
# Development and Deployment of an AI-Based Tool for Automated Histopathological Scoring of Human Articular Cartilage

Soroush Oskouei

Master's thesis

Master's degree program in medical physics

University of Eastern Finland

Department of Applied Physics

06.04.2022

University of Eastern Finland, Faculty of Science and Forestry

Department of Applied Physics

Medical Physics Degree Programme

Oskouei, Soroush: Development and Deployment of an AI-Based Tool for Automated Histopathological Scoring of Human Articular Cartilage

Thesis, 86 pages, 2 appendices (11 pages)

Thesis instructors: Academy research fellow Isaac Afara, postdoctoral researcher Jari Torniainen, and early-stage researcher Iman Kafian-Attari

Reviewer: postdoctoral researcher Jari Torniainen

April 2022

# Abstract

Current scoring systems for evaluating human articular cartilage classify it to discrete levels (scores) of degenerative diseases severity, hence, the issue in question is a classification problem. In this thesis work, an automated histopathological scoring system is developed. To develop this system, a set of 3102 histopathological images of articular cartilage sections from the knee joints (n=18) of nine human cadavers were used. Each histopathology image was graded using Mankin and Osteoarthritis Research Society International (OARSI) scoring systems. The original set of images was then augmented ten-fold by applying image manipulation techniques such as flipping and rotation, resulting in an augmented set of 31020 images. Different preprocessing algorithms, such as a bone-deletion algorithm that removes unwanted calcified parts of each image, were applied to these images. Several classification and regression models were developed to predict cartilage quality based on various metrics (integrity assessment, Mankin and OARSI score). These models use supervised deep learning methods, trained on the mentioned set of images. In some models, overlapping sliding windows were used to further increase the number of images by a factor of four. The effectiveness of the bone deletion algorithm and the windowing was investigated for the trained models.

Transfer learning based on various deep learning models (VGG19, ResNet18, ResNet50, Alexnet) were used to develop models for predicting Mankin and OARSI scores from the augmented set of images. Due to different camera settings, the original images consisted of two distinct groups with clearly different low-level image features (e.g., contrast and color). One group had a distinguishable RGB channel sensitivity and the other showed mixed green and blue channels. Models that were created and tested for the first set resulted in an accuracy of about 97% for integrity assessment and a mean squared error of 0.08 for OARSI scoring, while the models that were created and tested on both sets, although in an acceptable range, were not as accurate (about 85% for integrity classifications and a mean squared error of 0.16 for OARSI scoring).

The best performance was delivered by the models trained on the images that were not affected with the bone-deletion algorithm and validated/tested on the bone-deleted images. The best models were then extracted and deployed on a web application that can be accessed via:

https://share.streamlit.io/soroushoskouei/deephistology/DeepHisto.py.

## Acknowledgments

This thesis was carried out at the Department of Applied Physics, University of Eastern Finland. Firstly, I would like to thank the supervisors of this thesis: Academy Research Fellow, Docent Isaac Afara, Dr.  Jari Torniainen, and Mr. Iman Kafian-Attari.

Thank you for all your support and help. You have always given thorough and helpful comments and I consider myself privileged for having such dedicated and hard-working supervisors!

I want to extend my deepest gratitude to my family. Your sincere support is greatly appreciated, and you will always have my highest regards.

## Author’s contribution

Isaac Afara designed the general layout of the study. Soroush Oskouei, developed preprocessing methods for the initial images, developed the predictive regression and classification models, performed the analysis and evaluation of the models, and wrote the manuscript. Isaac Afara, Jari Torniainen, and Iman Kafian-Attari edited the text and suggested structural changes of the manuscript.

# Contents

# List of Abbreviations

| | |
|---|---|
| ACAN | Aggrecan |
| CAM | Class Activation Map |
| CNN | Convolutional Neural Networks |
| CSPCP | Cartilage-Specific Proteoglycan Core Protein |
| GANs | Generative Adversarial Networks |
| HHGS | Histological Histochemical Grading System |
| MR | Magnetic Resonance |
| MRI | Magnetic Resonance Imaging |
| MSEP | Mean Squared Error Percentage |
| NIN | Network In Network |
| OA | Osteoarthritis |
| OARSI | Osteoarthritis Research Society International |
| PG | Proteoglycan |
| PBS | Phosphate-Buffered Saline |
| RCNN | Region-Based Convolutional Neural Network |
| RGB | Red, Green, Blue |
| RNA | Ribonucleic Acid |
| SD | Standard Deviation |
| SVD | Singular Value Decomposition |
| SVM | Support Vector Machines |
| TMB | Tumor Mutation Burden |
| VGG | Visual Geometry Group |

# 1 Introduction

Osteoarthritis (OA), the most common form of arthritis, is the result of several disorders leading to structural and functional failure of articulating joints. OA is characterized by severe joint pain and, depending on the stage, might be accompanied by loss of joint functionalities and disability [1,2]. There are several methods for diagnosing OA following clinical evaluation, including synovial biopsy, x-ray, and magnetic resonance imaging (MRI), as well as arthroscopy, during the surgical intervention [1]. In the case of biopsy, a full investigation can be performed on tissue samples using histopathological approaches. This generally involves extracting and processing tissue sections using stains such as Safranin-O to reveal histopathological features. The stained sections are then scored by experts using standard histopathological grading systems, such as Mankin [3] and Osteoarthritis Research Society International (OARSI) [4] scoring systems, to assess the severity of tissue pathology and OA. These scoring systems rely on features such as structure, cellularity, staining, and integrity [5]. While these methods effectively assess the level of tissue degeneration, the outcome of histopathological scoring is highly subjective, with poor inter-observer reliability. In order to minimize this, the scoring is often done by multiple scorers, and the final results are averaged. However, aside from this method being extremely time-consuming, it also does not eliminate the issue of poor reliability. In this thesis, an automated approach based on artificial intelligence was developed to address these limitations.

Since histopathological scoring is performed on digital images of the stained tissue sections, this thesis aimed to develop an automated and reliable approach for accurately scoring articular cartilage sections. In this thesis, several deep neural networks were trained for reliable evaluation and automated scoring of existing histopathological images that have been previously scored using the aforementioned scoring systems. In order to minimize the impact of subchondral bone, image preprocessing algorithms (bone-deletion and rotation) were developed and applied to emphasize the important features for further analysis.

Following image preprocessing, two types of classifier neural networks were developed with the capability of classifying the integrity of the samples into one of three classes: *mild*, *moderate*, and *advanced*. One used a 3-class classification, and the other one applied nested binary classifications. These neural networks were followed by two types of regression models to predict the Mankin and OARSI scores of each sample. These

models were subsequently deployed online and made available for public use, taking into account all necessary ethical considerations.

While generating the models, various pre-trained Convolutional Neural Networks (CNNs) were investigated, each model showed non-identical performances in different tasks. These models use deep convolutional neural layers to improve accuracy. The input images for all the neural networks were transferred to 224*224*3 arrays. Furthermore, this thesis investigates the effects of the bone-deletion algorithm, windowing augmentations, model performances, and channel sensitivity.

## 1.1 Articular cartilage: composition structure and function

Articular cartilage (cartilage) is a type of connective tissue located in diarthrodial joints. Articular cartilage does not contain blood vessels, lymphatics, or nerves, thus a limited repair capacity. Due to its extreme biomechanical environment, the tissue's functional performance is characterized by features such as compressive stiffness (which is defined as resistance against deformation) and the coefficient of friction (which measures the amount of friction, indicating the force required for sliding) [6].

Articular cartilage is hyaline cartilage, 2-4 mm thick in humans, and is composed of a dense extracellular matrix, and a small number of chondrocytes. The extracellular matrix is mainly composed of water, collagen, and proteoglycans, with other non-collagenous proteins and glycoproteins [7].

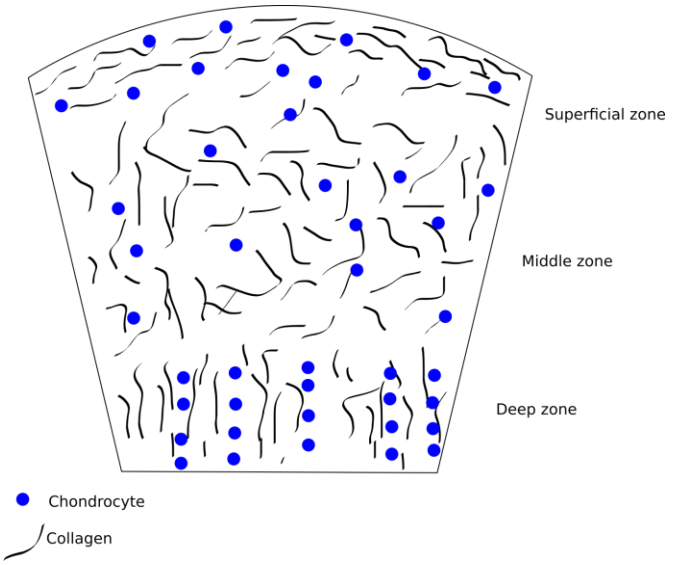


Figure 1.1: Articular cartilage structure.

The structure of the collagen network can be divided into three zones (figure 1.1): Superficial (tangential), middle (transitional), and deep (radial). The collagen fibers in the superficial zone are parallel to the surface. In the middle zone, however, the collagen fibers are oriented randomly. This zone covers about 40 to 60 percent of the total articular cartilage volume. Collagen fibers in the deep zone are perpendicular to the surface, thus providing the greatest resistance to compressive forces. Calcified cartilage and tidemarks can be seen below the deep zone. The deep zone also contains most of the collagen fibers, the highest proteoglycan, and the lowest water concentration in comparison to other zones [7].

Articular cartilage has a heterogeneous structure which enables it to withstand various forms of mechanical loading such as tension and compression and transfer them to the underlying bones. Articular cartilage supports a major part of the joint pressure by integrating the synovial fluid, which is incompressible. The extracellular matrix of articular cartilage possesses unique mechanical properties such as viscoelasticity and poroelasticity which in return contribute to the bulk mechanical properties of the tissue. [8].

## 1.2 Articular cartilage: Degeneration and characterization

Abnormal tissue remodeling in cartilage is driven by inflammation and undesired biomechanical conditions. Degeneration does not have an equal impact on the zonal structure of the cartilage. For instance, it has been shown that the deep zone is more susceptible to abnormal mechanical loading induced by degeneration. [9].

Several clinical conditions [7] are known to lead to the degeneration of articular cartilage, although it is not fully clear whether it is merely a result of aging, cumulative stress, traumatic injury, inflammatory events, or combination of all of the above.

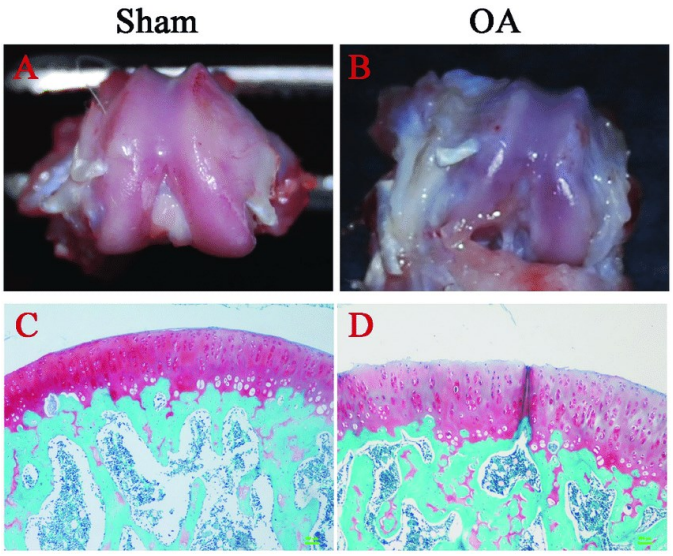


Figure 1.2: Degenerative changes in articular cartilage 4 weeks after surgery. Gross morphological observation of femoral condyles (A, B) and histological staining (Safranin-O/fast green) of articular cartilage (C, D) [10].

Aggrecan (ACAN), also known as Cartilage-Specific Proteoglycan Core Protein (CSPCP) or chondroitin sulfate proteoglycan, assists the articular cartilage with the ability to withstand compressive loads [11]. Aggrecans are usually found in the form of proteoglycan aggregates in the extracellular matrix of articular cartilage. Certain synthetic and degradative events can lead to alteration of the aggregate components. Loss of aggrecan and its fixed negative charges is known as one of the characteristics of early-stage articular cartilage degeneration. Figure 1.2 shows an example of degenerative changes that can occur post-surgery.

One of the other features related to articular cartilage degeneration is the loosening of the collagen network. Even though no collagen is lost, the alteration in the network organization of collagen can have a detrimental effect on the biomechanical properties [11-14].

# 2 Osteoarthritis and histopathological scoring

Osteoarthritis, the most common degenerative joint disease, is one of the major reasons for experiencing pain and disability in the adult population worldwide. There are many causes leading to OA, among them, injuries to the joint, aging, heredity, joint instability, joint inflammation, and obesity [15]. The molecular mechanisms responsible for the initiation or progression of the disease are not fully understood yet. Figure 2.1 shows the histological view of the features related to OA compared with a normal tissue section. In the OA-affected cartilage, loss of cells, disorder, disarray, and confusion of layers and matrix can be observed.

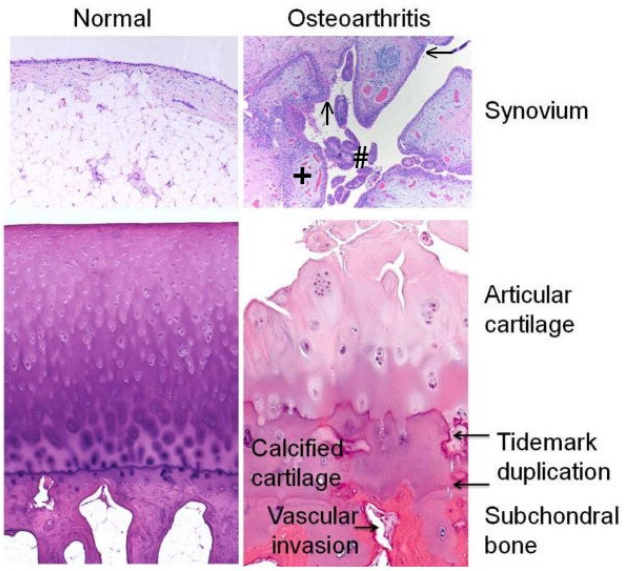


Figure 2.1: Comparing OA with a normal tissue [18].
The normal synovium is showing a thin lining layer, while in OA tissue, synovial villous hyperplasia (#), lining hyperplasia (arrows), increased vascularity (+), and perivascular mononuclear cell (inflammatory) infiltration can be seen.

The leading cause of OA in young adults is sports injuries. Also, patients with previous joint injuries or dislocations often develop OA. Joint stabilization can be affected by bone, cartilage, ligament, or meniscus damage, which are common sports traumas [16]. Although the process is complex and not completely understood, the significance of genetics in OA is apparent. Family-based studies have proved the effect of inheritance in OA [17]. In order to quantitatively determine the severity of OA in different patients, and for the sake of communication, it is important to have a standard, universally accepted system for grading OA severity. One of the most widely used systems for this purpose is the one proposed by Mankin [3].

## 2.1 Mankin scoring

The Histological-Histochemical-Grading-System (HHGS) or Mankin score for evaluation of osteoarthritic cartilage was originally proposed by Mankin in 1971. The main components that were investigated in the scoring system were the structure, cells, safranin-O staining, and tidemark integrity [3]. The data relating to the intra- and interobserver reproducibility of the Mankin scoring system was first reported on information gathered from an experimental animal model. However, another study conducted in 1997 showed the inadequacy of Manking scoring for evaluating the extent of OA severity in humans [19].

### 2.1.1 Components of Mankin scoring

The main components of the Mankin scoring system are structure, cells, safranin-O staining, and tidemark integrity. The total Mankin score varies in the range of (0, 14) where 0 is considered healthy/normal tissue whereas 14 is considered severe OA. Moreover, each of the Mankin components has its own sub-criteria that must be assessed. In the Mankin system, considering the tissue structure component, a smooth intact surface receives a zero score, and total disorganization would get a score of 6, meaning the worst case for this component. This component (surface structure integrity) has the most potent effect on the score. In the cell component, there are four stages: uniform distribution, diffuse cell proliferation, cell clustering, and cell loss (figure 2.2). The safranin-O staining component investigates the intensity of the safranin-O stain, which binds with the proteoglycans (PG) of the cartilage. This means that the low intensity of the stain is an indicator of proteoglycan loss in OA cartilage. A uniform and strong staining would suggest a healthy tissue with respect to this component, thus a zero score. As the discoloration increases, so does the attributed score. And the last component is the

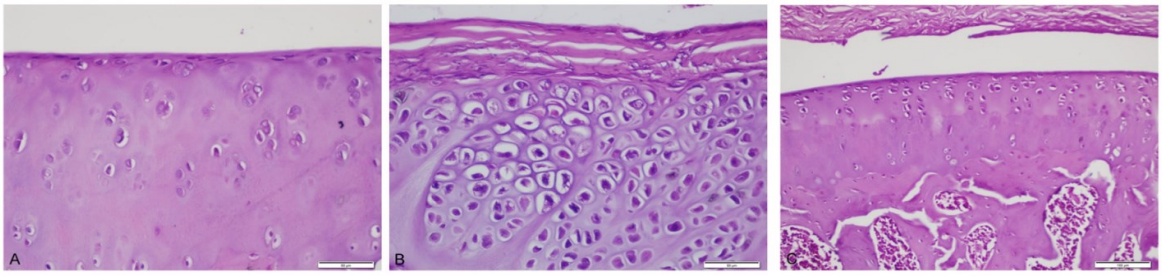


Figure 2.2: Microscopic evaluation of structure and cells subcategories of Mankin grading system. A: Intact articular surface with normal cellularity (Hematoxylin and eosin×400). B: Irregularity of the articular surface and chondrocyte cloning (Hematoxylin and eosin×400). C: Fissures and clefts of the articular surface to the calcified zone (Hematoxylin and eosin×200) [20].

tidemark integrity, where vascularity would suggest a score of one [3]. Table 2.1 shows the values assigned to the components.

Table 2.1: Mankin scoring system components.

| Component | Score | Histological finding |
|---|---|---|
| **Structure** | 0 | Smooth intact surface |
| | 1 | Slight surface irregularities |
| | 2 | Pannus/surface fibrillation |
| | 3 | Clefts into transitional zone |
| | 4 | Clefts into radial zone |
| | 5 | Clefts into calcified zone |
| | 6 | Total disorganization |
| **Cells** | 0 | Uniform cell distribution |
| | 1 | Diffuse cell proliferation |
| | 2 | Cell clustering |
| | 3 | Cell loss |
| **Safranin-o staining** | 0 | Uniform staining |
| | 1 | Minor discoloration |
| | 2 | Moderate discoloration |
| | 3 | Severe discoloration |
| | 4 | Total discoloration |
| **Tidemark integrity** | 0 | Intact |
| | 1 | Vascularity |

## 2.2 OARSI scoring

The grading methods that were developed for histopathological assessments evaluate the severity of the OA by reducing disparate histological information to quantifiable indicators. In general, the OA grading systems can be classified into two categories: an assessment system that is based on sequential stages of increasing OA severity [21, 22], or based on the sum of independent indicators of OA severity [3]. The grading systems of the second category can quantify different forms of OA by generating a total grade. However, this process does not reflect the contribution of different injury, repair, degenerative, and OA processes of the conditions. In other words, the overall score does not share any information about the component to which the damage is mostly related, leading to the case score. This means that two completely different degenerative pathways might lead to the same total histological grade in the previously mentioned systems.

To address these limitations, the OARSI scoring system offers seven different grades [4], where zero is for the normal tissue, grade 1 denotes the retention of the articular cartilage surface layer. Grade 2 indicates focal discontinuity of the cartilage superficial zone. Grade 3, vertical fissures formed by the extension of the matrix cracks into the middle zone. Grade 4, cartilage erosion, grade 5, denudation, and grade 6 that indicates changes in the contour of the cartilage surface (figure 2.3). With the OARSI system, OA severity is estimated based on the extent of the joint cartilage surface, area, or volume involved in the local OA process. Table 2.2 summarizes the details of the components used in this system.

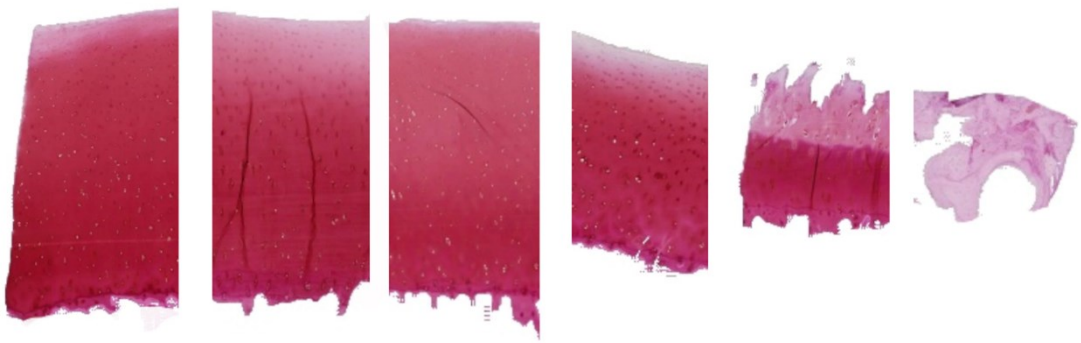

Figure 2.3: Microscopic evaluation samples of the OARSI system. From left to right, slices are given the grades of 0, 1, 2, 3, 4, 5.25. These are average scores given by expert observers.

