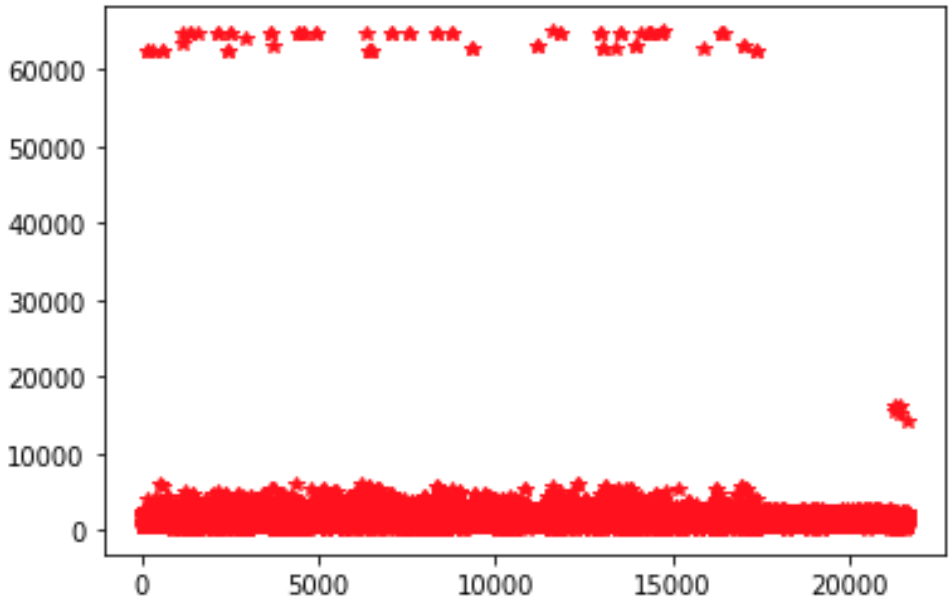


Figure 5.9: X-axis: image number, y-axis: pixel intensity value.

deviation in the direction of X and Y [87] with 0.3.

As the next pre-processing step, I adapted the contrast. Generally, a high contrast image uses the widest range of pixel intensity values possible, which makes the image clearer. Hence, the histogram of a well-contrasted image would be distributed over the whole range of pixel intensity values, as seen on the right histogram of Figure 5.11. The left histogram, which only covers a small range of pixel intensities, refers to an image with a bad contrast [94]. Histogram equalisation can be applied globally to enlarge the distribution of the intensity values over a wider range of the x-axis. A contrast rectified image is depicted in image c) on Figure 5.10. The original image with bad contrast corresponds to image a) on Figure 5.10. The disadvantage of the global histogram equalisation is the declined contrast in the bright regions of the image. This can be optimised by applying Adaptive Histogram Equalisation. Here, the image is split into smaller regions, for each of which the histogram is equalised independently [94]. Regions with a little range of intensity values would amplify noise. This issue can be solved by Contrast Limited Adaptive Histogram Equalisation (CLAHE). Applying this method, an upper threshold can be defined to limit the contrast above a certain value [94, 95, 96]. The improved effect of using CLAHE on X-ray images is also confirmed by Ikhsan et al. [97], who compared CLAHE to global histogram equalisation and gamma correction. The better results can also be seen in Figure 5.10 on image b).

Hence, I selected CLAHE to use as the contrast improving pre-processing method in order to keep the noise as low as possible and contrast the bone structure from the X-ray image, which enhances the learning process of the network [94, 95]. The parameters contrast limit and grid size can be set for this application. I tested both parameters with some values between 1 and 50, checked the result visually and used the best combination. The grid size resulted in 15 x 15, which defines the size of the image section, where the same equalisation is applied. The parameter contrast limit I set to 40, which is the default value of this function. Pixels with higher intensity values are treated as noise [94] and are redistributed to all other intensity values [95]. All pre-processing steps I used for all images are plotted in Figure 5.12. The previously explained steps I applied subsequently on image a). In image d) the application of Gaussian Blurring is very difficult to see. The effect of this step can be seen in image f), where I applied all pre-processing steps the same, but with no blurring. Thus Gaussian Blurring is also important to apply, since the bone structure on image e) can be perceived more clearly compared to image f).

Normalisation of the Numeric Data

On top of the image data, I also add numeric values to the input for the CNN models. This section deals with the normalisation of the numeric data, which means all values of different variables have to be converted to the same scale. This is very important because all variables are measured in different ranges. Normalisation helps to compare for example age scaled between 45 and 80 and the WOMAC pain score, which is scaled from 0 to 20.

To evaluate a difference in the performance of a model using data with different normalisation methods, I tested two different methods. Two very common and simple ones are

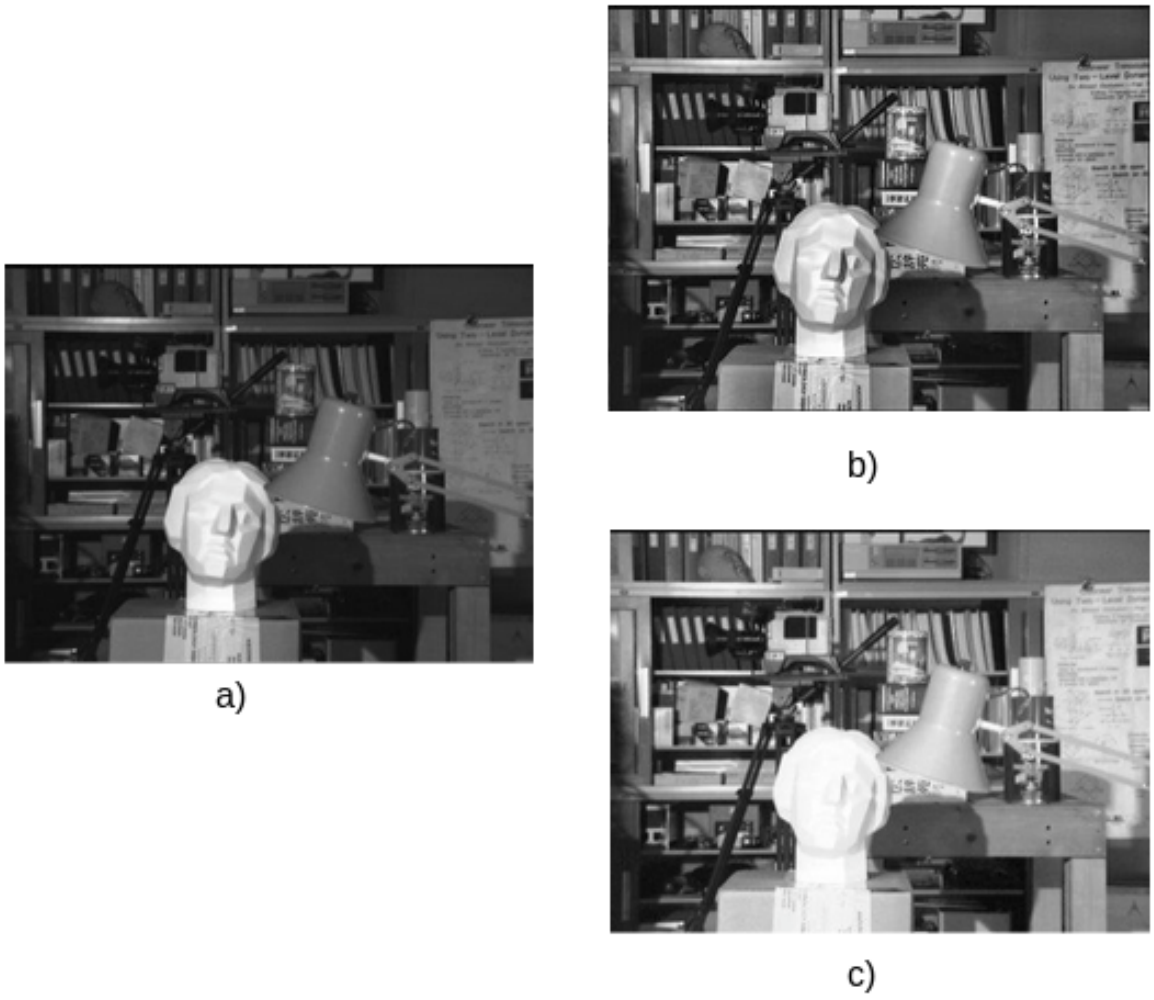


Figure 5.10: a) Original image. b) CLAHE applied image, where all regions can be recognized clearly. c) Global histogram equalisation, where some regions are overexposed. [94]

the Standard Scaler and the MinMax normalisation, which were also used in previous studies [98, 99, 100]. At first, I applied Standard Scaler normalisation, which calculates the standard score as seen in the following equation (Equation 5.2):

$$X_{std} = \frac{X - U}{S} \tag{5.2}$$

where X is the variable value, U the mean and S the standard deviation of all values of this variable [101]. The second method which I applied is the MinMax normalisation, as seen in Equation 5.3. X_{min} and X_{min} comply with the lowest and highest values of this feature. The scaled value is subsequently calculated in Equation 5.4 [102]. The desired

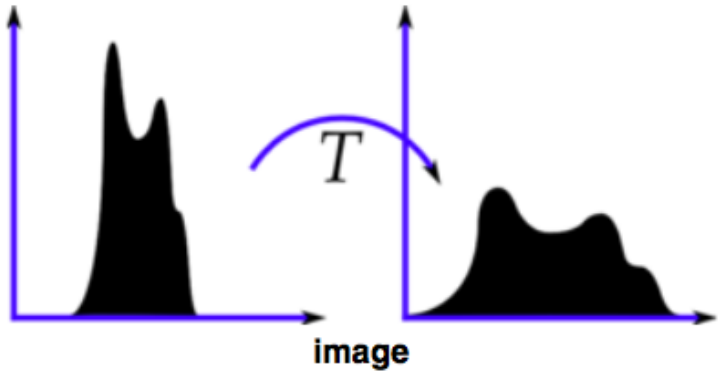


Figure 5.11: Left: example histogram of un pre-processed image. Right: example histogram of the same but histogram equalised image. [94]

new feature range between 0 and 1 corresponds to min and max.

$$X_{std} = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{5.3}$$

$$X_{scaled} = X_{std} * (max - min) + min \tag{5.4}$$

5.1.6 Batches of Data

Having now different class definitions and exclusion criteria, I created different batches of data to use for training my models. These batches I used in combination with the StandardScaler and the MinMax normalisation method for the numeric data (demographic and clinical information). For every batch, the images are preprocessed in the same way as mentioned in Section 5.1.5. The class definition and exclusion criteria I detailed in Section 5.1.3 and 5.1.4, respectively.

- **Batch 1:**
  - Class definition: 10% JSN
  - Exclusion criteria: Ex014
  - Number of images (including missing data): 21,139
  - Number of images (excluding missing data): 17,403
- **Batch 2:**
  - Class definition: 20% JSN
  - Exclusion criteria: Ex014
  - Number of images (including missing data): 27,432

  - Number of images (excluding missing data): 22,024

- **Batch 3:**
  - Class definition: 20% JSN
  - Exclusion criteria: Ex4
  - Number of images (including missing data): 44,538
  - Number of images (excluding missing data): 25,545

5.2 Implementation

This section covers the implementation of all models I created in this thesis to classify between slow- and fast-progressing KOA patients. First I will describe the implementation of the XGBoost model, which I used to test only numeric data to predict AKOA and to evaluate the importance of all features. I will then explain the implementation of the CNN models. I created these models to train with image data only and with the combination of image and numeric data. This section contains the description of all model structures and a detailed explanation of each training run. My implementation can be found in the GitHub repository with the following link: https://github.com/MAVOvo08/Masterarbeit_vogel

5.2.1 Implementation of the XGBoost Model

Model and Hyperparameter Tuning

In this section, I will examine the ability of the numeric data to predict fast-progressing KOA. I considered the XGBoost model as the best for classifying between fast and slow-progressing KOA, due to the high computational speed and the little complexity, which prevents overfitting [64, 65]. Besides, the XGBoost model proved its effectiveness in many machine learning competitions [103].

Hence, I trained an Extreme Gradient Boosting on the basis of the different batches of data and different normalisation methods, as I pointed out in the previous Sections 5.1.6 and 5.1.5. I split the data randomly into a training set and a test set, 75% and 25%, respectively.

The XGBoost model includes a large set of hyperparameters that need to be tuned, to end up with the highest possible performance. The learning rate defines the shrinkage size of the model during updating the weights. The maximum depth complies with the complexity of a tree. The deeper, the more complex the model gets, but also the higher the chance to overfit. The maximum number of leaves defines the highest number of nodes, which can be added to a tree. Gamma is the “minimum loss reduction required to make a further partition on a leaf node of the tree” [104]. The parameters alpha and lambda correspond both to some regularisation of the weights. For gamma, alpha and lambda, the higher the value, the more conservative the model gets [104].

Due to the high number of parameters, it is very difficult to find an optimal combination by testing manually [67]. I tuned the parameters listed in Figure 5.13 according to reports of being effective, referencing an online class of Nvidia [105]. After performing some grid search iterations with the first data batches, I did not further tune the values of the variables, which remained the same, during the subsequent iterations in order to decrease the processing time. All variable values, which I describe in the following, I identified as mentioned before. The subsample, which I set to 1, describes the fracture of data which will be randomly selected to be the input for the trees before expanding them. The optimal fraction of the first iterations was 1, which means all data will be used as input. The “scale_pos_weight" gives the ratio between the positive and negative weights, improving results for unbalanced classes, which I set to 2. Setting the tree method to “gpu_hist” enables the model to spread over several graphic processing units (GPU). Since this model only has access to one single GPU, this parameter does not influence the model’s performance. Due to the fact that the prediction of AKOA is a binary problem, I set the objective of the learning task to “binary: logistic”, which outputs probabilities. The values for the parameters, where I set the same value for all training runs, can be seen in Figure 5.13.

To tune all other parameters, I performed a grid search before every training run to iterate over learning rate, maximal depth, maximal leaves, alpha, gamma and eta in different ranges and to find the optimal combination of parameter values (see table 5.3). Grid search implies iterating over all variables in order to test all possible combinations of parameters for training the XGBoost model. So the outer loop refers to the iteration over the first variable. Inside of this loop, I iterate over the next variable and so on. With each combination, I trained again the XGBoost model and in case the performance of the model is better than that of the previous iteration, the results are saved or rather overwritten. Like this, I extracted the best combination of hyperparameters to train the XGBoost model to classify between slow and fast-progressing KOA. I minimised the processing time of the grid search by iterating only over three to six different values per parameter. In Figure 5.13 a summary of all parameters can be seen.

Variable	Iterating Range
max_depth	{4, 8, 32, 64, 128, 512}
max_leaves	{4, 8, 16}
alpha	{0.05, 0.1, 0.2, 0.3, 0.5, 0.7}
learning rate	{0.1, 0.2, 0.4, 0.6, 0.7}
gamma	{0.2, 0.5, 0.7, 0.9}
eta	{0.1, 0.3, 0.5, 0.6, 0.8}

Table 5.3: Parameter ranges of the grid search to tune the hyperparameter of the XGBoost model to find the best performance.

Training Runs

As seen in Table 5.4 the training of the XGBoost model is structured in blocks of three runs each. I performed the first blocks of training using the Standard Scalar normalised data of Batch 1, Batch 2 and Batch 3 (Run A to C, Run D to F and Run J to L) and removed all missing values. To figure out which normalisation method performs better, I trained batch 2 once with the StandardScalar and once with the MinMax normalised data (Run D to F and Run G to I, respectively). Due to equal results of both methods, I did not further use the MinMax normalised data and only documented the runs with batch 2 (Run G to I). I repeated all runs with the inclusion of missing values (Run M to O, Run P to R and Run S to U).

One training block contains three runs with different combinations of features. In addition to the numeric data, mentioned in Section 5.1.2, I also included the OARSI score of osteophytes and sclerosis, which I obtained by the KOALA software output. Hence, in each first run of a training block, I trained with all available variables (age, BMI, gender, knee injection, hip symptoms, three WOMAC scores, KL-grade, information about contralateral KL-grade, OARSI grade of osteophytes and sclerosis). Since the radiographic variables can already state the power of X-ray images for the classification task, I want to assess their importance. Therefore, I excluded the radiographic variables in the second run of each block. Although KL-grade is also a radiographic feature, this information is more often available than the OARSI grade of sclerosis and osteophytosis. Hence, I included the KL-grade for all training runs. Due to a high number of missing values of the variable knee injection, which would cause a lot of data to be lost for the runs where I excluded missing data, I removed this feature for the third run of one block of training. I did not use an imputation method, due to a high number of missing values for the knee injection variable and the imputed values would only be proxy values.

Thereupon, I calculated the importance of the features using the xgboost library. After the analysis of the less important features, I excluded the four criteria with less influence and performed new runs with every batch of data (Run V to X) with the exclusion of missing values. The hyperparameters were tuned using grid search before each run, as I pointed out in Section 5.2.1. All runs are summarised in Table 5.4. As a measurement of performance, I used the ROC-AUC, as detailed in Section 3.5.

This curve describes the true positive rate plotted over the false positive rate (the blue curve in Figure 3.14). The AUC can be calculated. The dashed black line represents random guessing with an AUC of 0.5. The closer the blue curve passes the upper left corner, the higher the AUC and the better the performance of the model. Depending on the classification thresholds of the model, a single point on the ROC curve is used to define the sensitivity and specificity [69, 70]. These can either be taken from the diagram or can be calculated with the number of TP, FP, TN and FN observations as seen in 3.7 and 3.8. Depending on the threshold which output between 0 and 1 corresponds to the classes, the sensitivity and specificity can be adapted depending on the desired usage of the model.

Run name	Data batch	Normalization method	Exclusion NaN values	Features
Run A	Batch 1	StandardScaler	yes	All
Run B	Batch 1	StandardScaler	yes	no radiographic
Run C	Batch 1	StandardScaler	yes	no radiographic, no knee_inj
Run D	Batch 2	StandardScaler	yes	All
Run E	Batch 2	StandardScaler	yes	no radiographic
Run F	Batch 2	StandardScaler	yes	no radiographic, no knee_inj
Run G	Batch 2	MinMax	yes	All
Run H	Batch 2	MinMax	yes	no radiographic
Run I	Batch 2	MinMax	yes	no radiographic, no knee_inj
Run J	Batch 3	StandardScaler	yes	All
Run K	Batch 3	StandardScaler	yes	no radiographic
Run L	Batch 3	StandardScaler	yes	no radiographic, no knee_inj
Run M	Batch 1	StandardScaler	no	All
Run N	Batch 1	StandardScaler	no	no radiographic
Run O	Batch 1	StandardScaler	no	no radiographic, no knee_inj
Run P	Batch 2	StandardScaler	no	All
Run Q	Batch 2	StandardScaler	no	no radiographic
Run R	Batch 2	StandardScaler	no	no radiographic, no knee_inj
Run S	Batch 3	StandardScaler	no	All
Run T	Batch 3	StandardScaler	no	no radiographic
Run U	Batch 3	StandardScaler	no	no radiographic, no knee_inj
Run V	Batch 1	StandardScaler	yes	no age, hip symptoms, WOMAC_dis, WOMAC_stiff
Run W	Batch 2	StandardScaler	yes	no age, hip symptoms, WOMAC_dis, WOMAC_stiff
Run X	Batch 3	StandardScaler	yes	no age, hip symptoms, WOMAC_dis, WOMAC_stiff

Table 5.4: Summary of all Training Runs of the XGBoost model using numeric data.

5.2.2 Implementation of the CNN

To now accomplish the final goal of my thesis, I implemented different CNN models to be able to include image data as input to predict fast-progressing KOA. To make use of the method of transfer learning, as discussed in Section 3.2, IBLab provided me with a pre-trained model, the ResNetClassifier. This model, trained by IBLab, will be the basis for all implemented CNN models in this work. The classifier encompasses as backbone a ResNet50 model, which is a deep CNN trained on the ImageNet dataset. It consists of 5 convolutional blocks and contains around 23.5 million trainable parameters. These weights were then used for the ResNetClassifier. The training included X-ray images of the wrist, knee, hip, hand, leg, spine and ankle from 10 different datasets to detect the body part and classify between the sagittal and frontal view. On the test set, a view accuracy of 0.9950 and a body part accuracy of 0.9998 were archived.

I selected this model as a base model, due to the general ability of the ResNet50 to classify between X-ray images. As well the large amount of data, which was used for training and the high performance made this model attractive for my application. The last layers of the ResNetClassifier can be seen in Figure 5.14 and a larger model summary is imaged in Appendix C.

Models

In order to receive a high accuracy in predicting fast progressors, I created different classification models and tested them with the different batches of data. My approach for creating the models can be seen in Figure 5.15. After the creation of the first model, I carried out training runs using different data batches and evaluated them subsequently. In the next step, I removed or added the new layers to create a new and better model. The first models could then be used as reference models for the following ones. For all models, I removed the last dense layer, which can be seen in the model summary in Figure 5.14. This layer is the output layer, which solves the final classification. Since I have a different classification task and require a smaller number of output nodes than the pre-trained model, I replace the last layer with different new layers. For the implementation, I used the Keras library of TensorFlow. Keras is an Application Programming Interface (API), which provides building blocks for machine learning problems [106].

I started with a very simple model to observe the possible increase in performance by adding a layer or the change of parameters. Thus, for Model 1 I added one dense layer with a softmax function behind the max-pooling layer and with 2 output nodes. I kept the pooling layer to transform the output shape of the pre-trained model to vectors [107]. One output node of the last dense layer is associated with class 0, the other one with class 1. The softmax function uses a certain threshold to assign the values between 0 and 1 to the right class. This activation function is used when the number of classes is equal to the number of output nodes [108].

For Model 2, I expanded the model with a new dense layer with 256 nodes and ReLU as an activation function. This number of nodes corresponds to the output size of the

	Type	Number of nodes	Activation function	Dropout
Model 1	Dense	2	Softmax function	no
Model 2	Dense	256	ReLU	no
	Dense	1	Sigmoid	no
Model 3	Dense	2048	LeakyReLU	no
	Dense	1024	LeakyReLU	no
	Dense	1	Sigmoid	no
Model 4	Dense	2048	LeakyReLU	0.3
	Dense	1024	LeakyReLU	0.3
	Dense	1	Sigmoid	0.3

Table 5.5: All CNN models are based on the ResNetClassifier. The last layer was dropped and the listed layers are added for the respective model.

pre-trained model. In literature, the ReLU function is considered to be the best activation function [108] for fully-connected layers, whereas the sigmoid function is used for the classification task [109]. I changed the output layer to a layer with only one output node and used sigmoid as an activation function. This activation function is used for binary classification when having one output node. Smaller networks, which come with fewer output nodes, mean less computational resources and hence fewer costs [110]. The use of a single output node is also confirmed in the Keras documentation [107]. When using just one output node, the output value will range between 0 and 1. This just removes one step at the end of the model, which allows an individual decision to be made about the threshold. For example, a threshold of 0.5 would imply that an output of 0.5 corresponds to class 0 and an output of 0.5 to class 1. For all subsequent models, this output layer remains unchanged.

In Model 3, I added three new dense layers and other activation functions. The first dense layer includes 2048 nodes, the second 1024 nodes. Both of these layers used the LeakyReLU activation function. This function is an improved version of the ReLU function and should prevent the neurons from dying as explained in section 3.1.1.

To build Model 4, I added an additional input layer behind the last convolutional layer of the last block of the ResNetClassifier to enable the model to train with image and numeric data. The next layers, which I added, are similar to the ones I added in Model 3. The first dense layer includes 2048 nodes and uses LeakyReLU as an activation function. This layer is concatenated with a dropout layer, which has a rate of 0.3 to minimise overfitting and regularise the model [107, 111]. Since literature suggests a dropout rate between 0.2 and 0.5 [59, 112], I first tested a dropout rate of 0.5 and 0.3. A rate of 0.3 provided a better performance of the model. A dropout of 1 would mean the network would react as without a dropout and all nodes would be used. A rate of 0 would mean this layer would not give any output [113]. Finally, I added a dense layer with 1024 nodes, LeakyReLU and a dropout layer with 0.3. A summary of all models can be found in table 5.5.

