
Title

Written by
Group 17gr7402

1. Semester

School of Medicine and Health

Biomedical Engineering and Informatics

Fredrik Bajers Vej 7

9220 Aalborg

Title:

title title

Theme:

Biomedical Signals and Information

Project Period:

Fall semester 2017

Project Group:

17gr7402

Abstract

Participants:

Birgithe Kleemann Rasmussen

Ignas Kupcikevičius

Linette Helena Poulsen

Mads Kristensen



Supervisor(s)

Shellie Boudreau

Lasse Riis Østergaard

Page Numbers:

Appendix:

Date of Completion: 20/12/2017

The content of this report is freely available, but publication may only be pursued due to agreement with the author.

Preface

Bla bæa

Contents

Chapter 1 Introduction	1
Chapter 2 Background	3
2.1 Anatomy of the Knee	3
2.2 Pain	5
2.3 Patellofemoral pain syndrome	6
2.4 Identify and interpret pain	7
2.4.1 Identify pain	7
2.4.2 Pain interpretation	7
2.5 Knee regions	9
2.6 Machine learning	10
2.6.1 Deep Learning	10
2.6.2 Back-propagation	11
2.6.3 Regularization	15
2.6.4 Core Neural Network layers	16
Chapter 3 Methodology	18
3.1 Data description	18
3.1.1 Software application: Navigate Pain	19
3.1.2 Data representations	20
3.2 Pre-analysis	21
3.2.1 Classification of data	21
3.2.2 Threshold selection	22
3.2.3 Simple regression models	24
3.3 Pre-processing	26
3.3.1 Morphology-representation	27
3.3.2 Regions-representation	28
3.3.3 Combined-representation	28
3.4 Neural network implementation and models	29
3.4.1 Software and hardware	29
3.4.2 Data handling and design choices of the models	30
3.4.3 Data handling in python	31
3.5 Optimization	32
3.5.1 Training and validation	33
3.5.2 Testing	34
Chapter 4 Description of models	35
4.1 Data handling and design choices of the models	35
4.1.1 Data handling in python	35
4.1.2 Optimization techniques	35

4.1.3	Training of the networks	37
4.2	Vector image model	37
4.3	Raw binary representation model	37
4.4	Combined representation model	39
Chapter 5	Results	40
Bibliography		41
Appendix A	Knee injury and Osteoarthritis Outcome Score (KOOS) [?]	45
Appendix B	Threshold analysis	49

Chapter 1

Introduction

Patellofemoral pain syndrome (PFPS) is a painful musculoskeletal condition that is presented as pain behind or around the patella [1, 2]. PFPS affects 6-7 % of adolescents, of whom two thirds are highly physically active [3]. Additionally the prevalence is more than twice as high for females than males [3, 4]. PFPS may be present over a longer period of time where a high number of individuals experience a recurrent or chronic pain [5]. Chronic pain may be maintained by the phenomenon central sensitization, which may result in increased areas of pain over longer periods of time. Furthermore, PFPS may lead to osteoarthritis [4, 6].

Patellofemoral pain (PFP) is often described as diffuse knee pain, that can be hard for individuals to explain and localize [5]. Despite the fact that individuals feel pain in the knee, there is no structural changes in the knee such as significant chondral damage or increased Q-angle. Because of this there is no definitive clinical test to diagnose PFPS and it is thereby often diagnosed based on exclusion criterias [4] to which PFPS is also described as an orthopaedic enigma, and is one of the most challenging pathologies to manage [7]. To assist diagnosis of PFPS, pain maps may be used as a helpful tool for the individuals to communicate their pain by drawing pain areas on a body outline [8].

A study by Boudreau et al. indicates, through the use of pain maps, that it is possible to find a correlation between the size of the pain and the pain duration as well as intensity for individuals with PFP longer than five years.[9] However, it is unknown whether the morphology and locations of the pain have an influence on the pain duration and intensity. It is assumed that relation between pain maps and pain duration or intensity is nonlinear, because the perceived PFP is subjective and is considered as multidimensional [?]. To investigate the nonlinear relation, deep learning is used, which is a method that has not been found used in this context before.

The goals of this project is to explore how accurate a deep learning model can classify pain maps according to pain duration and intensity by using a limited dataset. The pain maps are encoded into multiple data representations to investigate whether morphology and location have an influence on pain duration or intensity. Because of the imbalance in prevalence between females and males, the gender is included as a feature in the deep learning model.