Table 2.2: OARSI scoring system components.

| Grade | Histological finding |
|---|---|
| **0** | Normal (Healthy) |
| **1** | Cartilage surface intact |
| **2** | Cartilage surface discontinuity |
| **3** | Vertical fissures within the cartilage |
| **4** | Erosion |
| **5** | Denudation |
| **6** | Deformation |

## 2.3 Limitations of histopathological scoring systems

Previous surveys on the validity of the Mankin scoring system suggest, that since the system was developed and based on specimens with advanced OA, it might lead to difficulties in the mild and moderate classes of the OA [23]. There are also several features that are either missing or would mislead the overall score and cause the wrong classification. For instance, the horizontal extent of the cartilage surface that is affected by the OA is not considered in the system. Other examples are features such as 'pannus' and 'surface irregularities' that would lead to a higher score than the real value even though these features can be found in healthy or regenerative cartilage as well [24].

As mentioned before, other studies have questioned the reproducibility and validity of the Mankin scoring system [19, 24,25]. These studies suggest that the reliability of a grading system would be greatly enhanced if the observations to be scored are well defined and easy to assess. A set of more clear definitions of variables for evaluation can include clefts, clones/clusters of cells, number of cells, surface and tidemark, and the zones or

layers in a cartilage section. These assessment variables, should they be considered in a grading system, are to be independently reproducible.

In the original OARSI study on the human samples, interobserver reliability analysis for the histologic cartilage assessment was not established, however, the intra-observer reliability was found to be more than sufficient. Also, it has been reported that the OARSI system can be affected by the subjectivity of the reader, hence, an automated computer based OARSI grading system method could improve the subjectivity, process time, and throughput [26].

# 3 Automated grading models

Recent studies have used artificial intelligence for computer aided knee OA diagnosis using x-ray, magnetic resonance (MR), and histological images [27-39]. This is mostly done to reduce the uncertainties in the diagnosis that are related to human errors. The huge repositories of the related radiographic and MR images would help to develop efficient models to address this issue. Different types of artificial intelligence can be applied to this problem. One type that is mostly used with image data is the CNNs. Convolutional neural networks are a kind of deep learning which is itself a subcategory of artificial intelligence (Figure 3.1). Results derived from deep learning-based measurements are shown compatible with expert observations [28, 29].

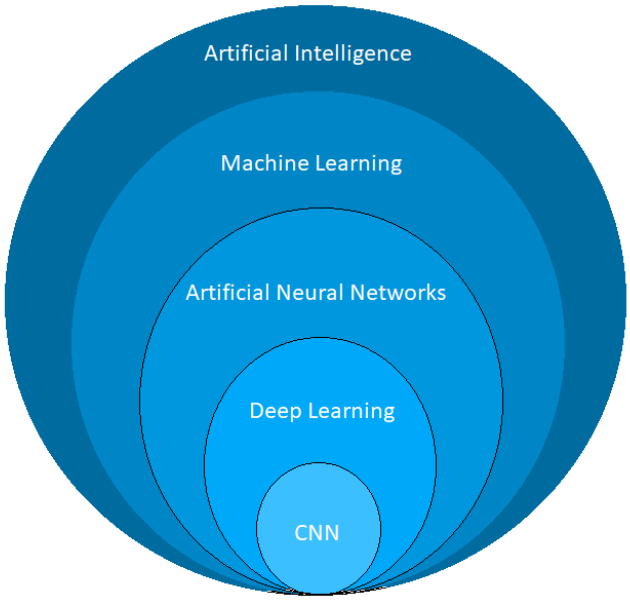


Figure 3.1: category of CNN under artificial intelligence.

The aim of this thesis is to develop a transfer-learning-based neural network model for automated histological scoring of human articular cartilage sections.

Automated evaluation methods that have been previously introduced mostly apply to radiographic and magnetic resonance images. Deep learning models, such as CNNs, can extract visual features using a combination of mappings in their model architectures. These are then used to by the network to “learn” the complex features.

When dealing with various large datasets to identify early OA biomarkers, it is more efficient to take an automated algorithm approach based on machine learning rather than traditional methods [30]. Within the machine learning domain, deep learning is a field that has become popular in research and clinical applications. It has revolutionized the diagnosis methods [31]. Conventional manual techniques will be less needed by the

emergence and improvements of the computer-aided diagnosis. It has been shown in the past few years that deep neural networks can be used for a wide range of medical image analysis tasks to achieve accurate results in classification, detection, and segmentation assignments for OA diagnosis [32]. Among different deep neural network architectures, the CNN architecture has received a significant amount of research interest [30, 26]. One of the advantages of this neural network form is the fewer parameters involved in the training which makes it more suitable and faster than other methods for image analysis tasks.

Although automated models are mostly developed for radiographic and magnetic resonance images, histological images are not neglected. Image analysis at the tissue level is focused on the definitive histopathological information. Two of the grading systems used for histological assessments are covered in chapter 2 and related features that can lead to model development are summarized in table 2.1 and table 2.2. Reproducibility limitations (see 2.3) along with the humane imperfections suggest the development of an automated tool. Selective applications have been used to analyze features in osteochondral histology. There have been previous works such as measurements on surface roughness, cell density, changes of cartilage thickness, [26] and estimation of glycosaminoglycan content [33, 34].

Rytky et al. presented the first ML-based automatic 3D histopathological osteoarthritis (OA) grading method in 2020 [35], in which they have used a four-stage grading system to describe the severity of the tissue degeneration. In their approach, they trained a linear regression model against the ground truth m CT grades. They used L2 (ridge) regularization with a coefficient of 0.1 and assumed a continuous outcome, and then a binary logistic regression (LR) model (also with L2 regularization) was trained to assess the sample's degeneration [35].

Power et al. also used Deep learning method for automation of the grading of engineered histological cartilage images [36]. In their approach they have tried different model trainings and development and implemented transfer learning [36].

A deep learning-based approach published in the Scientific Reports by Tuilpin et al. [37] shows a very accurate model based on radiographic images. Their approach is based on the Deep Siamese CNN architecture. Usually, the mentioned network consists of two branches, where each one corresponds to each input image. The novelty in their method was that they did not train their model to compare image pairs; rather, using the symmetry in the image, which allowed the architecture to learn identical weights for every image side.

The deep learning method can also be applied to different modalities of magnetic resonance images. Ashinsky et al. [38] have used machine learning methods to perform

classification and regression models on MRI images. They followed two methods to classify their dataset: “an analysis to measure separability of the two classes within the dataset itself and a standard leave-one-out cross validation to estimate osteoarthritis cartilage”.

Tissue based models based on histological images can also be developed to indicate the severity of cartilage degeneration. Mousavi-Harami et al. developed a custom image analysis program for automatic objective scoring of cartilage degeneration [39]. However, they have not used deep learning or transfer learning to develop their measuring system. In this thesis, we are aiming to develop a similar grading model by applying transfer learning to develop the wanted models.

# 4 Neural networks and deep learning

From the biological point of view, neurons are cells that are connected to each other forming a network. Each connection can transmit a signal to other neurons. One can consider the dendrites and axons of a neuron as input/output pathways for transmitting signals. Depending on variables related to synapses, the number of connections, and dendrite/axon diameters, these transmissions can be relatively strong or weak. The first mathematical model of a neuron was created in 1943, this model provided an abstract formulation for the functioning of a neuron but did not work on the learning concept [40]. Perceptron (the simplest form of a single-layer network) was established in 1957 by Rosenblat [41]. From this point forward, different types of networks were introduced that guided the concept through significant strides.

## 4.1 Single-layer perceptron

The basic simple entity of any neural network is based on the model of a neuron. A single-layer perceptron is a simple single-layer neural network in which the output is defined as the linear combination of the input variables plus a bias term. Before the output is generated, this sum is usually affected by an activation function (in this case a simple threshold function). This function determines whether the node should be activated or not. The output can be formed as:

$$p(x) = f_a(w^T x + b) \tag{4.1}$$

where the input is denoted by $x$, bias by $b$, weights by $w$, and activation function by $f_a$. The bias, b, is a scalar, whereas the input $x$ and the weights $w$, are vectors, i.e., $x \in \mathbb{R}^n$ and $w \in \mathbb{R}^n$ with $n \in \mathbb{N}$, representing the dimension of the input.

## 4.2 Supervised learning algorithm for a perceptron

In simple words, a supervised learning algorithm is a method of adjusting the values of the mentioned weights iteratively by using labeled data. In the case of the perceptron, this adjustment can be similar to this:

$$w \leftarrow w + \alpha(t - p(x))x \qquad 4.2$$

where $\alpha$ is called learning rate (which can be fixed or alternating), $t$ is the vector of true values (labels or targets), $p(x)$ is the output of the perceptron, and $x$ in the input vector. Due to its linear structure, a single-layer neural network is limited to classifying the data that are linearly separable.

## 4.3 Multi-layer perceptron

In order to overcome the limitation of the single-layer neural networks, one method is to use several layers and outputs and a combination of activation functions to generate the desired output regions in the target space. One can create a multi-layer perceptron by applying the principle of the single-layer perceptron in a combination of them to form several layers. In this model, the input layer results in several outputs that are themselves fed forward to the next layer (a feedforward neural network) and usually these outputs are affected by non-linear transformations (activation functions). The layers between the

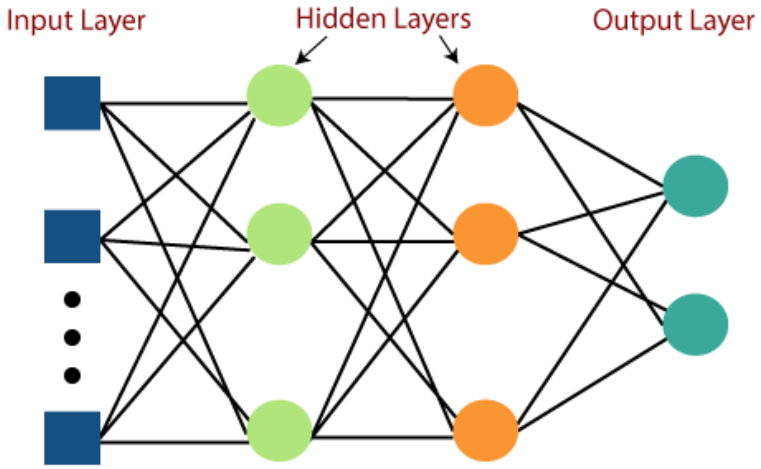


Figure 4.1: Multi perceptron structure [42].

input and final output layers are called the hidden layers (figure 4.1). Activation functions that are used in a multi-layer neural networks are more complicated than the ones used for single-layer neural networks.

## 4.4 Backpropagation

The backpropagation algorithm is a method used in training feedforward neural networks. Similar to the supervised learning algorithm that is used for the single-layer perceptron, the true values (targets) can be utilized to find the best fitting weights. A loss function is defined to represent the difference between the labels (t) and the neural network's output. In order to find the best set of weights in the network, one should minimize this loss function. This learning algorithm calculates the gradient of the loss function that is then used to change the weights and biases. This gradient can most efficiently optimize the loss function.

## 4.5 Batch size, iterations, and epochs

In modern neural networks, forward feeding (forward propagation) and backpropagation does not necessarily take all of the input data points at once. It is beneficial to batch the data points and run the forward and back propagations for all the batches. The word batch size simply refers to the number of data points that are taken for each batch. Iteration is the number of times that the propagations should occur for the whole data batches to be covered. The word "epochs" describes the number of times that all the iterations are complete, and all batches of samples are propagated.

## 4.6 Overfitting

When fitting a model to a particular dataset, there is always the possibility that some features are not included. In other words, it is possible that the "test" data (not included in the training data) contains features that are not considered in the model. It is also possible that the model focuses excessively on the training data. These would lead to poor performance of the model over the test data, even though a perfect performance might be achieved over the training set. This problem is referred to as the overfitting problem and it is highly probable when the training set is relatively small.

## 4.7 Machine learning and deep learning

Machine learning is a type of artificial intelligence, an automated method of data analysis, that applies the training (learning) concept to the recognition of patterns in data. In general, this learning process can be divided into four types: supervised, semi-

supervised, unsupervised, and reinforcement. Supervised learning is when labeled data is used for training the model. Semi-supervised is when both labeled and unlabeled data are utilized. An unsupervised learning algorithm studies unlabeled data to identify patterns. In reinforcement learning, a “policy network” is used to “reward” or “penalize” an agent (operating on the input dataset) for each particular input without using prior labeled targets. Some of the traditional algorithms in machine learning include decision trees, Bayesian networks, Support Vector Machines (SVM), etc.

Artificial neural networks are methods in machine learning that can be used in both supervised and unsupervised approaches. A neural network can be “shallow” or “deep”. Shallow neural networks contain a few hidden layers, whereas deep neural networks contain many hidden layers. The word “deep” refers to the number of layers. When a deep neural network is applied for machine learning, it is called *deep learning*. Deep learning algorithms build a model based on sample data in order to make complex predictions, models, or decisions "without being explicitly programmed to do so" [43, 44].

## 4.8 Computer vision and neural networks

Basically, images are stored as a matrix of pixel values. Mathematically, a black and white image can be considered a function that maps from $\mathbb{R}^2$ to $\mathbb{R}$. This function gives the intensity value of the pixels. This intensity value ranges from 0 to 255. A color image is a stack of three functions that form a 3$^{rd}$-order tensor, thus the mapping is from $\mathbb{R}^2$ to $\mathbb{R}^3$. Hence, the value at each pixel is a vector with three components corresponding to the intensities received by each color channel (red, green, and blue). In computer vision problems, one usually seeks to recognize patterns in the images. That is, the data in the represented matrix can be used to classify parts of the image or the whole image into desired categories. One might also seek to extract continuous values from an image, for instance, measuring the area covered by cells in microscopic images, or predicting age from facial images. This means the computer vision tasks that are related to recognition can be interpreted as classification and regression problems. If one attempts to solve these problems with a multi-layer perceptron as explained in previous sections, an important issue will arise. Due to a large number of data points, the number of parameters (weights and biases) would increase significantly. Another noticeable issue is that the data would be translated into a vector, which might destroy some 2D-related or structural information. This means that it is better to take another approach so that 2D information is preserved and the number of parameters is not large. This approach is utilizing a Convolutional Neural Network.

## 4.9 Convolutional neural networks

A CNN uses local connectivity between neurons, meaning that a neuron is only connected to nearby neurons in the next layer. This would lead to a significant decrease in the total number of parameters in the network. This type of neural network is a great tool for generating models for learning from image data. CNN accepts matrices as input and preserves spatial structures. In this type of neural network, a set of convolution windows (also called kernels) are defined that slide over the input dataset [45]. Each of these kernels has specific weight values and the convolved matrix is then affected by a non-linear transformation (activation function). The outputs from these kernels are then stacked to form the output, which is of the size $k \times H_o \times W_o$, where k is the number of kernels, and $H_o$ and $W_o$ are the height and width of the output. This makes one layer of the network. The output of each layer is then used as input for the next layer with a different set of kernels. In general, CNNs are constructed with the following types of layers (figure 4.2):

- Convolutional layers
- Fully connected layers
- Activation layers
- Pooling layers

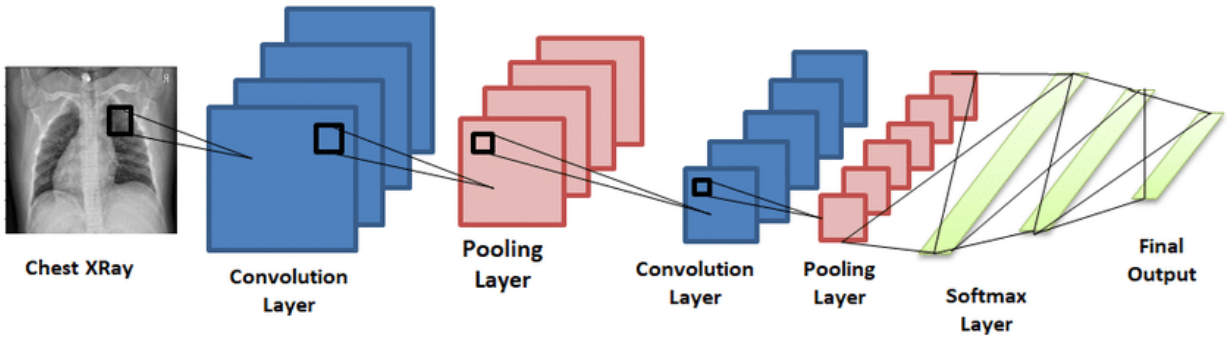


Figure 4.2: A sample CNN structure [46].

### 4.9.1 Convolutional layer

This layer performs a dot product of a kernel matrix with a part of an image to produce and output. This is repeated to cover the whole image (hence the term convolution). The input of a convolutional layer is an array of the size $n \times H_i \times W_i$, where *n* is the number of channels, *H* is the height of the input data, and *W* is the width of the data [45]. The parameters of this layer are the number of kernels, kernel size, strides, and padding value.

The word “strides” refers to the amount of shift in pixels for a kernel after each operation which determines the size of overlapping in the output calculations.

### 4.9.2 Fully connected layer

This layer is similar to a simple neural network layer where each node is connected to all the nodes in the previous layer. This layer is often called the dense or linear layer [45].

### 4.9.3 Activation layer

After going through the mentioned layers, the weighted outputs are passed through non-linear transformations that are the explained activation functions. Some common types of activation functions are Sigmoid, ReLU, Tanh, and Softmax [47].

### 4.9.4 Pooling layer

The pooling layer performs sampling over local regions (maximum or average) to reduce the size of an input [48]. This layer is usually inserted between 2 or 3 convolution layers and decreases the requirement for parameters.

### 4.9.5 Dropout

Dropout is a technique to prevent overfitting. It refers to deactivating (ignoring) some of the nodes (neurons) during the training [49]. Dropout layer usually selects randomly a designated portion of the nodes and freeze them. Thus, the frozen nodes would not have an impact on the feed-forwad process.

## 4.10 Popular CNN model architectures

The architecture of a network is the framework, in which, distinct layers are arranged between the input and output layers. Several parameters determine the form of an architecture, including the function of the layers, order of the layers, and connectivity between nodes and between layers. Various possibilities in choosing these parameters results in various architectures.

### 4.10.1 AlexNet

AlexNet is a type of CNN network with a ReLU activation function after convolutional and fully-connected layers (excluding the last fully-connected layer), and a dropout of 0.5 for hidden layers. AlexNet is composed of a relatively simple structure, yet it prohibits overfitting due to above features (figure 4.3) [50].


Figure 4.3: Alexnet architecture [51].

### 4.10.2 VGG

VGG neural network was developed in the Visual Geometric Group at Oxford University and has two available versions: VGG16 and VGG19, numbers16 and 19 refer to the number of hidden layers, exclusive of the max pooling and softmax layers. The novelty in this network was depth, which was increased by adding more convolution layers with 3x3 kernel size (figure 4.4) [52]. Table 4.1 shows the differences between these two pre-trained models [53].

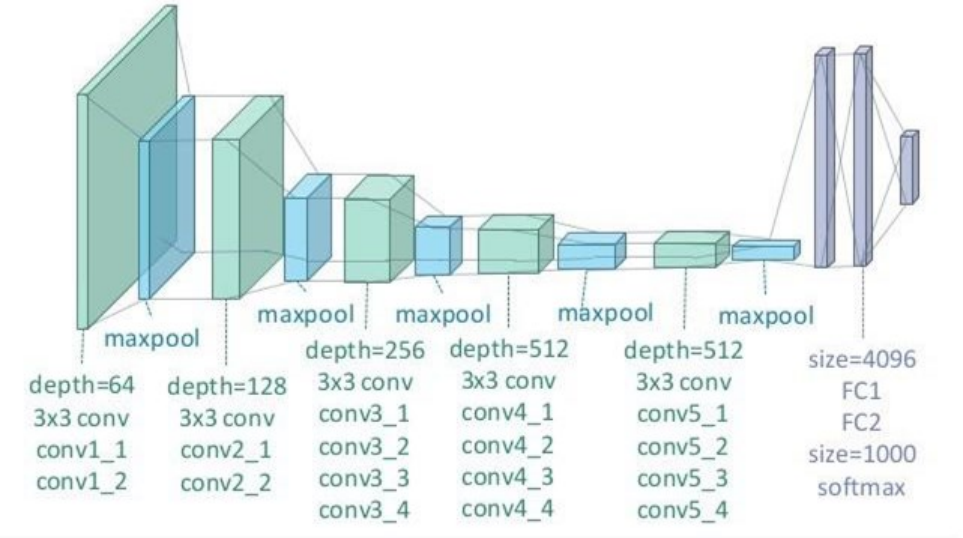


Figure 4.4: VGG architecture [54].

Table 4.1: Differences between VGG16 and VGG19 models.

| Layer | VGG16 | VGG19 |
|---|---|---|
| **Size of layer** | 41 | 47 |
| **Convolutional layer** | 13 | 16 |
| **Filter size** | 64, 128 | 64, 128, 256, 512 |
| **ReLU** | 5 | 18 |

### 4.10.3 ResNet

After reaching a certain depth, adding layers to feed-forward CNN results in higher errors both in the training and validation sets. This is due to two problems, first, in forward propagation the information in the last layers of the network is greatly dependent on the middle layers and not directly on the original image. Second, during the backpropagation the early layers near the input almost receive no update in gradient. In the ResNet CNN model, developers have added a new concept called residual block, meaning connections that can skip some layers (figure 4.5) [55]. ResNet stands for residual network. In this

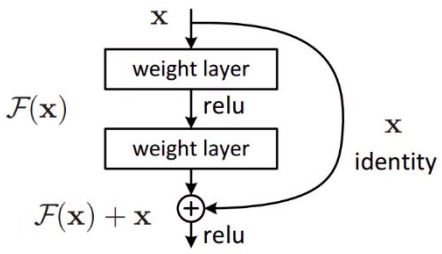


Figure 4.5: Resnet residual block architecture [54].

method raw information from earlier layers are passed to the later ones, this feature helps the desired information to be better preserved and addresses the mentioned issues. Residual networks usually contain the following layers: convolution, batch normalization, ReLU, MaxPooling, ResNet blocks, average pooling, and fully connected layers.