Training Runs

One training run is defined as one training using one model, a specific set of hyperparameters and one batch of data. All training runs, which are relevant for this work, I will describe in the following. I performed all runs using the ADAM optimizer, which has already proven its worth in previous studies [11, 27, 40, 41]. Since the AKOA prediction model is a binary classifier, I used the binary-cross-entropy as the loss function [38]. To evaluate the model's performance, I used AUC as a metric, which is a very common metric function for binary classification [5, 11, 22, 27, 32]. I set the training batch size to 4 and applied no augmentation on the images for all runs. Image augmentation usually reduces overfitting by, for example, cropping or rotating images or changing contrast randomly. These methods prevent the CNN from learning noise on the images for their classification task [114]. Training with data augmentation I leave for future work since this was out of the scope of my work.

To check the informational content of a single X-ray image about the progression of disease I used exclusively image data for Run 1 to 5. I applied transfer learning, as explained further in Section 3.2, to make use of the knowledge of the pre-trained ResNet50 model (see Section 5.2.2) for Run 1 to 8 and Run 10. This method also diminishes training time and resources compared to training from scratch [59]. Hence for these runs, I used the architecture and the weights of the base model. For the first run (Run 1) I used Model 1. I trained this model with the first batch of data. After loading the first 188 layers of Model 1 with the weights of the previously trained model, I froze all of them by setting them to "not trainable" in order to keep the initial weights and make use of the extracted features of the pre-trained model [58]. To specify my model on my data, I trained only the newly added dense layer with a learning rate of 1^{-04}. This learning rate is large enough to have a reasonable time of the training process and low enough to avoid unstable training [115]. To examine the effect of fine-tuning, I froze only 153 layers in the next run with Model 1 (Run 2), meaning in addition to the new layer all layers of block 3 were trainable now. As said in the literature [59, 107], I reduced the learning rate to 1^{-05}. Other than that, all other parameters I took over. Run 3 I performed on Model 2. Again, I started with training only the new layers and set all 188 layers of the base model to not-trainable. The newly added layers I trained again with a learning rate of 1^{-04}. The parameters of Run 4 remained the same as for Run 3 but I used Model 3.

For the next training runs, I added the numeric data. Because the CNN is not able to handle missing values, I deleted all rows containing any NaN values from the input data, resulting in 16,941 images. Still using Model 3, I carried out the next run (Run 6) with the original numeric data (not normalised) to be able to see the effect of normalisation. I decreased the learning rate even further to 1^{-06} to examine a change in performance. Due to the small amount of remaining data, after the exclusion of all NaN values, I identified the variable with the most missing values. Since this was the criteria "knee injection", I eliminated this variable from the dataset. I now used the remaining 22,432 images for the following training run (Run 8) on Model 4, with a reduced Dropout of 0.3. For Run 8 I used the numeric data, which I normalised with the standard scalar function

as explained in Section 5.1.5 and froze again the whole base model. The newly added layers I trained with a learning rate of 1^{-06}. For Run 1 to Run 10 as well as for Run 14 and 15, I used the data from OAI, MOST and CHECK and split it randomly into training, tuning and test set. The training set is composed of 80% of the total mixed data, from which 15% correspond to the tuning dataset. For testing, I used 20% of the mixed data.

Since the training data of the pre-trained model also included the data I used for training, I want to rule out that the images were detected as a whole instead of classifying them with the trained knowledge. Hence, I performed the following training runs using the ResNet50 architecture but not the pre-trained weights. Thus, during Run 9 and Run 11 to 15, I set all layers of Model 4 as trainable since I do not use pre-trained weights. For Run 9, the parameters and the data remained the same, but I unfreeze the whole model. Subsequently, for Run 11, 12 and 13 I performed experiments separating my data in the three original datasets. These results show the ability of the network to classify new datasets. Since images, which originate from the same dataset, show several same characteristics, the network could learn the wrong features. For Run 11, the OAI and MOST data composed the training set and the CHECK data the testing set, which corresponds to 2.1% of the total data (abbreviated in Table 5.6 with OMC). Run 12 trained with OAI and CHECK data and tested with the MOST dataset, which covers 21.4% of the total data (OCM). The same for Run 13, where MOST and CHECK data provided the training set and OAI the testing set (MCO), which is contrary to all other runs larger than the training set (76.6%). For all three runs, 15% of the respective training set served as tuning data. To limit the training time, I reduced the learning rate to 1^{-05} and did not further reduce it.

Referring to the Results of the XGBoost model, which I used to predict AKOA using only numeric data, I excluded the five least important features in order to minimise the size and complexity of the model and hence reduce training time and overfitting, which I expect to lead to higher performance [62]. I applied this to all three data batches. Run 16 corresponds to the training of Model 4 using Batch 1, which includes the image data and only the four most important features. I used the same setting during the training with Batch 2 (Run 15) and Batch 3 (Run 17). The learning rate remained 1^{-05} and I used the standardised data for both runs. All training runs are summed up in figure 5.6.

	Model	Data batch	Numeric data	Number of frozen layers	Learning rate	Amount of data
Run 1	Model 1	Batch 1	No	188	1^{-04}	21,139
Run 2	Model 1	Batch 1	No	153	1^{-05}	21,139
Run 3	Model 2	Batch 1	No	188	1^{-04}	21,139
Run 4	Model 3	Batch 1	No	188	1^{-04}	21,139
Run 5	Model 3	Batch 3	No	188	1^{-04}	16,941
Run 6	Model 3	Batch 2	original	188	1^{-06}	16,941
Run 7	Model 4	Batch 2	original	188	1^{-08}	16,941
Run 8	Model 4	Batch 2	standardised	188	1^{-06}	22,304
Run 9	Model 4	Batch 2	standardised	0	1^{-06}	22,304
Run 10	Model 4	Batch 2	standardised	153	1^{-06}	22,304
Run 11	Model 4	Batch 2 (OMC)	standardised	0	1^{-06}	22,304
Run 12	Model 4	Batch 2 (OCM)	standardised	0	1^{-05}	22,304
Run 13	Model 4	Batch 2 (MCO)	standardised	0	1^{-05}	22,304
Run 16	Model 4	Batch 1	standardised	0	1^{-05}	21,139
Run 15	Model 4	Batch 2	standardised	0	1^{-05}	22,304
Run 17	Model 4	Batch 3	standardised	0	1^{-05}	16,941

Table 5.6: Summary of all training runs with only image data (Run 1 – 5) and numeric and image data in combination (Run 6 – 17). Run 15 - -17 include only the four most important numeric features. Original numeric data means no normalised data was used. For standardised numeric data, I used the standard scaler normalisation method. CNN Models 1–5 are described in Section 5.2.2. OMC: trained with OAI & MOST data and tested with CHECK. OCM: trained with OAI & CHECK and tested on MOST. MCO: trained with MOST & CHECK and tested on OAI. I trained and tested all other runs with all three datasets.

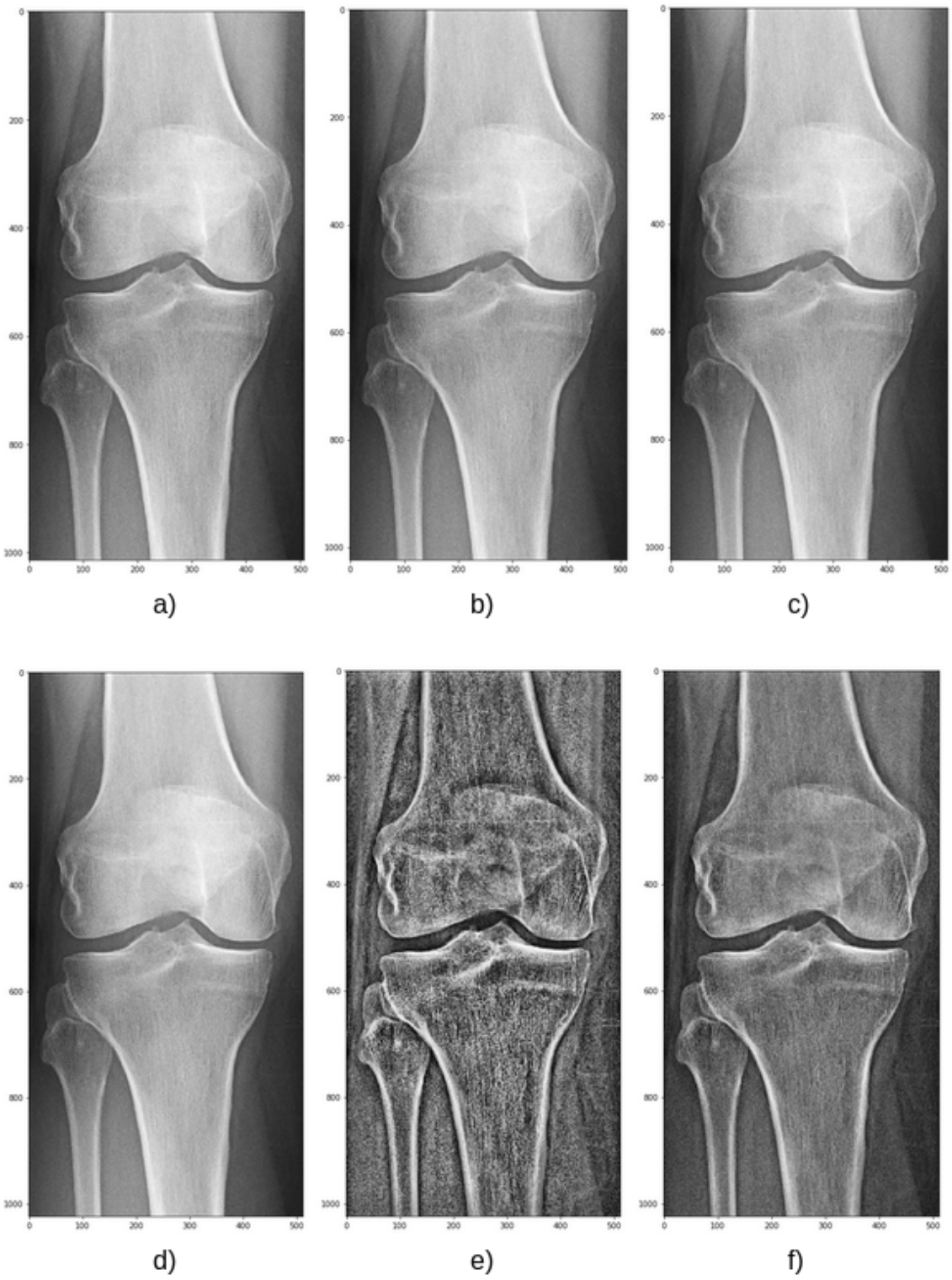


Figure 5.12: Example of a pre-processing sequence of a knee X-ray taken out of the OAI study. a) Original image. b) Resized to 1024, 512 pixels. c) Normalised pixel intensity to values between 0 and 20 000. d) Gaussian Blurring applied. e) CLAHE applied. f) All pre-processing steps without Gaussian Blurring.

```
params = {
    'num_rounds':         75,
    'max_depth':          max_depth,
    'max_leaves':         2**max_leaves,
    'alpha':              alpha,
    'eta':                eta,
    'gamma':              gamma,
    'learning_rate':      lr,
    'subsample':          1,
    'reg_lambda':         1,
    'scale_pos_weight':   2,
    'tree_method':        'gpu_hist',
    'objective':          'binary:logistic',
    'verbose':            True
}
```

Figure 5.13: Parameters of the XGBoost model. Not defined values I optimised using grid search

```
res5c_branch2c (Conv2D)          (None, 32, 16, 2048) 1048576     res5c_branch2b_relu[0][0]
bn5c_branch2c (BatchNormalizati (None, 32, 16, 2048) 8192        res5c_branch2c[0][0]
res5c (Add)                      (None, 32, 16, 2048) 0           bn5c_branch2c[0][0]
                                                                  res5b_relu[0][0]
res5c_relu (Activation)          (None, 32, 16, 2048) 0           res5c[0][0]
Dropout_0.3_0 (Dropout)          (None, 32, 16, 2048) 0           res5c_relu[0][0]
Final_conv_0 (Conv2D)            (None, 32, 16, 512)  1049088     Dropout_0.3_0[0][0]
MaxPool_0 (GlobalMaxPooling2D)   (None, 512)          0           Final_conv_0[0][0]
dense_48 (Dense)                 (None, 2)            1026        MaxPool_0[0][0]
==================================================================================================
Total params: 24,604,994
Trainable params: 24,551,874
Non-trainable params: 53,120
```

Figure 5.14: The last layers of the model summary of the ResNetClassifier created by IBLab. A ResNet50 based model to classify between X-rays of wrist, knee, hip, hand, leg, spine and ankle and between sagittal and frontal view.

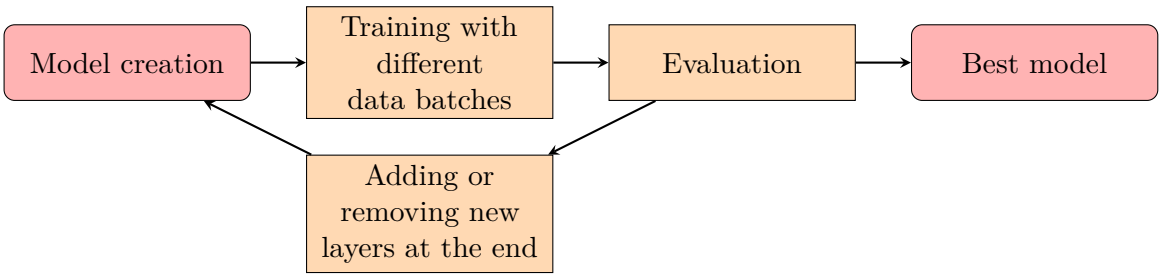


Figure 5.15: Process of creating the different CNN models.

CHAPTER 6

Results and Discussion

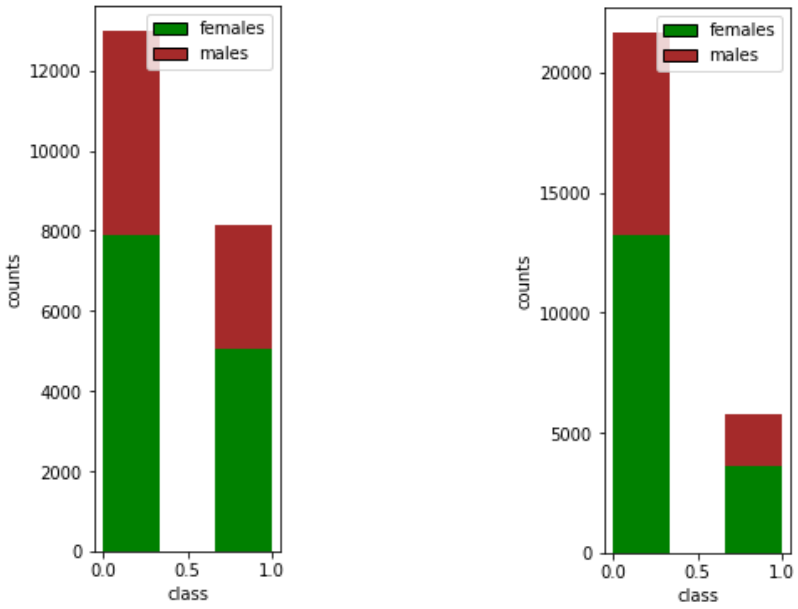
This chapter deals with the analysis of the data and the presentation and discussion of all results, which I obtained with the XGBoost model and the CNN models.

6.1 Data Analysis

This section deals with the analysis of the data considering Batch 1 and Batch 2. Previous studies suggest age, gender and BMI be associated with AKOA [6, 11, 13]. Thus, I will take a closer look at the correlation between these factors and AKOA. The three datasets provide me with a total number of 80,234 unilateral knee images, 40,120 among them are left knee images and 40,114 are right knee images. 66.16% of the total data origins from the OAI dataset, 24.57% from MOST and 9.27% from the CHECK study. Due to the high proportion of OAI data, the distributions of the OAI dataset correspond mainly to the plots including all datasets. Patients with remaining KL-grade 0 and 1, as well as KL-grade 4 at the baseline visit, were excluded for both data batches. Batch 1, with the class definition of 10% JSN, contains a total number of 21,139 images and Batch 2, for which I used the class definition of 20% JSN, a total number of 27,432 images.

Comparing the data of Batch 1 and Batch 2 regarding the class distribution, a significantly higher imbalance between the two classes can be seen for Batch 2 (Figure 6.1). Defining class 1 with more than 10% JSN, AKOA patients covers 38.6% of the total data of Batch 1. When increasing the threshold to 20% JSN, as expected, the ratio of fast-progressing patients decreases to 21.05%. The distribution of female and male participants is approximately the same for each class in both data batches (around 60% women and 40% men). The class distribution is reflected in Figure 6.1. Previous studies show arbitrary results. In the study of Halilaj et al., more women are represented among the group of non-progressors than in the group of progressors. Here the groups are also clustered by JSN [22]. In the study of Raynauld et al., 73% of fast progressors are women, versus 48% women among non-progressors [29]. These numbers are similar to the ones in

the study from Bartlett et al., where females represent 71% of fast progressors compared to 64% of women among non-progressors [30].



(a) 10% JSN **(b)** 20% JSN

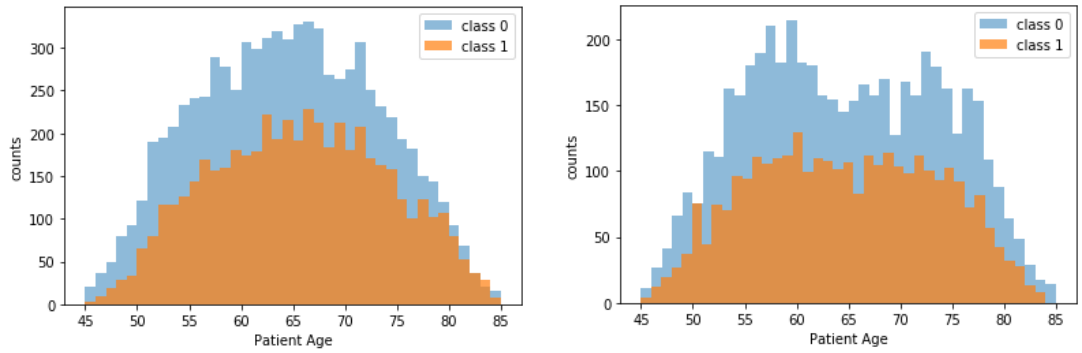
Figure 6.1: Class distribution for OAI, MOST and CHECK dataset. Left: JSN 10%. Right: JSN 20%. Green corresponds to female patients and red to male.

The age distributions for both class definitions including all three datasets are plotted for women in Figure 6.2 and men in Figure 6.3. The left diagrams reflect the female cohort and the right ones the male cohort. Concerning women of Batch 1 (Figure 6.2a), the distribution is a normal distribution around about 67 years for both classes. The orange curve of class 1 is slightly shifted to the higher age, which results in an increased ratio of class 1 to class 0 with higher age. For example, around 50% of women older than 82 develop fast-progressing KOA, whereas only around 35% of women aged 57 exhibits AKOA.

The female age distribution curve of 20% JSN in Figure 6.3a, also shows a normal distribution, but around 65 years. Here the proportion of class 1 to class 0, especially among the 51 to 80 years aged women, is very much lower than for the other class definition. This means a large amount of the female cohort in this age range develops KOA with a JSN per two years between 10 and 20%. The curve of class 1 is also shifted slightly to a higher age. The class 1 to class 0 ratio decreases until the age of 65 and then starts to increase again for higher ages.

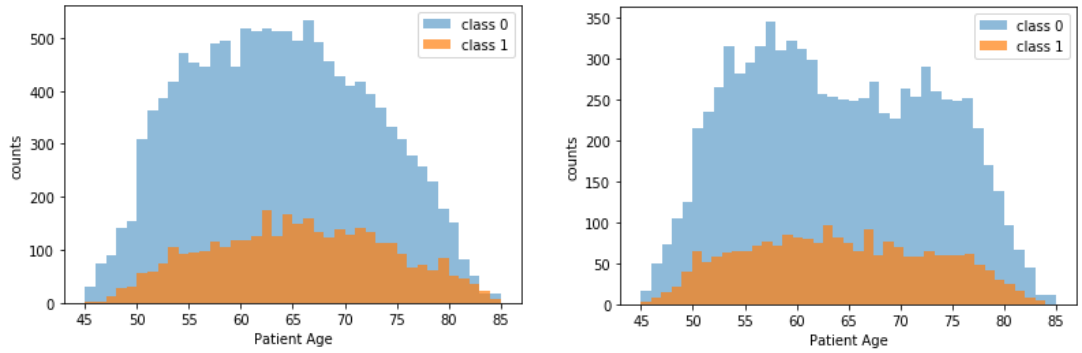
The male cohort of the total data shows a bimodal age distribution in both data batches (Figure 6.2b and 6.3b). This can be seen more clearly for the blue curves, the slow progressors, and only slightly for the distributions of class 1. One peak at around 59

years and the other at around 73 years. The class 1 distribution is in contrast to the female cohort very centred with the curve of class 0. The ratio of class 1 to class 0 is also much lower for the class definition of 20%.



(a) Female **(b)** Male

Figure 6.2: Age distribution for OAI, MOST and CHECK dataset, JSN 10 %. Blue corresponds to class 0, orange corresponds to 1.



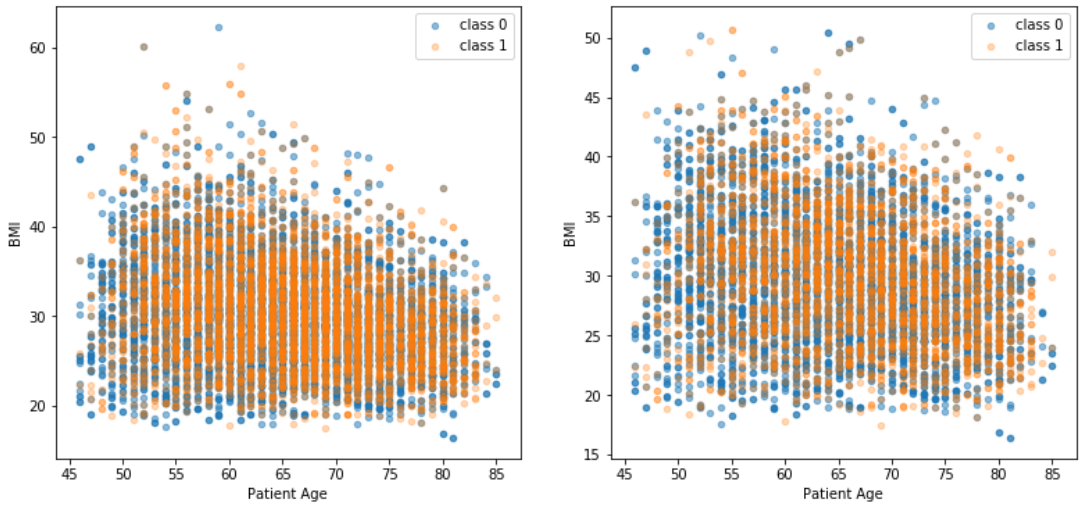
(a) Female **(b)** Male

Figure 6.3: Age distribution for OAI, MOST and CHECK dataset, JSN 20 %. Blue corresponds to class 0, orange corresponds to 1.

In Figure 6.4, the female participant's age and BMI of Batch 1 are plotted. The female cohort is imaged on the diagram in Figure 6.4a, considering the OAI, MOST and CHECK data, and on the diagram in Figure 6.4b, considering only the OAI data. All observations of the blue class 0 concentrate in both diagrams mainly between the age of 50 and 80 years with a BMI between 20 and 37 kg/m^2. For observations of class 1, the points concentrate between 53 and 80 years and 21 and 48 kg/m^2. Figure 6.5 represents the female cohort of Batch 2. The observations are distributed the same way as for Batch 1, except the density of the orange class 1 datapoints is generally less. Women with an age higher than 70, exhibit generally lower BMI. One reason for this may be that overweight people have a higher risk of developing other serious diseases in old age, which may take

the focus away from knee problems and prevent them from participating in a study like OAI.