The aim of this study is to explore classification performance of a deep learning model, using PFP pain maps and gender as input to classify either pain duration or intensity.

It is hypothesized that a deep learning model that uses pain maps and gender as input parameter has a higher performance when classifying according to pain duration than pain intensity.

The secondary aim is to investigate if multiple pain map representations, which reflect the

morphology and location of the pain, affect the deep learning model classification performance.

It is hypothesized that different data representations of pain maps, reflecting morphology and location of pain, affect the performance accuracy of a deep learning model when classifying according to pain duration or intensity.

Chapter 2

Background

This chapter presents the background knowledge that optimizes the understanding of essential topics in this project, such as patellofemoral pain and deep learning. Regarding patellofemoral pain it is relevant to get knowledge about the anatomy of the knee as well as pain and pain measurements if a deeper understanding of the syndrome is considered necessary. Furthermore, the chapter is essential for getting a basic understanding of some properties and optimizers in the neural network models used in this project.

2.1 Anatomy of the Knee

The knee is the largest synovial joint in the body and consists of a hinge and a gliding joint. The hinge joint is placed between the lateral and medial femoral condyles and the lateral and medial tibial condyles. Between the patella and femur is the gliding joint formed. The structure of the knee is illustrated in figure 2.1.[10]

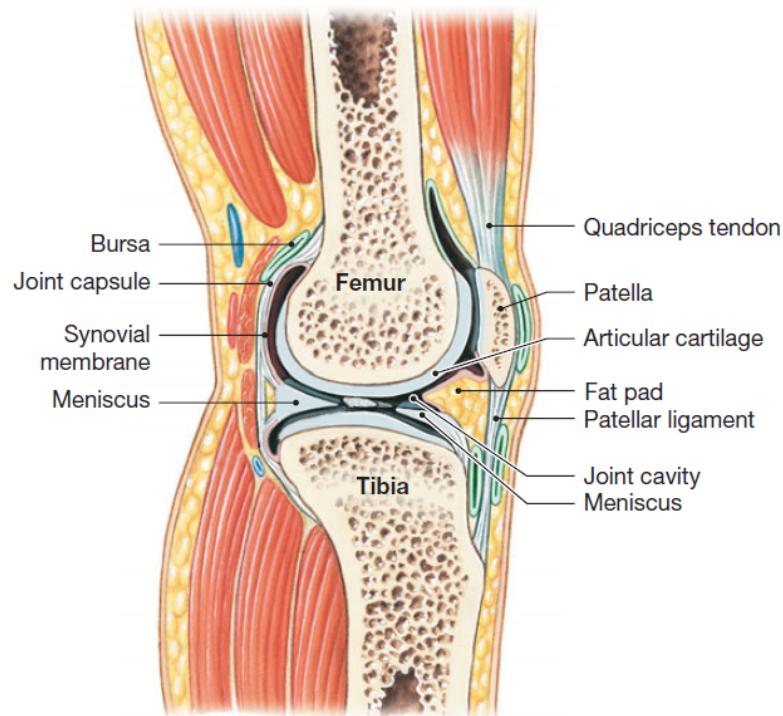


Figure 2.1: The anatomy of the knee. Edited [10].

It is shown in figure 2.1 that the patella is a sesamoid bone. At birth the patella consists of cartilaginous and ossifies when the child's extremities gets stronger, which typically proceeds

between age two or three and the beginning of puberty.

The patella is surrounded by the tendon of the quadriceps femoris. Quadriceps femoris is the muscles which controls the extending of the knee. The quadriceps tendon is combined to the surface anterior and superior of patella. Tibia is combined to the anterior and inferior surface of the patella by the patellar ligament. The bones, tibia and femur, are covered by articular cartilage with the purpose of protecting the bones from friction. The articular cartilage on the two bones are separated from one another by synovial membranes that contains synovial fluid, that further reduce the friction. The primary functions of the synovial fluid is to lubricate, distribution of nutrient and absorption of shock.[10]

The fat pads and menisci are placed between the articular cartilages. The fat pads' function is to protect the cartilage and fill out space as result of the joint cavity changes. The menisci stabilize the knee and acts like pads, that conform shape when femur moves. In addition to fat pads and menisci, the bursa acts as friction minimization between patella and tissues.[10] There are three separate articulations in the knee joint. The first is between the patella and the patellar surface of the femur and the rest are between the femoral and tibial condyles. Additionally, the knee consist of seven major ligaments that stabilize the knee joint, which are shown in figure 2.2.[10]