## 4.11 Machine learning for automated histopathological grading

Utilization of machine learning for automated histopathological grading can suggest several methods with various learning paradigms. The goal of the supervised machine learning approach is to infer a function that is able to map the input data to their appropriate labels by utilizing the training data. On the other hand, the goal of unsupervised methods mostly is to infer a function that can describe hidden structures from unlabeled images [56].

Unsupervised learning models contain methods such as clustering, dimensionality reduction, and latent variables. One of the benefits of these methods is that there is no need for supervision (or prior knowledge of the output). Most important disadvantage of this approach is its suboptimal performance in terms of accuracy and explicitness. Semi-supervised learning can also suffer from the same problems.

The advantage of supervised learning is that models will be task specific and more reliable. Also, in general, supervised learning methods require less data than unsupervised learning methods [56]. Among the machine learning methods based on supervised learning, traditional methods (also referred as shallow-learning methods) include decision trees, support vector machines, Naive Bayes classifiers, and various linear and non-linear regression algorithms. These methods, although simple to implement, would require high computing power, high memory usage, and very large, complete, and known feature set, especially when dealing with image data [56]. Some of these methods can also be very sensitive to outliers or exceptional cases. In these methods the evaluation of bias–variance tradeoff would be essential and hard to implement.

Deep learning algorithms can be beneficial compared to other types of machine learning methods if the key features are not known, if the number of input images is large, and if sufficient computational power are provided. Deep neural networks can be applied in both supervised and unsupervised fashions. The best approach can be chosen based on the data available and suitability of each method. It is important to note that a preprocessing step on the input dataset is required before feeding them to the model. In other words, important and relevant features of the data shall be extracted prior to being fed to network.

Learning methods (supervised or unsupervised) that are used on image datasets rely on quantifiable features in the images. In digital pathology, quantifiable features can be modeled by approaches that depend on the type of features in question: engineered or unsupervised features. Generally, engineered features are those connected to specific measurable attributes in the image and have some degree of interpretability. Unsupervised feature approaches (deep learning-based methods for example) are less intuitive and rely on filter responses extracted from large numbers of training input. Deep

learning paradigms offer end-to-end unsupervised feature generation methods that take advantage of large amounts of training data in combination with a multi-layered neural network. In both approaches of feature selection – engineered and unsupervised – there are several strengths and weaknesses. Engineered features provide more transparency and subsequently will usually appear more instinctive to the end-user – the pathologist or clinician as the case might be. The unsupervised feature generation-based approach of deep learning strategies, however, can be applied quickly and seamlessly to any domain or problem, but suffer from the lack of feature interpretability [57].

## 4.12 Transfer learning

Transfer learning is a method in machine learning that uses the knowledge gained while solving one problem and applies it to a different but related problem. There are many types of transfer learning, such as adversarial, relation-based, feature-based, instance-based, and model-based, to name a few. In this thesis, model-based or (parameter-based) transfer learning is used. In most model-based transfer learning algorithms, it is assumed that the transfer is of the inductive type, where some labeled instances are considered to be available in the target domain [58].

### 4.12.1 Theory and background

A major challenge that exists in machine learning methods in practice, is that they are not competently applicable in new task domains. The reason for this incompetency can root in several challenges: small amount of data, changes of circumstances and major changes of tasks or features [58]. Where other methods fail, transfer learning can be auspicious. Transfer learning extracts knowledge from one or more source domains to improve the learning in a target domain [59].

Transfer learning can help promote AI where not much data is available, however, if the distance between two domains is large (quantifiable features and tasks vary significantly), then one better not use transfer learning since it can induce a negative effect on the target domain model. To the contrary, if two domains are “relatively close,” transfer learning would be favorable [58].

### 4.12.2 Definitions

In this section, important concepts in transfer learning are defined following the notations used in [59].

**Domain:** A domain *D* is considered a set of two components: a feature space *X* and a marginal probability distribution $P^X$. In this definition, each input instance is a member of the feature space.

$$D = \{X, P^X\} \quad 4.3$$

In general, if two domains are different, this difference can root in any or both of the two components.

**Task:** A task *T* consists of two components: a label space *Y* and a function $f(\cdot)$. Consider the set of unknown instances $\{x^*\}$, the function *f* can map these instances to elements in the label space. $f(x)$ can also be denoted as $P(y|x)$.

$$T = \{Y, f(\cdot)\} \quad 4.4$$

A two-domain scenario (source and target domains) can be described as follows:

$$D_s = \left\{\left(x_{s_i}, y_{s_i}\right) \mid \forall\, i \in [1, n_s], x_{s_i} \in X_s, y_{s_i} \in Y_s\right\} \quad 4.5$$

$$D_t = \left\{\left(x_{t_i}, y_{t_i}\right) \mid \forall\, i \in [1, n_t], x_{t_i} \in X_t, y_{t_i} \in Y_t\right\} \quad 4.6$$

**Transfer learning:** The obvious assumption behind transfer learning is that the domains and tasks of the two (or more spaces) are not equal ($D_s \neq D_s$ and $T_s \neq T_t$). Knowing this inequality, transfer learning can be defined as any form of using the knowledge in $D_s$ and $T_s$ to improve the predictive function in the target domain $f_t(\cdot)$.

**Homogeneous transfer learning:** Homogeneous transfer learning assumes the following conditions:

$$X_s \cap X_t \neq \emptyset \quad 4.7$$

$$Y_s = Y_t \quad 4.8$$

**Heterogeneous transfer learning:** Assumes that the set of features in source domain(s) and the set of features in target domain(s) share no subset. However, the assumption does not deny the possibility of translators (functions that apply transformation within or between certain spaces) between the two spaces to enable successful transfer learning.

### 4.12.3 Transfer learning types

**Instance-based:** The transferred knowledge is related to the weights attached to source instances. This approach assumes that part of the source domain labeled data can be reused for the target domain after reweighting or resampling. In this method, “knowledge” is considered the source domain labeled instances with large weights. In instance-based transfer learning it is assumed that the set of features in source domain(s) and the set of features in target domain(s) have many mutual members [58,60].

**Feature-based:** The transferred knowledge is related to a subspace of the domains that can be spanned by features in both domains. The idea behind feature-based approaches is to identify a proper feature representation for both domains so that by projecting data onto the new representation, the source domain labeled data can be reused for training in the target domain [58,60].

**Model-based:** The transferred knowledge is included in part of the source domain models. This method assumes that the source and target domains have some mutual parameters in their learning models. In this approach, the aim is to use a well-trained source model and transfer it to train a target domain model that is more precise. Transferred knowledge, in this method, is embedded in the model parameters and is domain-invariant [58,60].

**Relation-based:** The transferred knowledge is in the relations between the entities in the source domain. This method assumes and extracts relationships between instances that are (assumably) similar in both domains and / or tasks [58,60].

**Adversarial:** There are various methods used as an adversarial transfer learning, one of the most popular ones is the domain-adversarial transfer learning. This method applies a discriminative module on the extracted features between the source and target domain, and a feature extracting module that operates against the previous module’s objective [58, 61].

### 4.12.4 Instance-based transfer learning

Instance-based transfer learning approaches tend to use the labeled data from the source domain to train a more precise model for the target domain tasks. In case the two domains have very short distance, the data in both domains can be merged, thus the problem would become a machine learning problem in a singular space. However, this approach would not always result in practical and favorable solutions for target domain tasks [58, 62].

The idea behind the instance-based transfer learning is that some labeled data in source domain are still useful for training the model in target domain. Choosing the right data from the source domain is important and bias-variance analysis can be used for verifying the similarity between data distributions.

This approach is very useful especially if the dataset in target domain is small and model's variance is large. Adding some of the data from the source domain that shares some similarity in distribution would reduce the variance of the model. If the data distribution of the two domains are very different, the new learning model might have a large bias [58, 62].

Two important issues raise when using this transfer learning approach. First, how to identify the similar labeled instances, and second, how to utilize them in an algorithm for a more accurate model.

Common assumptions behind this approach are that features for most instances (in source and target domain) have similar range of values, and the output labels in both domains are the same. Considering that the task $T = \{Y, P^{Y|X}\}$ has two components, the labeled space $Y$ and the conditional probability distribution $P^{Y|X}$, one can conclude that the difference in this transfer learning approach lies in the second component of the task. If the $P_s^X \neq P_t^X$ but $P_s^{Y|X} = P_t^{Y|X}$ the problem is referred to as noninductive transfer learning, and when $P_s^{Y|X} \neq P_t^{Y|X}$ the problem is called inductive transfer learning. In the inductive transfer learning approaches, due to difference in the conditional probabilities, at least a small set of target domain labeled data are required for training an accurate predictive model [58, 62].

### 4.12.5 Feature-based transfer learning

In many real-world scenarios, features in the source domain and target domain are not completely overlapping. This is where feature-based transfer learning can be used. This approach allows transfer learning to be operable in an abstract feature space.

One approach to this problem would be to identify mapping functions $\{\varphi_s(\cdot), \varphi_t(\cdot)\}$ to map data from both domains to a common feature space, where the difference between domains can be reduced (or be minimum). After that, a target classifier is trained on the new feature space with the mapped data.

In many real-world cases, observed data in a high dimensional space can be mostly described by a set of latent factors or principal components constructed upon those factors. The mentioned factors can be considered as features. The difference that exists

between the domains is caused by a subset of these features. By identifying the latent features that do not cause the difference between the domains and representing the data by them, one would be able to train an accurate classifier in the target domain from the source domain training data [58, 62].

### 4.12.6 Model-based transfer learning

In this approach, which is mostly popular when combined with well-trained deep learning models, the assumption is that the transferred knowledge is encoded into the model parameters. This approach reuses the model trained in the source domain to learn the new tasks in the target domain more efficiently. Model-based transfer assumes that the source model $\theta_s$ is well-trained, meaning that it has learned a lot of the structure from data. Thus, for a related task in the target domain, this structure can be transferred for training a precise target model $\theta_t$ with labeled data in the target domain. This would result in a more powerful model in the target domain and avoids overfitting when there is limited labeled data available [58, 62].

Based on different assumptions, two model-based categories can be proposed: knowledge transfer by sharing model components and knowledge transfer by regularization. The first category is referred to algorithms that reuse some components or some hyperparameters of the source-domain model to create a target-domain model [63-65]. The second category is referred to algorithms that apply constraints on parameters based on prior hypotheses. This is an approach used to solve ill-posed machine learning problems. When deep learning models are used, parameters of a pre-trained deep learning model can be used for initialization of the target domain model(s).

**Transfer through shared model components:**

1. Transfer via Gaussian process:

This process is used to model the data distribution with a Gaussian prior probability distribution. When the data is labeled, the modeled distribution can be used for the prediction of test data by the similarities that is measurable in the training set. The prior distribution over the latent variables can be described by a Gaussian prior as

$$p(z|X,\theta) = N(0,K) \qquad 4.8$$

where $\theta$ denotes the parameters and *K* is the covariance function to depict a multivariate normal distribution. Considering the latent variables to be $z = [z_1, z_2 \dots z_N]^T$ The joint likelihood of overall data can be formulated as

$$p(y, z|X, \theta) = p(z|X, \theta) \prod p(y_i|z_i) \quad 4.9$$

Using equation 4.9 and assuming M different related tasks on the corresponding training data, one can define a multitask Gaussian process by constraining the covariance matrix K to be a block diagonal matrix. Also, to increase the training speed, information vector machines can be used for conducting a sparse representation [66].

2. Transfer via Bayesian Models

Beside the Gaussian, other distributions can also be used for model-based transfer learning. One study [63] proposes an algorithm that transfers the priors to estimate the parameter distribution of some target domain objects in images based on a Bayesian method. Transferring the prior would also reduce the amount of necessary labeled data so that learning a new category can be achieved by only single or a few examples.

3. Transfer via Deep Models

In this method one would average over the softmax of all activations of source examples in the category *k* to distill a soft label *l*. Then the training will be accomplished using these soft labels, hence, the relationship between the instances in the feature space would be closer than the situation where soft labels are not used [58, 62].

**Transfer through regularization:**

The standard form of regularization in a model is:

$$\tilde{J}(\theta; x, y) = J(\theta, x, y) + \alpha\Omega(\theta) \quad 4.10$$

where $J$ is the original objective function and $\tilde{J}$ is the regularized objective function with the regularization term $\Omega(\cdot)$ and $\alpha$ is the regularization weight. One can divide parameters into a task-specific part and a task-invariant part [67], therefore:

$$\begin{cases} \theta_s = \theta_0 + v_s \\ \theta_t = \theta_0 + v_t \end{cases} \quad 4.11$$

$\theta_0$ denotes the task-invariant part and $v$ the task-specific parameters. By identifying and using the task-invariant parameters, one can transfer the knowledge and improve the target model.

**Fine-tuning approaches for deep models**

One method for fine-tuning is to first initialize specific tasks by applying the parameters trained using unsupervised learning [68], and then fine-tuning with the supervised instances.

Another option would be to use dropouts and batch normalization to train all (or certain) layers using the new data. One can directly train the whole model or to select specific layers to conduct the training by a stable optimization method and large number of labeled instances. The transferability of different layers of a pretrained convolutional neural network CNN model has been investigated in a study by Yosinski et al. [69].

### 4.12.7 Relation-based transfer learning

There are many real-world situations, in which domains often contain some structures among the data instances. This means there are relational structures in the domains. In such domains, instances are related with multiple relations. When information can be acquired from related domains, the need for large data for training a new model will be banished and the knowledge from the relations will be useful to transferred for improving the learning process in the target domain [58, 62].

Relation-based transfer learning seeks forming the mapping of the relational knowledge between the source relational domain and the target relational domain. It assumes common regularities between relations among the source domain and the target domain data [58].

Two approaches can be taken for a relation-based transfer learning, first order relation-based and second-order relation-based. First-order transfer assumes that related domains may share some similar relations among data instances. These similar relations can then be transferred across the domains. Second-order transfer assumes that two related relational domains share some similar relation-independent structural regularities that can be extracted from the source domain [58]. The transferred knowledge is within the regularities that would be assigned in the target domain.

### 4.12.8 Adversarial transfer learning

Generative modeling in machine learning can lead to adversarial models. There are, usually, abundant unlabeled data available in the source domain, and the labeled data in the target domain may be limited. If the representations are to be obtained, unsupervised

learning can be used on unlabeled data. In order to transfer the knowledge of the learned features, generative models can be applied [70, 71].

Generative models can be presented in two styles, explicit and implicit. Explicit models have specified density functions and their parameters are estimated via the principle of maximum likelihood. Implicit models act like a simulator; they follow the underlying data distribution by generating new samples.

One of the successful generative models that have been applied to many tasks is the generative adversarial networks (GANs) model [72].

### 4.12.9 Similar studies

Many studies have been conducted to apply transfer learning in histological image classification and regression analysis; some of these studies and their methodology will be mentioned in this section.

Wang et al. [73] introduced a novel approach combining transfer learning and deep learning to predict Tumor Mutation Burden (TMB) from histological images. Their proposed method includes four stages. First, the non-synonymous mutations in the coding region are calculated for obtaining the TMB value for each patient. Second, tumor patches are tagged. Third, a convolutional neural network based on transfer learning is established to classify patients' patches. Finally, by calculating the ratio of TMB-high patches to all patches, prediction results are achieved.

Vesal et al. [74] used transfer learning with two pretrained CNNs Google's Inception-V3 [75] and ResNet50 [55] to classify breast cancer histology images. They have applied fine-tuning to update the initialized weights of the models to enable the network to learn features specific to their task of interest. They have reported successful models with accuracies of approximately 97%.

Hosseinzadeh Kassani et al. [76] also used transfer learning for breast cancer diagnosis using histology images. They have proposed a method based on deep convolution neural network. In their first step they have used a preprocessing technique for stain normalization. Following the normalization, they have used data augmentation procedures to address the issue of limited size of dataset and improve the training. Then high-level features are extracted from the preprocessed images by applying proposed network architecture from well-stablished deep CNNs. Then these features are used as an input to a standard multilayer perceptron classifier. Finally, the test image dataset is used for evaluation of the model performance.

Ohata et al. [77] have also used transfer learning for the classification of histological images of colorectal cancer. Similar to the previously mentioned study, they have used different pretrained CNNs for the purpose of feature extraction and then applied well-known classifiers (Naive Bayes, multilayer perceptron, k-nearest neighbor, random forest, and SVM) for classification using the quantified extracted features.

# 5 Aims and hypothesis of the thesis

A study conducted in 2020 stated that the global incidence of knee OA was 203 per 10,000 person-years in individuals aged 20 and over [78]. This means that there are around 86·7 million individuals (20 years and older) with incident knee OA. This highlights the importance of early-stage diagnosis of this disease. It is also known that arthroscopic synovial biopsy is simple and easy to perform technique and is an important investigative method that may give conclusive and definitive diagnosis [79].

In this thesis, model-based transfer learning through deep learning and fine-tuning was used on pretrained CNN models (VGG, Resnet, and AlexNet) to predict the severity of OA in human cadaveric cartilage tissues digitized by histology imaging. Models were further trained and fine-tuned on a dataset of 31020 histology images (after augmentation) with a relatively large variation in the OA stage (0-6 in OARSI system scale). The model is deployed online and publicly accessible via a web app.

This thesis hypothesizes that an automated and reliable grading system can be developed for predicting the severity of OA using histological images. Main objectives of this thesis are to develop models with significant performance which are less prone to the existing limitations. In addition to the main objectives, a novel bone-deletion algorithm is developed to remove the calcified parts of the tissue sections in histology images. The impact of this algorithm on the neural network model's performance was investigated and compared in various scenarios. In order to conduct further analysis, the whole dataset was constructed in two styles: using sliding windows on images and using full histological sections. The effects of both assemblies were investigated and compared.

# 6 Methods

## 6.1 Histopathological grading

### 6.1.1 Sample collection

The samples used in this study were harvested from nine human cadaver knee joints. In the harvesting process, osteochondral (i.e., cartilage-on-bone) samples were collected from different anatomical locations, including the tibia, femur, and patella. An experienced orthopedic surgeon at the University of Eastern Finland performed the extraction. During the procedure, knee joints were distended with saline solution (25 ℃, 0.9% NaCl concentration). After removing the osteochondral blocks, they were frozen at -20 ℃ in phosphate-buffered saline (PBS) for future histology assessment [80].

### 6.1.2 Histology assessment

To subject the osteochondral samples to histology assessment, they were first thawed to room temperature. Afterward, they were fixed in a solution containing formaldehyde (4%, Merck, Darmstadt, Germany) and ethylenediaminetetraacetic acid (EDTA, 10%, Merck, Darmstadt, Germany) at room temperature for 21 days. The purpose of this fixation was to decalcify the underlying bone of the osteochondral samples for a better sample sectioning. Following decalcification, the samples were processed and embedded in paraffin [81]. 3-µm thick histological sections were cut perpendicular to the articular surface and stained with Safranin-O [81]. Samples were then transferred for scanning and scoring. Each section was then scanned and digitized (Hamamatsu-C12000-02-NDP scan, lens magnification=35.16), resulting in a total of 3102 images.

### 6.1.3 Histology scoring

Scoring of the sections was accomplished by four trained experts using Mankin and OARSI systems for their evaluations. To reduce the inter-observer variability and bias in

their scores, images were labeled randomly and scoring procedures were repeated three times with a two-week time gap. This means for each histology image, there was three observations from each scorer resulting in 12 individual scores. The final score of the section and its true condition were considered the average of these scores.

## 6.2 Preprocessing

A series of preprocessing steps were applied to all histology images prior to developing the predictive models. These steps include rotation, bone-deletion, augmentation, and windowing. All the computational analyses were performed using Python (v.3.8.10) and related libraries: OpenCV (v.4.5.3.56), [82], NumPy (v.1.19.5) [83], PIL (v.7.0.0) [84], PyTorch (v.1.9.0) [85], and Fastai (v.1.0.61) [86]. The effectiveness of using these preprocessing methods is fully investigated.

### 6.2.1 Rotation

The bone-deletion algorithm and model development can both benefit from horizontally arranged histology images. This was not practically achievable during the scanning process. Hence, a post-processing step was needed to adjust them accordingly for further analysis. To this end, a sequence of image processing algorithms was utilized to obtain the best rotation angle, then applied on all the input images to rotate them to an acceptably horizontal orientation. Original sizes of the input images were initially very large, presenting a very good resolution. Since that resolution is not required for finding the angle of rotation, the sequence begins with reducing the image size to 8 percent of the original size. The sequence is then followed by translating the image into an 8-bit grayscale image, then blurring by a Gaussian filter (kernel size 7*7 with mirrored boundaries) and a median filter (kernel size 5*5). Next, a threshold is set on the image, and a morphological gradient is applied. The morphological gradient is the difference between dilation and erosion of an image. In other words, it would provide a shallow structure of the subject. Afterward, a dilation is used on the morphological gradient. Then contours are employed to find the smallest rectangle that covers the

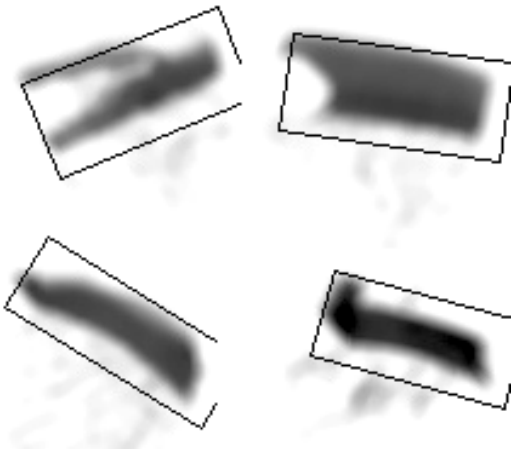

Figure 6.1: Drawing the box on the blurred image for angle detection.

structure. This rectangle can suggest the angle of the structure (figure 6.1). The angle between the lower edge of the rectangle and the horizontal line is the structure's orientation angle. Lastly, this angle is used to rotate the image. It is noteworthy that the threshold level helps to ignore the subchondral part. Since the pictures are in JPG format, the alpha channel cannot be applied for them during the rotation. Thus, background is modified accordingly.