Another aspect can be seen in the plots 6.4 and 6.5. Slightly fewer younger women with less weight develop AKOA. In Figure 6.4a the density of data points of class 1 for women younger than 57 and BMI lower than 30 kg/m^2 is lower than for women younger than 57 and BMI higher than 30 kg/m^2. Thus, for women aged under 57, a higher BMI has a larger impact on the development of AKOA, compared to older women. The study of Driban et al. also reports a low risk for women younger than 63.5 years to be a fast progressor, except when having a high BMI [6]. Besides, the centre of data points corresponding to class 1 is more oriented to a higher age, resulting in a higher proportion of AKOA patients with increasing age. This trend applies for both data batches but can be seen more clearly for Batch 2 in Figure 6.5b. In Batch 1 outliers with a BMI higher than 45 kg/m^2 correspond more likely to class 1, the fast-progressing patients. The association of BMI and AKOA is also confirmed by the studies [6, 9, 13]. All distributions of the male cohort do not show significant correlations between age or BMI and AKOA. An example of this is imaged in Figure 6.6, where I took all three datasets into account to plot the male distribution for age and BMI of Batch 1.

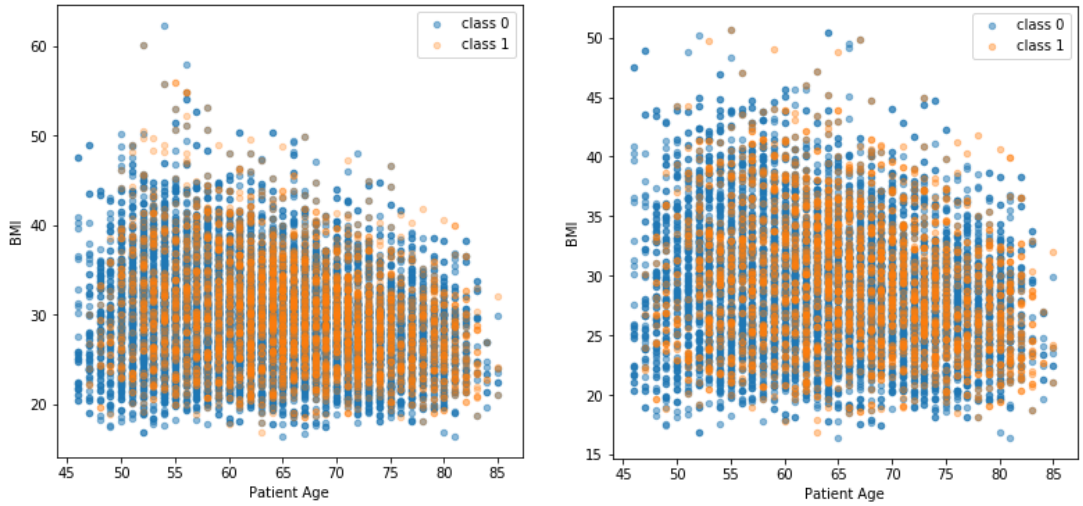


(a) OAI, MOST and CHECK **(b)** OAI

Figure 6.4: Age and BMI distribution for the female cohort with 10% JSN. Blue corresponds to class 0, orange corresponds to 1. (a) total data of OAI, MOST and CHECK. (b) only OAI data considered.

A further indicator for AKOA could also be the current KL-grade of the knee as seen in Figure 6.7. Due to the fact that the KL-grade distribution of class 0 has its maximum at KL-grade 2, whereas the class 1 distribution at KL-grade 3, patients with baseline KL-grade 3 are more likely to be a fast progressor than being a slow progressor.

Comparing now the different datasets, the MOST cohort shows some differences. When



(a) OAI, MOST and CHECK **(b)** OAI

Figure 6.5: Age and BMI distribution for the female cohort with 20% JSN. Blue corresponds to class 0, orange corresponds to 1. (a) total data of OAI, MOST and CHECK. (b) only OAI data considered.

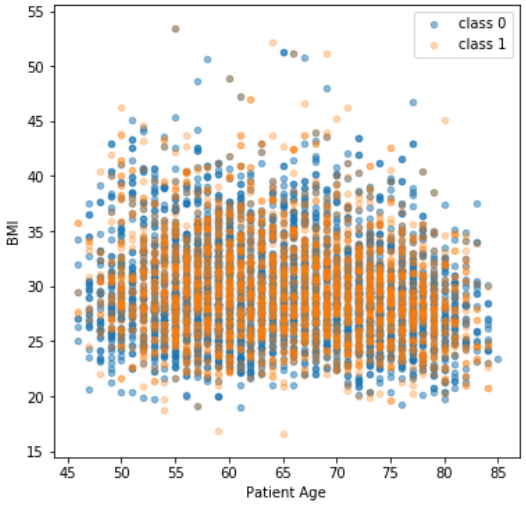


Figure 6.6: Plot of the age and BMI distribution of the male cohort of Batch 1. OAI, I considered MOST and CHECK data.

using the threshold of 10% JSN, the data is distributed very equally over both classes (Figure 6.8a) compared to the distribution of the total amount of data (Figure 6.1a). This does not hold for the class distribution of MOST data in Batch 2, which shows a similar distribution as for the total amount of data of Batch 2. That proves the presence of many participants exhibiting a JSN larger than 10 but lower than 20% JSN among the MOST cohort. The MOST cohort also shows a different concentration of observations of

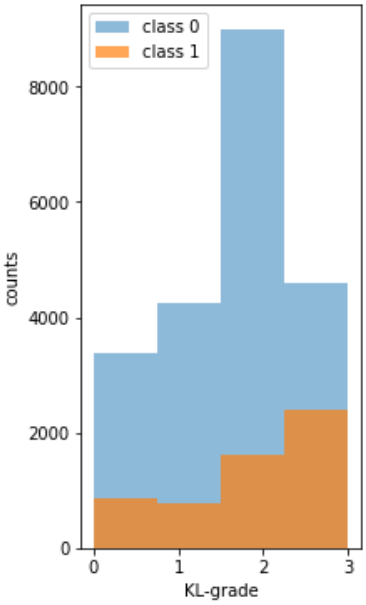
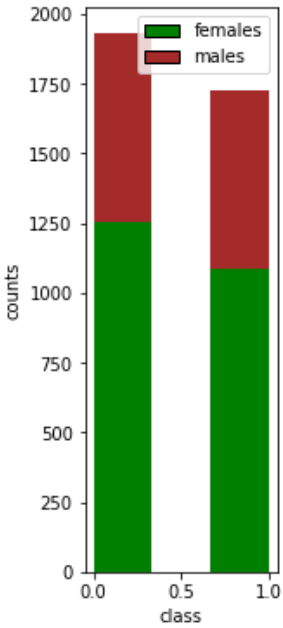
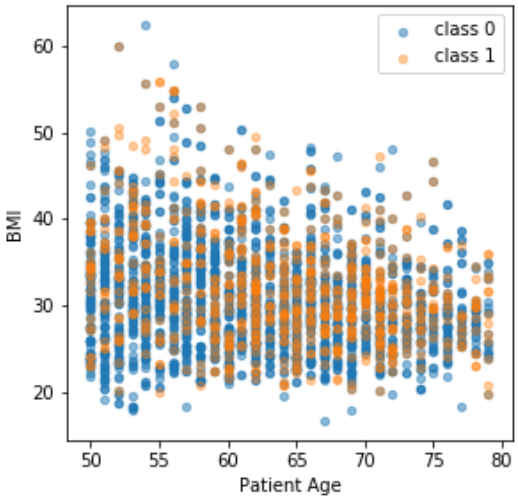


Figure 6.7: Plot of the KL-grade distribution of Batch 2 including female and male of the OAI, MOST and CHECK cohort. Blue: class 0. Orange: class 1.



(a) Class distribution of Batch 1. Green: females. Red: males.



(b) Age and BMI distribution of the female cohort of Batch 2. Blue: class 0. Orange: class 1.

Figure 6.8: Data distribution of the MOST cohort.

class 1 for Batch 2. As seen in Figure 6.8b the orange data points of class 1 accumulate around the age of 60 to 75 years. This is a smaller range compared to the total data distribution as seen in Figure 6.5a. Since the amount of data for OAI is the largest one, as mentioned before, the distributions are very similar to the ones for the total data. Due to the small CHECK cohort, these results did not show any specific characteristics. In general, I can say that women younger than around 60 years have a lower risk to develop AKOA, except if they are overweight. Correlations between single features and AKOA can only be seen slightly. Therefore, it must be a combination of several parameters which is able to make a statement about the risk of developing AKOA.

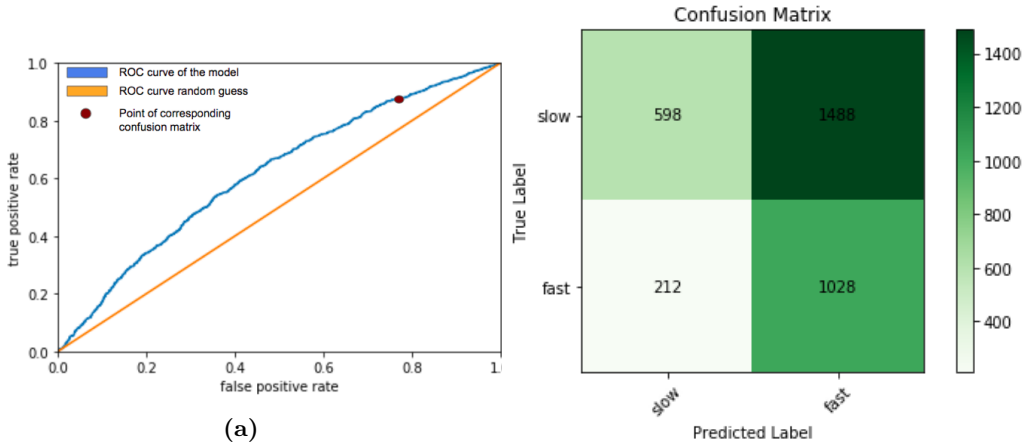
6.2 XGBoost Model Results

This section covers the results of the hyperparameter tuning of the XGBoost model and the appropriate training runs. Because the comprehension of a reasonable set of parameter values is very time-consuming [67], I applied grid searching for every single run, to find the optimal model parameter values (listed on the right side of Table 6.2). The individual tuning is very important due to the different data and feature combinations I used for each run. For all runs, the optimal learning rate was either 0.1 or 0.2. All other parameters showed larger variations between the different runs, for example, gamma, which varies between 0.2 and 0.9. The highest value of alpha was 0.5. As an evaluation tool, I use the AUC, where 1 accounts for the best result.

For training with Batch 2, the best practice was performed by including all features and missing values with an AUC of 0.6616 (Run P). The same applies to Batch 3, where I achieved the highest results with Run S (AUC: 0.7308). Considering training runs of Batch 2 and Batch 3, the AUC increased compared to Run D (AUC: 0.6469) and Run J (AUC: 0.6953), for which I excluded missing data. This could result from the fact of training with more data. On the contrary, training runs using Batch 1 showed opposite results. The value of AUC decreased or remained the same while considering observations with missing values. The best performance using Batch 1 was an AUC of 0.616 (Run A) including all features but excluding NaN values.

The ROC curves of the best practices for Batch 1 (Run A), Batch 2 (Run P) and Batch 3 (Run S) are plotted in Figure 6.9a, 6.10a and 6.11a, respectively. For every point on the ROC curve, a confusion matrix exists with respective specificity and sensitivity. Depending on the application of the model, a threshold can be chosen. If the goal is to detect fast-progressing patients as much as possible, the sensitivity should be as high as possible and the specificity as low as acceptable. Since there is no treatment for fast-progressing KOA patients, which would harm slow-progressing patients, I accept a high number of wrongly identified as being a fast progressor instead of a high number of wrongly identified as being a slow progressor. The red point on the ROC corresponds to the confusion matrix on the right side. Suitable confusion matrices for high sensitivity of Run A, P and S can be seen in Figure 6.9b, 6.11b and 6.10b, where the x-axis represents the label, which the model predicts, and the y-axis represents the ground truth. Positive

labels correspond to class 1, the fast progressors, negative labels to class 0. Looking closer at Run S, the model detected among 1,585 knees 1,472 knees as fast progressors. Among 9,550 slow- or non-progressing knees, 7,264 knees were falsely detected as fast progressors. In Table 6.1 I listed the specificity and sensitivity of the best practice of each batch (Run A, P and S) using the classification thresholds of 0.47, 0.21 and 0.13, respectively.

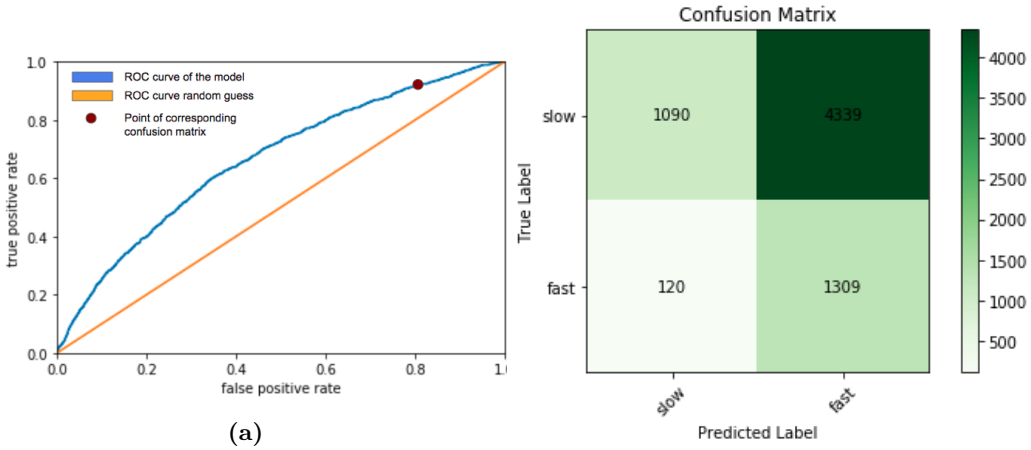


(a) (b)

Figure 6.9: XGBoost model results from Run A. (a) The x-axis represents the false positive rate, the y-axis the true positive rate. The blue graph corresponds to the ROC curve of Run A, the orange graph represents a random guess. (b) The confusion matrix of Run A taken at the red point on the ROC curve (threshold: 0.47). The x-axis represents the predicted labels, the y-axis the true labels. The higher the number of observations in one field, the darker the background of this field.

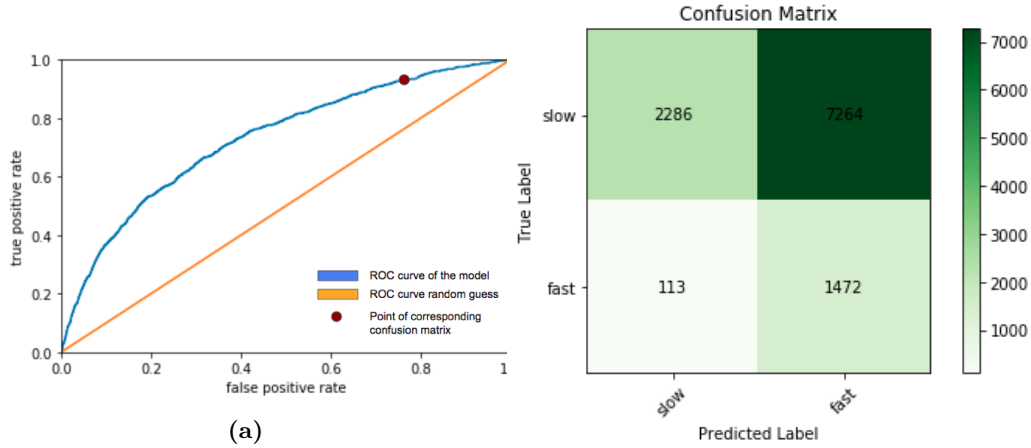
Defining AKOA with 20% JSN per two years resulted in a more extreme group of patients considered as class 1, compared to class 1 of Batch 1 where I used the threshold of 10% JSN. Thus, observations of class 1 are more sharply demarcated from class 0, which facilitates the learning process for the model and results in better performance. The results in Table 6.2 confirm this. All training runs using Batch 2 resulted in higher results than using Batch 1. Similar applies to the training results using Batch 3, for which I considered patients with remaining KL-grade 0 and KL-grade 1. This increases the number of observations for class 0 as well as the disparity between class 0 and class 1. This could be the reason for Batch 3, which predicts the fast-progressing patients among the slow- and non-progressing ones, exceeding all results I achieved with Batch 1 or Batch 2.

Thereupon, I inspected the importance of the numeric features I used for the XGBoost model. The three most relevant features for training were very similar for all runs. Including radiographic information, except for Run S, T and U (Batch 3), sclerosis always showed the highest impact on the prediction of AKOA. Considering all features or just



(a) (b)

Figure 6.10: XGBoost model results from Run P. (a) The x-axis represents the false positive rate, the y-axis the true positive rate. The blue graph corresponds to the ROC curve of Run P, the orange graph represents a random guess. (b) The confusion matrix of Run P taken at the red point on the ROC curve (threshold: 0.21). The x-axis represents the predicted labels, the y-axis the true labels. The higher the number of observations in one field, the darker the background of this field.



(a) (b)

Figure 6.11: XGBoost model results from Run S. (a) The x-axis represents the false positive rate, the y-axis the true positive rate. The blue graph corresponds to the ROC curve of Run S, the orange graph represents a random guess. (b) The confusion matrix of Run S taken at the red point on the ROC curve (threshold: 0.13). The x-axis represents the predicted labels, the y-axis the true labels. The higher the number of observations in one field, the darker the background of this field.

excluding the OARSI grade of sclerosis and osteophytosis, the KL-grade of the index knee, contralateral KOA (a KL-grade larger than 2 at the contralateral knee) and injection

of arthritis medication were very important criteria for all of these runs. The KL-grade was more important for Batch 2 and 3, whereas the contralateral KOA had a higher influence for Batch 1. Generally, the KL-grade makes sense to have a high impact on the prediction of AKOA since Dam et al. mentioned that the relevance of clinical factors in the prediction task could depend on the patient's current state of disease, which is expressed by the KL-grade itself [10]. After excluding subsequently the feature knee injection, the patient's BMI (Run F, O, R and U) and gender (Run B and C) happened to be the third most important criteria. The WOMAC score measuring the pain level was only in the top three important features of Run L. Due to the relativity and subjectivity of pain, the WOMAC score of pain suffers from incomparability and could be influenced by personal characteristics. Nevertheless, Davis et al. [23] found at least some single WOMAC criteria, like pain for walking or straightening the leg, to be associated with AKOA, whereas Raynault et al. [13] reported no correlation between the WOMAC score and AKOA. Whether pain is an indicator for AKOA could therefore be more or less a coincidence. Although osteophytes are a radiographic symptom of AKOA [18], this feature shows a small impact on the classification task, which is also confirmed by Felson et al. [19]. For the runs using Batch 1, all features were nearly on the same level of importance (see Figure 6.12a), whereas especially for Run Q and T using Batch 2 and Batch 3 respectively, the most important feature was by far the most important one (see Figure 6.12b and 6.12c).

Except for Batch 2, removing features always reduces the performance of the model. To use the information about the relevance of the features, I trained the model excluding the four least important features for every data batch. For all three batches, the least important features were the patient's age, hip symptoms and the WOMAC score for disability and stiffness. Hence, I performed Run V, W and X including only gender, BMI, knee injection, contralateral KOA, WOMAC pain score and the OARSI score of sclerosis and osteophytosis. In the following, I compare these runs to the other runs of the same batch with the exclusion of the missing data. Run V, for which I used Batch 1, yielded an AUC of 0.608. This result is lower, compared to the AUC of Run A (AUC: 0.616), where I took all features into account, but higher than the results of Run B (AUC: 0.6042), for which I removed both OARSI scores, and C (AUC: 0.5908), excluding OARSI scores and knee injection. I achieved lower AUC values for the training of Batch 2 when excluding the least important features (Run W, AUC: 0.6374), compared to the other runs with Batch 2 (AUC: 0.6402 - 0.6521). The same applies to Batch 3 (Run X). Hence, excluding features with low impact could not improve the performance of the XGBoost model.

6.2.1 Summary

To sum up, I achieved the best AUC value of 0.7308 when including non-progressors (Batch 3), all observations with missing values and all features (Run S). Even if for this run the model classifies the fast-progressing participants among slow and non-progressing ones, which was not the final goal of this thesis, the runs of Batch 3 help to compare my model with previous studies. To the best of my knowledge, there is no previous

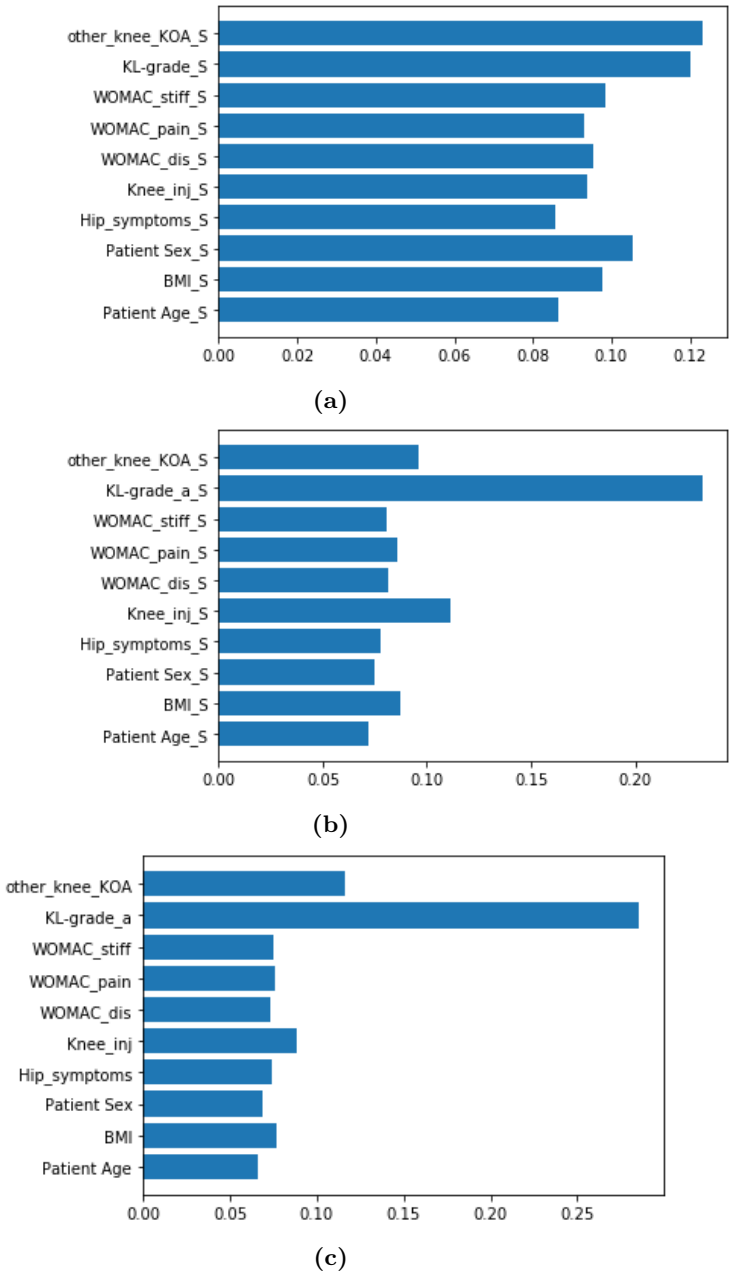


Figure 6.12: Plots of the most important features. In (a) the features of Run B, where I used Batch 1, in (b) the features of Run Q, where I used Batch 2 and in (c) the features of Run T, where I used Batch 3, are plotted.

study, which discussed the problem of classifying between slow and fast progressors. Tiulpin et al. [11] achieved similar results as I did for classifying between progressors and non-progressors. Considering Age, Sex, BMI, KL-grade, previous surgery, knee injury

Run name	Sensitivity (TP)	Specificity (TN)
Run A	0.829	0.287
Run P	0.916	0.201
Run S	0.929	0.239

Table 6.1: Sensitivity (TP rate) and Specificity (TN rate) results of all runs of the XGBoost model

and the WOMAC score, they reached an AUC of about 0.76 with a Gradient Boosting Machine [11]. Guan et al. [32] also classified progressors and non-progressors with an AUC of 0.66 using age, gender, ethnicity, BMI, history of knee injuries, KL-grade and the tibiofemoral angle as input for an artificial neural network model [32]. My results were also higher compared to the study of Halilaj et al. [22]. They used a LASSO regression model to predict progressing KOA among non-progressors with an AUC of 0.6 [22]. Even with the more difficult classification task, classifying between slow and fast progressors, I yielded higher AUC values than Guan et al. and Halilaj et al. [22, 32] and speaks for a high classification quality of my models.