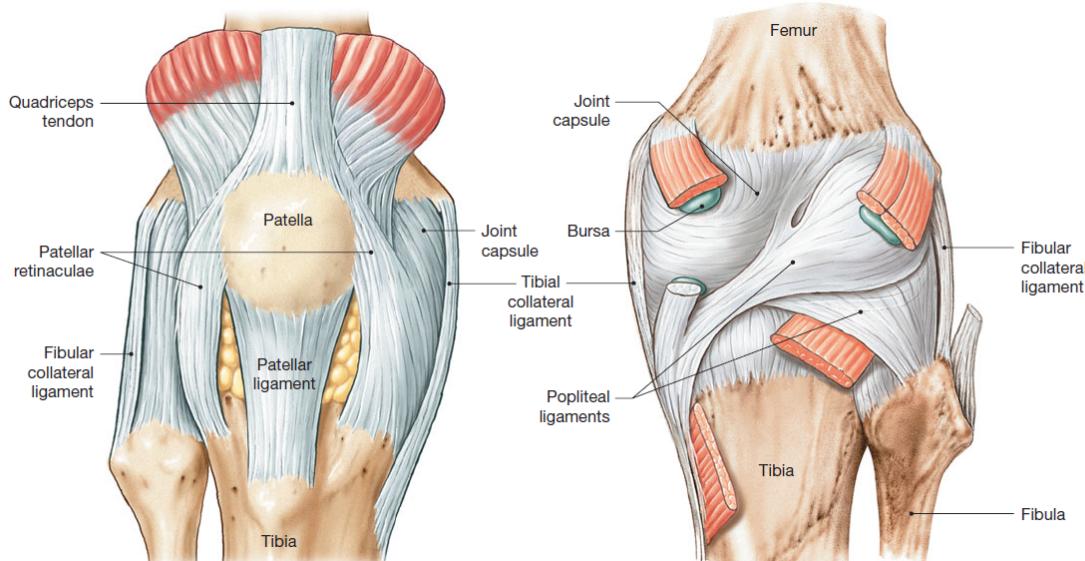


Figure 2.2: The anatomy of the knee with focus on the ligaments. Edited [10].

The ligaments patellar retinaculae and patellar ligament support the anterior surface of the knee. When the knee is fully extended, the tibial and fibular collateral ligament are responsible for stabilizing the joint. Between femur and the two lower bones in the leg, tibia and fibula, is the location of the two popliteal ligaments, which stabilize the posterior surface of the joint. In addition to the visible ligaments in figure 2.2 there are the anterior cruciate ligament (ACI) and posterior cruciate ligament (PCL) in the joint capsule. The two ligaments cross each other and are connected to the tibial and femoral condyles, which reduce the movement, anterior and posterior.[10]

2.2 Pain

The International Association for the Study of Pain (IASP) has defined pain as being “an unpleasant sensory and emotional experience associated with actual or potential tissue damage or described in terms of such damage” [11, 12].

Humans are aware of the surroundings and threats to their bodies because of the pain. The pain indicates that there might be a risk for permanent damage on the body, which refrain humans from danger and therefore increases the chances of survival.

Pain can be either nociceptive or neuropathic. Nociceptive pain is associated with tissue damage. This type of pain is related to the nociceptors, which are receptors with a high threshold that when stimulated may give the perception of pain in tissues [13]. Neuropathic pain occurs central from the nervous system. This pain can be caused by illness or physical damage.[13]

Furthermore, pain can be divided into three categories: acute pain (less than three months), persistent or chronic pain and cancer pain.[14] Additionally, the pain can be divided into qualities, which is shown in figure 2.3.