### 6.2.2 Bone and Artifact Deletion

The subchondral bone attached to the cartilage, does not carry any important information about the severity of OA. The pixels including these parts can distract activation sites leading to less accurate prediction. Also, some of the images involved artifacts with unusual color values that could similarly affect the predictions. Hence, one of the critical parts of improving the classification and regression models was to eliminate parts that were unnecessary to our problem. An algorithm was developed and applied to all input images to delete the subchondral bone from cartilage tissues in the histology images. The algorithm is based on the color, structure, color saturation, and relative location of the elements. The impact of this algorithm on the performance of the models is investigated and reported in the results and discussion sections. The algorithm uses singular value decomposition (SVD) with a negative filter on the SVD values (changing values greater than zero to a negative value). Then it reconstructs a simpler version of the image applying 1 or 2 rows of the resulting values (figure 6.2) and uses that reconstruction to find the border separating the subchondral bone from the cartilage. The cross-section of the components with darker pixels is where cartilage is mostly present in the image. Usually, the bone segments are below this line. The reconstructed image is only used to find these separating lines with acceptable accuracy Then it replaces the pixels below this line (that have color values lower than a certain threshold) with white pixels. These separating lines are also used to crop the image to remove the unnecessary white space. Rotation and bone deletion algorithms are combined and uploaded to google colab for open access use:

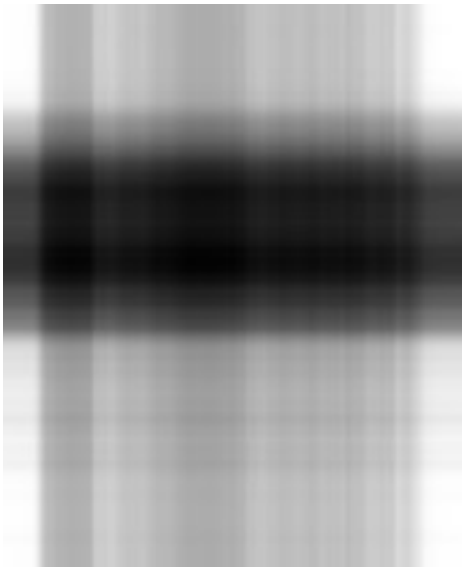

Figure 6.2: A sample SVD reconstruction.

https://colab.research.google.com/drive/1D6nTOvIhRJE3GTQ5-Ha7GHN9l-ipjWjQ?usp=sharing

### 6.2.3 Augmentation and Windowing

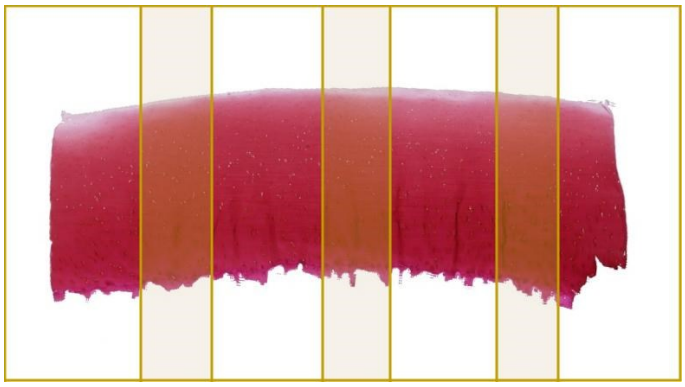

Figure 6.3: Windowing of the processed image. Highlighted sections indicate the overlapping parts.

Dataset augmentation has been shown to reduce the risk of overfitting in deep learning models [87]. Augmentation typically consists of adding rotated and flipped copies of the original images into the training dataset. In this thesis, we performed a tenfold augmentation by five ±1 and ±2 degrees of rotation along with respective flipped images. Various augmentation sizes were tested using cross validation, the tenfold size proved to avoid overfitting and showed the best results. Previously mentioned method of augmentation – sliding windows – will be called windowing throughout this text. To increase the number of input images even more, windowing has been performed on each augmented section, so that each image is divided into *n* overlapping parts (figure 6.3). The width of each window is set to be one-third of the input image (wl = width/3). Thus, the size of the overlaping section of the windows would be equal to

$$a = \frac{(n \times wl - width)}{n-1} \qquad 6.1$$

Where n is the number of windows. Using cross validation over all input images, the best value of n was found to be 4. Different models were created within different scenarios, some of which did not contain windowing as a part of their preprocessing.

## 6.3 Deep learning-based scoring

To develop an automated histology scoring system, predictive classification and regression models based on pre-trained CNN models including VGG [52], ResNet [55], and AlexNet [50], were developed using transfer learning. In these models, the histology images were input while their associated histology scores (Mankin and OARSI) were output. To achieve the best models, different scenarios were tested. These scenarios are

including or excluding the followings: windowing, bone-deletion, and the image sets with the abnormal low-level features. In each scenario, the CNN models were trained via a cross-validation scheme to ensure the prevention of overfitting. Moreover, various metrics including Mean Squared Error Percentage (MSEP), accuracy, and precision were utilized to assess the performance of the CNN models in these scenarios. Furthermore, the best-performing models were packaged for web deployment. The details of the employed dataset, cross-validation scheme, performance metrics, and web deployment are presented in the following subsections.

### 6.3.1 Dataset

Our dataset consisted of 3102 histopathological images. These images were gathered from knee joints of nine cadavers (nine groups of images). The images from cadavers 8 and 9 were recorded using a different camera setting, hence showing clearly different low-level image features (e.g., contrast and color).

These images were labeled by Mankin and OARSI scoring systems and all possible scores were observed in the dataset. Regression models were developed to predict a real number value in the range of these scoring systems (0-14 for Mankin, 0-6 for OARSI). Figure 6.4 shows the distribution of the Mankin scores in the dataset. The classification model was developed for the assessment of tissue integrity into three different classes based on specific thresholds applied to the Mankin score (mild: scores<4, moderate: 4<scores<7, advanced: scores>7). To investigate the effect of preprocessing and camera settings, several scenarios were generated:

- Scenario 1: Only images from the first seven cadavers that showed similar camera features were included (groups 1-7), windowing, rotation, and bone-deletion, were applied on all images. Training data consisted of 80 percent of the images. Training/validation split was performed randomly (plus shuffling) from all seven groups (optimal validation size was achieved by cross-validation).
- Scenario 2: All nine groups of images were included, windowing, rotation, and bone-deletion were applied on all images. The validation set was made from groups 5 and 8.
- Scenario 3: All nine groups of images were included; windowing was applied on all images. The validation set was made from groups 5 and 8.
- Scenario 4: All nine groups of images were included, rotation and bone-deletion were applied on all images. The validation set was made from groups 5 and 8.
- Scenario 5: All nine groups of images were included, none of windowing, rotation, or bone-deletion was applied. The validation set was made from groups 5 and 8.

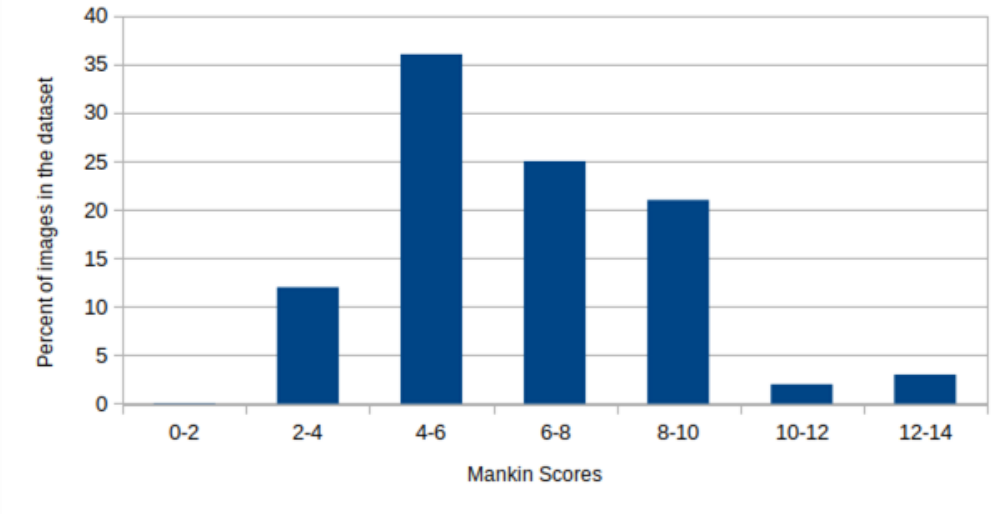


Figure 6.4: Distribution of scores in the dataset.

### 6.3.2 Learning procedure

The models used in this study include VGG16, VGG19, ResNet18, ResNet50, and AlexNet. Due to the nature of parameter-base transfer learning, in a series of attempts different layers of the pre-trained models were “frozen” (not to be affected by the new training procedure). Different epoch values for each model were tested and chosen properly to avoid overfitting. The test was conducted with up to 24 epochs. Overfitting could be observed for almost all models at epochs greater than 5. A constant size of 224x224 pixels was used as the input for all images, and different values were tested for each model's batch. The Fastai library has a built-in function for the selection of the optimal batch size based on an initial evaluation. These values did not always show the best performance in several trials for different models; hence, the best batch size value was selected after numerous trials. After this initial learning that adjusts the weights in the unfrozen layers, fine-tuning was performed in a series of 1, 2, and 3 repetitions. We adjusted the weights of the new CNN, built on the pre-trained model, using the unfreeze function and the fit cycle. The purpose of this action is to hyper-tune all the layers of the CNN instead of keeping some of the earlier layers fixed (due to overfitting) and only fine-tune some higher layers of the CNN. This is motivated by the observation that the first layers of a CNN include more generic features which are helpful to various machine vision applications, whereas the last layers of the CNN tend to detect more specific features in the dataset.

Two approaches were taken for the classification problem:

a) Multi-class (3-class classification), and

b) Nested binary (2 + 2) classification.

The schematic of these approaches was depicted in Figure 6.5. In the 3-class classification approach, images were labeled as one of the three classes of "mild", "moderate", and "advanced". While in the second approach, the histology images were first classified as “healthy” and “unhealthy”, then at the second stage the unhealthy images were classified as “moderate” and “advanced”.

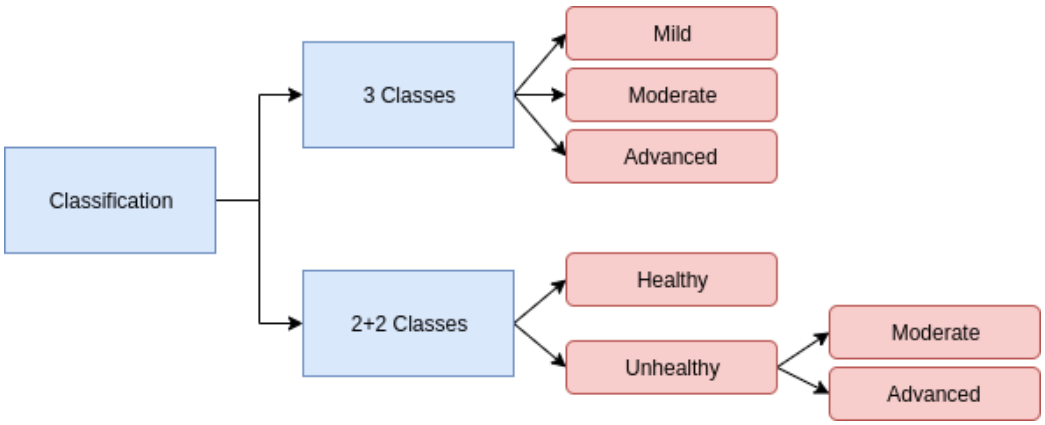


Figure 6.5: Classification methods.

The set of images consisted of two distinct groups with different low-level image features. Based on this difference, two types of tests were conducted, one using only the group with similar features that made 7 out of 9 groups of cadaver images, and the other with all image sets (including the two abnormal groups). In the scenario where only normal groups were evaluated, training/validation splits were followed with 20% for validation and 80% for training, a value that was achieved from fixed-sized standard cross-validation (this was done to include an acceptable distribution of scores within the training set). All images were separated based on their scores; hence, this splitting would result in a stratified division based on their scores. Where the abnormal groups were included (scenarios 2 to 5), the split was conducted differently. The validation and training data each had one set belonging to the abnormal groups, with other images from the normal groups. Also, to avoid biased models, training and validation sets were built upon different cadaver images.

After generating the dropout layers, all the classification models feedworad their output to the softmax layer. The softmax layer at the end uses the softmax function, also known as softargmax, or normalized exponential function, whichis a generalization of the logistic

function to multiple dimensions. The output layer of the models includes a softmax activation function to map the output of the network to a probability distribution over the target classes. The softmax function receives a vector of real numbers as input and normalizes it into a probability distribution consisting of K (the number of classes) probabilities proportional to the exponentials of the input numbers. Before the softmax layer is applied, some vector components might be negative or greater than 1, but after applying softmax, the components will be normalized onto the interval of [0, 1] with their sum equal to 1, thus they can be interpreted as probabilities. The larger input components will correspond to larger probabilities. Before this function is applied, the vector of the numbers is not limited to any class and is distributed within a certain linear range, at this point (before the softmax layer), this vector can be extracted to be regarded as a regression output.

## 6.4 Model performance analysis

In classification problems, related metrics measure how well the model can assign true labels to the data. In regression problems, the metrics measure how compatible the predicted values are with true scores. To assess the performance of the models, the following metrics have been utilized. The best-performing model was chosen to have the best performance in these metrics.

### 6.4.1 Classification

Precision:

Precision is defined as follows:

$$precision = \frac{tp}{(tp + fp)} \tag{6.2}$$

where *tp* represents the number of true positives and *fp* is the number of false positives. To give an intuitive definition of precision one can interpret it as the ability of the classifier not to label as positive a sample that is negative. This score also returns a value between 0 and 1 where 1 would be the best possible outcome.

Accuracy and error rate:

Both represent the same concept; the accuracy measures the proportional number of correct predictions (that are equal to the true scores), and the error rate is defined as:

$$error = 1 - accuracy \quad (6.3)$$

When the 2+2 class classification is used, accuracy is reported for each classification step. Since the classification is nested, the accuracy of moderate/advanced (second part) is reported multiplied by the accuracy of healthy/unhealthy classification.

### 6.4.2 Regression

To evaluate the performance of the regression model, two metrics were used: mean squared error percentage (MSEP) and standard deviation (SD) of the difference between the true and predicted scores. MSEP calculates the error between predicted and true values over the max possible value of the scoring range according to the equation:

$$MSEP = \left(\frac{1}{N}\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2\right)/R \quad (6.4)$$

Where N is the number of data points, $Y_i$ and $\hat{Y}_i$ are the true and predicted scores, respectively, and R is the scoring system range. Standard deviation is calculated according to the equation:

$$SD = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2} \quad (6.5)$$

Where N is the number of data points, $x_i$ is the absolute difference between the given score and the true score for the slice *i*, and $\mu$ is the average of the difference array. The true score is considered to be the average of all scores given by the observers.

## 6.5 CAMs and visualization of activations

### 6.5.1 Class activation maps

Convolutional units of various layers of CNNs can be used as object detectors without providing any information about the location of the object [88]. This ability, however, is lost when fully-connected layers are used for classification. Recently, new architectures such as the Network in Network (NIN) [89] and GoogLeNet [90] have been proposed that avoid the use of fully-connected layers to minimize the number of parameters while maintaining high performance.

By using a concept called global average pooling, one study [91] successfully used advantages of this global average pooling layer for localization ability until the last layer. This would allow identification of regions in a single forward-pass for a wide variety of tasks, even those that the network was not originally trained for. This method is called class activation mapping and it points to the most important regions leading to a particular network output (e.g., a class).

### 6.5.2 Visualization of activation layers

After the image is fed to the network, the representation of the layer is represented and forwarded through the softmax function, then the activation probabilities of the layer are obtained. Neuron activations in fully connected layers are visualized for specific filters and layers [92]. It can be used to distinguish features that lead to certain outputs.

## 6.6 Standard deviation of difference

Standard deviation is calculated according to the equation:

$$SD = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2} \qquad 6.6$$

Where N is the number of data points, $x_i$ is the absolute difference between the given score and the true score for the slice *i*, and $\mu$ is the average of the difference array. True score is considered to be the average of all scores given by the observers.

## 6.7 Model deployment

The best performing models were deployed online using Python and Streamlit [93]. Streamlit allows the creation of web applications via a Python integrated development environment. Moreover, it is based on the Tornado and Flask web frameworks. The web application was first deployed and tested on a local machine and then transferred to an online server. Streamlit framework allows some manipulation of the front-end to a limited extent. Further adjustments were made using the built-in markdown function. Models are exported to the streamlit.io servers where the application is hosted. The models are then loaded to the application for prediction when a user runs the function. streamlit framework also provides an API for front-end web-design materials, however, many components had to be added manually using Javascript, CSS, and HTML codes. The application has been

deployed to streamlit.io servers where free machine learning and data analytics applications can be served for free. Many user interface components are made directly by streamlit utilities such as st.sidebar, st.write, and st.selectbox. Limitations on the file upload and used libraries had to be introduced to the serving host directly. Necessary codes and files were pushed to a GitHub page and deployed to the server (figure 6.6). The client is provided with basic image processing and data analysis utilities such as image resizing, and data preprocessing. The application has also a size limitation for image uploading.

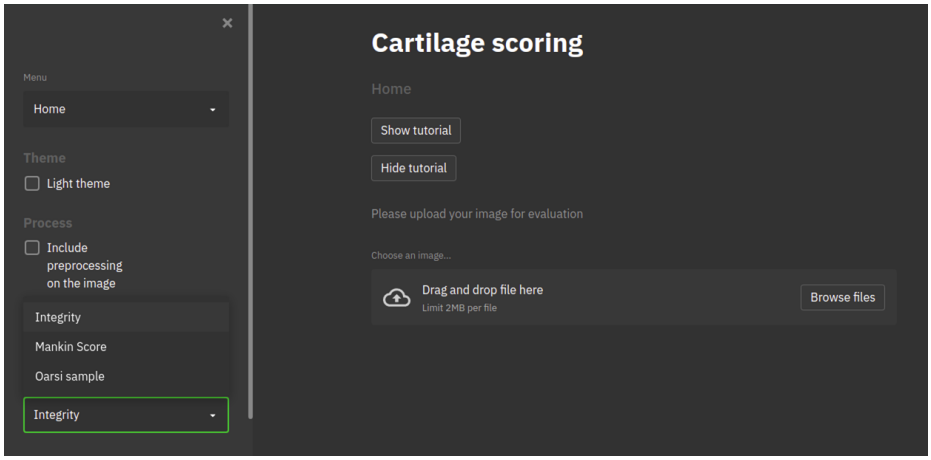


Figure 6.6: Illustration of the web application.

# 7 Results

## 7.1 Preprocessing

The rotation algorithm successfully adjusted the angle of 95% of the images to achieve horizontal surface, the remaining 5% were adjusted manually. Figure 7.1 shows a sample set of images after they were corrected by this algorithm.

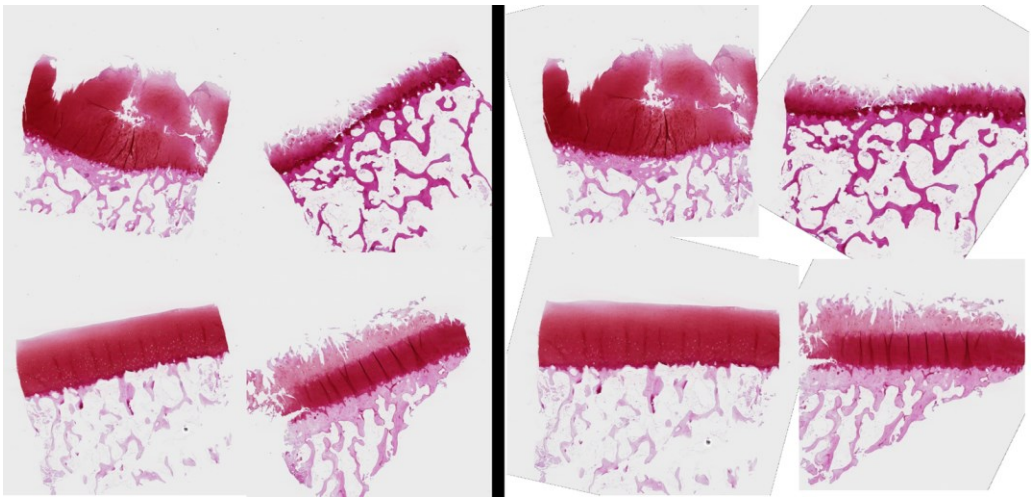

Figure 7.1: Before and after the rotation algorithm was

The bone deletion algorithm successfully removed the subchondral bone from the samples in 2341 images. However, few stained parts of the bone were not distinguishable from cartilage in 761 images, either the deletion was incomplete or had affected the uncalcified parts. In case of incomplete deletion, they were corrected manually. Figure 7.2 shows one sample of this algorithm's result.