Run name	No. of images	AUC	Most important features	Learning rate	Max depth	Max leaves	alpha	gamma
A	13,304	0.6160	1. sclerosis 2. contralateral KOA 3. gender	0.1	8	4	0.1	0.9
B	13,304	0.6042	1. contralateral KOA 2. KL 3. gender	0.1	8	4	0.1	0.7
C	17,403	0.5908	1. contralateral KOA 2. KL 3. gender	0.2	8	4	0.5	0.2
D	16,941	0.6469	1. sclerosis 2. KL 3. osteophytes	0.1	8	4	0.2	0.7
E	16,941	0.6402	1. KL 2. contralateral KOA 3. gender	0.1	8	4	0.5	0.2
F	22,304	0.6521	1. KL 2. contralateral KOA 3. BMI	0.1	8	4	0.5	0.9
G	16,941	0.6469	1. sclerosis 2. KL 3. osteophytes	0.1	8	4	0.2	0.7
H	16,941	0.6402	1. KL 2. contralateral KOA 3. gender	0.1	8	4	0.5	0.2
I	22,304	0.6521	1. KL 2. contralateral KOA 3. BMI	0.1	8	4	0.5	0.9
J	25,545	0.6953	1. sclerosis 2. KL 3. contralateral KOA	0.1	64	16	0.1	0.7
K	25,545	0.6922	1. KL 2. contralateral KOA 3. knee_inj	0.1	32	16	0.5	0.9
L	32,539	0.6901	1. KL 2. contralateral KOA 3. WOMAC_pain	0.1	64	16	0.1	0.9

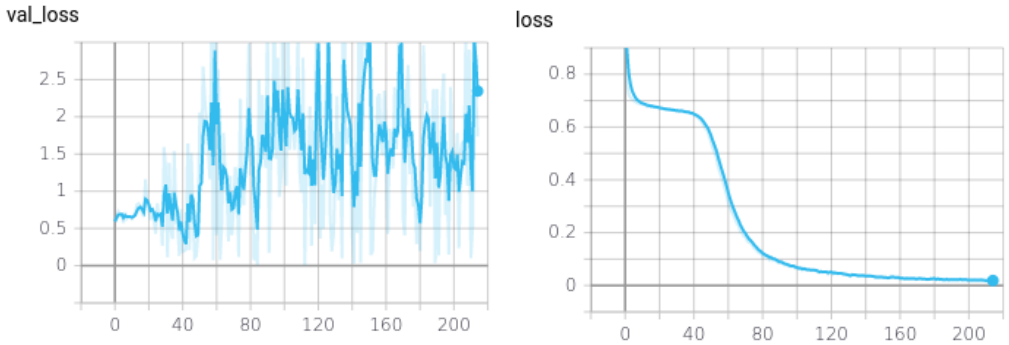
Run name	Image amount	AUC	Most important features	Learning rate	Max depth	Max leaves	alpha	gamma
M	21,139	0.6101	1. sclerosis 2. contralateral KOA 3. knee_inj	0.1	4	8	0.2	0.9
N	21,139	0.6100	1. contralateral KOA 2. KL 3. knee_inj	0.1	8	4	0.5	0.2
O	21,139	0.5939	1. contralateral KOA 2. KL 3. BMI	0.1	8	4	0.2	0.2
P	27,432	0.6616	1. sclerosis 2. KL 3. contralateral KOA	0.1	8	4	0.1	0.5
Q	27,432	0.6556	1. KL 2. knee_inj 3. contralateral KOA	0.2	8	4	0.2	0.7
R	27,432	0.6493	1. KL 2. contralateral KOA 3. BMI	0.2	8	4	0.2	0.2
S	44,538	0.7308	1. KL 2. sclerosis 3. contralateral KOA,	0.2	8	4	0.1	0.2
T	44,538	0.7268	1. KL 2. contralateral KOA 3. knee_inj	0.2	8	4	0.2	0.2
U	44,538	0.7237	1. KL 2. contralateral KOA 3. BMI	0.1	8	4	0.2	0.2
V	21,139	0.6080	1. sclerosis 2. knee_inj 3. contralateral KOA	0.2	8	4	0.5	0.9
W	27,432	0.6374	1. sclerosis 2. KL 3. knee_inj	0.1	8	4	0.2	0.5
X	44,538	0.6733	1. sclerosis 2. KL 3. contralateral KOA,	0.1	8	4	0.2	0.2

Table 6.2: Summary of the XGBoost training runs. Left part of the table shows the results. The right part shows the used hyperparameters of the training run.

6.3 CNN Results

Due to the better performance of the XGBoost model using as well the radiographic information, I expect the CNN to find even more information in a knee radiograph about the development of AKOA. Tiulpin et al. [11] also reported good results for using only radiographs as input. The following section discusses the results of my different CNN models. I trained these models, which are listed and described in Section 5.2.2, on the different data batches using once only image data and once image data with complementation of the numeric data. To evaluate these, I used the sklearn metrics library to calculate the AUC and plot the ROC curve. In Table 6.3, the AUC result for each run is listed.

For every training, I assigned up to 600 epochs but interrupted the training if the validation loss started to oscillate very strongly. An example of this reflects Figure 6.13. Here the loss function for validation and training of Run 14 is plotted. The validation loss function starts to oscillate strongly, simultaneously with a decrease of the negative slope of the training loss function around epoche 40. This could imply that the model found some new features to increase training performance (decrease of the training loss function) but overfit too strongly and the performance of the classification task could not increase (no simultaneous decrease of the validation loss function). Hence, I stopped training runs earlier, when recognizing a strong oscillation of the validation loss.



(a) (b)

Figure 6.13: CNN model results using Model 4 (Run 14). The x-axis corresponds to the number of epoche and the y-axis to the loss. (a) reflects the loss function of the validation set. (b) reflects the loss function of the training set.

I achieved very low AUC results with Model 1 and Model 2 in combination with Batch 1, where AKOA was defined with at least 10% JSN per two years (AUC: 0.500 - 0.551). Here I used only image data. Model 1 performed better for Run 2, where I set some of the last layers (35 layers) of the pre-trained model as trainable and increased the learning rate to 1^{-06} (AUC: 0.5514). After increasing the number of nodes from 256 to 2048 and adding one more dense layer, the new model performed only slightly better with an AUC of 0.5626. I could not increase the performance significantly of Model 3 by adding the not

normalised numeric data and using Batch 2 (Run 6) instead of Batch 1 (AUC: 0.5706). This low improvement may result from using not normalised data, where each variable scales in different ranges and is therefore hard to compare.

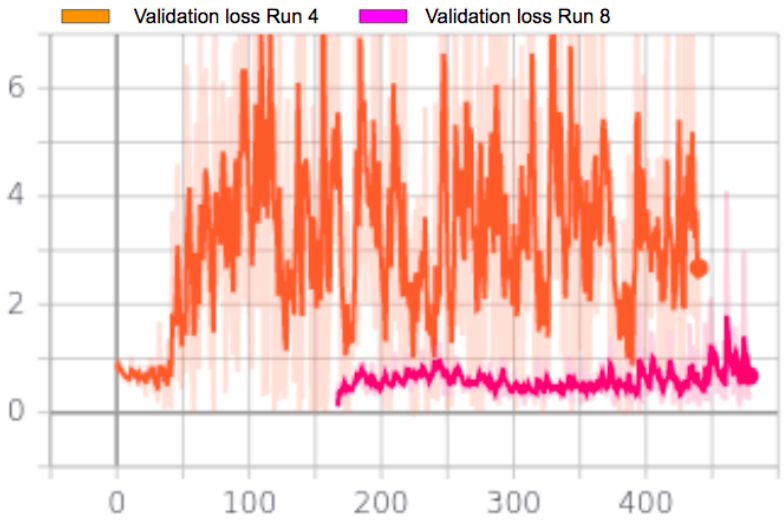


Figure 6.14: Plot of the validation loss of Run 4 (orange) and Run 8 (pink). The x-axis corresponds to the number of epoche and the y-axis to the loss.

Since these results are fairly low and very close to a random guess, I tried to optimise the parameters of the model and modify the input data further. Run 8, for which I added a dropout rate of 0.3 and used the standardised data as input, yielded an AUC of 0.663. Because of the simultaneous change of both parameters, I could not identify if the addition of the dropout or the modification of the input was the reason for this improvement. Run 8 also shows reduced overfitting, which is indicated by a less oscillating validation loss function (the pink graph in Figure 6.14). This oscillation is significantly less, compared to the orange graph. The orange graph corresponds to the validation loss function of Run 4, for which I used a model without a dropout rate (Model 3). Another improvement of the AUC of Run 8 could result from the higher amount of data compared to previous runs. Due to the elimination of the variable knee injection, fewer observations had to be excluded because of missing values, resulting in 30% more data. For all further runs, I kept this feature excluded.

By means of the following runs, I discuss the impact of using the pre-trained weights to train the model. Run 8, for which I loaded and froze the first 188 layers of Model 4 loaded with the pre-trained weights, yielded an AUC of 0.663. Freezing only the first 153 layers (Run 10) the performance decreased to 0.5483 but increased again to 0.584 when training without any pre-trained weights (Run 9). Comparing Run 8 and Run 9, I could observe a decrease in the AUC. I achieved contrary results for Run 18 and 15. As before, all model parameters remained the same for both runs, except the number of frozen pre-trained layers. The performance of Run 15 without any pre-trained weights (AUC: 0.6878) was

slightly higher compared to Run 18 with 188 frozen layers (AUC: 6749). Hence, I can rule out the fact that the model recognizes images as a whole since the model did not classify significantly better with the inclusion of the pre-trained knowledge. The best practice I achieved was a model trained from scratch. Nevertheless, I could not find any other correlation between the usage of the pre-trained weights and the performance of the model. Since the pre-trained model was trained on X-ray images of all body parts, only the basic knowledge about X-ray images could be used in my models, which could be the reason for no improvement in performance using the pre-trained weights.

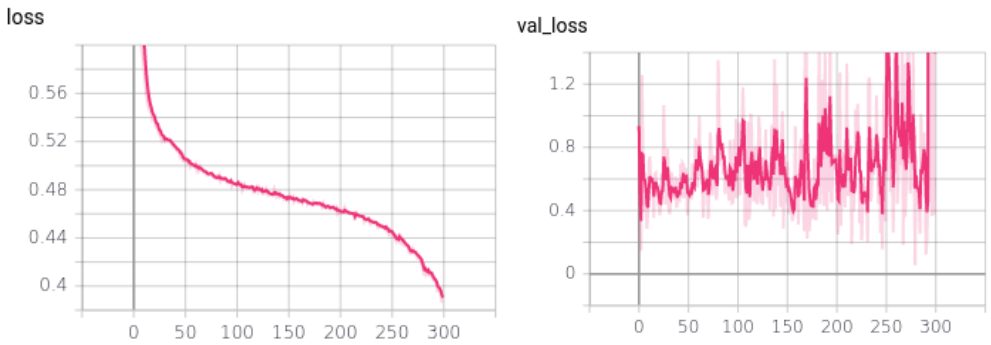
The experiments, for which I used two of the three datasets as training and tuning data and the third one as a testing set, achieved similar results as the previous runs. The highest AUC of 0.575 I yielded with Run 12. Here I used a mix out of OAI and CHECK data for training and the MOST dataset as a testing set. The testing with the CHECK dataset (Run 11) yielded the lowest results (AUC: 0.5197). Possibly, the OAI and MOST datasets are the most similar ones, which would result in a low AUC when using a testing dataset with new image characteristics.

Taking the results of the XGBoost model into consideration, where the most important features of the numeric data were identified, the CNN model results did not show a consistent behaviour regarding the number of numeric features. For all Batch 1 and Batch 2, the least important features were the patient's age, the occurrence of hip symptoms and the WOMAC score for physical function, stiffness and pain. Apart from the patients' age, all variables are very subjective and hard to compare, which could explain the little correlation to the development of KOA. Training with Batch 2 and the four most important features (BMI, gender, contralateral KOA and KL-grade) the fast progressors could be identified among slow progressors with an AUC of 0.6878 (Run 15). This shows a significant increase in performance compared to Run 9 (AUC: 0.584), in which I trained with all numeric features. The same experiment with Batch 1 exhibited contrary results. Run 16, where I used only the four most important features, achieved an AUC of 0.5809, whereas Run 14, including all numeric features, yielded slightly higher results with an AUC of 0.5918. Hence, using Batch 2, the higher number of less important features disturbed the classification task of the model, whereas the exclusion of less important features in Batch 1 did not show large changes in performance. These findings could correlate with the diagrams in Figure 6.12a and 6.12b. Because for Batch 2 large differences between the most important feature and the other less important ones can be seen, which is not for Batch 1, the impact of using just important features could be higher for Batch 2. This confirms the larger change in performance of Model 4.

Considering now the same parameters of the model but a change of the data batch, I compared Run 16, 15 and 17 with AUC results of 0.5809, 0.6878 and 0.7646, respectively. As seen in Table 6.3 I achieved the highest performance using Batch 3 and the lowest using Batch 1. The better results of Batch 2 compared to Batch 1 confirmed my expectations. The class definition of 20% JSN includes only the more severe cases of AKOA in class 1, which could show more clear characteristics in the X-ray images or even in the numeric data. So, the classification between slow- and fast-progressing participants could be

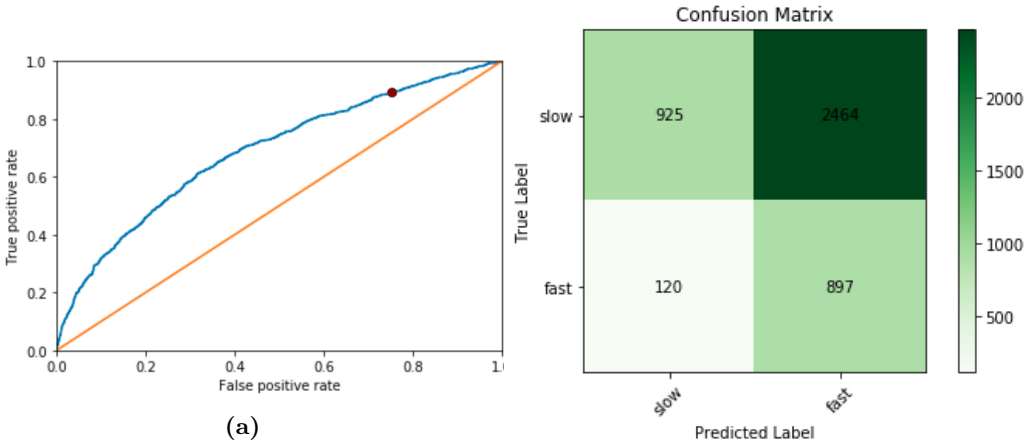
facilitated using Batch 2 (Run 15, AUC: 0.6878). A similar amount of increase I observed as well for the training run, for which I used Batch 3 (Run 17, AUC: 0.7646). Due to the inclusion of non-progressing patients, in this trial, the model predicts fast-progressing KOA among non, slow and fast progressors. This results on one hand in a higher amount of data and on the other hand in an increased image entropy among the total data, which could both be the reason for an improvement of the classification performance of the model. The ROC curve and the confusion matrix of Run 15 is plotted in Figure 6.16 and in Figure 6.18for Run 17. The confusion matrices correspond to the red point in the respective ROC curve. As previously explained in Section 6.2, I chose a sensitivity as high as possible and a specificity as low as acceptable. Figure 6.15 and 6.17 show the loss function of the training and the validation set for Run 15 and Run 17. The start of a strong oscillation of the validation loss and a simultaneously increased negative slope of the training loss indicates overfitting of the model at around epoche 150 for Run 15 and at epoche 40 for Run 17. Most of the training runs were affected by overfitting. From the point of strong oscillation of the validation loss, the performance did not increase and I stopped the run. A possible reason for the overfitting results from a bad regularisation of the models and a too large amount of model parameters. It was out of the scope of my work to analyse this further.

Further, I compare my results to the literature. The results of Batch 3 come close to the ones achieved in the work from Tiulpin et al. [11], where an AUC of 0.8 was yielded as best practice. A model from Guan et al. even performed with an AUC of 0.86 [32]. The difference to my work is that both other studies merged the class of slow and fast progressors and then classified between non-progressing and progressing KOA patients [11, 32]. Besides, AKOA was defined by a KL-grade change in the following years [11] and with a JSN of more than 0.7 mm per 2 years. Tiulpin et al. [11] confirmed a high performance using only the X-ray image as input. I could not confirm this result with my models and my different classification tasks, where the best practice using only radiographs was an AUC of 0.5626 for classifying between slow and fast progressors.



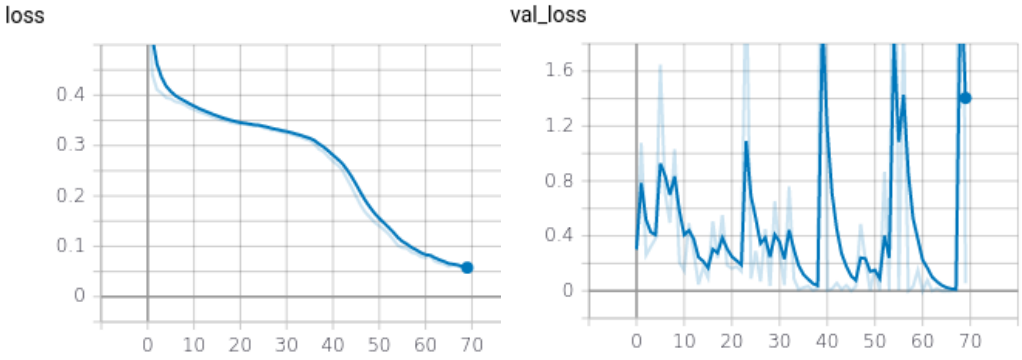
(a) (b)

Figure 6.15: CNN model results using Model 4. The x-axis corresponds to the number of epoche and the y-axis the loss. In (a) the training loss function of Run 15 and in (b) the validation loss function of Run 15 is plotted.



(a) (b)

Figure 6.16: CNN model results using Model 4 (Run 15). In (a) the blue graph reflects the ROC curve of Run 15. The true positive rate is plotted over the false negative rate. The orange curve represents the ROC curve of a random guess. In (b) the Confusion Matrix is reflected corresponding to the red point on the graph in (a). The x-axis corresponds to the predicted labels and the y-axis to the true labels.



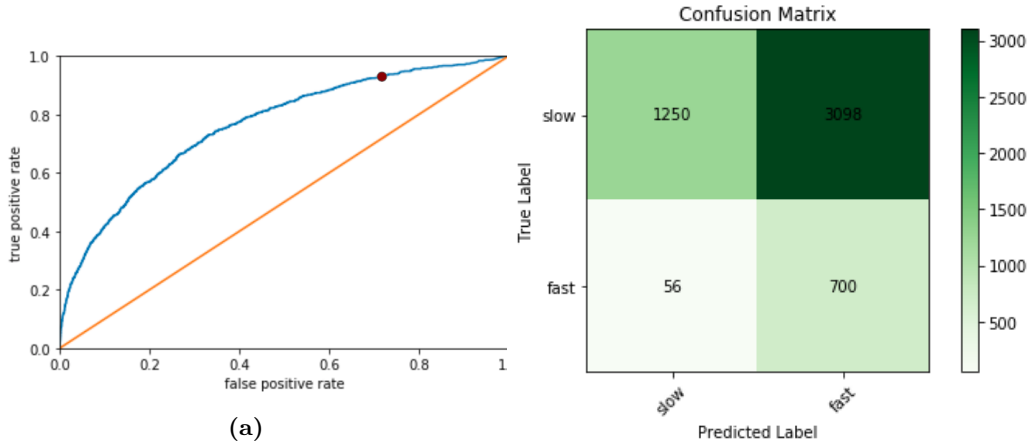
(a) (b)

Figure 6.17: CNN model results using Model 4. The x-axis corresponds to the number of epoche and the y-axis the loss. In (a) the validation loss function of Run 17 and in (b) the loss function of the training of Run 17 is plotted.

6.4 Summary

In the following, I sum up the main findings of the data analysis, XGBoost model and CNN model training runs:

- Data of Batch 2 has 21.05% of fast progressors, compared to Batch 1 with 38.6%.



(a) **(b)**

Figure 6.18: CNN model results using Model 4 (Run 17). In (a) the blue graph reflects the ROC curve of Run 17. The true positive rate is plotted over the false negative rate. The orange curve represents the ROC curve of a random guess. In (b) the Confusion Matrix is reflected corresponding to the red point on the graph in (a). The x-axis corresponds to the predicted labels and the y-axis to the true labels.

- For women, younger than 57 years, the BMI has a higher influence regarding the development of AKOA.
- Women younger than 60 have a decreased risk of being a fast progressor.
- Patients with KL3 at baseline are more likely to develop AKOA.
- Best practice with XGBoost model: AUC 0.6616 (Batch 2) and AUC 0.7308 (Batch 3).
- The most important features were: KL-grade of the index knee and KOA of the contralateral knee, OARSI-grade of sclerosis and osteophytosis.
- Best practice with CNN model: AUC 0.6878 (Batch 2) and AUC 0.7646 (Batch 3).
- Performance was highest for Batch 3 and lowest for Batch 1 and higher for JSN% than for JSN%, due to higher image entropy..
- No significant difference in performance with and without weights of the pre-trained model
- Best performance with split datasets was training with OAI and CHECK and testing with MOST: AUC 0.575.

Therefore, for the future I recommend to use Model 4, where I added 3 dense layers in total and used the LeakyReLU as an activation function. Here, the best results were achieved to classify between KOA and AKOA.

Run name	Model	Data batch	Numeric data	No. of frozen layers	AUC
Run 1	1	Batch 1	No	188	0.5043
Run 2	1	Batch 1	No	153	0.5514
Run 3	2	Batch 1	No	188	0.5002
Run 4	3	Batch 1	No	188	0.5626
Run 6	3	Batch 2	original	188	0.5706
Run 8	4	Batch 2	stand.	188	0.6630
Run 9	4	Batch 2	stand.	0	0.5840
Run 10	4	Batch 2	stand.	153	0.5483
Run 11	4	Batch 2 (OMC)	stand.	0	0.5197
Run 12	4	Batch 2 (OCM)	stand.	0	0.5750
Run 13	4	Batch 2 (MCO)	stand.	0	0.5333
Run 14	4	Batch 1	stand.	0	0.5918
Run 15	4	Batch 2 (BMI, gender, contralateral KOA, KL-grade)	stand.	0	**0.6878**
Run 18	4	Batch 2 (BMI, gender, contralateral KOA, KL-grade)	stand.	188	0.6749
Run 16	4	Batch 1 (BMI, gender, contralateral KOA, KL-grade)	stand.	0	**0.5809**
Run 17	4	Batch 3 (BMI, gender, contralateral KOA, KL-grade)	stand.	0	**0.7646**

Table 6.3: Summary of all training runs with only image data (Run 1 - 5) and numeric and image data in combination (Run 6 - 17). Original numeric data means I used no normalised data. For standardised numeric data (stand.) I used the StandardScaler normalisation method. OMC: trained with OAI & MOST data and tested with CHECK. OCM: trained with OAI & CHECK and tested on MOST. MCO: trained with MOST & CHECK and tested on OAI. I trained and tested all other runs with all three datasets. AUC: Area Under the Curve.

CHAPTER 7

Conclusion and Future Work

7.1 Summary

In this thesis, I investigated the ability of a Convolutional Neural Network to classify between slow- (class 0) and fast-progressing (class 1) KOA patients. AKOA should be defined by JSN per time. I tested two different thresholds for the definition of both classes. Once I defined AKOA by at least 10% and once by at least 20% of JSN per two years. As input data for the network, I considered a single radiograph complemented with demographic and clinical data, such as patient's gender, age, BMI, the presence of hip symptoms and contralateral KOA, the WOMAC score of disability, pain and stiffness and the KL-grade of the index knee. I carried out a data analysis of the data, which is composed of the OAI, MOST and CHECK study dataset. The cohort shows an equal ratio of women and men for both classes, which is about 60% to 40%. The class definition of 20% JSN resulted in a decreased proportion of class 1, from 38.6% to 21.05%. Adding also non-progressors to class 0, the fraction of AKOA knees decreased to 14.23%. Considering the age and BMI of the female cohort, women younger have less risk to develop AKOA, than older females or females with higher BMI, which is also confirmed by Driban et al. [6]. Besides, patients exhibiting KL-grade 3 at the baseline visit are more likely to develop fast-progressing AKOA than patients, which exhibit KL-grade 2, 1 or 0 at baseline.