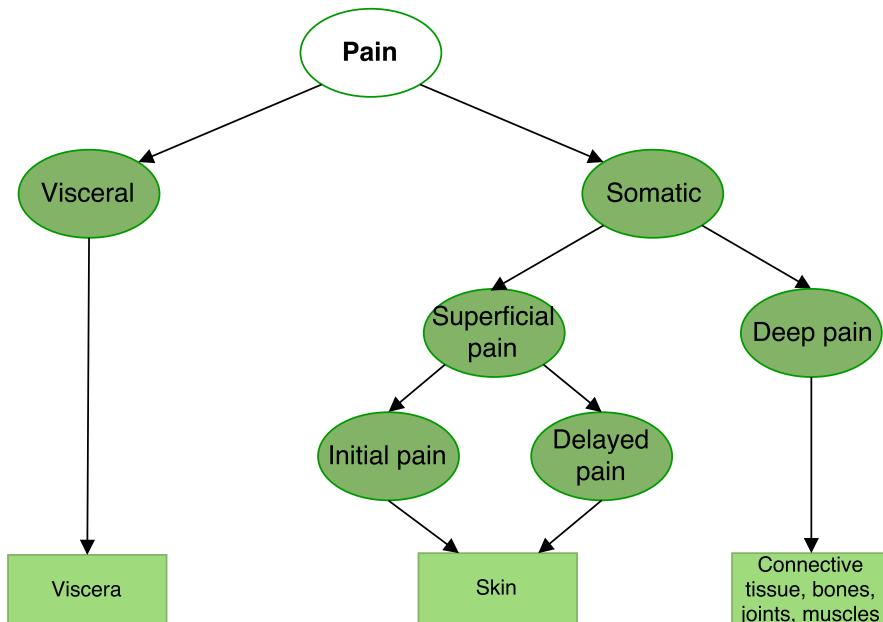


Figure 2.3: Model of pain qualities. Ovals with green background represent qualities of pain. The rectangles show where the pain originates. Edited [13].

Pain can be divided into two qualities; visceral and somatic pain. Examples of visceral pain include pain associated with gallstone and appendicitis. This pain can be characterised as a dull or diffuse feeling. Somatic pain is subdivided into superficial pain and deep pain. If the pain derives from the skin it is superficial pain, which furthermore is divided into initial pain and delayed pain. The initial pain is the first pain that is received, and characterised as sharp and localizable. The delayed pain, also known as the second pain, is a dull or burning pain that occur after a half to one second. This pain is more difficult to localise than the initial pain and lasts longer.[12, 13] The other type of somatic pain is deep pain, which is associated with pain from the muscles, bones, joints and connective tissue. This pain is described as a dull pain and it radiates into the surrounding tissue, which makes the exact pain area hard to point out.[12, 13]

2.3 Patellofemoral pain syndrome

Patellofemoral pain syndrome (PFPS) is a painful musculoskeletal condition [1, 2], which presents pain behind or around the patella. The patellofemoral pain (PFP) is often known as anterior knee pain and runner's knee. The pain is often described as diffuse knee pain, which is provoked by patellofemoral loaded activities like climbing stairs, running on hard or slanted surfaces, hiking, squatting or just prolonged sitting in the same position.[2, 6, 9, 15] However, this pain is not caused by previous trauma [6]. Knee pain is not the only symptom of PFPS, the individuals often complain about knee stiffness, patellofemoral crepitus, swelling knee and having trouble with common daily activities [6, 10]. The individuals may limit or stop physical activity because of the pain, and that can lead to an increase in weight [4, 6]. Physiologically PFPS is associated with incorrect movement of the patellar, that occurs when the patella moves outside of its ordinary track, which for instance can be movement in lateral direction instead of movement in superior-inferior direction.[10]

PFP is mostly prevalent in adolescents and younger adults who are physically active, but it can affect people of all ages and activity levels [1, 6, 15]. Additionally, females are affected about more than twice as often as males [4]. Furthermore, the PFPS may persist for up to 20 years, and thereby categorised as a chronic pain, and may lead to osteoarthritis [4, 6]. Despite the fact that individuals feel pain in the knee, there is not any structural changes in the knee such as significant chondral damage or increased Q-angle [4]. The Q-angle, also known as the quadriceps angle, is the angle between a line that follows the longitudinal axis of femur, and a line that follows a line from the tibial tubercle through the center of patella [16].

There is no definitive clinical test to diagnose PFP, but there is a test to elicit the knee pain by doing a squatting manoeuvre. The PFP is evident in 80% of people who are tested positive in this test.[6, 15] Therefore the diagnosis PFPS is often based on exclusion [4]. After the diagnosis, evidence based treatments may reduce pain and improve function that allows individuals to maintain physical activity [15]. The aetiology of PFPS still remains unclear [2].