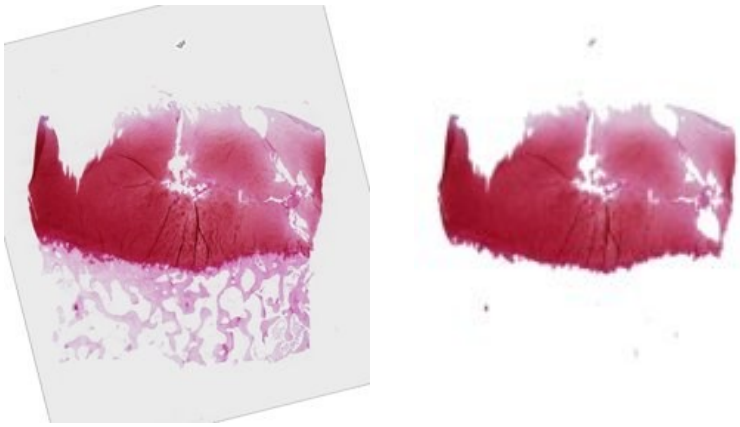

Figure 7.2: Before and after the bone deletion algorithm was applied.

## 7.2 Excluding the abnormal groups

Developing the 3-class classifier excluding the abnormal groups resulted in 88% accuracy with the ResNet18-based model following scenario 1. When the nested binary classification approach was taken, the accuracy increased by 8 percent and reached about 96% using the AlexNet-based model.

The regression model based on ResNet50 for OARSI score prediction resulted in a 1% MSEP.

Also, for the Mankin score predictions, the VGG19-based model achieved 2.1% MSEP after 8 epochs following the similar method (figure 7.3).

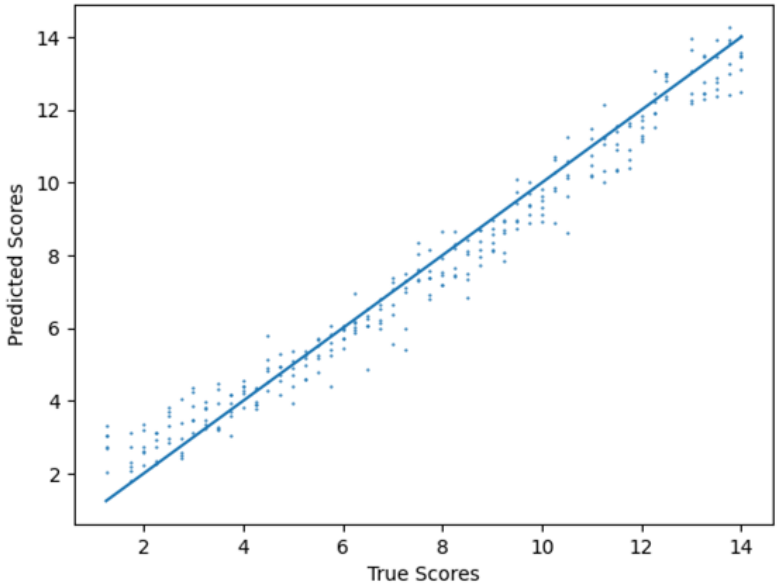


Figure 7.3: Performance of a regression model based on VGG19 (predicting Mankin scores), following the first scenario instructions (section 6.3.1).

## 7.3 Including the abnormal groups

When the abnormal groups are included, models are not as accurate as the contrary situation. However, since this situation is common in real-world problems, it is investigated more thoroughly. Following sections summarize the results related to scenarios 1-4. After an initial evaluation, best models were chosen for further investigation. Thus, parameters that could improve the model were thoroughly analyzed and details are provided in appendix I.

### 7.3.1 Classification

Among all investigated scenarios that included the abnormal group (section 6.3.1), the fourth scenario resulted in the best performing models. Detailed results of each scenario are provided in appendix II.

Table 7.1: Result (validation sets) comparison of the predictive classifier models based on VGG19, AlexNet, and ResNet. M/A indicates moderate/advanced, and H/U indicates healthy/unhealthy classes.

| 3 class | | | | | 2+2 class | | | |
|---|---|---|---|---|---|---|---|---|
| | preprocessing | | | | preprocessing | | | |
| **Model** | none | rotation and bone-deletion | windowing | all included | none | rotation and bone-deletion | windowing | all included |
| **VGG19** | 0.62 | 0.69 | 0.58 | 0.67 | M/A: 0.68<br>H/U: 0.71 | M/A: 0.87<br>H/U: 0.88 | M/A: 0.62<br>H/U: 0.64 | M/A: 0.80<br>H/U: 0.84 |
| **ResNet18** | 0.66 | 0.74 | 0.59 | 0.68 | M/A: 0.72<br>H/U: 0.79 | M/A: 0.87<br>H/U: 0.88 | M/A: 0.63<br>H/U: 0.66 | M/A: 0.78<br>H/U: 0.85 |
| **ResNet50** | 0.63 | 0.72 | 0.61 | 0.65 | M/A: 0.71<br>H/U: 0.73 | M/A: 0.86<br>H/U: 0.85 | M/A: 0.68<br>H/U: 0.70 | M/A: 0.79<br>H/U: 0.84 |
| **AlexNet** | 0.65 | 0.78 | 0.63 | 0.69 | M/A: 0.71<br>H/U: 0.75 | M/A: 0.88<br>H/U: 0.94 | M/A: 0.66<br>H/U: 0.73 | M/A: 0.80<br>H/U: 0.91 |

### 7.3.2 Regression

Similar to classifications, among the scenarios that included the abnormal group, the fourth scenario resulted in the best performing models. Detailed results of each scenario are provided in appendix II.

Table 7.2: Result (validation sets) comparison of the predictive regressor models based on VGG19, AlexNet, and ResNet.

| | Mankin | | | | OARSI | | | |
|---|---|---|---|---|---|---|---|---|
| | preprocessing | | | | preprocessing | | | |
| **Model** | none | rotation and bone-deletion | windowing | all included | none | rotation and bone-deletion | windowing | all included |
| **VGG19** | 13.0% | 6.6% | 17.6% | 8.6% | 4.0% | 3.0% | 9.1% | 7.6% |
| **ResNet18** | 13.0% | 6.5% | 18.0% | 11.6% | 4.2% | 2.8% | 9.3% | 7.6% |
| **ResNet50** | 12.7% | 5.7% | 17.3% | 11.7% | 4.1% | 2.5% | 8.9% | 8.0% |
| **AlexNet** | 13.8% | 6.8% | 18.6% | 11.8% | 3.6% | 3.0% | 8.9% | 6.8% |

Figure 7.4 shows the performance of the ResNet50 model created with the non-windowed approach. The true score is considered to be the average of all observations made by experts.

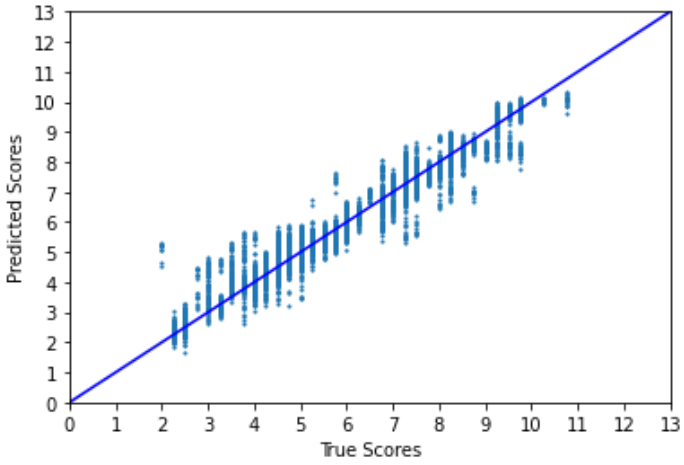


Figure 7.4: Performance of the regression model transferred from ResNet50.

### 7.3.3 Preprocessing effectiveness

Figure 7.5 shows the comparison of the effectiveness of the rotation and bone-deletion algorithm on the MSEP when applied to non-windowed cases for data related to Mankin and OARSI scoring models.

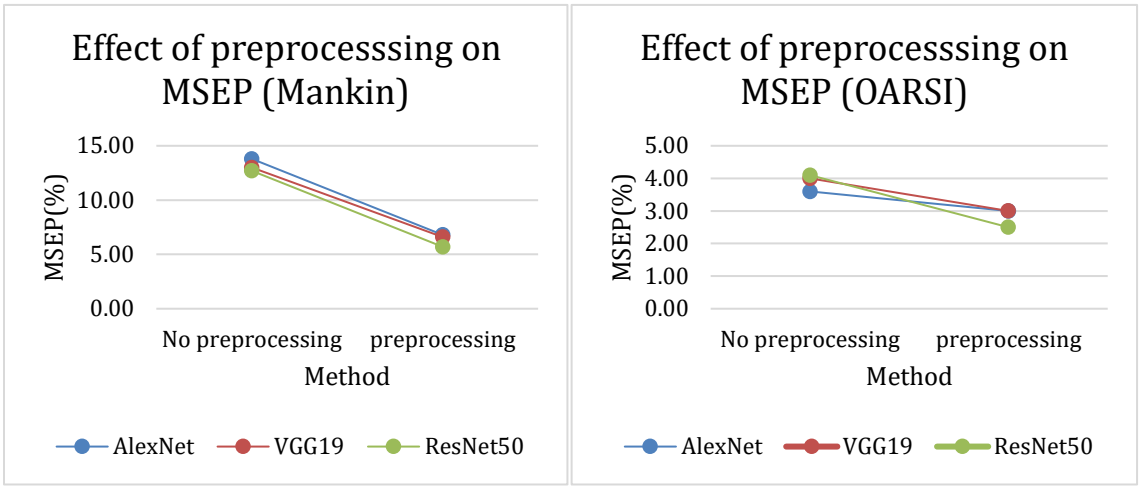


Figure 7.5: Effectiveness of the bone deletion on Mankin and OARSI models.

## 7.4 Visualization of activations and CAMs

Figures 7.6 shows the activations of the middle layer of the AlexNet-based model developed for the classification problem. Activations are compared using two images, each being presented with and without the preprocessing. Figure 7.7 shows the class activation maps and indicate how the preprocessing is affecting the activation sites. Most of the activation sites are originating form subchondral bone sections when bone deletion

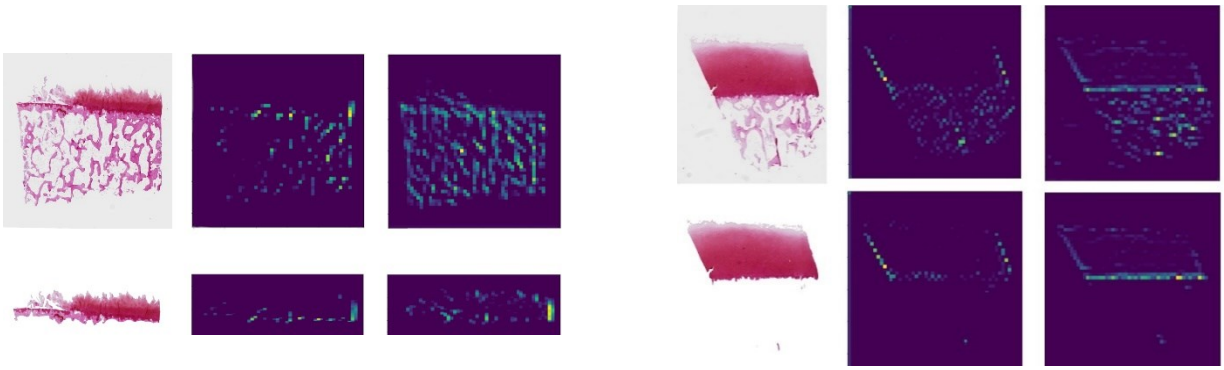

Figure 7.6: Visualization of activations of the middle layer of the AlexNet model.

is not applied. Figures 7.8 and 7.9 visualize activations in some layers in ResNet50-based models for Mankin and OARSI predictions respectively.

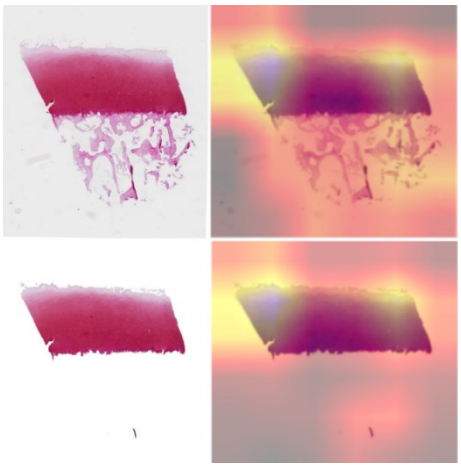

Figure 7.7: Class activation maps
before and after preprocessing.

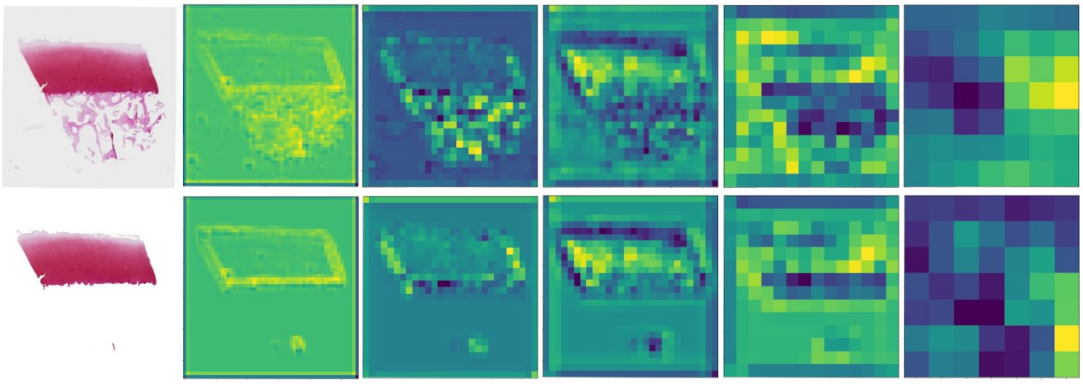

Figure 7.8: Visualization of activations in different layers (Mankin).

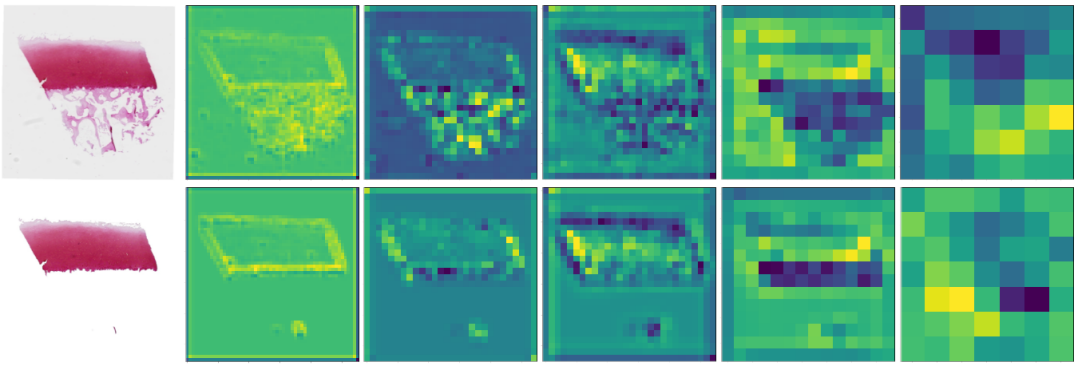

Figure 7.9: Visualization of activations in different layers (OARSI).

## 7.5 Inter-observer variability and comparison with the developed models

To investigate how consistent the human-observer given scores are, the average score of each section given by each observer was compared with other observers' average values, and with the true values. True values are considered to be the average of all the scores given by all observers (Figure 7.10, Figure 7.11).

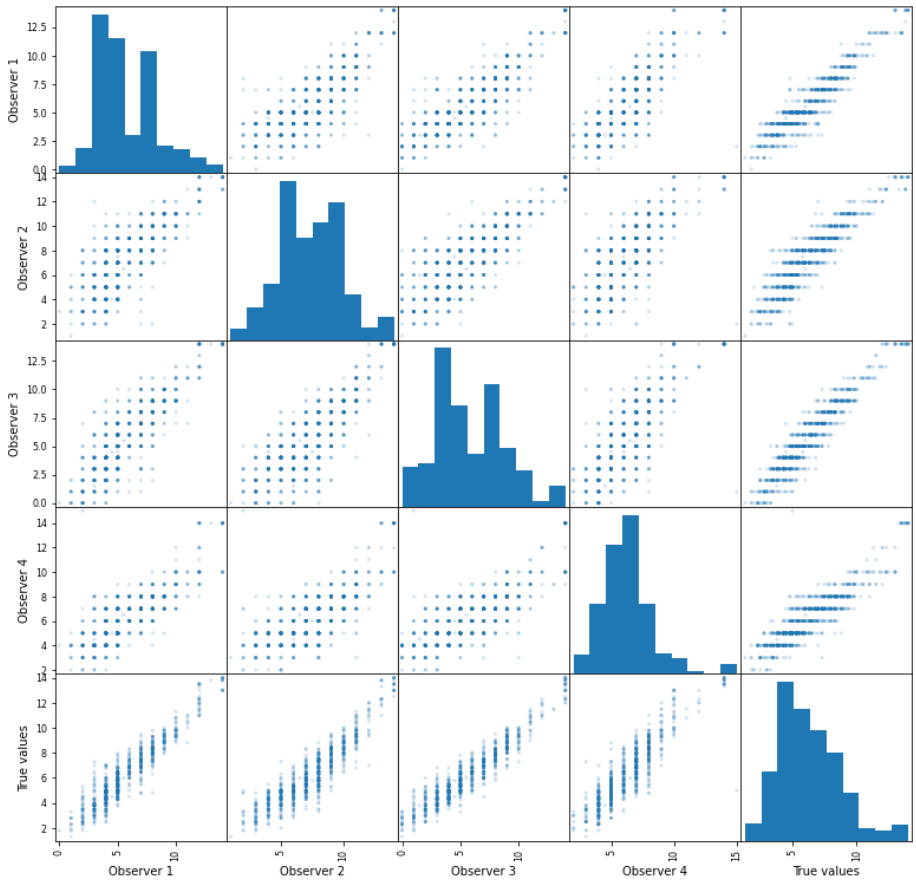


Figure 7.10: Predicted values of the Mankin scores by transferred ResNet50 model versus true scores.

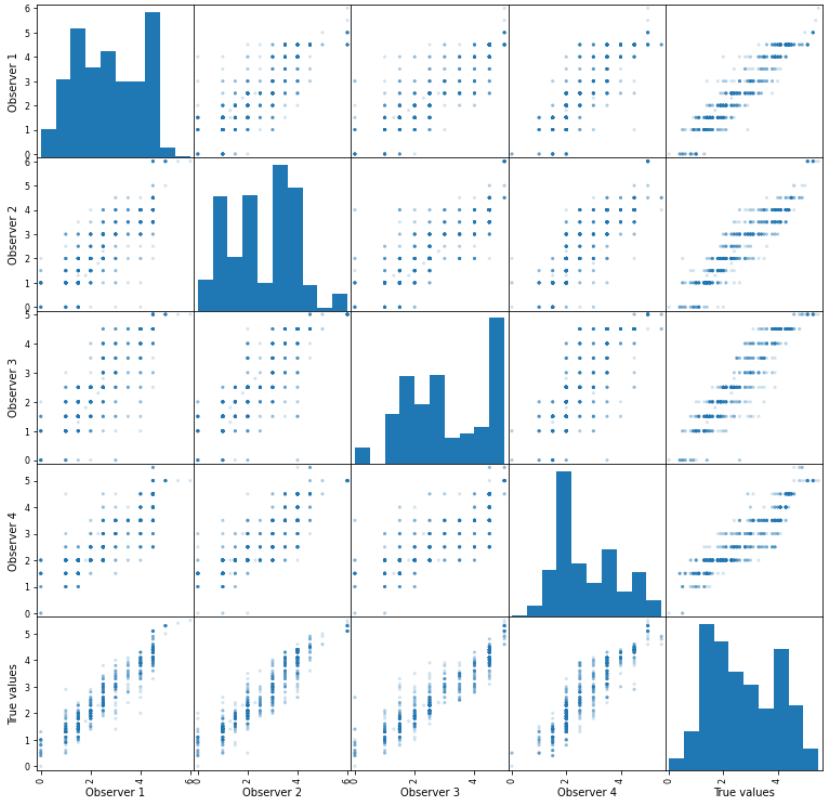


Figure 7.11: Predicted values of the OARSI scores by transferred ResNet50 model versus true scores.

In addition, to evaluate the reliability of the AI-driven models, the scores of these models on the test set were also compared with the true values (Figure 7.12). Table 7.3 represents the SD values of the difference between each scorer's results and the true value.

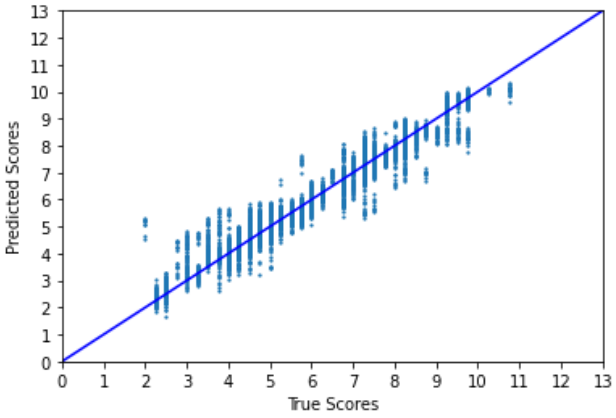


Figure 7.12: Predicted values of the Mankin scores by transferred ResNet50 model versus true scores.