The fundamental research question of my thesis is: **"Is it possible to classify KOA progression into fast and slow progression (defined by JSN per year) using Convolutional Neural Networks?"**. I answered this question using different approaches. With an XGBoost model I trained with only numeric data and for the training of image data only and the combination of both, I implemented multiple CNN models. I started with training an XGBoost model to predict AKOA, using all above mentioned numeric factors, including the OARSI grade of sclerosis and osteophytosis. Sclerosis, KL-grade and contralateral KOA, which are all radiographic characteristics, proved to

be the numeric features with the highest impact on the classification task, which is also confirmed in the studies [7, 15, 33]. I obtained the best classification performance using only numeric data with the classification threshold of 20% JSN. After including the non-progressors to class 0, the XGBoost model is able to predict with an AUC of 0.7308 fast-progressing KOA. Excluding the non-progressors from this data, the model yielded an AUC of 0.6616. With the 10% JSN as class definition, the model is able to classify with an AUC of 0.616 between fast and slow progressors. These results are in the same range as the ones from similar studies (AUC: 0.6 - 0.75) [11, 22, 32].

In order to use the X-ray images in combination with numeric features as input, I created four different convolutional neural network models, which are based on the architecture of a ResNet50. For the first runs, I applied the transfer learning method and used the weights of a pre-trained ResNet50, which was trained on X-ray images of different body parts. Since the data I used for validation was also used to train the pre-trained model, I had to rule out the possibility of the network to recognise the images as a whole. Hence, for all other runs, I trained the model from scratch. I expanded the model based on the ResNet50, which performed best, with two dense layers. Both with a LeakyReLU as activation function and a dropout rate of 0.3, the first with 2048 nodes and the second with 1024 nodes. I added one output node and a sigmoid function. As the best practice of this model, I achieved an AUC of 0.6878 for classifying between fast and slow progressors. Here I used 20% as the class definition threshold and considered as input, next to the image data, BMI, gender, the KL-grade and the information about contralateral KOA. To generate a reference value, I trained this model also including non-progressing KOA patients to class 0 and resulted in an AUC of 0.7646. This good performance could result from an increased amount of training data and a higher image entropy between class 0 and class 1.

I compared my results to similar studies, which were about classifying between non-progressors and progressors [11, 32]. Tiulin et al. achieved an AUC of 0.8 using a change of the KL-grade in the following years as the definition of progressing KOA [11]. Defining progressors with more than 0.7 *mm* JSN per 2 years, Guan et al. obtained an AUC of 0.863 [32]. My results, for which I also considered the non-progressing patients, are below the performance of previous work, which could result from the different definitions of AKOA between the studies. However, since the inclusion of non-progressing patients delivered better results, I suggest the classification between slow and fast progressors to be a more difficult task compared to the classification between progressors and non-progressors.

To conclude, the classification between slow and fast progressors exhibits good performance, when defining AKOA with 20% of JSN. Hence, although previous studies achieved higher results when classifying between progressors and non-progressors, my research is more relevant due to the low availability of knee radiographs of non-progressing patients, which mostly do not exhibit any symptoms or pain. Also, the definition of AKOA by JSN is simple and easy to measure. Besides, all input data, which are required for my models, are easy to obtain. It is a big achievement to be able to predict AKOA among progressing patients better than random guessing, since there is no prediction method available yet.

My classification model, which is able to classify slow- and fast-progressing patients with an AUC of 0.68, could definitely serve as a good and comfortable decision supporting tool for physicians in diagnosis. A risk assessment of AKOA would happen earlier, faster and more reliably. High-risk patients can be observed more specific, modifiable risk-factors can be changed (like reduction of BMI) and more time can be used to find alternative medical interventions to avoid a knee replacement surgery ending up in less pain and costs.

7.2 Limitations and Future Work

Starting from the definition of the fast-progressing KOA patients, the percentage of JSN per time seems to be in general a good choice regarding its simplicity. Other studies, such as those from Guan et al. or Bartlett et al. also achieved good classification results using this way of definition [30, 32]. The fraction of AKOA in this work (21% of the used data) is significantly higher than the number of 3.4%, which can be found in literature and describes the proportion of AKOA patients among KOA. Even after considering that I did not include asymptomatic patients, my definition of AKOA could still be too slight. Thus, it would also be interesting to compare the performance of my classification models when using different definitions of AKOA such as more extreme thresholds of JSN or the change of the KL-grade, as in other previous work [6, 7, 15, 23]. Not only new definitions of AKOA or progression, but future work could also test the same model classifying first between progressors and non-progressors and subsequently predicting AKOA among these progressing patients. The model can then be compared to previous studies and could then be improved by small modifications.

All CNN models started to overfit at some point during the training, which could indicate a bad regularisation of the model and a too high number of model parameters. To counteract the overfitting of the models, image augmentation, which is random processing of the image data, can be applied or noise can be added to the training data to result in better-regularised models. Besides, for the CNN models, I handled the missing values by removing these observations completely. To increase the amount of data, imputation methods could be used in further experiments. Another prediction improvement method could be to use instead of a single X-ray image a sequence of images, which would deliver more information about the changes in bone structure and the current progression of KOA, as suggested in previous work by Halilaj et al. [22]. Nevertheless, this would be hard to realise for very fast-progressing KOA patients.

The most important limitation of training with the image data was the fact of training from scratch. The pre-trained model, which was available to me, was trained with the same images that I used as validation data. Since the task of this classifier should be to work on completely new images, I could not rely on the outcomes of the transfer learning method in this work. Although I obtained similar results for experiments with and without the pre-trained model weights, I could not figure out the impact of the pre-known images. Besides, the ResNetClassifier, which was trained on seven different

body parts, contains many weights without relevance to my classification task. Networks, pre-trained on a high number of only knee radiographs, which are not used for subsequent validation, could provide more useful information for the classification task of this work.

Nevertheless, the results of my work were a good contribution to the research of AKOA prediction methods. To the best of my knowledge, there is no other study, which tested one model involving once only slow and fast progressors and once adding non-progressing patients. Hence, this work can serve as a good fundament for further research regarding a prediction method of AKOA.

APPENDIX A

Data Analysis

		Number of images	percentage
Total		21 139	100 %
Class 0		12 985	61.4 %
	female	7 885	60.72 %
	male	5 100	39.28 %
Class 1		8 154	38.6%
	female	5 054	61.98 %
	male	3 099	38.02 %

Table A.1: Summary of the data with class definition of 10 % JSN per 1 or 2 years, exclusion criteria of remaining KL 0 or remaining KL 1 from OAI, MOST and CHECK

		Number of images	percentage
Total		16 988	100 %
Class 0		10 775	63.4 %
	female	6 407	59.46 %
	male	4 368	40.54 %
Class 1		6 213	36.6%
	female	3 799	61.15 %
	male	2 414	38.85 %

Table A.2: Summary of the data with class definition of 10 % JSN per 1 or 2 years, exclusion criteria of remaining KL 0 or remaining KL 1 from OAI

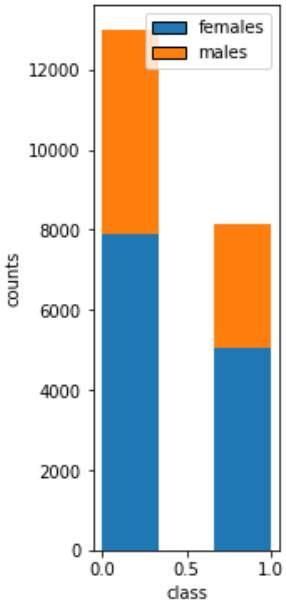
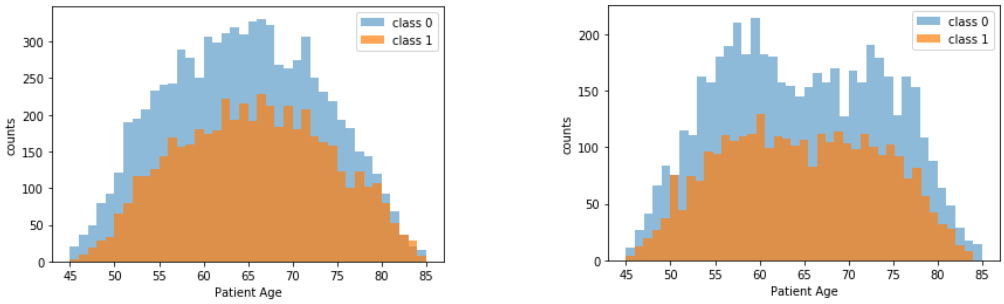


Figure A.1: Class distribution (0 and 1) of all datasets. Blue: females, orange: males

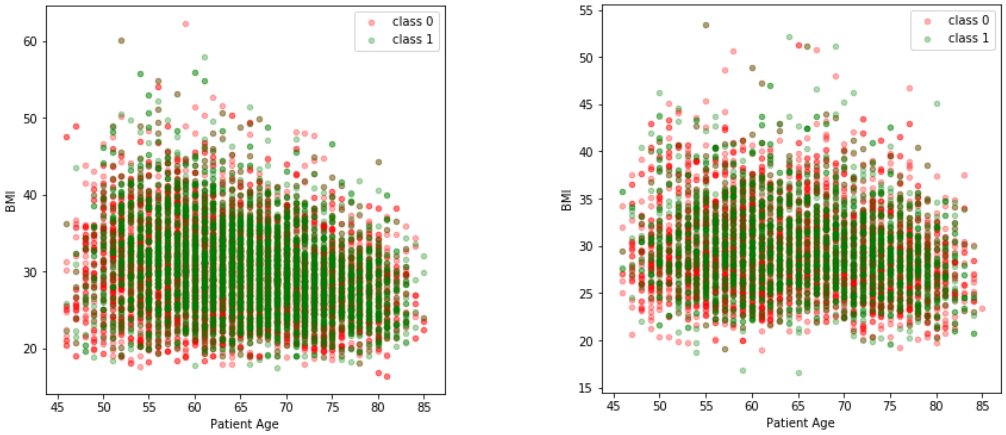


(a) Female **(b)** Male

Figure A.2: Age distribution for OAI, MOST and CHECK dataset, JSN 10 %. Blue corresponds to class 0, orange corresponds to 1.

		Number of images	percentage
Total		3 653	100 %
Class 0		1 930	52.83 %
	female	1 252	64.87 %
	male	678	35.13 %
Class 1		1 723	47.17%
	female	1 084	62.91 %
	male	639	37.09 %

Table A.3: Summary of the data with class definition of 10 % JSN per 1 or 2 years, exclusion criteria of remaining KL 0 or remaining KL 1 from MOST



(a) Female **(b)** Male

Figure A.3: Age and BMI distribution for OAI, MOST and CHECK dataset, JSN 10 %. Red dots correspond to class 0, green dots correspond to class 1.

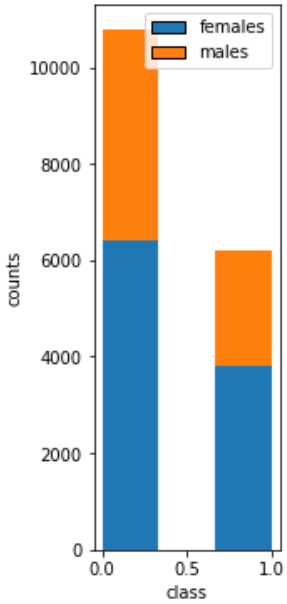
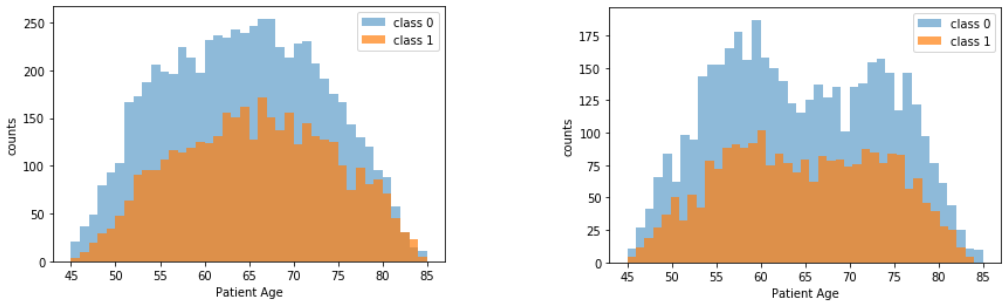
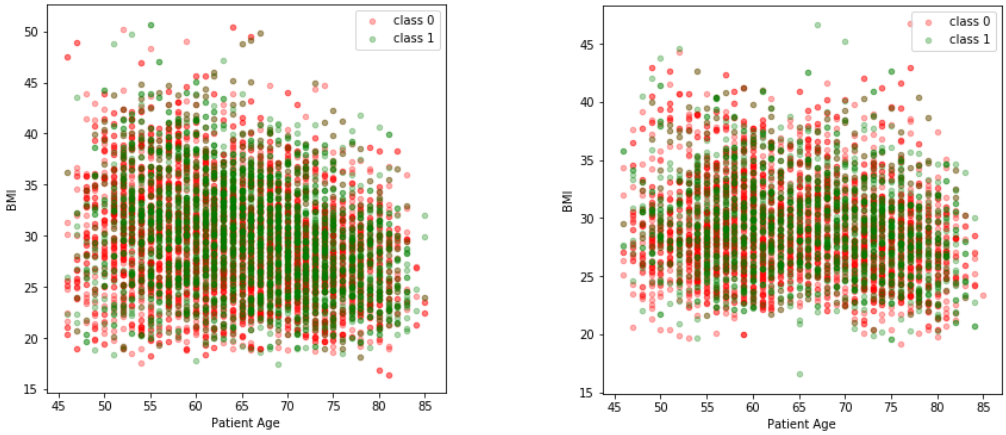


Figure A.4: Class distribution (0 and 1) of OAI datasets, JSN of 10 %. Blue: females, orange: males



(a) Female **(b)** Male

Figure A.5: Age distribution for OAI, JSN 10%. Blue corresponds to class 0, orange corresponds to 1.



(a) Female **(b)** Male

Figure A.6: Age and BMI distribution for OAI, JSN 10 %. Red dots correspond to class 0, green dots correspond to class 1.

		Number of images	percentage
Total		498	100 %
Class 0		280	56.22 %
	female	226	80.71 %
	male	54	19.29 %
Class 1		218	43.78%
	female	171	78.44 %
	male	46	21.56 %

Table A.4: Summary of the data with class definition of 10 % JSN per 1 or 2 years, exclusion criteria of remaining KL 0 or remaining KL 1 from CHECK

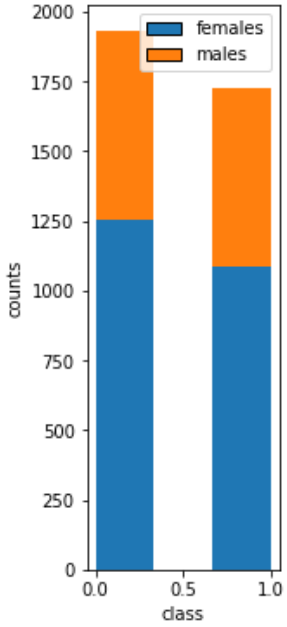
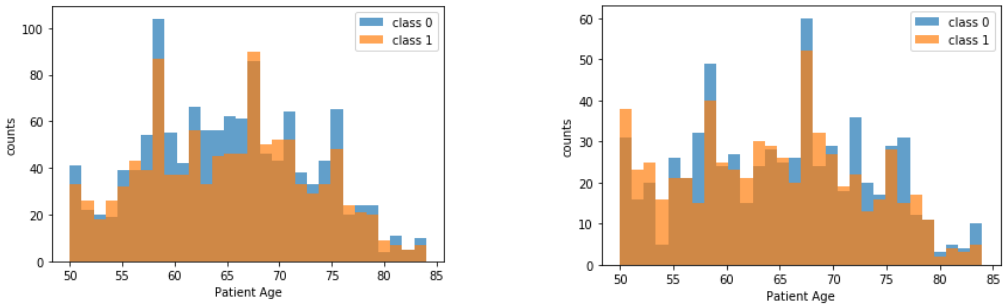


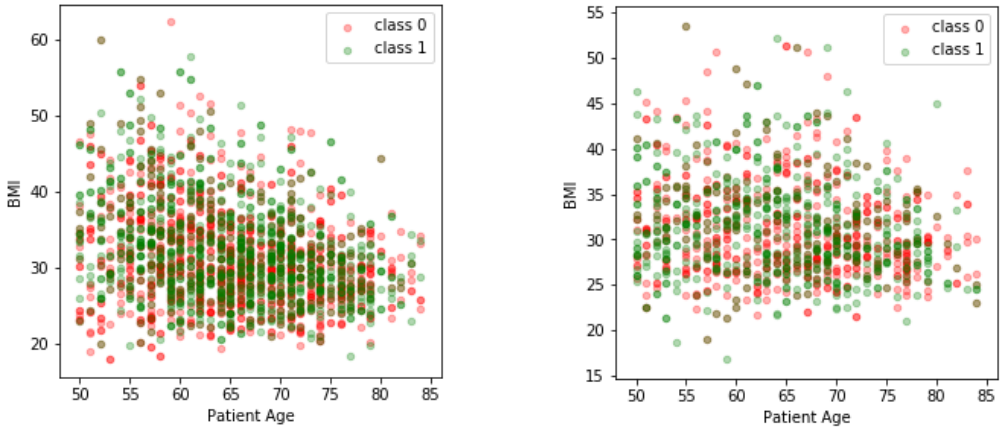
Figure A.7: Class distribution (0 and 1) of MOST datasets, JSN of 10 %. Blue: females, orange: males



(a) Female

(b) Male

Figure A.8: Age distribution of MOST dataset, JSN 10%. Blue corresponds to class 0, orange corresponds to 1.



(a) Female **(b)** Male

Figure A.9: Age and BMI distribution of MOST dataset, JSN 10 %. Red dots correspond to class 0, green dots correspond to class 1.

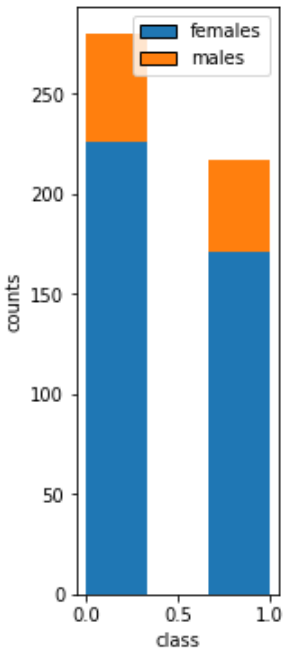
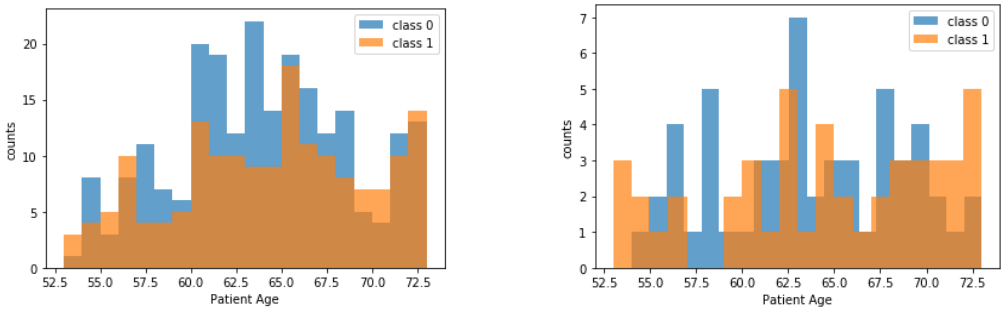
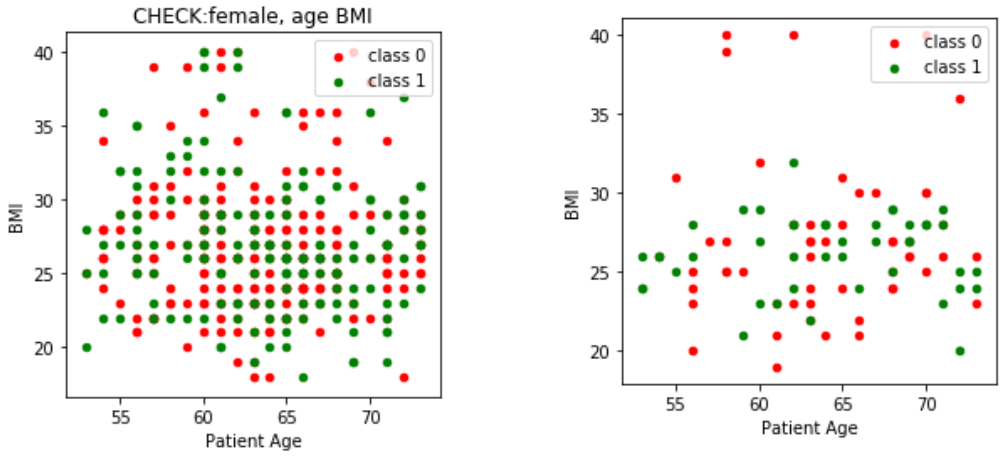


Figure A.10: Class distribution (0 and 1) of CHECK datasets, JSN of 10 %. Blue: females, orange: males



(a) Female **(b)** Male

Figure A.11: Age distribution of CHECK dataset, JSN 10%. Blue corresponds to class 0, orange corresponds to 1.

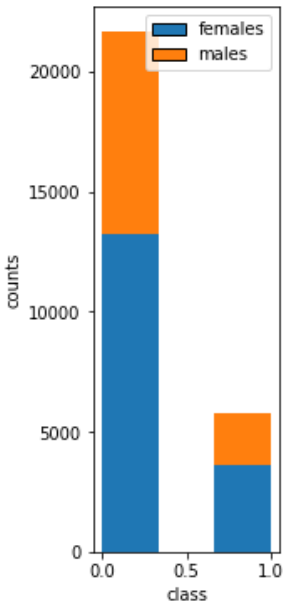


(a) Female **(b)** Male

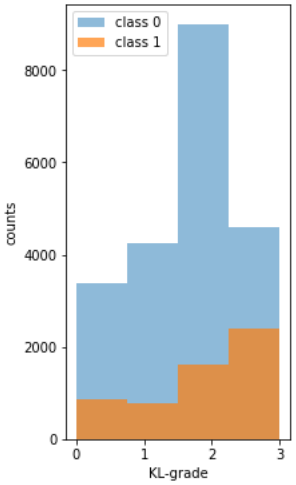
Figure A.12: Age and BMI distribution of CHECK dataset, JSN 10 %. Red dots correspond to class 0, green dots correspond to class 1.

		Number of images	percentage
Total		27 432	100 %
Class 0		21 658	78.95 %
	female	13 234	61.10 %
	male	8 423	38.90 %
Class 1		5 774	21.05 %
	female	3 579	61.98 %
	male	2 195	38.02 %

Table A.5: Summary of the data with class definition of 20 % JSN per 1 or 2 years, exclusion criteria of remaining KL 0 or remaining KL 1 from OAI, MOST and CHECK

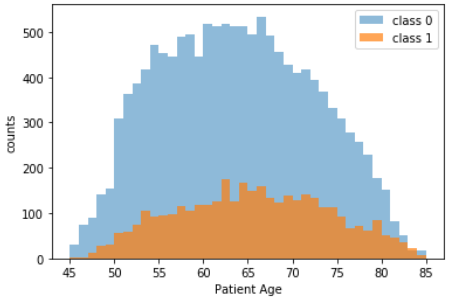


(a) Female

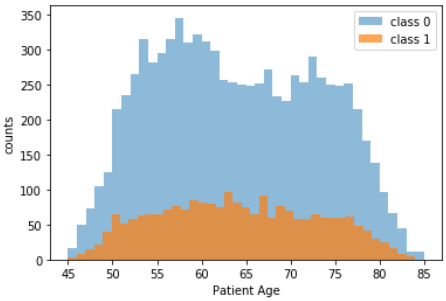


(b) Male

Figure A.13: (a) Age distribution of OAI, MOST and CHECK dataset, JSN 20 %. Blue corresponds to class 0, orange corresponds to 1. (b) KL-grade distribution of OAI, MOST and CHECK dataset, JSN 20 %. Blue corresponds to class 0, orange corresponds to 1.