Central sensitization

Central sensitization (CS) may give an evidence-based explanation of PFP, which is an unexplained chronic musculoskeletal pain [? ?]. CS is not completely biological, but may further depend on psychosocial factors [?].

When a peripheral injury occurs, signals are send to the central nervous system, that respond with a sensory stimuli which results in pain in the injured area. CS occurs when the central nervous system continues sending sensory stimuli despite that the injury is healed. The longer period of time the pain persists, the greater influence on the central neuroplasticity, which may lead to increased activity in the transmission cells. Thereby a longer duration may expand the area of pain, also known as widespread pain.[? ? ?]

CS has mainly two characteristics allodynia, which is pain occurred from a normally nonpainful stimulus, and hyperalgesia, which is excessive sensitivity to normally painful stimulus [? ? ?].

2.4 Identify and interpret pain

There are many ways to identify and interpret pain. To identify pain and find some physical damage that causes the knee pain, objective methods may be used. Subjective methods are used to interpret pain for collecting knowledge of the individual's pain intensity, behavior and how it is experienced.[17]

2.4.1 Identify pain

An objective pain measurement is often used when an individual experiences knee pain where a clinical examination of the knee can occur. This examination involves i.a. provocative tests, such as anterior and posterior drawer test, Lachman's test and pivot test that examines the integrity of the ACL and PCL. Furthermore is McMurray test which tests for meniscal tear.[18] Illustrations of the tests are shown in figure 2.4.

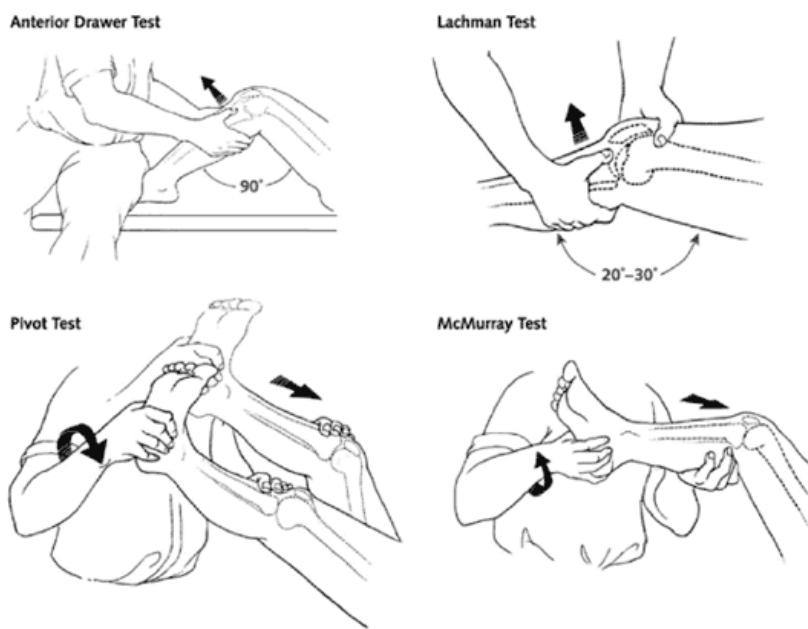


Figure 2.4: Clinical examination with provocative tests: Anterior Drawer Test, Lachman Test, Pivot Test and McMurray Test.[18]

In addition to clinical tests there is some paraclinical tests such as X-ray and MRI, but as mentioned PFPS does not show any structural changes in the knee [4], which makes it difficult for healthcare personnel to treat the individuals.

2.4.2 Pain interpretation

Pain is experienced and perceived subjectively [11, 17] and is dependent on personality and character [13], which is why it is important to measure the pain from the individual's perspective.

One of the most commonly method used to measure pain intensity is Visual Analogue Scale (VAS) [19]. VAS is often used in clinical and research settings, where the individuals mark their pain on a scale from no-pain to the worst pain they can imagine.[20] An illustration of a VAS is shown in figure 2.5.

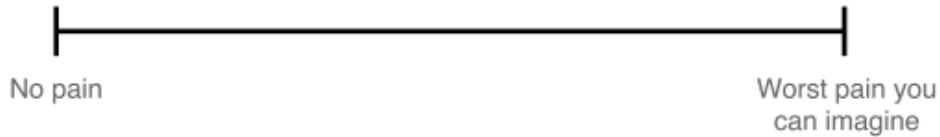


Figure 2.5: Visual Analogue Scale (VAS). Edited [20].