Table 7.3: Standard deviation of absolute difference for each scorer.

| Scorer | Mankin scoring system | OARSI scoring system |
|---|---|---|
| **Human observer #1** | 0.63 | 0.30 |
| **Human observer #1** | 0.95 | 0.30 |
| **Human observer #1** | 0.85 | 0.37 |
| **Human observer #1** | 0.64 | 0.31 |
| **AI-driven model** | 0.34 | 0.15 |

## 7.6 Web deployment

The application has been deployed to the following URL:

https://share.streamlit.io/soroushoskouei/deephistology/DeepHisto.py.

Figure 7.13 is a screenshot that shows different parts of the application home page. The web application features a tutorial section, and below that is an upload button. In the left-hand bar, the user is asked if preprocessing is desired on the image. Below this checkbox, the desired output and the type of sample can be selected. The bovine sample option will be developed and selectable later. After uploading the image, a reduction option can be found, and after clicking on the predict button the result can be observed at the end of the page (figure 7.14).

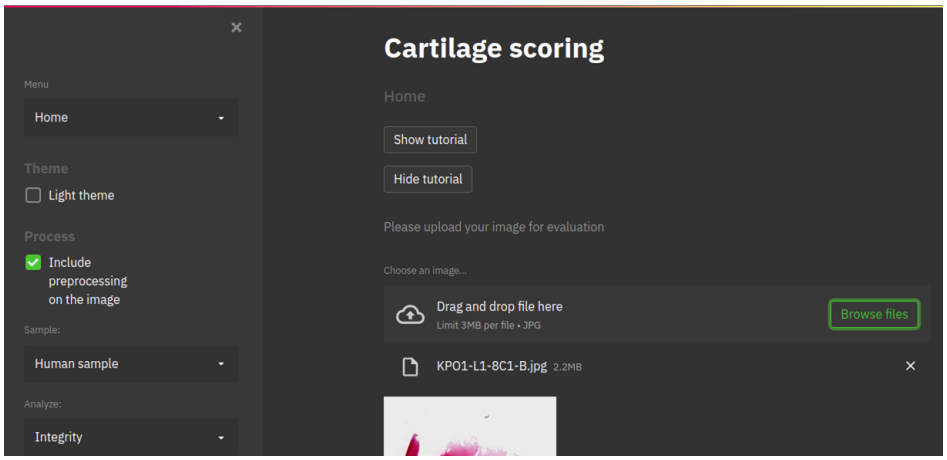


Figure 7.13: Application results.

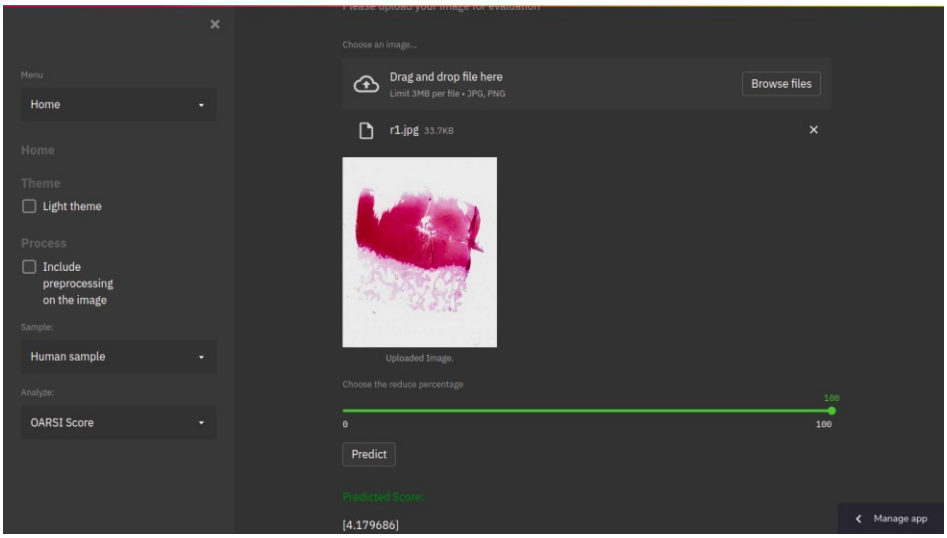


Figure 7.14: Application results.

# 8 Discussion

## 8.1 Learning process

The training data for this thesis work was not as large as those of popular, well-built, and pre-trained CNNs. In addition, some of the necessary filters, features, and parameters can be found in most of the early layers of these pre-trained models. Hence, it was justifiable to use these pre-trained, tested, and high performing neural networks to take advantage of the already learned parameters and many of the achieved weights and build new parameters upon the latter layers and fine-tune the weights using our own training set (target domain).

## 8.2 Model performance evaluation

The performance of the CNN models in different scenarios and classification/regression problems clearly indicates that the accuracy of the models in predicting the histopathological scores strongly depends on the preprocessing, dataset's coherency in image features, and the architecture of the classifying model. It is shown that using the same parameters and setup (in the camera) to result in consistency in the images, can lead to very accurate scoring. The 2+2 classification models resulted in better accuracy and precision than the 3-class classification model. Also, using the bone-deletion technique improved the error rates significantly. Furthermore, the MSEP errors of the models developed for predicting OARSI scores were better than the MSEP error of the models developed for predicting Mankin scores. Thus, it suggests that the OARSI scoring system is more amenable to automatic implementation using deep learning approaches.

Additionally, it is noteworthy that the bone-deletion process had almost similar impact on both regression models when it was not involved in the training (i.e., only used on the validation set). We speculate this is due to transfer learning and the use of pre-trained models. It is, however, essential, and much effective for the model performance that rotation and bone-deletion are used on the validation set. This makes sure that during the transfer learning and adjusting the weights, relevant regions and features in the target domain are affecting the adaptation.

## 8.3 Model comparison

In general, the AlexNet-based model shows much better results than other models for classification problems. This emphasizes the importance of the dropouts, perhaps reducing the effect of the white areas. It also implies that the depth and complexity of the architecture would not necessarily increase the accuracy. The better performance of the VGG19-based compared to the VGG16-based model could be traced to its architecture as it contains a higher number of convolution and ReLU layers. However, this is not the case for the ResNet models used in the classification problems and almost in all regression modelings. In the classification problem, ResNet18-based model showed better results than ResNet50-based model. This variation stems from the difference in the architecture of these two models. It seems that the additive feature of the ResNet can compensate for the number of convolution layers, increasing the convolution layer number would cause an over-fitting and higher error rate during the training as that was the case for ResNet50. However, in our case of regression problems, this trend was not observed for the ResNet model. This might be because of the increased number of possible outputs (before and without applying the softmax layer). In that case, we can see that ResNet50 had much better performance. This can suggest the importance of the initial layers in the network that cover the basic generic features.

## 8.4 Bone-deletion algorithm effectiveness

The bone deletion technique had an almost similar impact on the performance of the CNN models when applied only on the validation set, in comparison to when it was applied on both training and validation datasets. Good effectiveness for classification problems has been reported before when image enhancement is applied on both training and validation sets [94]. We expected a similar result utilizing this algorithm since the calcified parts were not affecting the scores and the network is to learn to ignore these pixels. Considering the MSEP values for the models that were not using this algorithm, this ignorance is assumed to be generally achieved by the pre-trained models. The explained performance of the bone-deletion algorithm might be caused by accidental deletion of cartilage parts (false detection of the cartilage parts with poor staining as calcified parts or as background) or leaving some of the calcified parts undeleted when it cannot be properly detected (figure 8.1).

Figure 8.1: Left image: the algorithm falsely deletes some parts of the cartilage that contains features important for prediction. Right image: the algorithm fails to detect calcified parts due to the strong staining and similarity to the cartilage.

## 8.5 Impact of windowing on model performance

Many studies have shown the advantage of the windowing technique for enlarging the training set, in addition to less frequent improvement of accuracy in the classification models. After the windowing is applied, an ensemble is used over all sections to find the class of the whole histopathological image [95 - 99]. This method can be useful in the case where all the slices (windows) contain a segmentation that would truly be classified in the same group as the original image. If the mentioned condition cannot be satisfied for any reason, the windowing procedure can lead to confusion and false classification. In the process that was followed in this work, after flipping and adding small degree rotations there were more than 31020 images ready for windowing; and when the windowing with n=4 is applied, the number of images increased to 124080. This means that it would not be easy to score all these sections manually, and since there is no program available to perform the scoring, each window would inherit the class or score from the original image. This can introduce erroneous data to the model (Figure 8.2).

Figure 8.2: Different windows derived from an image. These windows will not all be categorized with the same score as the original image, the section at the right is a healthier section than the one in the left window. Predicted OARSI scores for slices from left to right are: 2.15, 2.73, 3.12, and 1.69. The true OARSI score of the whole histological section is 2.87.

## 8.6 Including the abnormal sets

Knee joints from nine human cadavers were used for generating the dataset. The 8th and the 9th cadavers were imaged using different camera settings. When these two aberrant sets (the 8th and 9th sets) are not used, the accuracy of the best 2+2-classification

Figure 8.3: Sample images from the normal and abnormal groups. Right: Normal image, Left: Abnormal image.

method is more than 97%. Also, the OARSI regression model generated without these two sets has a mean squared error of 0.08. Since the staining color values are an important feature in this problem, it would not be wise to normalize (equalize) the distribution of color histograms since that might also change the true values of the feature areas containing important pixels. These variations in the image output are significant enough to not only affect the neural network but also to be used to identify the camera

model [100]. Figure 8.3 shows two images selected from the normal and abnormal groups to demonstrate this difference which could be hard to notice since the dominant color is red in both images.

To point out the effect of this difference, 50 random images were selected to compare the color value histograms from each group. In 100% of the samples from the abnormal groups, the peak channels of the green and blue were indistinguishable. In contrast, these channels were easily distinguishable for all samples from the normal groups. This means that the sensitivity in these two channels is different in the used cameras. Figure 8.4 shows sample histograms that clearly show the difference between the two groups.

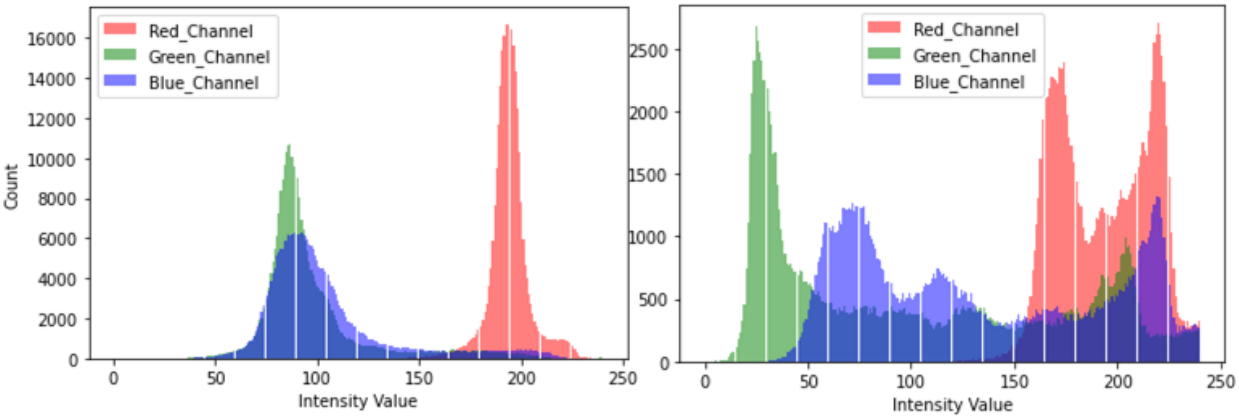


Figure 8.4: Sample images from the normal and abnormal groups. Histograms of color value distributions. Right: Normal image, Left: Abnormal image.

This would imply that using different camera settings (sensitivities) can lead to a less accurate prediction.

## 8.7 Model losses

Using the attention heatmap generated from the interpretation function of the fastai library we were able to locate which parts of the validation images are leading to a false classification of the image. When the bone deletion is not used, the calcified parts are the major cause of the error (Figure 8.5).

Figure 8.5 shows a heatmap generated for a model failure when the bone-deletion is applied. One element that affects the misclassification in many images is structural disarray. Another factor is the white areas fed into the neural network layers although they lack any feature or importance.

Many of the prediction failures are shown to be caused by the model searching through the white areas; although other works have used and shown the effectiveness of replacing pixels with white pixels [101], it is shown here that it can also have its downsides. Also, another misleading feature of this method is the calcified parts that are missed by the bone-deletion algorithm.

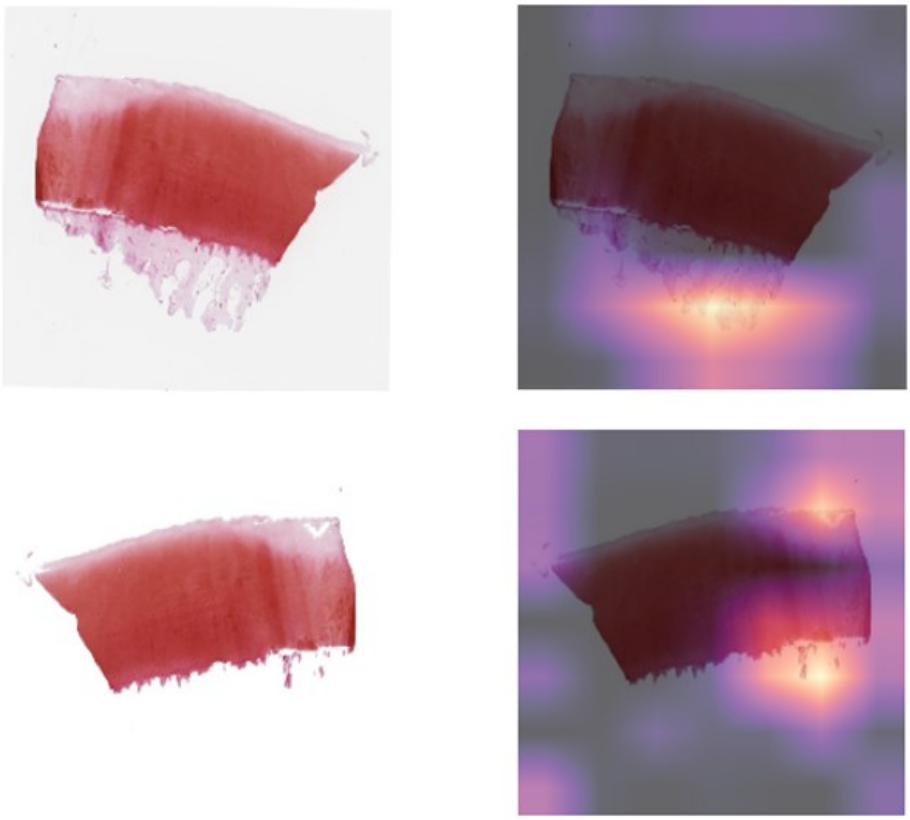

Figure 8.5: Model failure if bone-deletion algorithm is applied.
Top left: the original image. Top right: The heatmap showing which parts are considered for prediction. Bottom left: Bone-deletion algorithm applied on a flipped version of the original image. Bottom right: The heatmap showing which parts are considered for prediction.

## 8.8 Previous works

Power et al. also used deep learning method in 2021 for automating the grading of histological images of engineered cartilage [36]. The reported RMSE in their study is about 0.50 ± 0.05 in the best user reported case (grading in range 0 to 6 in modified Bern scoring system). For consistent comparison with our results, the estimated MSEP of their model would be about 11.7% ± 0.6%. The ResNet50 model in our study shows much

better performance when preprocessing is used. Similar to our models, this paper also shows a better distinction for advanced degeneration than mild ones.

Mousavi-Harami et al. developed a custom image analysis program for automatic objective scoring of cartilage degeneration [39]. We estimated that comparable MSEP for their model would be approximately 13%, hence, the models developed in this study for Mankin scoring system, show much better performance, provided that the preprocessing is used.

By comparing the results, it can be stated that automated prediction systems based on microscopic images can lead to more accurate (overall) evaluations than the MRI-based and radiographic-based models [35, 37, 38], provided that proper preprocessing is applied. Also, model-based transfer learning via deep learning and fine-tuning proved to be beneficial for such tasks and is suggested for similar histopathological evaluations. Standard deviation of difference between predicted and true values indicated that the proposed models in this thesis present superior reliability in comparison with human observers and previous models.

# 9 Conclusion and suggestions

The aim of the thesis was to address the limitations that exist in current scoring systems and methods caused by human errors and inefficacies. To this end, we followed three scoring objectives, one integrity assessment classification and two quantitative score regressions. We developed sixteen different models in total, that were based on pretrained CNN models (VGG19, ResNet18, ResNet50, AlexNet). We evaluated each model's performance in five different scenarios. For each scoring objective we chose the model showing best performance.

In the integrity assessment classification objective, the 2+2 class approach showed better results than the 3-class approach. The AlexNet-based model showed an accuracy of 88% and 94% for moderate/advanced and healthy/unhealthy classes, respectively. In the Mankin and OARSI score prediction regressors, ResNet50-based models presented lowest MSEPs (5.7% and 2.5% respectively).

A novel bone-deletion algorithm was developed, applied, and evaluated for each model in all objectives. In general, this preprocessing method resulted in better performance than other methods that were investigated in this thesis. Conversely, studying the effect of windowing revealed that it would diminish the performance of models. With proper preprocessing in use, results show that the developed models in this thesis have better performance than two of the similar works previously proposed for automating the scoring by Mousavi-Harami et al. [39] and Power et al. [36].

Inter-observer assessments of the received scores from the human scorers shows very poor consistency between them. Comparing the SD values of differences revealed that the AI-driven models in this thesis have much less variability and proved that well-trained models can be more reliable than human observers.

Considering model losses, we can first suggest the development of an improved bone-deletion algorithm. Second, a new model with a novel structure shall be trained to be able to ignore alpha channels (or white pixels) and run the convolution only through the meaningful pixels, instead of covering all the white pixels that are affecting the results. This is different than the concept of RCNN (Region-Based Convolutional Neural Network) that selects a certain shape as the region for further computations in the proceeding layers. Third, for similar histological evaluations, we can suggest a combined method of

transfer learning to be used, so that adversarial transfer learning and model-based transfer learning would be applied cooperatively. This might reduce the number of labeled data needed.

Overall, our results show that an accurate automated scoring system is achievable if the explained considerations are taken into account.

# References

1- Hunter, D.J. and Felson, D.T. (2006). Osteoarthritis. BMJ, 332(7542), pp. 639–642.

2- Sen, R. and Hurley, J. (2021). Osteoarthritis, StatPearls Publishing LLC. [online] Ncbi.nlm.nih.gov. Available at: https://www.ncbi.nlm.nih.gov/books/NBK482326/.

3- Mankin, H., Dorfman, H., Lippiello, L. and Zarins, A. (1971). Biochemical and Metabolic Abnormalities in Articular Cartilage from Osteo-Arthritic Human Hips. The Journal of Bone & Joint Surgery, 53(3), pp. 523-537.

4- Glasson, S., Chambers, M., Van Den Berg, W. and Little, C. (2010). The OARSI histopathology initiative – recommendations for histological assessments of osteoarthritis in the mouse. Osteoarthritis and Cartilage, 18, pp. 17-23.

5- Pauli, C., Whiteside, R., Heras, F.L., Nesic, D., Koziol, J., Grogan, S.P., Matyas, J., Pritzker, K.P.H., D'Lima, D.D. and Lotz, M.K. (2012). Comparison of cartilage histopathology assessment systems on human knee joints at all stages of osteoarthritis development. Osteoarthritis and Cartilage, 20(6), pp. 476–485.

6- Lakin, B., Snyder, B. and Grinstaff, M. (2017). Assessing Cartilage Biomechanical Properties: Techniques for Evaluating the Functional Performance of Cartilage in Health and Disease. Annual Review of Biomedical Engineering, 19(1), pp. 27-55.

7- Sophia Fox, A., Bedi, A. and Rodeo, S. (2009). The Basic Science of Articular Cartilage: Structure, Composition, and Function. Sports Health: A Multidisciplinary Approach, 1(6), pp. 461-468.

8- Mostakhdemin, M., Nand, A. and Ramezani, M. (2021). Articular and Artificial Cartilage, Characteristics, Properties and Testing Approaches—A Review. Polymers, 13(12), p. 2000.

9- Nebelung, S., Post, M., Knobe, M., Shah, D., Schleich, C., Hitpass, L., Kuhl, C., Thüring, J. and Truhn, D. (2019). Human articular cartilage mechanosensitivity is related to histological degeneration – a functional MRI study. Osteoarthritis and Cartilage, 27(11), pp. 1711-1720.

10- Mei, L., Shen, B., Ling, P., Liu, S., Xue, J., Liu, F., Shao, H., Chen, J., Ma, A. and Liu, X. (2017). Culture-expanded allogenic adipose tissue-derived stem cells attenuate cartilage degeneration in an experimental rat osteoarthritis model. PLoS One, 12(4), p.e0176107.

11- Roughley, P. and Mort, J. (2014). The role of aggrecan in normal and osteoarthritic cartilage. Journal of Experimental Orthopaedics, pp. 1-11.

12- Bank, R., Bayliss, M., Lafeber, F., Maroudas, A. and Tekoppele, J. (1998). Ageing and zonal variation in post-translational modification of collagen in normal human articular cartilage: The age-related increase in non-enzymatic glycation affects biomechanical properties of cartilage. Biochemical Journal, 330(1), pp. 345-351.

13- Mlynárik, V. and Tratting, S. (2000). Physicochemical Properties of Normal Articular Cartilage and Its MR Appearance. Investigative Radiology, 35(10), pp. 589-594.

14- Aigner, T. and Stöve, J. (2003). Collagens—major component of the physiological cartilage matrix, major target of cartilage degeneration, major tool in cartilage repair. Advanced Drug Delivery Reviews, 55(12), pp. 1569-1593.

15- Moskowitz R.W. (2009). The burden of osteoarthritis: clinical and quality-of-life issues. The American Journal of Managed Care, 15(8), pp. 223-229.