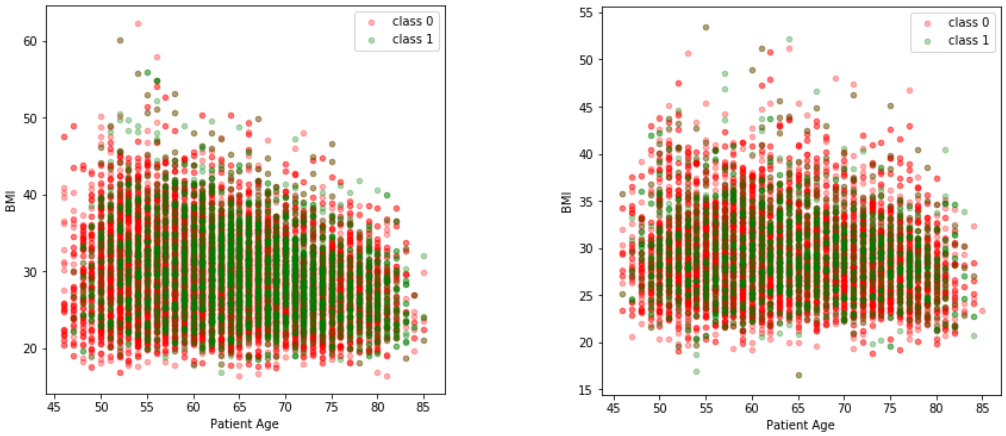


(a) Female



(b) Male

Figure A.14: Age distribution of OAI, MOST and CHECK dataset, JSN 20 %. Blue corresponds to class 0, orange corresponds to 1.



(a) Female **(b)** Male

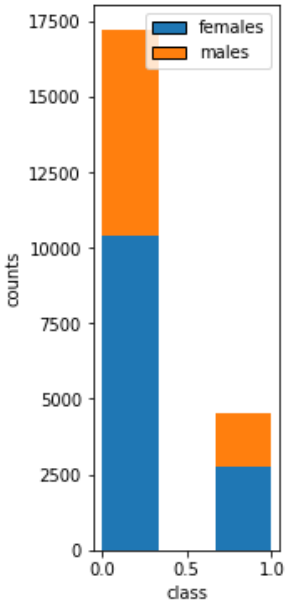
Figure A.15: Age and BMI distribution of OAI, MOST and CHECK dataset, JSN 20 %. Red dots correspond to class 0, green dots correspond to class 1.

		Number of images	percentage
Total		21 698	100 %
Class 0		17 196	79.25 %
	female	10 387	60.40 %
	male	6 809	39.60 %
Class 1		4 502	20.75 %
	female	2 734	60.73 %
	male	1 768	39.27 %

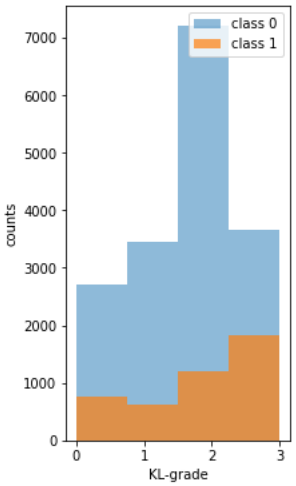
Table A.6: Summary of the data with class definition of 20 % JSN per 1 or 2 years, exclusion criteria of remaining KL 0 or remaining KL 1 from OAI

		Number of images	percentage
Total		5 259	100 %
Class 0		4 091	77.79 %
	female	2 539	62.06 %
	male	1 552	37.94 %
Class 1		218	22.21%
	female	761	65.15 %
	male	407	34.85 %

Table A.7: Summary of the data with class definition of 20 % JSN per 1 or 2 years, exclusion criteria of remaining KL 0 or remaining KL 1 from MOST

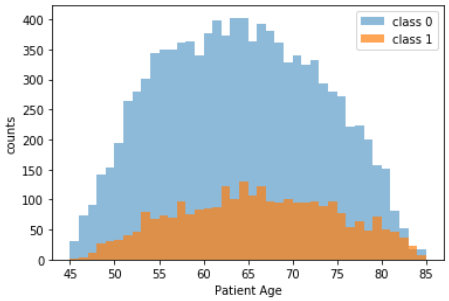


(a) Female

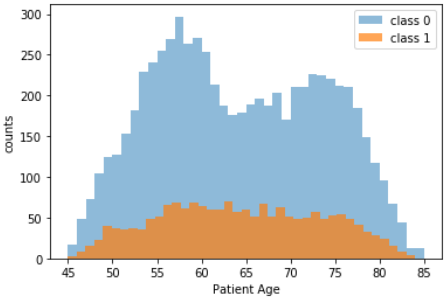


(b) Male

Figure A.16: (a) Age distribution of OAI dataset, JSN 20 %. Blue corresponds to class 0, orange corresponds to 1. (b) KL-grade distribution of OAI dataset, JSN 20 %. Blue corresponds to class 0, orange corresponds to 1.

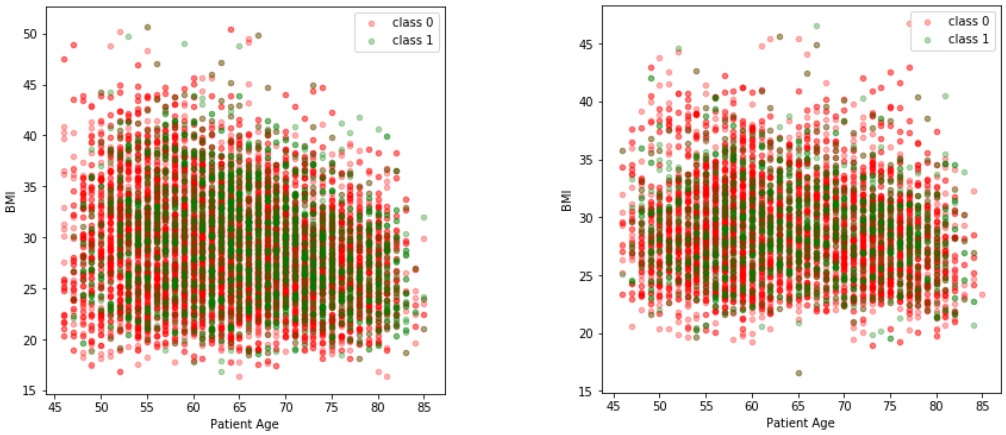


(a) Female



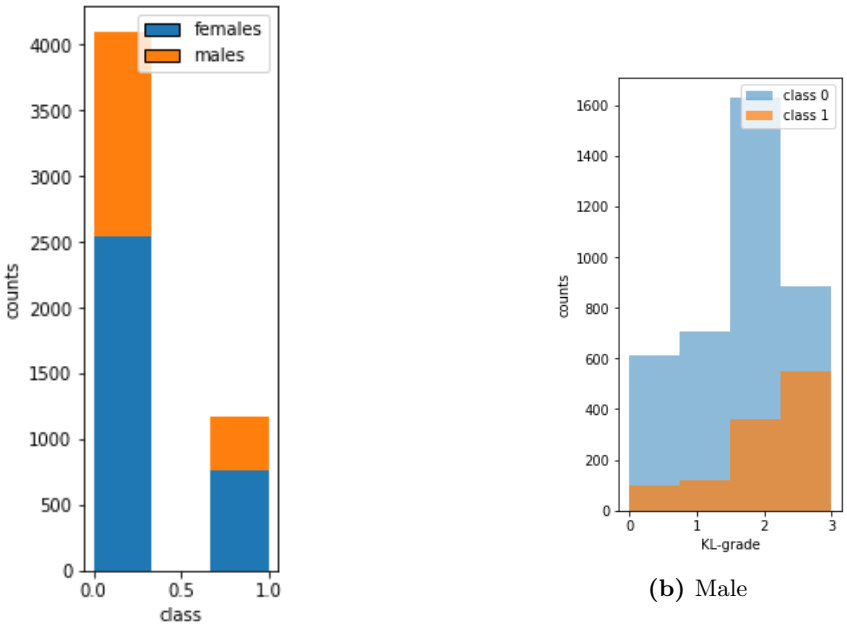
(b) Male

Figure A.17: Age distribution of OAI dataset, JSN 20 %. Blue corresponds to class 0, orange corresponds to 1.



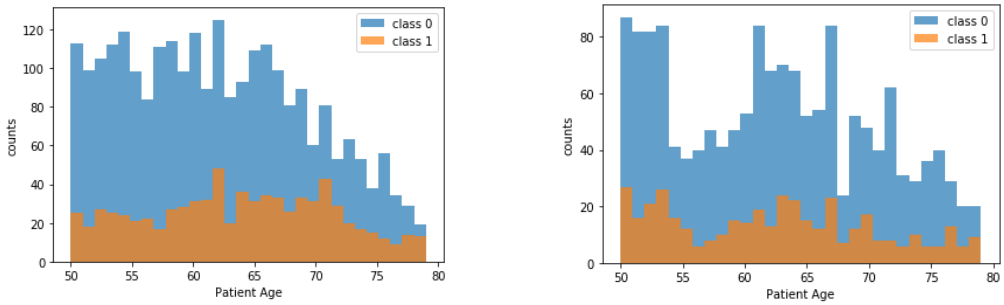
(a) Female **(b)** Male

Figure A.18: Age and BMI distribution of OAI dataset, JSN 20 %. Red dots correspond to class 0, green dots correspond to class 1.



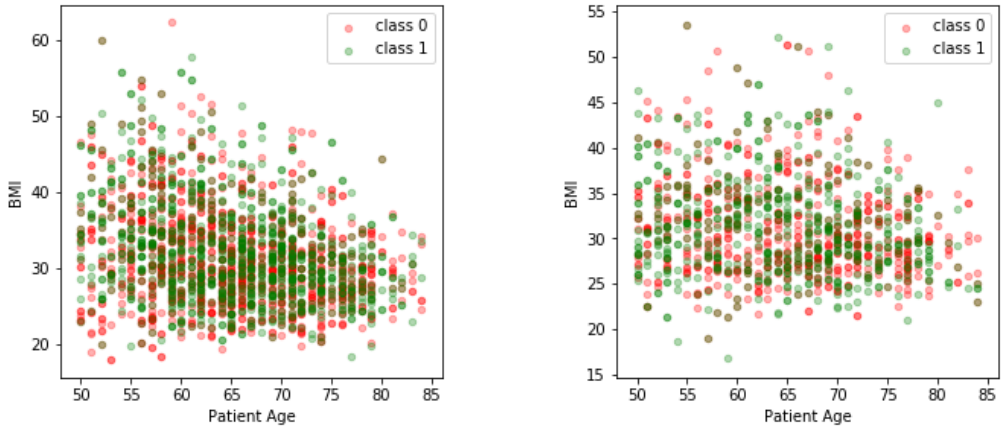
(a) Female **(b)** Male

Figure A.19: (a) Age distribution of MOST dataset, JSN 20 %. Blue corresponds to class 0, orange corresponds to 1. (b) KL-grade distribution of MOST dataset, JSN 20 %. Blue corresponds to class 0, orange corresponds to 1.



(a) Female **(b)** Male

Figure A.20: Age distribution of MOST dataset, JSN 20 %. Blue corresponds to class 0, orange corresponds to 1.

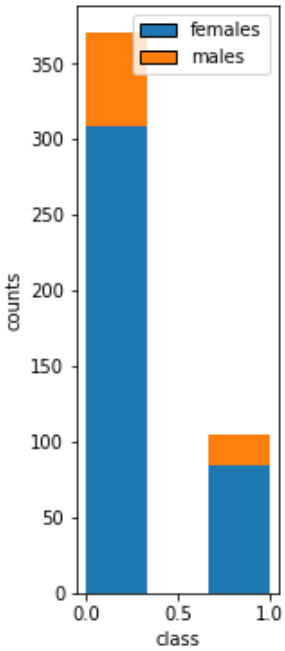


(a) Female **(b)** Male

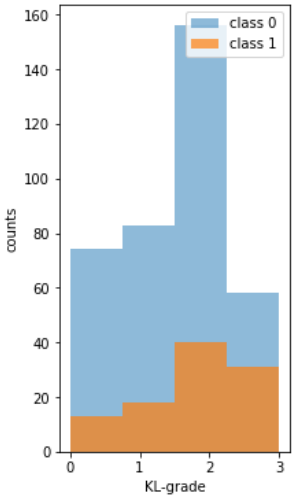
Figure A.21: Age and BMI distribution of MOST dataset, JSN 20 %. Red dots correspond to class 0, green dots correspond to class 1.

		Number of images	percentage
Total		475	100 %
Class 0		371	78.11 %
	female	308	83.02 %
	male	62	16.98 %
Class 1		104	21.89%
	female	84	80.77 %
	male	20	19.23 %

Table A.8: Summary of the data with class definition of 20 % JSN per 1 or 2 years, exclusion criteria of remaining KL 0 or remaining KL 1 from CHECK

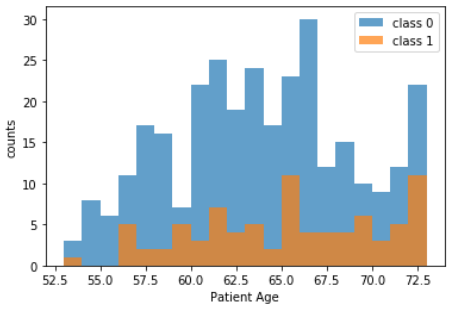


(a) Female

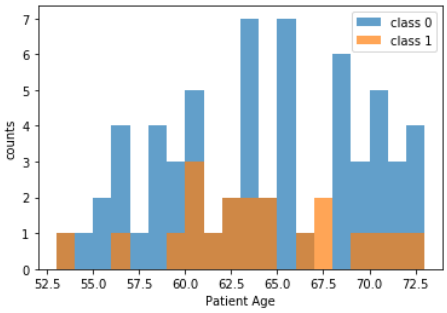


(b) Male

Figure A.22: (a) Age distribution of CHECK dataset, JSN 20 %. Blue corresponds to class 0, orange corresponds to 1. (b) KL-grade distribution of CHECK dataset, JSN 20 %. Blue corresponds to class 0, orange corresponds to 1.

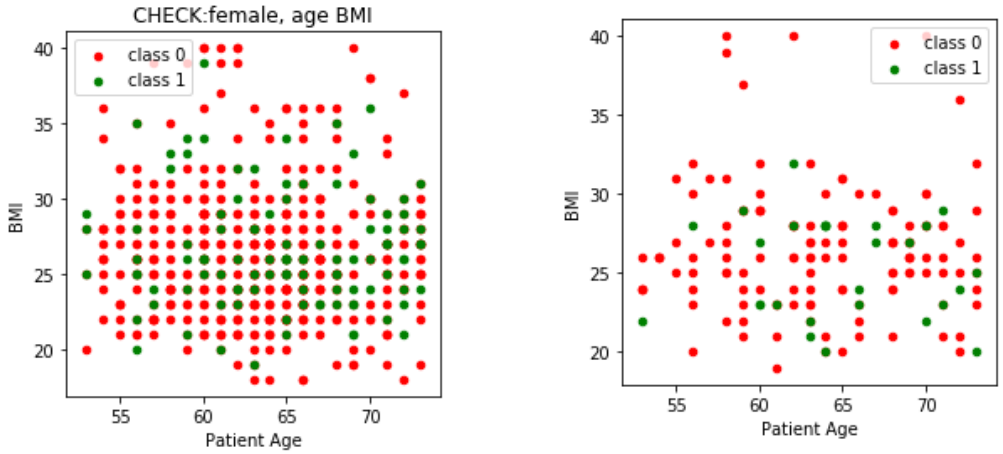


(a) Female



(b) Male

Figure A.23: Age distribution of CHECK dataset, JSN 20 %. Blue corresponds to class 0, orange corresponds to 1.



(a) Female

(b) Male

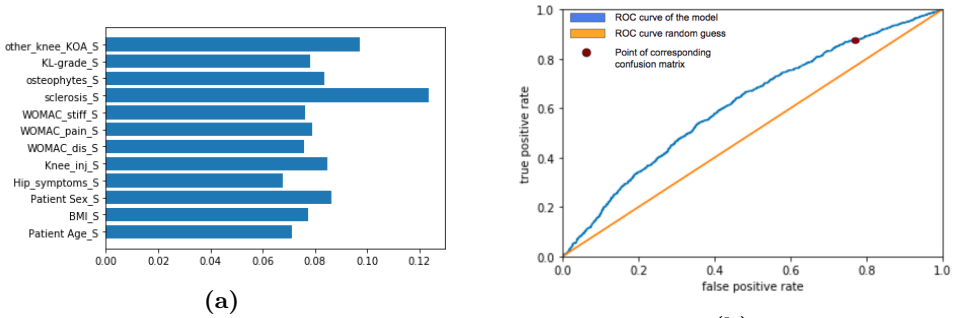
Figure A.24: Age and BMI distribution of CHECK dataset, JSN 20 %. Red dots correspond to class 0, green dots correspond to class 1.

	Number of images
right knee images	11 071
left knee images	10 068
medial slow progressors	13 313
lateral slow progressors	14 337
medial fast progressors	5 364
lateral fast progressors	3 846
whole knee slow progressors	**12 985**
whole knee fast progressors	**8 153**

Table A.9: Summary of the training data

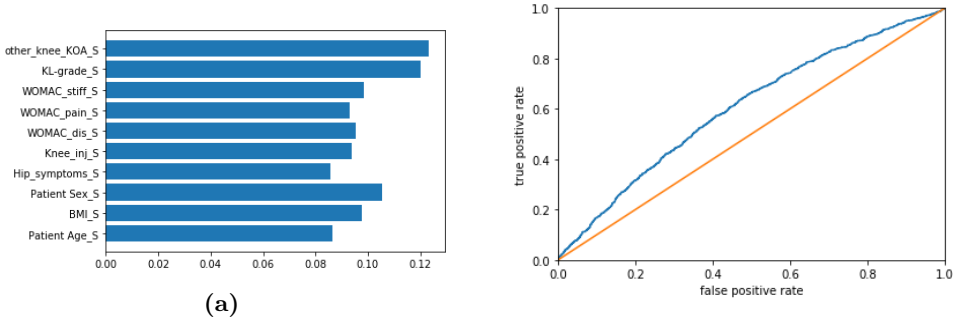
APPENDIX B

XGBoost Results



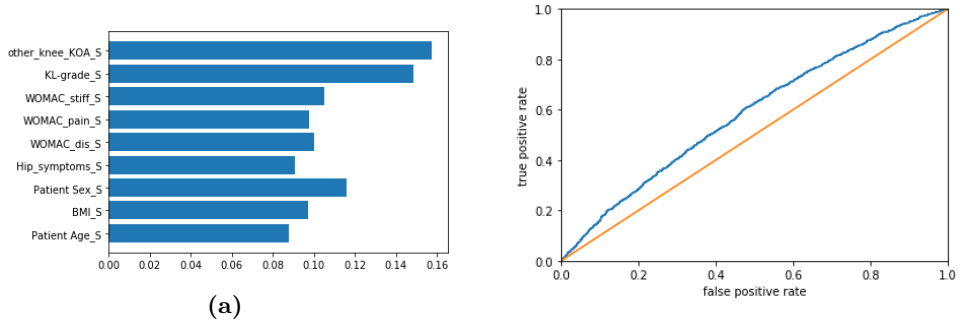
(a) **(b)**

Figure B.1: (a) Most important features of Run A for the XGBoost model. (b) ROC curve of Run A (blue curve). X-axis represents the FP rate, y-axis the TP rate. The orange curve represents a random guess.



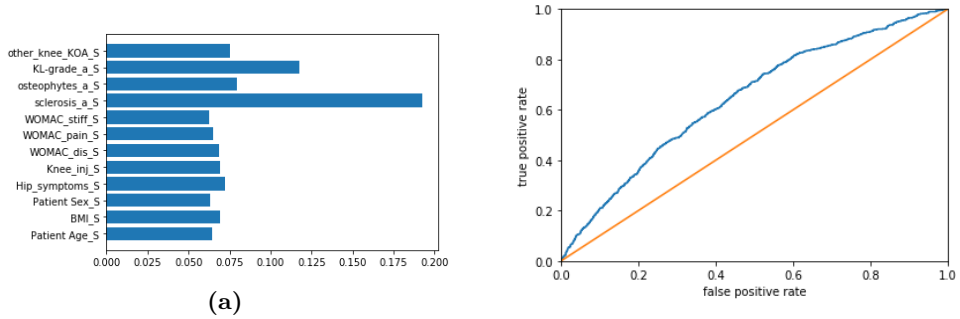
(a) **(b)**

Figure B.2: (a) Most important features of Run B for the XGBoost model. (b) ROC curve of Run B (blue curve). X-axis represents the FP rate, y-axis the TP rate. The orange curve represents a random guess.



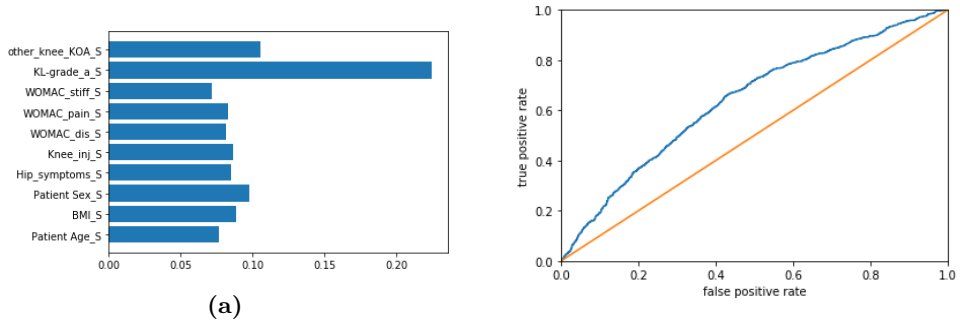
(a) **(b)**

Figure B.3: (a) Most important features of Run C for the XGBoost model. (b) ROC curve of Run C (blue curve). X-axis represents the FP rate, y-axis the TP rate. The orange curve represents a random guess.



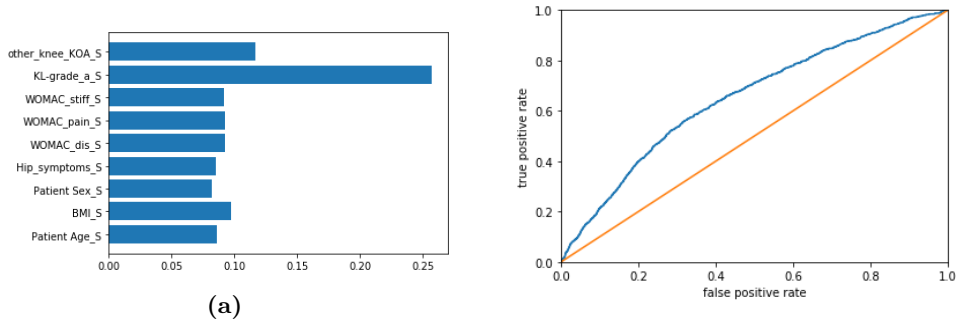
(a) **(b)**

Figure B.4: (a) Most important features of Run D for the XGBoost model. (b) ROC curve of Run D (blue curve). X-axis represents the FP rate, y-axis the TP rate. The orange curve represents a random guess.



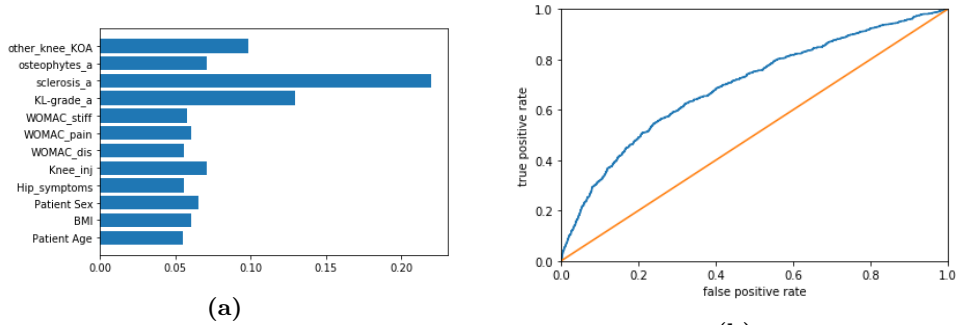
(a) **(b)**

Figure B.5: (a) Most important features of Run E for the XGBoost model. (b) ROC curve of Run E (blue curve). X-axis represents the FP rate, y-axis the TP rate. The orange curve represents a random guess.



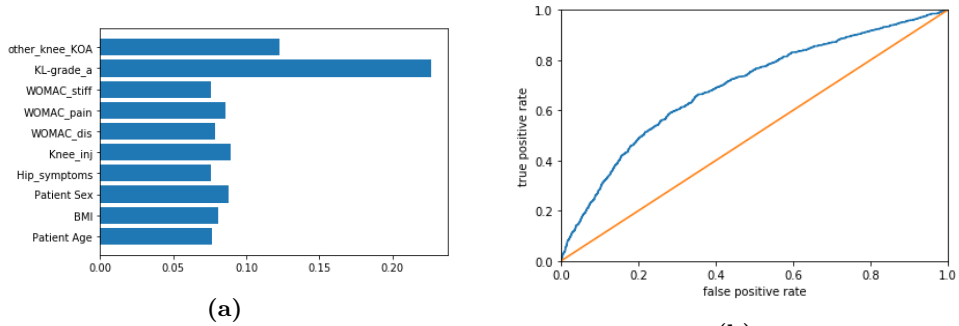
(a) **(b)**

Figure B.6: (a) Most important features of Run F for the XGBoost model. (b) ROC curve of Run F (blue curve). X-axis represents the FP rate, y-axis the TP rate. The orange curve represents a random guess.



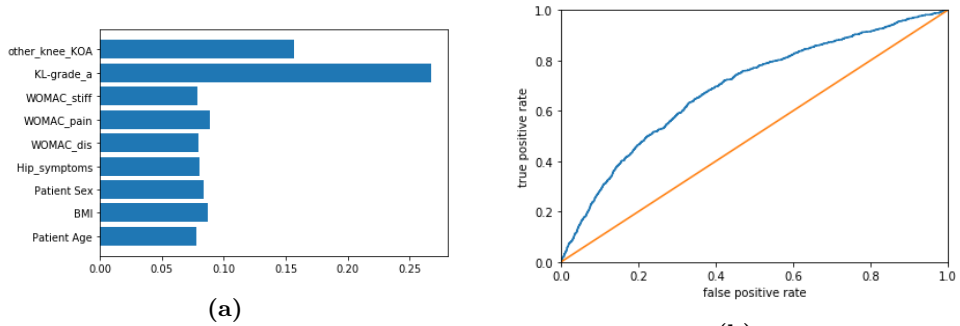
(a) **(b)**

Figure B.7: (a) Most important features of Run J for the XGBoost model. (b) ROC curve of Run J (blue curve). X-axis represents the FP rate, y-axis the TP rate. The orange curve represents a random guess.



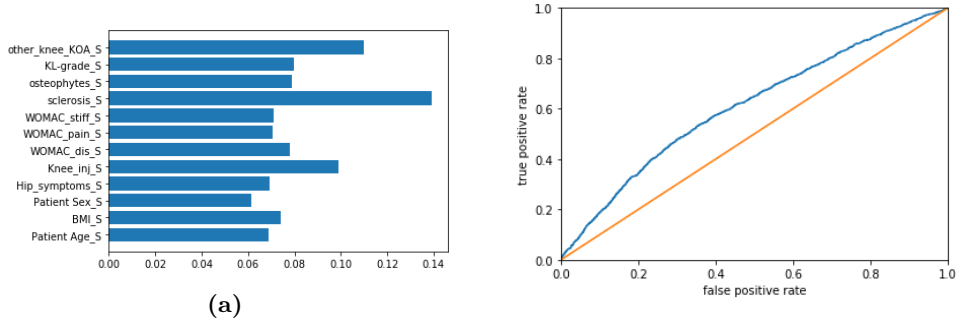
(a) **(b)**

Figure B.8: (a) Most important features of Run K for the XGBoost model. (b) ROC curve of Run K (blue curve). X-axis represents the FP rate, y-axis the TP rate. The orange curve represents a random guess.



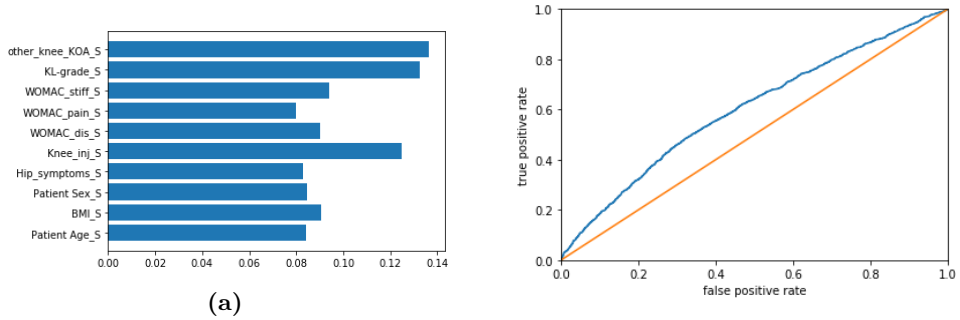
(a) **(b)**

Figure B.9: (a) Most important features of Run L for the XGBoost model. (b) ROC curve of Run L (blue curve). X-axis represents the FP rate, y-axis the TP rate. The orange curve represents a random guess.



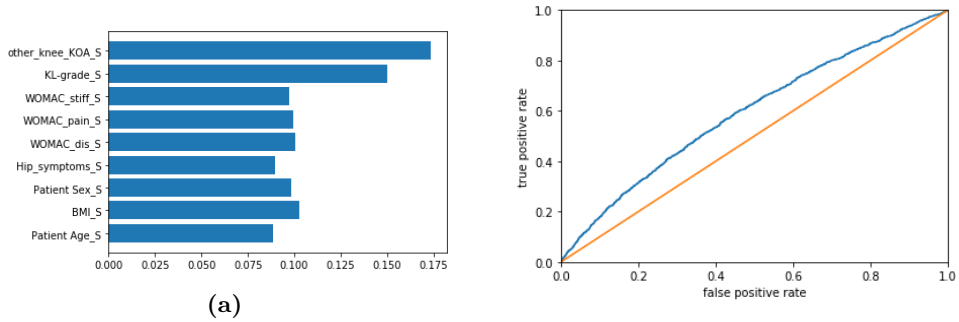
(a) **(b)**

Figure B.10: (a) Most important features of Run M for the XGBoost model. (b) ROC curve of Run M (blue curve). X-axis represents the FP rate, y-axis the TP rate. The orange curve represents a random guess.



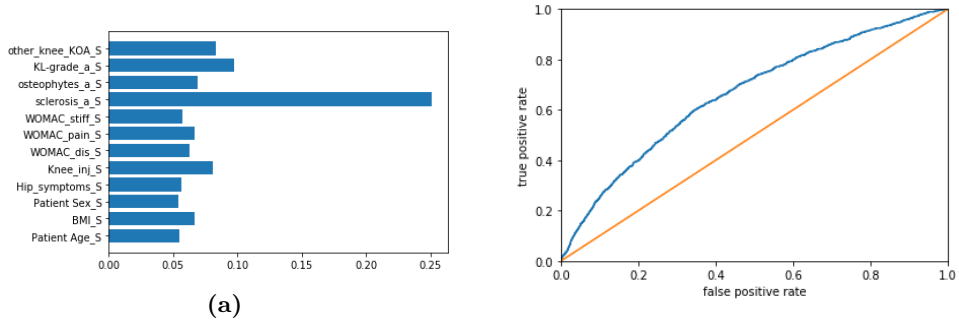
(a) **(b)**

Figure B.11: (a) Most important features of Run N for the XGBoost model. (b) ROC curve of Run N (blue curve). X-axis represents the FP rate, y-axis the TP rate. The orange curve represents a random guess.



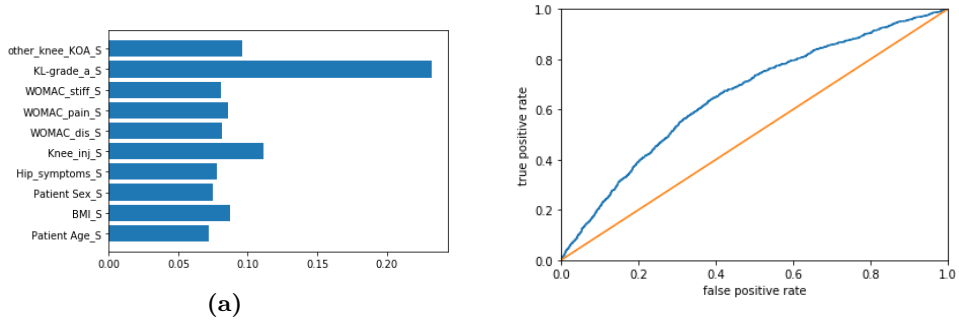
(a) **(b)**

Figure B.12: (a) Most important features of Run O for the XGBoost model. (b) ROC curve of Run O (blue curve). X-axis represents the FP rate, y-axis the TP rate. The orange curve represents a random guess.



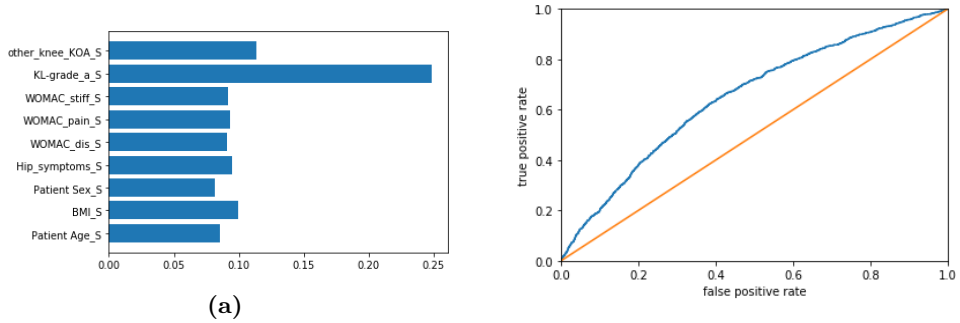
(a) **(b)**

Figure B.13: (a) Most important features of Run P for the XGBoost model. (b) ROC curve of Run P (blue curve). X-axis represents the FP rate, y-axis the TP rate. The orange curve represents a random guess.



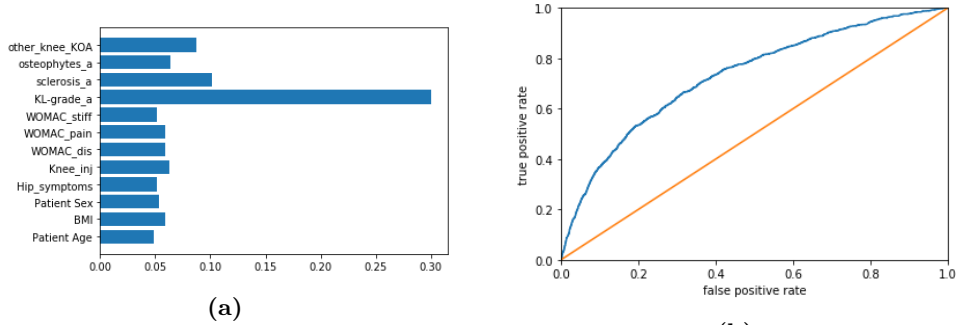
(a) **(b)**

Figure B.14: (a) Most important features of Run Q for the XGBoost model. (b) ROC curve of Run Q (blue curve). X-axis represents the FP rate, y-axis the TP rate. The orange curve represents a random guess.



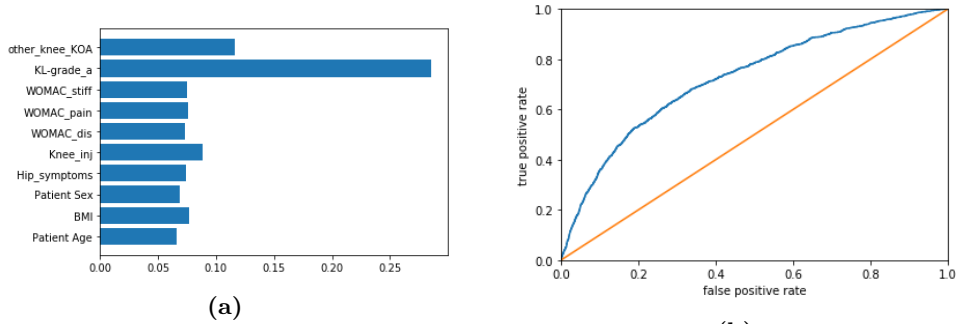
(a) **(b)**

Figure B.15: (a) Most important features of Run R for the XGBoost model. (b) ROC curve of Run R (blue curve). X-axis represents the FP rate, y-axis the TP rate. The orange curve represents a random guess.



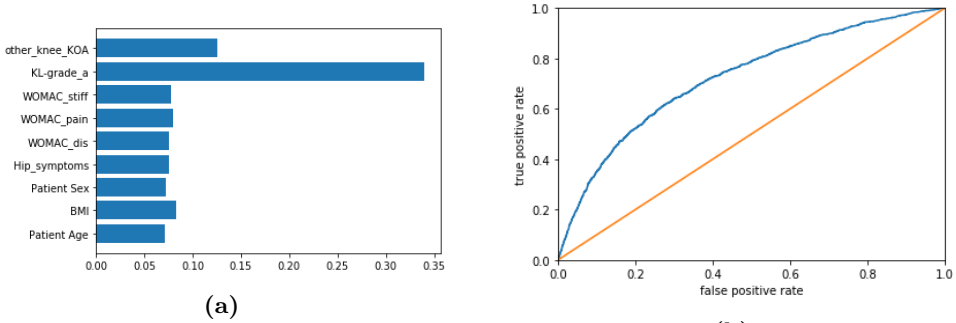
(a) **(b)**

Figure B.16: (a) Most important features of Run S for the XGBoost model. (b) ROC curve of Run S (blue curve). X-axis represents the FP rate, y-axis the TP rate. The orange curve represents a random guess.



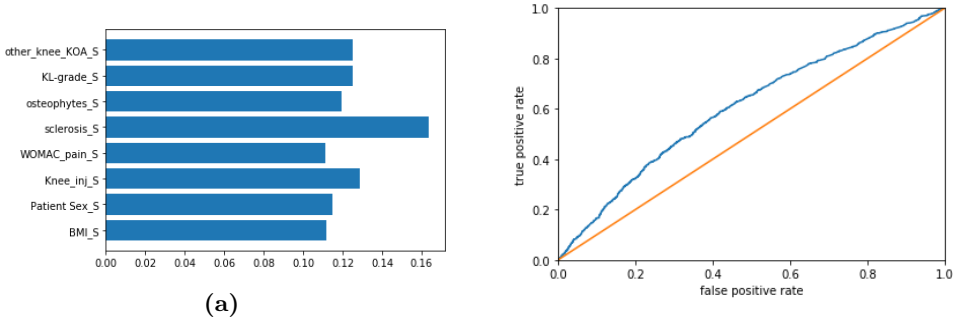
(a) **(b)**

Figure B.17: (a) Most important features of Run T for the XGBoost model. (b) ROC curve of Run T (blue curve). X-axis represents the FP rate, y-axis the TP rate. The orange curve represents a random guess.



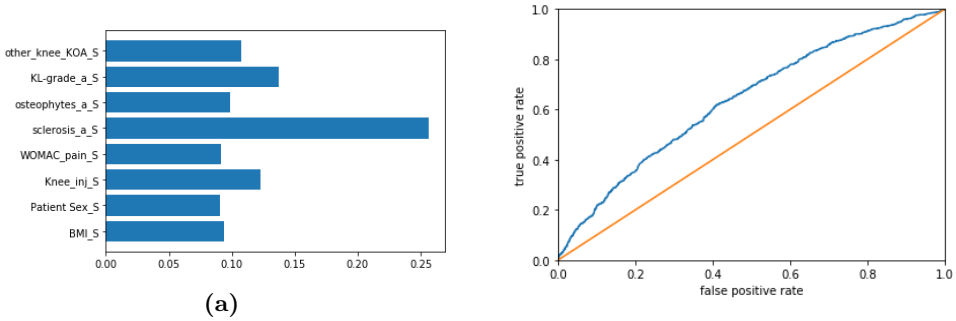
(a) **(b)**

Figure B.18: (a) Most important features of Run U for the XGBoost model. (b) ROC curve of Run U (blue curve). X-axis represents the FP rate, y-axis the TP rate. The orange curve represents a random guess.



(a) **(b)**

Figure B.19: (a) Most important features of Run V for the XGBoost model. (b) ROC curve of Run V (blue curve). X-axis represents the FP rate, y-axis the TP rate. The orange curve represents a random guess.



(a) **(b)**

Figure B.20: (a) Most important features of Run W for the XGBoost model. (b) ROC curve of Run W (blue curve). X-axis represents the FP rate, y-axis the TP rate. The orange curve represents a random guess.

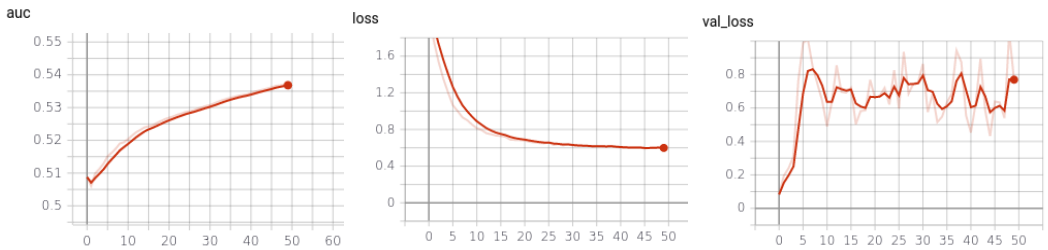
APPENDIX C

CNN Results

bn5c_branch2b (BatchNormalizati	(None, 32, 16, 512)	2048	res5c_branch2b[0][0]
res5c_branch2b_relu (Activation	(None, 32, 16, 512)	0	bn5c_branch2b[0][0]
res5c_branch2c (Conv2D)	(None, 32, 16, 2048)	1048576	res5c_branch2b_relu[0][0]
bn5c_branch2c (BatchNormalizati	(None, 32, 16, 2048)	8192	res5c_branch2c[0][0]
res5c (Add)	(None, 32, 16, 2048)	0	bn5c_branch2c[0][0] res5b_relu[0][0]
res5c_relu (Activation)	(None, 32, 16, 2048)	0	res5c[0][0]
Dropout_0.3_0 (Dropout)	(None, 32, 16, 2048)	0	res5c_relu[0][0]
Final_conv_0 (Conv2D)	(None, 32, 16, 512)	1049088	Dropout_0.3_0[0][0]
MaxPool_0 (GlobalMaxPooling2D)	(None, 512)	0	Final_conv_0[0][0]
numeric (InputLayer)	(None, 4)	0	
dense_5 (Dense)	(None, 2048)	1050624	MaxPool_0[0][0]
dense_3 (Dense)	(None, 128)	640	numeric[0][0]
leaky_re_lu_1 (LeakyReLU)	(None, 2048)	0	dense_5[0][0]
dropout_1 (Dropout)	(None, 128)	0	dense_3[0][0]
dropout_2 (Dropout)	(None, 2048)	0	leaky_re_lu_1[0][0]
concatenate_1 (Concatenate)	(None, 2176)	0	dropout_1[0][0] dropout_2[0][0]
dense_6 (Dense)	(None, 1024)	2229248	concatenate_1[0][0]
leaky_re_lu_2 (LeakyReLU)	(None, 1024)	0	dense_6[0][0]
dropout_3 (Dropout)	(None, 1024)	0	leaky_re_lu_2[0][0]
jsn (Dense)	(None, 1)	1025	dropout_3[0][0]

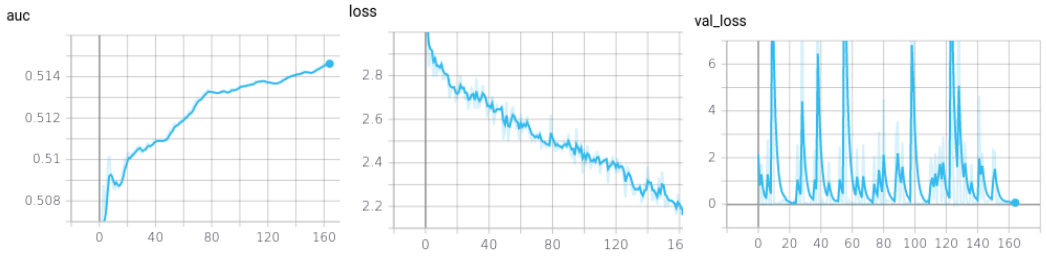
Total params: 27,885,505
Trainable params: 27,832,385
Non-trainable params: 53,120

Figure C.1: Model summary of Model 4.



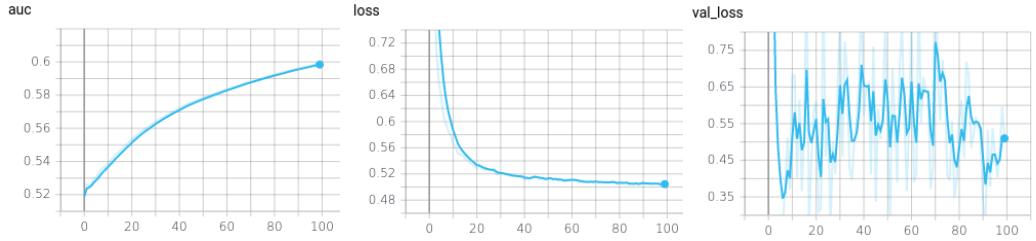
(a) **(b)** **(c)**

Figure C.2: Results of the CNN model of Run 6. (a) AUC (b) loss-function (c) validation loss function.



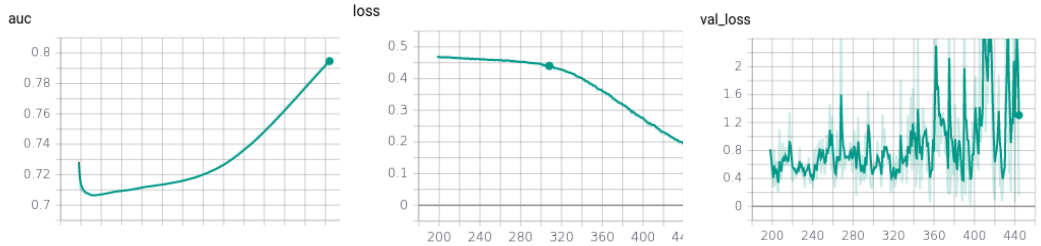
(a) **(b)** **(c)**

Figure C.3: Results of the CNN model of Run 7. (a) AUC (b) loss-function (c) validation loss function.



(a) **(b)** **(c)**

Figure C.4: Results of the CNN model of Run 9. (a) AUC (b) loss-function (c) validation loss function.

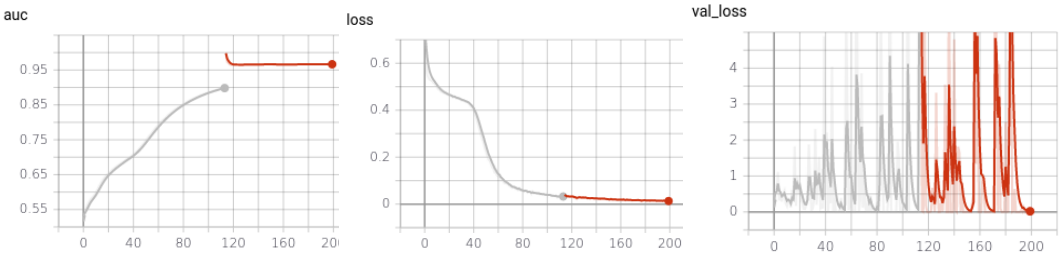


(a) **(b)** **(c)**

Figure C.5: Results of the CNN model of Run 10. (a) AUC (b) loss-function (c) validation loss function.

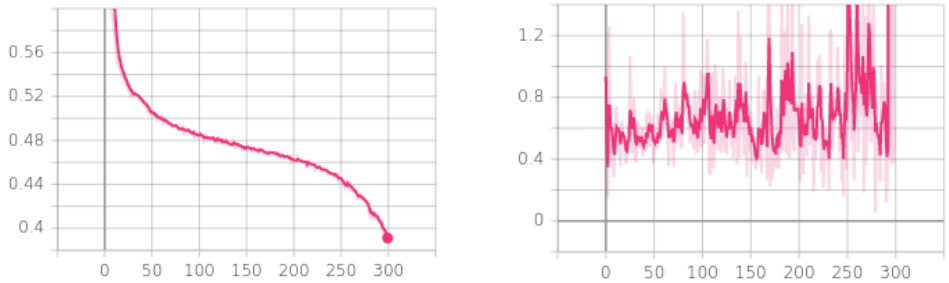
(a) **(b)** **(c)**

Figure C.6: Results of the CNN model of Run 11. (a) AUC (b) loss-function (c) validation loss function.