16- Roos, E. (2005). Joint injury causes knee osteoarthritis in young adults. Current Opinion in Rheumatology, 17(2), pp. 195-200.

17- Spector, T., Cicuttini, F., Baker, J., Loughlin, J. and Hart, D. (1996). Genetic influences on osteoarthritis in women: a twin study. BMJ, 312(7036), pp. 940-943.

18- Loeser, R., Goldring, S., Scanzello, C. and Goldring, M. (2012). Osteoarthritis: A disease of the joint as an organ. Arthritis & Rheumatism, 64(6), pp. 1697-1707.

19- Ostergaard, K., Petersen, J., Andersen, C., Bendtzen, K. and Salter, D. (1997). Histologic/histochemical grading system for osteoarthritic articular cartilage. Reproducibility and validity. Arthritis & Rheumatism, 40(10), pp. 1766-1771.

20- Ozkan, F.U., G., Turkmen, I., Yıldız, Y., Senol, S., Ozkan, K., Türkmensoy, F., Ramadan, S., Aktas, I. (2015). Intra-articular hyaluronate, tenoxicam and vitamin e in a rat model of osteoarthritis: Evaluation and comparison of chondroprotective efficacy. International journal of clinical and experimental medicine. 8, pp. 1018-1026.

21- Pritzker, K., Gay, S., Jimenez, S., Ostergaard, K., Pelletier, J., Revell, P., Salter, D. and van den Berg, W. (2006). Osteoarthritis cartilage histopathology: grading and staging. Osteoarthritis and Cartilage, 14(1), pp. 13-29.

22- McElligott, T. and Collins, D. (1960). Chondrocyte Function of Human Articular and Costal Cartilage Compared by Measuring the In Vitro Uptake of Labelled (35S) Sulphate. Annals of the Rheumatic Diseases, 19(1), pp. 31-41.

23- Ostergaard, K., Andersen, C., Petersen, J., Bendtzen, K. and Salter, D. (1999). Validity of histopathological grading of articular cartilage from osteoarthritic knee joints. Annals of the Rheumatic Diseases, 58(4), pp. 208-213.

24- Van der Sluijs, J.A., Geesink, R.G.T., van der Linden, A.J., Bulstra, S.K., Kuyer, R. and Drukker, J. (1992). The reliability of the mankin score for osteoarthritis. Journal of Orthopaedic Research, 10(1), pp. 58–61.

25- Moody, H., Heard, B., Frank, C., Shrive, N. and Oloyede, A. (2012). Investigating the potential value of individual parameters of histological grading systems in a sheep model of cartilage damage: the Modified Mankin method. Journal of Anatomy, 221(1), pp. 47-54.

26- Tiulpin, A. and Saarakkala, S. (2020). Automatic Grading of Individual Knee Osteoarthritis Features in Plain Radiographs Using Deep Convolutional Neural Networks. Diagnostics, 10(11), p.932.

27- Guan, B., Liu, F., Haj-Mirzaian, A., Demehri, S., Samsonov, A., Neogi, T., Guermazi, A. and Kijowski, R. (2020). Deep learning risk assessment models for predicting progression of radiographic medial joint space loss over a 48-month follow-up period. Osteoarthritis and Cartilage, 28(4), pp. 428–437.

28- Gan, H.S., Ramlee, M.H., Wahab, A.A., Lee, Y.S. and Shimizu, A. (2020). From classical to deep learning: review on cartilage and bone segmentation techniques in knee osteoarthritis research. Artificial Intelligence Review, 54(4), pp. 2445–2494.

29- Schwartz, A.J., Clarke, H.D., Spangehl, M.J., Bingham, J.S., Etzioni, D.A. and Neville, M.R. (2020). Can a Convolutional Neural Network Classify Knee Osteoarthritis on Plain Radiographs as Accurately as Fellowship-Trained Knee Arthroplasty Surgeons? The Journal of Arthroplasty, 35(9), pp. 2423–2428.

30- Chaudhari, A.S., Kogan, F., Pedoia, V., Majumdar, S., Gold, G.E. and Hargreaves, B.A. (2019). Rapid Knee MRI Acquisition and Analysis Techniques for Imaging Osteoarthritis. Journal of Magnetic Resonance Imaging, 52(5), pp. 1321–1339.

31- Yeoh, P.S.Q., Lai, K.W., Goh, S.L., Hasikin, K., Hum, Y.C., Tee, Y.K. and Dhanalakshmi, S. (2021). Emergence of Deep Learning in Knee Osteoarthritis Diagnosis. Computational Intelligence and Neuroscience, [online] 2021, p.e4931437. Available at: https://www.hindawi.com/journals/cin/2021/4931437.

32- Nasser, Y., Jennane, R., Chetouani, A., Lespessailles, E. and Hassouni, M.E. (2020). Discriminative Regularized Auto-Encoder for Early Detection of Knee OsteoArthritis: Data from the Osteoarthritis Initiative. IEEE Transactions on Medical Imaging, 39(9), pp. 2976–2984.

33- Ebrahimkhani, S., Jaward, M.H., Cicuttini, F.M., Dharmaratne, A., Wang, Y. and de Herrera, A.G.S. (2020). A review on segmentation of knee articular cartilage: from conventional methods towards deep learning. Artificial Intelligence in Medicine, 106, p. 101851.

34- Lim, J., Kim, J. and Cheon, S. (2019). A Deep Neural Network-Based Method for Early Detection of Osteoarthritis Using Statistical Data. International Journal of Environmental Research and Public Health, 16(7), p. 1281.

35- Rytky, S.J.O., Tiulpin, A., Frondelius, T., Finnilä, M.A.J., Karhula, S.S., Leino, J., Pritzker, K.P.H., Valkealahti, M., Lehenkari, P., Joukainen, A., Kröger, H., Nieminen, H.J. and Saarakkala, S. (2020). Automating three-dimensional osteoarthritis histopathological

grading of human osteochondral tissue using machine learning on contrast-enhanced micro-computed tomography. Osteoarthritis and Cartilage, 28(8), pp. 1133–1144.

36- Power, L., Acevedo, L., Yamashita, R., Rubin, D., Martin, I. and Barbero, A. (2021). Deep learning enables the automation of grading histological tissue engineered cartilage images for quality control standardization. Osteoarthritis and Cartilage, 29(3), pp. 433–443.

37- Tiulpin, A., Thevenot, J., Rahtu, E., Lehenkari, P. and Saarakkala, S. (2018). Automatic Knee Osteoarthritis Diagnosis from Plain Radiographs: A Deep Learning-Based Approach. Scientific Reports, 8(1), pp. 17-27.

38- Ashinsky, B.G., Coletta, C.E., Bouhrara, M., Lukas, V.A., Boyle, J.M., Reiter, D.A., Neu, C.P., Goldberg, I.G. and Spencer, R.G. (2015). Machine learning classification of OARSI-scored human articular cartilage using magnetic resonance imaging. Osteoarthritis and Cartilage, 23(10), pp. 1704–1712.

39- Moussavi-Harami, S.F., Pedersen, D.R., Martin, J.A., Hillis, S.L. and Brown, T.D. (2009). Automated Objective Scoring of Histologically Apparent Cartilage Degeneration Using a Custom Image Analysis Program. Journal of Orthopaedic Research, 27(4), pp. 522–528.

40- McCulloch, W.S. and Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. The Bulletin of Mathematical Biophysics, 5(4), pp. 115–133.

41- Rosenblatt, F. (1957). The perceptron, a perceiving and recognizing automaton. Buffalo, Ny Cornell Aeronautical Laboratory, Report 85-60-1.

42- www.javatpoint.com. Multi-layer Perceptron in TensorFlow - Javatpoint. [online] Available at: https://www.javatpoint.com/multi-layer-perceptron-in-tensorflow.

43- Koza, J.R., Bennett III, F.H., Andre, D. and Keane, M.A. (2000). Synthesis of topology and sizing of analog electrical circuits by means of genetic programming. Computer Methods in Applied Mechanics and Engineering, 186(2-4), pp. 459–482.

44- LeCun, Y., Bengio, Y. and Hinton, G. (2015). Deep Learning. Nature, 521(7553), pp. 436–444.

45- LeCun, Y., Boser, B., Denker, J.S., Henderson, D., Howard, R.E. and Hubbard, W.E. (1990). Proc. Advances in Neural Information Processing Systems, pp. 396-404.

46- Hashmi, M.F., Katiyar, S., Keskar, A.G., Bokde, N.D. and Geem, Z.W. (2020). Efficient Pneumonia Detection in Chest Xray Images Using Deep Transfer Learning. Diagnostics, 10(6), p. 417.

47- Alzubaidi, L., Zhang, J., Humaidi, A.J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaría, J., Fadhel, M.A., Al-Amidie, M. and Farhan, L. (2021). Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions. Journal of big Data, 8(1), pp. 1-74.

48- Gholamalinezhad, H. and Khosravi, H. (2020). Pooling methods in deep neural networks, a review. arXiv preprint arXiv:2009.07485.

49- Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. The journal of machine learning research, 15(1), pp. 1929-1958.

50- Krizhevsky, A., Sutskever, I. and Hinton, G.E. (2017). ImageNet classification with deep convolutional neural networks. Communications of the ACM, 60(6), pp. 84–90.

51- Shafiq, S. and Azim, T. (2021). Introspective analysis of convolutional neural networks for improving discrimination performance and feature visualisation. PeerJ Computer Science, 7, p.e497.

52- Ox.ac.uk. (2014). Visual Geometry Group - University of Oxford. [online] Available at: https://www.robots.ox.ac.uk/~vgg/research/very_deep/.

53- Setiawan, W. and Damayanti, F. (2020). Layers Modification of Convolutional Neural Network for Pneumonia Detection. Journal of Physics: Conference Series, 1477, pp. 52-55.

54- Zheng, Y., Yang, C. and Merkulov, A. (2018). Breast cancer screening using convolutional neural network and follow-up digital mammography. Computational Imaging III, Proc. SPIE 1066905.

55- He, K., Zhang, X., Ren, S. and Sun, J. (2016). Deep Residual Learning for Image Recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770–778.

56- Komura, D. and Ishikawa, S. (2018). Machine Learning Methods for Histopathological Image Analysis. Computational and Structural Biotechnology Journal, [online] 16, pp.34–42. Available at: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6158771.

57- Madabhushi, A. and Lee, G. (2016). Image analysis and machine learning in digital pathology: Challenges and opportunities. Medical Image Analysis, 33, pp.170–175.

58- Yang, Q., Zhang, Y., Dai, W. and Pan, S. (2020). Transfer learning.

59- Pan, S.J. and Yang, Q. (2010). A Survey on Transfer Learning. IEEE Transactions on Knowledge and Data Engineering, [online] 22(10), pp.1345–1359. Available at: http://home.cse.ust.hk/~qyang/Docs/2009/tkde_transfer_learning.pdf.

60- Pinto, G., Wang, Z., Roy, A., Hong, T. and Capozzoli, A. (2022). Transfer learning for smart buildings: A critical review of algorithms, applications, and future perspectives. Advances in Applied Energy, 5, p.100084.

61- Ahmed, A.M.A., Zhang, Y. and Eliassen, F. (2020). Generative Adversarial Networks and Transfer Learning for Non-Intrusive Load Monitoring in Smart Grids. 2020 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm), pp. 1-7.

62- Weiss, K., Khoshgoftaar, T.M. and Wang, D. (2016). A survey of transfer learning. Journal of Big Data, 3(9), pp. 1-40.

63- Li Fei-Fei, Fergus, R. and Perona, P. (2006). One-shot learning of object categories. IEEE Transactions on Pattern Analysis and Machine Intelligence, 28(4), pp. 594–611.

64- Tommasi, T., Orabona, F. and Caputo, B. (2010). Safety in numbers: Learning categories from few examples with multi model knowledge transfer. 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 3081-3088.

65- Jie, L., Tommasi, T. and Caputo, B. (2011). Multiclass transfer learning from unconstrained priors. 2011 International Conference on Computer Vision, pp. 1863–1870.

66- Lawrence, N.D. and Platt, J.C. (2004). Learning to learn with the informative vector machine. Twenty-first international conference on Machine learning - ICML ’04, p. 65.

67- Evgeniou, T. and Pontil, M. (2004). Regularized multi--task learning. Proceedings of the 2004 ACM SIGKDD international conference on Knowledge discovery and data mining - KDD ’04, pp. 109-117.

68- Biengo, Y. (2012). Deep learning of representations for unsupervised and transfer learning. Proceedings of ICML Workshop on Unsupervised and Transfer Learning, 27, pp. 17-37.

69- Yosinski, J., Clune, J., Bengio, Y. and Lipson, H. (2014). How transferable are features in deep neural networks? In Proceedings of the 27th International Conference on Neural Information Processing Systems, 2 (NIPS'14). MIT Press, Cambridge, pp. 3320–3328.

70- Zhu, X. (2005), Semi-Supervised Learning Literature Survey, Tech. Report, Computer Sciences TR 1530, University of Wisconsin-Madison.

71- Bengio, Y., Courville, A. and Vincent, P. (2013). Representation Learning: A Review and New Perspectives. IEEE Transactions on Pattern Analysis and Machine Intelligence, 35(8), pp. 1798–1828.

72- Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. and Bengio, Y. (2014). Generative adversarial nets. In Proceedings of the 27th International Conference on Neural Information Processing Systems, 2 (NIPS'14). MIT Press, Cambridge, pp. 2672–2680.

73- Wang, L., Jiao, Y., Qiao, Y., Zeng, N. and Yu, R., 2020. A novel approach combined transfer learning and deep learning to predict TMB from histology image. Pattern Recognition Letters, 135, pp. 244-248.

74- Vesal, S., Ravikumar, N., Davari, A., Ellmann, S. and Maier, A. (2018) June. Classification of breast cancer histology images using transfer learning. In International conference image analysis and recognition. Springer, Cham, pp. 812-819.

75- Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J. and Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 2818-2826.

76- Kassani, S.H., Kassani, P.H., Wesolowski, M.J., Schneider, K.A. and Deters, R. (2019) October. Breast cancer diagnosis with transfer learning and global pooling. In 2019 International Conference on Information and Communication Technology Convergence (ICTC) IEEE, pp. 519-524.

77- Ohata, E.F., Chagas, J.V.S.D., Bezerra, G.M., Hassan, M.M. and de Albuquerque, V.H.C. (2021). A novel transfer learning approach for the classification of histological images of colorectal cancer. The Journal of Supercomputing, 77(9), pp. 9494-9519.

78- Cui, A., Li, H., Wang, D., Zhong, J., Chen, Y. and Lu, H. (2020). Global, regional prevalence, incidence and risk factors of knee osteoarthritis in population-based studies. EClinicalMedicine, 29-30, p. 100587.

79- Singhal, O., Kaur, V., Singhal, M., Machave, Y., Gupta, A. and Kalhan, S. (2012). Arthroscopic synovial biopsy in definitive diagnosis of joint diseases: An evaluation of efficacy and precision. International Journal of Applied and Basic Medical Research, 2(2), p.102.

80- Prakash, M., Joukainen, A., Torniainen, J., Honkanen, M.K.M., Rieppo, L., Afara, I.O., Kröger, H., Töyräs, J. and Sarin, J.K. (2019). Near-infrared spectroscopy enables quantitative evaluation of human cartilage biomechanical properties during arthroscopy. Osteoarthritis and Cartilage, 27(8), pp. 1235–1243.

81- Király, K., Lapveteläinen, T., Arokoski, J., Törrönen, K., Módis, L., Kiviranta, I. and Helminen, H.J. (1996). Application of selected cationic dyes for the semiquantitative estimation of glycosaminoglycans in histological sections of articular cartilage by microspectrophotometry. The Histochemical Journal, 28(8), pp. 577–590.

82- Bradski, G. (2000). The OpenCV Library. Dr. Dobb's Journal of Software Tools.

83- Harris, C.R. et al., (2020). Array programming with NumPy. Nature, 585, pp. 357–362.

84- Clark, A. (2015). Pillow (PIL Fork) Documentation, readthedocs. Available at: https://buildmedia.readthedocs.org/media/pdf/pillow/latest/pillow.pdf.

85- Paszke, A. et al. (2019). PyTorch: An Imperative Style, High-Performance Deep Learning Library. In Advances in Neural Information Processing Systems 32. Curran Associates, Inc., pp. 8024–8035.

86- fastai1.fast.ai | fastai. [online] Available at: https://fastai1.fast.ai/.

87- Shorten, C. and Khoshgoftaar, T.M. (2019). A survey on Image Data Augmentation for Deep Learning. Journal of Big Data, 6(60), pp. 1-48.

88- Zhou, B., Khosla, A., Lapedriza, A., Oliva, A. and Torralba, A. (2015). Object Detectors Emerge in Deep Scene CNNs. arXiv:1412.6856 [cs]. [online] Available at: https://arxiv.org/abs/1412.6856.

89- Lin, M., Chen, Q. and Yan, S. (2013). Network In Network. [online] arXiv.org. Available at: https://arxiv.org/abs/1312.4400.

90- Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V. and Rabinovich, A. (2015). Going Deeper with Convolutions. Cv-foundation.org, [online] pp.1–9. Available at: https://www.cv-foundation.org/openaccess/content_cvpr_2015/html/Szegedy_Going_Deeper_With_2015_CVPR_paper.html.

91- Zhou, B., Khosla, A., Lapedriza, A., Oliva, A. and Torralba, A. (2015). Learning Deep Features for Discriminative Localization. [online] arXiv.org. Available at: https://arxiv.org/abs/1512.04150.

92- Wang, L. and Chen, P. (2019). Neurons Activation Visualization and Information Theoretic Analysis. arXiv:1905.08618 [cs]. [online] Available at: https://arxiv.org/abs/1905.08618.

93- Streamlit.io. (2021). Streamlit • The fastest way to build and share data apps. [online] Available at: <https://streamlit.io/>.

94- Ostergaard, K. and Salter, D.M. (1998). Immunohistochemistry in the Study of Normal and Osteoarthritic Articular Cartilage. Progress in Histochemistry and Cytochemistry, 33(2), pp. 93-168.

95- Lu, X., You, Z., Sun, M., Wu, J. and Zhang, Z. (2021). Breast cancer mitotic cell detection using cascade convolutional neural network with U-Net. Mathematical Biosciences and Engineering, 18(1), pp. 673–695.

96- Hägele, M., Seegerer, P., Lapuschkin, S., Bockmayr, M., Samek, W., Klauschen, F., Müller, K.-R. and Binder, A. (2020). Resolving challenges in deep learning-based analyses of histopathological images using explanation methods. Scientific Reports, 10(1), pp. 1-12.

97- Alqudah, A. and Alqudah, A.M. (2021). Sliding window based deep ensemble system for breast cancer classification. Journal of Medical Engineering & Technology, 45(4), pp. 313–323.

98- Iizuka, O., Kanavati, F., Kato, K., Rambeau, M., Arihiro, K. and Tsuneki, M. (2020). Deep Learning Models for Histopathological Classification of Gastric and Colonic Epithelial Tumours. Scientific Reports, 10(1), p. 1504.

99- Haryanto, T., Suhartanto, H., Arymurthy, A.M. and Kusmardi, K. (2021). Conditional sliding windows: An approach for handling data limitation in colorectal histopathology image classification. Informatics in Medicine Unlocked, 23, p. 100565.

100- Bondi, L., Güera, D., Baroffio, L., Bestagini, P., Delp, E. and Tubaro, S. (2017). A Preliminary Study on Convolutional Neural Networks for Camera Model Identification. Electronic Imaging, 2017(7), pp. 67–76.

101- Boulze, H., Korosov, A. and Brajard, J. (2020). Classification of Sea Ice Types in Sentinel-1 SAR Data Using Convolutional Neural Networks. Remote Sensing, 12(13), p. 2165.

# Appendix I

## A.I.1 AlexNet-based model analysis in 2+2 classification

### A.I.1.1 The effect of batch size

Different batch sizes resulted in different accuracies showing a nonlinear behavior (Figure A.I.1). The best accuracy was achieved by the batch size of 32.

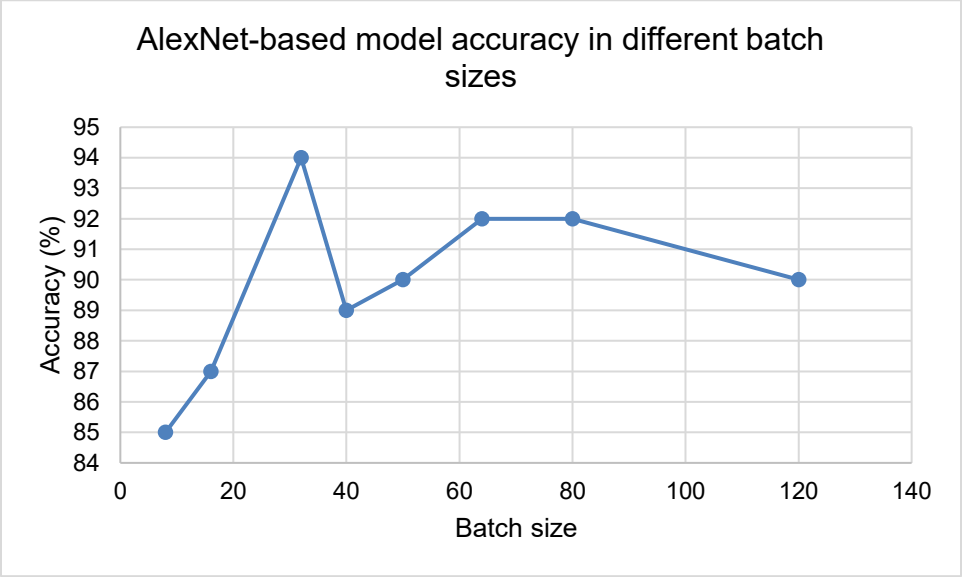


Figure A.I.1: Effect of batch size in accuracy (AlexNet).