(a) **(b)** **(c)**

Figure C.7: Results of the CNN model of Run 12. (a) AUC (b) loss-function (c) validation loss function.



(a) **(b)**

Figure C.8: Results of the CNN model of Run 16. (a) loss-function (b) validation loss function.

List of Figures

List of Tables

Acronyms

ADAM Adaptive Moment Estimation. 13, 56

AKOA Accelerated Knee Osteoarthritis. ix–xiii, 1–3, 5, 7–10, 12, 20, 25–27, 29, 31, 33, 35, 37, 38, 44, 50, 51, 56, 57, 63, 64, 66, 69, 70, 72, 77, 79, 80, 82, 85–88

AUC Area Under the Curve. ix, xi, xiii, 3, 13, 22, 23, 26, 27, 52, 56, 69, 72, 74–80, 82, 83, 86, 87, 123

BMI Body Mass Index. ix–xi, 2, 7, 27, 28, 35, 37, 52, 63, 65–68, 72–76, 79, 82, 83, 85–87, 117

CHECK Cohort Hip and Cohort Knee Study. 3, 34, 35, 38, 57, 58, 63–69, 79, 82, 83, 85, 117, 123

CLAHE Contrast Limited Adaptive Histogram Equalisation. 47, 48, 59, 117

CNN Convolutional Neural Network. ix, xi, xiii, xiv, 2, 3, 11–19, 27, 30, 31, 33, 34, 37, 47, 50, 54–56, 58, 61, 63, 77, 79–82, 85, 87, 115–119, 123

FN false negative. 23, 24, 52

FP false positive. 22–24, 52

GBM Gradient Boosting Machine. 26, 27, 30, 116

IBLab Image Biopsy Lab. 3, 18, 33, 35–37, 54, 60, 116, 117

JSN Joint Space Narrowing. ix, xi, 1–3, 5, 7–10, 26, 27, 31, 36–40, 49, 50, 63–67, 70, 77, 79, 80, 82, 85–87, 89, 102, 116, 117, 120, 123, 124

JSW Joint Space Width. 7, 10, 36–38, 40–42, 116

KL Kellgren-Lawrence. ix–xi, 3, 8–10, 25–27, 35–37, 40, 41, 43, 52, 63, 66, 68, 70–76, 79, 80, 82, 83, 85–87, 89, 90, 92, 95, 116, 117, 123, 124

KOA Knee Osteoarthritis. ix–xii, 1–3, 5, 7–10, 12, 23, 25–29, 33–35, 37, 40, 41, 50, 51, 54, 64, 69, 71, 72, 74–76, 79, 80, 82, 83, 85–87

KOALA Knee Osteoarthritis Labeling Assistent. 36, 40–44, 52, 116, 117

MOST Multicenter Osteoarthritis Study. 3, 27, 34, 35, 38, 57, 58, 63–68, 79, 82, 83, 85, 89, 91, 117, 119, 123, 124

MRI Magnetic Resonance Image. 44

NaN Not a Number. 40, 42, 56, 69

OA Osteoarthritis. ix, xi, 1, 5–9, 34, 42, 115

OAI Osteoarthritis Initiative. 1, 3, 26–28, 34, 35, 38, 39, 41, 43, 45, 57–59, 63–69, 79, 82, 83, 85, 91, 116, 117, 119, 123

OARSI Osteoarthritis Research Society International. ix, xi, 9, 27, 36, 37, 52, 71, 72, 82, 85, 116

ReLU Rectified Linear Unit. 12, 15, 16, 19, 20, 54, 55, 82, 86, 115, 116

ResNet Residual Network. ix, xi, xiii, 18, 19, 21, 54, 56, 57, 60, 86, 116, 117

ROC Receiver Operating Characteristic. ix, xiii, 3, 22, 23, 52, 69–71, 77, 80–82, 116, 118, 119

tanh hyperbolic tangent. 15, 16, 115

TN true negative. 23, 24, 52, 74, 123

TP true positive. 22, 23, 52, 74, 123

WOMAC Western Ontario and McMaster Universities Arthritis Index. ix, xi, 9, 25, 27, 35, 36, 47, 52, 53, 72, 74, 75, 79, 85

XGBoost Extreme Gradient Boosting. ix, xi, xiii, xiv, 3, 11, 20, 21, 33, 34, 50–53, 57, 60, 63, 69–77, 79, 81, 82, 85, 86, 117, 118, 123

Bibliography

[1] Flexikon. *https://flexikon.doccheck.com/de/Osteoarthritis, accessed: 2021-09-13.*

[2] Sion Glyn-Jones et al. "Osteoarthritis". In: *The Lancet* 386.9991 (2015), pp. 376–387.

[3] WebMD. *https://www.webmd.com/osteoarthritis/ostearthritis-of-the-knee-degenerative-arthritis-of-the-knee, accessed: 2021-08-29.*

[4] Aiyong Cui et al. "Global, regional prevalence, incidence and risk factors of knee osteoarthritis in population-based studies". In: *EClinicalMedicine* 29 (2020), p. 100587.

[5] Afshin Jamshidi et al. "Identification of the most important features of knee osteoarthritis structural progressors using machine learning methods". In: *Therapeutic advances in musculoskeletal disease* 12 (2020), p. 1759720X20933468.

[6] Jeffrey B Driban et al. "Risk factors can classify individuals who develop accelerated knee osteoarthritis: data from the osteoarthritis initiative". In: *Journal of Orthopaedic Research®* 36.3 (2018), pp. 876–880.

[7] Matthew S Harkey et al. "Early pre-radiographic structural pathology precedes the onset of accelerated knee osteoarthritis". In: *BMC musculoskeletal disorders* 20.1 (2019), pp. 1–10.

[8] Matthew S Harkey et al. "Diffuse tibiofemoral cartilage change prior to the development of accelerated knee osteoarthritis: data from the osteoarthritis initiative". In: *Clinical Anatomy* 32.3 (2019), pp. 369–378.

[9] Jeffrey B Driban et al. "The incidence and characteristics of accelerated knee osteoarthritis among women: the Chingford cohort". In: *BMC musculoskeletal disorders* 21.1 (2020), pp. 1–6.

[10] Erik B Dam et al. "Identification of progressors in osteoarthritis by combining biochemical and MRI-based markers". In: *Arthritis research & therapy* 11.4 (2009), pp. 1–11.

[11] Aleksei Tiulpin et al. "Multimodal machine learning-based knee osteoarthritis progression prediction from plain radiographs and clinical data". In: *Scientific reports* 9.1 (2019), pp. 1–11.

[12] MSD Manuals. *https://www.msdmanuals.com/de/heim/knochen-,-gelenk-und -muskelerkrankungen/gelenkerkrankungen/osteoarthrose-oa, accessed: 2021-09-07.*

[13] Jean-Pierre Raynauld et al. "Long term evaluation of disease progression through the quantitative magnetic resonance imaging of symptomatic knee osteoarthritis patients: correlation with clinical symptoms and radiographic changes". In: *Arthritis research & therapy* 8.1 (2005), pp. 1–12.

[14] Michelle J Lespasio et al. "Knee osteoarthritis: a primer". In: *The Permanente Journal* 21 (2017).

[15] Jeffrey B Driban et al. "Coronal tibial slope is associated with accelerated knee osteoarthritis: data from the Osteoarthritis Initiative". In: *BMC musculoskeletal disorders* 17.1 (2016), pp. 1–7.

[16] Wise-geek. *https://www.wise-geek.com/what-is-the-medial-meniscus.htm, accessed: 2021-11-15.*

[17] Hunter Hsu and Ryan M Siwiec. "Knee osteoarthritis". In: *StatPearls [Internet]* (2020).

[18] Tadashi Hayami et al. "The role of subchondral bone remodeling in osteoarthritis: reduction of cartilage degeneration and prevention of osteophyte formation by alendronate in the rat anterior cruciate ligament transection model". In: *Arthritis & Rheumatism: Official Journal of the American College of Rheumatology* 50.4 (2004), pp. 1193–1206.

[19] DT Felson et al. "Osteophytes and progression of knee osteoarthritis". In: *Rheumatology* 44.1 (2005), pp. 100–104.

[20] Web MD. *https://www.webmd.com/osteoarthritis/osteoarthritis-subchondral-sclerosis, accessed: 2021-10-13.*

[21] Spring Loaded. *https://springloadedtechnology.com/guide-to-severe-knee-osteoarthritis/, accessed: 2021-11-15.*

[22] Eni Halilaj et al. "Modeling and predicting osteoarthritis progression: data from the osteoarthritis initiative". In: *Osteoarthritis and cartilage* 26.12 (2018), pp. 1643–1650.

[23] Julie Davis et al. "Knee symptoms among adults at risk for accelerated knee osteoarthritis: data from the Osteoarthritis Initiative". In: *Clinical rheumatology* 36.5 (2017), p. 1083.

[24] Jeffrey B Driban et al. "Association of knee injuries with accelerated knee osteoarthritis progression: data from the Osteoarthritis Initiative". In: *Arthritis care & research* 66.11 (2014), pp. 1673–1679.

[25] Wikipedia. *https://en.wikipedia.org/wiki/Body_mass_index, accessed: 2021-12-14.*

[26] Radiopaedic. *https://radiopaedia.org/articles/kellgren-and-lawrence-system-for-classification-of-osteoarthritis?lang=us, accessed: 2021-09-13.*

[27] Aleksei Tiulpin and Simo Saarakkala. “Automatic grading of individual knee osteoarthritis features in plain radiographs using deep convolutional neural networks”. In: *Diagnostics* 10.11 (2020), p. 932.

[28] Physiopedia. *https://www.physio-pedia.com/WOMAC_Osteoarthritis_Index, accessed: 2021-10-03.*

[29] Jean Pierre Raynauld et al. “Quantitative magnetic resonance imaging evaluation of knee osteoarthritis progression over two years and correlation with clinical symptoms and radiologic changes”. In: *Arthritis & Rheumatism: Official Journal of the American College of Rheumatology* 50.2 (2004), pp. 476–487.

[30] Susan J Bartlett et al. “Identifying common trajectories of joint space narrowing over two years in knee osteoarthritis”. In: *Arthritis care & research* 63.12 (2011), pp. 1722–1728.

[31] T Neogi et al. “Identifying trajectories of medial joint-space width loss and associated risk factors”. In: *Osteoarthritis and Cartilage* 20 (2012), S182–S183.

[32] Bochen Guan et al. “Deep learning risk assessment models for predicting progression of radiographic medial joint space loss over a 48-MONTH follow-up period”. In: *Osteoarthritis and cartilage* 28.4 (2020), pp. 428–437.

[33] Jeffrey B Driban et al. “Association of knee injuries with accelerated knee osteoarthritis progression: data from the Osteoarthritis Initiative”. In: *Arthritis care & research* 66.11 (2014), pp. 1673–1679.

[34] Daniel Hain and Roman Jurowetzki. “Introduction to Rare-Event Predictive Modeling for Inferential Statisticians–A Hands-On Application in the Prediction of Breakthrough Patents”. In: *arXiv preprint arXiv:2003.13441* (2020).

[35] Towards Data Science. *https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53, accessed: 2021-10-13.*

[36] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. “Deep learning”. In: *nature* 521.7553 (2015), pp. 436–444.

[37] Peltarion. *https://peltarion.com/knowledge-center/documentation/modeling-view/build-an-ai-model/loss-functions/binary-crossentropy, accessed: 2021-11-30.*

[38] Keras. *https://keras.io/api/losses/probabilistic_losses/binary_crossentropy-function, accessed: 2021-11-30.*

[39] MachineLearningMastery. *https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/, accessed: 2021-11-21.*

[40] Sebastian Bock, Josef Goppold, and Martin Weiß. “An improvement of the convergence proof of the ADAM-Optimizer”. In: *arXiv preprint arXiv:1804.10587* (2018).

[41] Sebastian Ruder. “An overview of gradient descent optimization algorithms”. In: *arXiv preprint arXiv:1609.04747* (2016).

[42] Github. *https://github.com/Vercaca/NN-Backpropagation, accessed: 2021-12-12.*

[43] TowardsDataScience. *https://towardsdatascience.com/train-validation-and-test-sets-72cb40cba9e7, accessed: 2022-01-05.*

[44] Towards Data Science. *https://towardsdatascience.com/intuitively-understanding-convolutions-for-deep-learning-1f6f42faee1, accessed: 2021-10-08.*

[45] Medium. *https://medium.com/technologymadeeasy/the-best-explanation-of-convolutional-neural-networks-on-the-internet-fbb8b1ad5df8, accessed: 2021-10-14.*

[46] Towards Data Science. *https://towardsdatascience.com/convolutional-neural-networks-explained-9cc5188c4939, accessed: 2021-10-08.*

[47] Nvidia. *https://developer.nvidia.com/blog/deep-learning-nutshell-core-concepts/, accessed: 2021-10-14.*

[48] Medium. *https://medium.com/ml-cheat-sheet/understanding-non-linear-activation-functions-in-neural-networks-152f5e101eeb, accessed: 2021-10-17.*

[49] TowardDatascience. *https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6, accessed: 2021-11-21.*

[50] Medium. *https://medium.com/@danqing/a-practical-guide-to-relu-b83ca804f1f7, accessed: 2021-11-21.*

[51] Deepai. *https://deepai.org/machine-learning-glossary-and-terms/sigmoid-function, accessed: 2022-07-14.*

[52] Tensorflow. *https://www.tensorflow.org/api_docs/python/tf/keras/layers/Dropout, accessed: 2021-11-27.*

[53] Github. $https://ivansanchezfernandez.github.io/TSC\_supplementary_methods/, accessed : 2021 - 10 - 08.$

[54] ResearchGate. $https://www.researchgate.net/figure/Illustration-of-Max-Pooling-and-Average-Pooling-Figure-2-above-shows-an-example-of-max_fig2_333593451, accessed : 2021 - 11 - 22.$

[55] Zhenhua Song et al. "A sparsity-based stochastic pooling mechanism for deep convolutional neural networks". In: *Neural Networks* 105 (2018), pp. 340–345.

[56] Deep AI. *https://deepai.org/machine-learning-glossary-and-terms/softmax-layer, accessed: 2021-10-14.*

[57] Iván Sánchez Fernández et al. "Deep learning in rare disease. Detection of tubers in tuberous sclerosis complex". In: *PloS one* 15.4 (2020), e0232376.

[58] Punyanuch Borwarnginn et al. "Breakthrough Conventional Based Approach for Dog Breed Classification Using CNN with Transfer Learning". In: (2019), pp. 1–5.

[59] Neptune. *https://neptune.ai/blog/transfer-learning-guide-examples-for-images-and-text-in-keras, accessed: 2022-05-04.*

[60] Kaiming He et al. "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[61] Viso AI. *https://viso.ai/deep-learning/resnet-residual-neural-network/, accessed: 2021-10-08.*

[62] Lu Han et al. "A new method of mixed gas identification based on a convolutional neural network for time series classification". In: *Sensors* 19.9 (2019), p. 1960.

[63] Dongxian Wu et al. "Skip connections matter: On the transferability of adversarial examples generated with resnets". In: *arXiv preprint arXiv:2002.05990* (2020).

[64] Yung-Chia Chang, Kuei-Hu Chang, and Guan-Jhih Wu. "Application of eXtreme gradient boosting trees in the construction of credit risk assessment models for financial institutions". In: *Applied Soft Computing* 73 (2018), pp. 914–920.

[65] Adeola Ogunleye and Qing-Guo Wang. "XGBoost model for chronic kidney disease diagnosis". In: *IEEE/ACM transactions on computational biology and bioinformatics* 17.6 (2019), pp. 2131–2140.

[66] Nvidia. *https://en.wikipedia.org/wiki/Overfitting, accessed: 2022-01-04.*

[67] Wenjiang Jiao, Xingwei Hao, and Chao Qin. "The Image Classification Method with CNN-XGBoost Model Based on Adaptive Particle Swarm Optimization". In: *Information* 12.4 (2021), p. 156.

[68] XGBoost. *https://xgboost.readthedocs.io/en/stable/faq.html, accessed: 2021-12-14.*

[69] Wikipedia. *https://en.wikipedia.org/wiki/Receiver_operating_characteristic, accessed: 2022-01-22.*

[70] MedCalc. *https://www.medcalc.org/manual/roc-curves.php, accessed: 2022-01-22.*

[71] Paweł Widera et al. "Multi-classifier prediction of knee osteoarthritis progression from incomplete imbalanced longitudinal data". In: *Scientific reports* 10.1 (2020), pp. 1–15.

[72] Frank W Roemer et al. "Tibiofemoral joint osteoarthritis: risk factors for MR-depicted fast cartilage loss over a 30-month period in the multicenter osteoarthritis study". In: *Radiology* 252.3 (2009), pp. 772–780.

[73] et al. M. C. Nevitt D.T. Felson. "THE OSTEOARTHRITIS INITIATIVE - Protocol of the cohort study". In: *Osteoarthritis Initiative: A Knee Health Study* (2003). DOI: 10.1186. URL: https://doi.org/10.1038/s41598-019-56527-3.

[74] Dimitrios Kollias and Stefanos Zafeiriou. "Training deep neural networks with different datasets in-the-wild: The emotion recognition paradigm". In: *2018 International Joint Conference on Neural Networks (IJCNN)*. IEEE. 2018, pp. 1–8.

[75] Niams. *https://www.niams.nih.gov/grants-funding/funded-research/osteoarthritis-initiative, accessed: 2021-10-29.*

[76] Neil A Segal et al. "The Multicenter Osteoarthritis Study (MOST): opportunities for rehabilitation research". In: *PM & R: the journal of injury, function, and rehabilitation* 5.8 (2013).

[77] Janet Wesseling et al. “Cohort profile: cohort hip and cohort knee (check) study”. In: *International journal of epidemiology* 45.1 (2016), pp. 36–44.

[78] Jelle Wesseling et al. “CHECK (Cohort Hip and Cohort Knee): similarities and differences with the Osteoarthritis Initiative”. In: *Annals of the rheumatic diseases* 68.9 (2009), pp. 1413–1419.

[79] Andrew J Metcalfe et al. “Is knee osteoarthritis a symmetrical disease? Analysis of a 12 year prospective cohort study”. In: *BMC musculoskeletal disorders* 13.1 (2012), pp. 1–8.

[80] ImageBiopsyLab. *https://www.imagebiopsy.com/product/koala-ce, accessed: 2022-05-06.*

[81] Ambrahealth. *https://ambrahealth.com/directory/koala-from-imagebiopsy-lab/, accessed: 2022-05-06.*

[82] Roboflow. *https://blog.roboflow.com/you-might-be-resizing-your-images-incorrectly/, accessed: 2022-05-10.*

[83] OpenCV. *https://docs.opencv.org/5.x/da/d54/group___imgproc___transform.htmlgga5bb5a1fea74e accessed: 2021-11-14.*

[84] Inside-MachineLearning. *https://inside-machinelearning.com/en/why-and-how-to-normalize-data-object-detection-on-image-in-pytorch-part-1/, accessed: 2022-05-10.*

[85] Medium. *https://medium.com/analytics-vidhya/a-tip-a-day-python-tip-8-why-should-we-normalize-image-pixel-values-or-divide-by-255-4608ac5cd26a, accessed: 2022-05-10.*

[86] Medium. *https://wuecampus2.uni-wuerzburg.de/moodle/mod/book/view.php?id=958001chapterid= accessed: 2022-05-12.*

[87] OpenCV. *https://docs.opencv.org/3.4/d4/d13/tutorial_py_filtering.html, accessed: 2021-11-26.*

[88] Wikipedia. *https://en.wikipedia.org/wiki/Gaussian_blur, accessed: 2021-11-26.*

[89] Ngoc Thanh Nguyen et al. *Intelligent information and database systems*. Springer, 2016.

[90] Siddharth Misra and Yaokun Wu. “Machine learning assisted segmentation of scanning electron microscopy images of organic-rich shales with feature extraction and feature ranking”. In: *Machine Learning for Subsurface Characterization* 289 (2019).

[91] Zayed M Ramadan. “Effect of kernel size on Wiener and Gaussian image filtering”. In: *Telkomnika* 17.3 (2019), pp. 1455–1460.

[92] Hackaday. *https://hackaday.com/2021/07/21/what-exactly-is-a-gaussian-blur/, accessed: 2022-05-16.*

[93] Anil K Bharodiya and Atul M Gonsai. “An improved edge detection algorithm for X-Ray images based on the statistical range”. In: *Heliyon* 5.10 (2019), e02743.

[94] OpenCV. *https://docs.opencv.org/4.x/d5/daf/tutorial_py_histogram_equalization.html, accessed: 2021-11-14.*

[95] Wikipedia. *https://en.wikipedia.org/wiki/Adaptive_histogram_equalizationContrast _Limited_AHE, accessed: 2022-05-16.*

[96] C Rubini and N Pavithra. "Contrast enhancement of MRI images using AHE and CLAHE techniques". In: *International Journal of Innovative Technology and Exploring Engineering* 9.2 (2019), pp. 2442–2445.

[97] Ili Ayuni Mohd Ikhsan et al. "An analysis of x-ray image enhancement methods for vertebral bone segmentation". In: *2014 IEEE 10th International Colloquium on Signal Processing and its Applications*. IEEE. 2014, pp. 208–211.

[98] Puneet Misra and Arun Singh Yadav. "Impact of Preprocessing Methods on Healthcare Predictions". In: *Proceedings of 2nd International Conference on Advanced Computing and Software Engineering (ICACSE)*. 2019.

[99] CD Anisha and N Arulanand. "Early Prediction of Parkinson's Disease (PD) Using Ensemble Classifiers". In: *2020 International Conference on Innovative Trends in Information Technology (ICITIIT)*. IEEE. 2020, pp. 1–6.

[100] Christoph Smaczny. "Feature preprocessing in HEP at the example of a SUSY classification problem". In: (2018).

[101] Scikit-Learn. *https://scikit-learn.org/stable/modules/generated/sklearn. preprocessing.StandardScaler.html, accessed: 2021-11-14.*

[102] Scikit-Learn. *https://scikit-learn.org/stable/modules/generated/sklearn. preprocessing.MinMaxScaler.html, accessed: 2021-11-14.*

[103] Setthanun Thongsuwan et al. "ConvXGB: A new deep learning model for classification problems based on CNN and XGBoost". In: *Nuclear Engineering and Technology* 53.2 (2021), pp. 522–531.

[104] XGBoost. *https://xgboost.readthedocs.io/en/latest/parameter.html, accessed: 2021-11-30.*

[105] Nvidia. *https://www.nvidia.com/en-us/training/, accessed: 2021-12-21.*

[106] XGBoost. *https://keras.io/about/, accessed: 2021-12-07.*

[107] Keras. *https://keras.io/guides/transfer_learning/, accessed: 2022-05-16.*

[108] Sagar Sharma, Simone Sharma, and Anidhya Athaiya. "Activation functions in neural networks". In: *towards data science* 6.12 (2017), pp. 310–316.

[109] Ali Narin, Ceren Kaya, and Ziynet Pamuk. "Automatic detection of coronavirus disease (covid-19) using x-ray images and deep convolutional neural networks". In: *Pattern Analysis and Applications* 24.3 (2021), pp. 1207–1220.

[110] Mingchen Yao et al. "Binary output of multiple linear perceptrons with three hidden nodes for classification problems". In: *Proceedings on the international conference on artificial intelligence (ICAI)*. The Steering Committee of The World Congress in Computer Science, Computer ... 2014, p. 1.

[111] Coimbatore Peelamedu. “CNN BASED CLASSIFIER FOR IDENTIFICATION OF CANINE BREEDS”. In: ().

[112] MachineLearningMastery. *https://machinelearningmastery.com/dropout-regularization-deep-learning-models-keras/, accessed: 2022-05-16.*

[113] MachineLearningMastery. *https://machinelearningmastery.com/dropout-for-regularizing-deep-neural-networks/, accessed: 2022-05-16.*

[114] Qi Xu et al. “Overfitting remedy by sparsifying regularization on fully-connected layers of CNNs”. In: *Neurocomputing* 328 (2019), pp. 69–74.

[115] MachineLearningMastery. *https://machinelearningmastery.com/understand-the-dynamics-of-learning-rate-on-deep-learning-neural-networks/, accessed: 2022-05-16.*