### A.I.1.2 The effect of epochs

Overfitting was observed for epochs greater than 2 (Figure A.I.2).

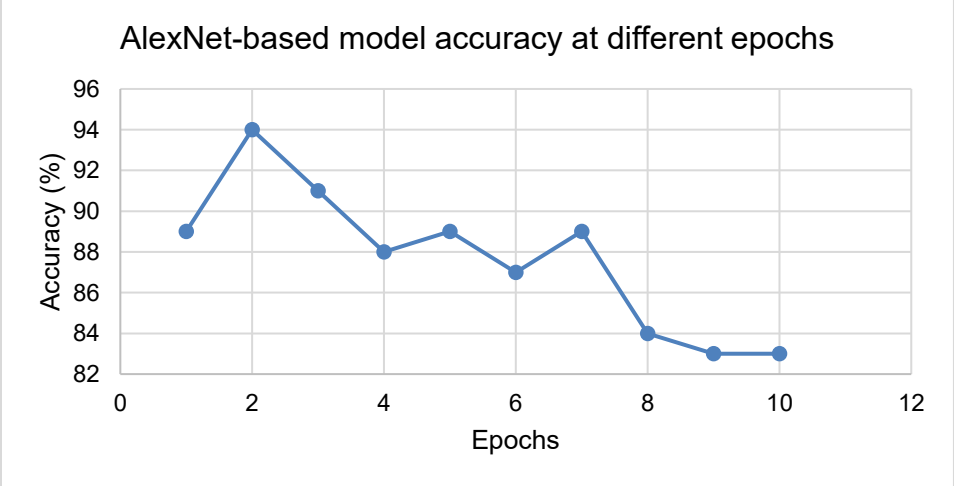


Figure A.I.2: Effect of epochs on accuracy (AlexNet).

### A.I.1.3The effect of learning rate

The two classifications of healthy-unhealthy and moderate-advanced have the following learning rate graphs.

Healthy-unhealthy:

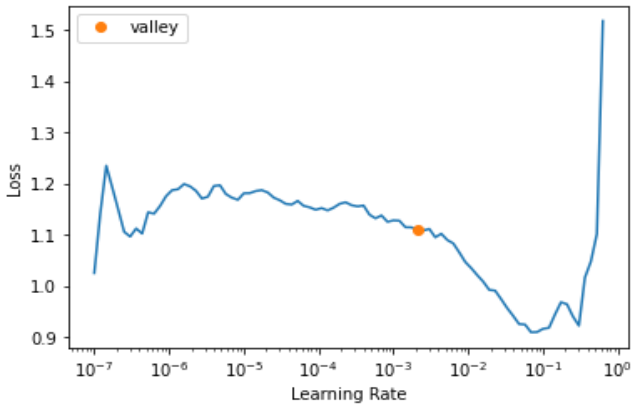


Figure A.I.3: Effect of learning rate on model's loss.

Moderate-advanced:

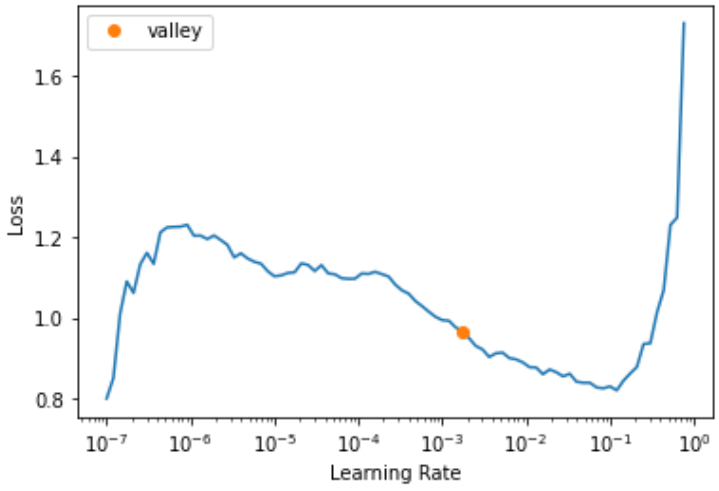


Figure A.I.4: Effect of learning rate on model's loss.

## A.I.2 ResNet50-based model analysis in regression (Mankin)

### A.I.2.1 The effect of batch size

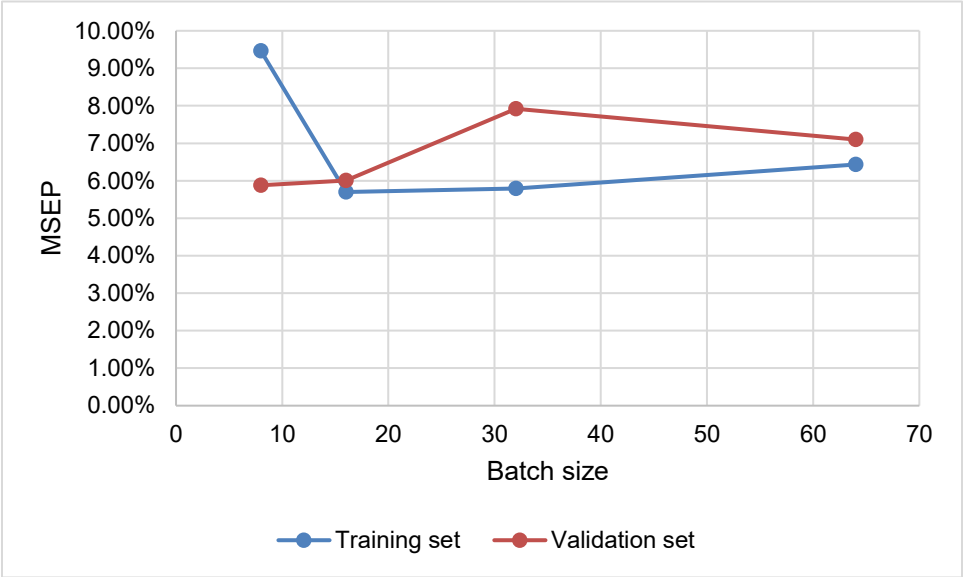


Figure A.I.5: Effect of batch sizes (8, 16, 32, 64) on model performance.

### A.I.2.2 The effect of epochs

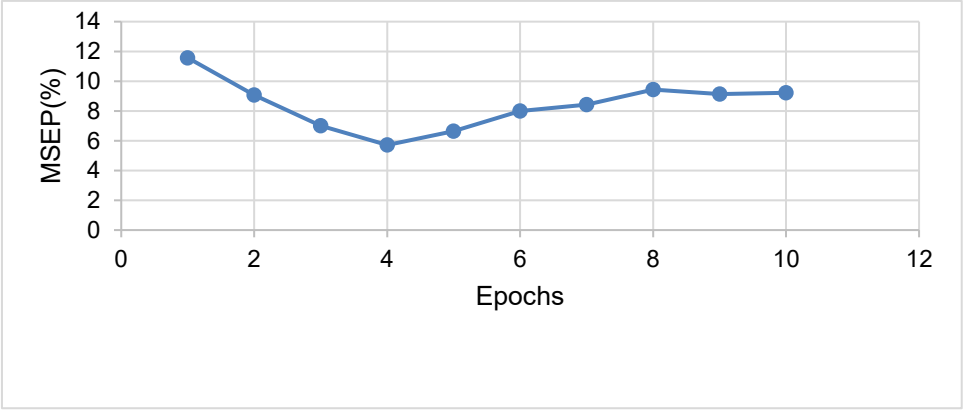


Figure A.I.6: The effect of epochs on model performance.

### A.I.2.3The effect of learning rate

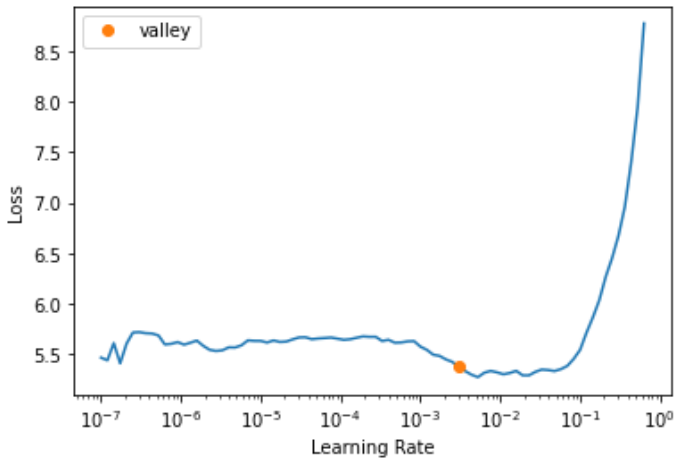


Figure A.I.7: The effect of learning rate on model performance.

## A.I.3 ResNet50-based model analysis in regression (OARSI)

### A.I.3.1 The effect of batch size

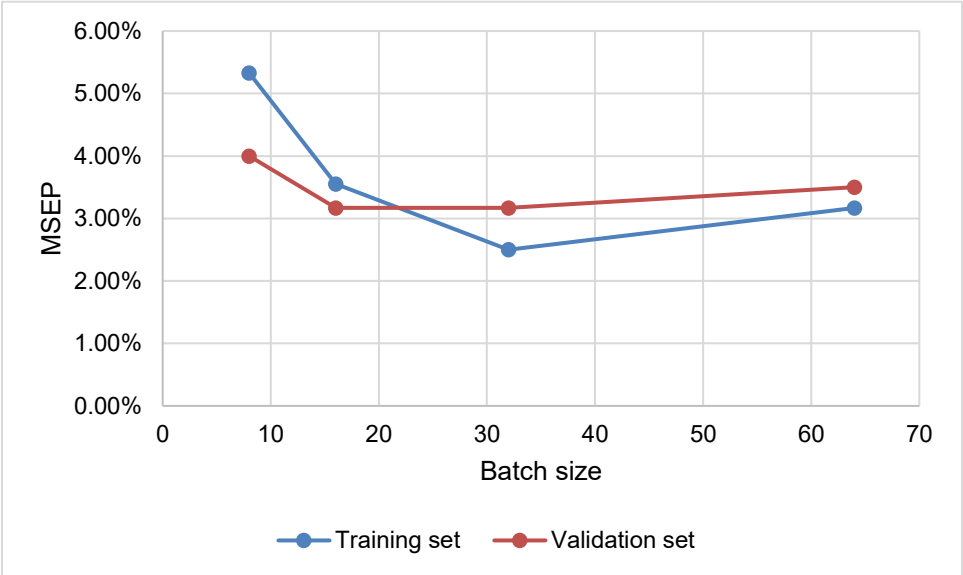


Figure A.I.8: Effect of batch sizes on model performance.

### A.I.3.2The effect of epochs

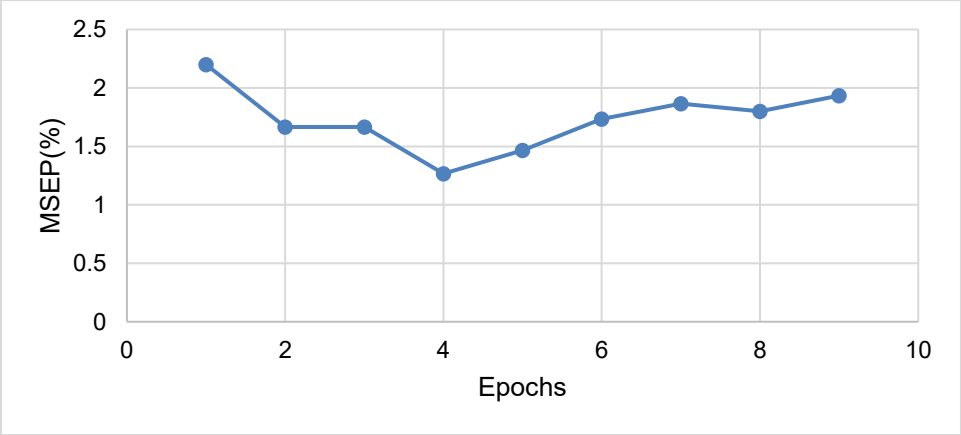


Figure A.I.9: Effect of epochs on model performance.

### A.I.3.3The effect of learning rate

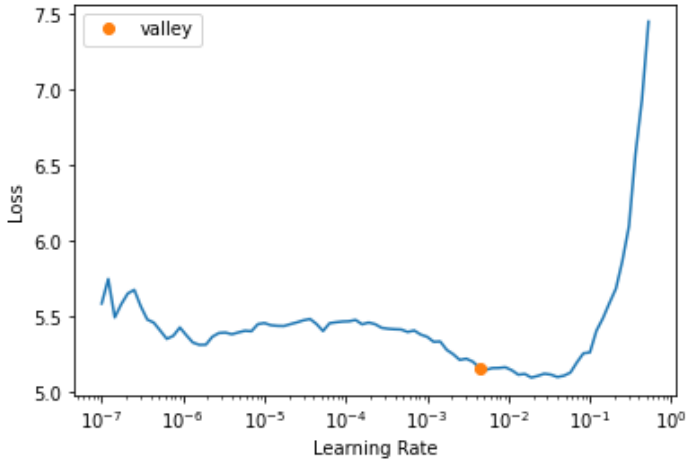


Figure A.I.10: Effect of learning rate on model performance

# Appendix II

Table A.II.1: Result (validation sets) comparison of the predictive classifier models based on VGG19, AlexNet, and ResNet using windowed images (bone-deletion and rotation were applied).

| Pretrained model | Accuracy (3-class) | Precision (3-class) | Accuracy (2+2 class) | Precision (2+2 class) |
|---|---|---|---|---|
| **VGG19** | 0.67 | Advanced: 0.87<br>Moderate: 0.49<br>Mild: 0.79 | Moderate/Advanced: 0.80<br>Healthy/Unhealthy: 0.84 | Advanced: 0.71<br>Moderate: 0.64<br>Mild: 0.97 |
| **ResNet18** | 0.68 | Advanced: 0.82<br>Moderate: 0.45<br>Mild: 0.75 | Moderate/Advanced: 0.78<br>Healthy/Unhealthy: 0.85 | Advanced: 0.71<br>Moderate: 0.60<br>Mild: 0.83 |
| **ResNet50** | 0.65 | Advanced: 0.80<br>Moderate: 0.42<br>Mild: 0.73 | Moderate/Advanced: 0.79<br>Healthy/Unhealthy: 0.84 | Advanced: 0.71<br>Moderate: 0.62<br>Mild: 0.92 |
| **AlexNet** | 0.69 | Advanced: 0.89<br>Moderate: 0.5<br>Mild: 0.81 | Moderate/Advanced: 0.80<br>Healthy/Unhealthy: 0.91 | Advanced: 0.81<br>Moderate: 0.65<br>Mild: 0.92 |

Table A.II.2: Result (validation sets) comparison of the predictive classifier models based on VGG19, AlexNet, and ResNet using non-windowed images (bone-deletion and rotation were applied).

| Pretrained model | Accuracy (3-class) | Precision (3-class) | Accuracy (2+2 class) | Precision (2+2 class) |
|---|---|---|---|---|
| **VGG19** | 0.69 | Advanced: 0.88<br>Moderate: 0.54<br>Mild: 0.82 | Moderate/Advanced: 0.87<br>Healthy/Unhealthy: 0.88 | Advanced: 0.76<br>Moderate: 0.76<br>Mild: 0.92 |
| **ResNet18** | 0.74 | Advanced: 0.84<br>Moderate: 0.49<br>Mild: 0.77 | Moderate/Advanced: 0.87<br>Healthy/Unhealthy: 0.88 | Advanced: 0.80<br>Moderate: 0.73<br>Mild: 0.88 |
| **ResNet50** | 0.72 | Advanced: 0.82<br>Moderate: 0.54<br>Mild: 0.76 | Moderate/Advanced: 0.86<br>Healthy/Unhealthy: 0.85 | Advanced: 0.71<br>Moderate: 0.63<br>Mild: 0.83 |
| **AlexNet** | 0.78 | Advanced: 0.91<br>Moderate: 0.62<br>Mild: 0.83 | Moderate/Advanced: 0.88<br>Healthy/Unhealthy: 0.94 | Advanced: 0.78<br>Moderate: 0.75<br>Mild: 0.88 |

Table A.II.3: Result (validation sets) comparison of the predictive classifier models based on VGG19, AlexNet, and ResNet using windowed images (bone-deletion and rotation were **not** applied).

| Pretrained model | Accuracy (3-class) | Precision (3-class) | Accuracy (2+2 class) | Precision (2+2 class) |
|---|---|---|---|---|
| **VGG19** | 0.58 | Advanced: 0.69<br>Moderate: 0.48<br>Mild: 0.54 | Moderate/Advanced: 0.62<br>Healthy/Unhealthy: 0.64 | Advanced: 0.62<br>Moderate: 0.55<br>Mild: 0.72 |
| **ResNet18** | 0.59 | Advanced: 0.70<br>Moderate: 0.48<br>Mild: 0.53 | Moderate/Advanced: 0.63<br>Healthy/Unhealthy: 0.66 | Advanced: 0.66<br>Moderate: 0.50<br>Mild: 0.75 |
| **ResNet50** | 0.61 | Advanced: 0.70<br>Moderate: 0.50<br>Mild: 0.58 | Moderate/Advanced: 0.68<br>Healthy/Unhealthy: 0.70 | Advanced: 0.69<br>Moderate: 0.63<br>Mild: 0.77 |
| **AlexNet** | 0.63 | Advanced: 0.73<br>Moderate: 0.52<br>Mild: 0.56 | Moderate/Advanced: 0.66<br>Healthy/Unhealthy: 0.73 | Advanced: 0.72<br>Moderate: 0.59<br>Mild: 0.78 |

Table A.II.4: Result (validation sets) comparison of the predictive classifier models based on VGG19, AlexNet, and ResNet using non-windowed images (bone-deletion and rotation were **not** applied).

| Pretrained model | Accuracy (3-class) | Precision (3-class) | Accuracy (2+2 class) | Precision (2+2 class) |
|---|---|---|---|---|
| **VGG19** | 0.62 | Advanced: 0.75<br>Moderate: 0.50<br>Mild: 0.57 | Moderate/Advanced: 0.68<br>Healthy/Unhealthy: 0.71 | Advanced: 0.75<br>Moderate: 0.54<br>Mild: 0.78 |
| **ResNet18** | 0.66 | Advanced: 0.79<br>Moderate: 0.52<br>Mild: 0.58 | Moderate/Advanced: 0.72<br>Healthy/Unhealthy: 0.79 | Advanced: 0.72<br>Moderate: 0.58<br>Mild: 0.78 |
| **ResNet50** | 0.63 | Advanced: 0.74<br>Moderate: 0.50<br>Mild: 0.60 | Moderate/Advanced: 0.71<br>Healthy/Unhealthy: 0.73 | Advanced: 0.69<br>Moderate: 0.52<br>Mild: 0.77 |
| **AlexNet** | 0.65 | Advanced: 0.77<br>Moderate: 0.57<br>Mild: 0.63 | Moderate/Advanced: 0.71<br>Healthy/Unhealthy: 0.75 | Advanced: 0.69<br>Moderate: 0.53<br>Mild: 0.80 |

Table A.II.5: Result comparison of the predictive regression models of Mankin scoring based on VGG19, AlexNet, and ResNet (bone-deletion and rotation were applied).

| Pretrained Model | MSEP (training set) windowed | MSEP (validation set) windowed | MSEP (training set) non-windowed | MSEP (validation set) non-windowed |
|---|---|---|---|---|
| **VGG19** | 5.8% | 8.6% | 3.4% | 6.6% |
| **ResNet18** | 6.0% | 11.6% | 3.1% | 6.5% |
| **ResNet50** | 5.1% | 11.7% | 3.0% | 5.7% |
| **AlexNet** | 8.0% | 11.8% | 4.5% | 6.8% |

Table A.II.6: Result comparison of the predictive regression models of Mankin scoring based on VGG19, AlexNet, and ResNet (bone-deletion and rotation were **not** applied).

| Pretrained Model | MSEP (training set) windowed | MSEP (validation set) windowed | MSEP (training set) non-windowed | MSEP (validation set) non-windowed |
|---|---|---|---|---|
| **VGG19** | 10.5% | 17.6% | 6.7% | 13.0% |
| **ResNet18** | 11.1% | 18.0% | 6.8% | 13.0% |
| **ResNet50** | 10.2% | 17.3% | 6.3% | 12.7% |
| **AlexNet** | 11.1% | 18.6% | 8.1% | 13.8% |

Table A.II.7: Result comparison of the predictive regression models of OARSI scoring based on VGG19, AlexNet, and ResNet (bone-deletion and rotation were applied).

| Pretrained Model | MSEP (training set) windowed | MSEP (validation set) windowed | MSEP (training set) non-windowed | MSEP (validation set) non-windowed |
|---|---|---|---|---|
| **VGG19** | 4.0% | 7.6% | 1.8% | 3.0% |
| **ResNet18** | 1.6% | 7.6% | 1.8% | 2.8% |
| **ResNet50** | 3.3% | 8.0% | 1.5% | 2.5% |
| **AlexNet** | 3.1% | 6.8% | 3.3% | 3.0% |

Table A.II.8: Result comparison of the predictive regression models of OARSI scoring based on VGG19, AlexNet, and ResNet (bone-deletion and rotation were **not** applied).

| Pretrained Model | MSEP (training set) windowed | MSEP (validation set) windowed | MSEP (training set) non-windowed | MSEP (validation set) non-windowed |
|---|---|---|---|---|
| **VGG19** | 4.2% | 9.1% | 2.8% | 4.0% |
| **ResNet18** | 4.5% | 9.3% | 2.9% | 4.2% |
| **ResNet50** | 4.4% | 8.9% | 2.6% | 4.1% |
| **AlexNet** | 4.3% | 8.9% | 3.7% | 3.6% |