Additionally to mark pain on a scale are questionnaires used to define individual's pain. An example on a questionnaire is Knee injury and Osteoarthritis Outcome Score (KOOS), which contains questions about symptoms, stiffness, pain, daily living, function, sports and recreational activities and quality of life. When the individuals fill the scheme a score between zero and one hundred is achieved. A score at zero represents extreme knee problems, whereas a score at one hundred represents no knee problems.[21] The KOOS questionnaire can be seen in Appendix A.

Pain mapping

Individuals indicate their pain by 'placing both hands over their knees', because of the PFP characteristics, which makes it hard to precisely communicate their pain. Thereby pain mapping is a method for individuals to better indicate and communicate their pain. Pain mapping is a technique, that Harold Palmer introduced in 1949 [22], which is used to transfer an individual's perceived pain into an objective graph or map by drawing the pain area. Pain drawings can be made by the individuals who draw their pain areas on a body outline. Pain drawings can also be made by observers who observe the individuals and then draw from the signs the individuals are showing. An example of a body outline is shown in figure 2.6. Sometimes a subjective questionnaire is added to the pain drawings to get a more detailed overview of the pain to determine parameters associated with the pain. These parameters can also be useful in determining the source of the pain.[23]



Figure 2.6: An anterior body outline of pain drawing taken from the software application Navigate Pain.

Pain mapping are commonly used in clinical practice [23], and can be useful for individuals when they try to describe their pain. Pain maps may also be helpful in diagnosing individuals and follow-ups during or after treatment to get an indicator of the individual's response to the treatment.[8] According to Schott, there are some issues with the graphical representations of pain, some of which are problems with drawing a three-dimensional feeling of pain on a two-dimensional surface, and distinguishing between internal and external perceived pain on a map [23].

2.5 Knee regions

When looking at pain drawing samples from multiple individuals it is evident that there is a high variability in the distribution of pain patterns across different areas of the knee, which might be related to the PFP characteristics.

To distinguish between different pain areas, the knee can be divided into various regions as seen in figure 2.7, where the division of the left and right anterior knees are illustrated.

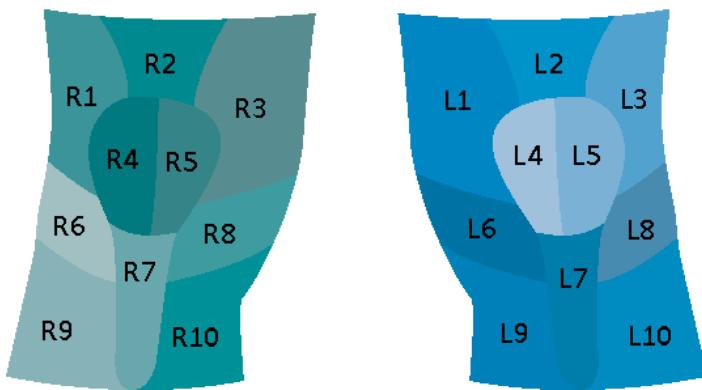


Figure 2.7: The regions of the left and right knees, where each knee is split into ten regions. Edited [24].

The divisions are inspired by Photographic Knee Pain Map (PKPM) which are designed to categorise the location of knee pain, for diagnostic and research purposes. PKPM represents both knees that makes it possible to identify unilateral and bilateral pain.[24]

The regions are based on the anatomical structures according to the areas where individuals often indicate pain. There are ten regions, where region 1 and 3 represent the superior lateral and superior medial areas for patella. Region 2 refers to quadriceps tendon. The patella is divided into lateral and medial regions, which are region 4 and 5. Region 6 and 8 are lateral and medial joint line areas. Patella tendon is region 7 and the two last regions, 9 and 10, are tibia lateral and medial.[24]

2.6 Machine learning

Machine learning describes the use of algorithms to make a system able to identify different data types, like images or text, for transcription of speech into text, matching news items, posts or selection of relevant results of search [25]. Machine learning identifies rules in a dataset from given input and output. If a computer learns this feature, it can be used to make intelligent decisions and predict specific outcomes.[26] Machine learning is a field that has seen a lot of progress over the past decades, partially because developers recognize the ease in training a system only using examples of the desired in- and output behavior. By using this method it is easier than trying to manually write a piece of code that anticipates different scenarios from different input types.[27]

2.6.1 Deep Learning

Deep learning is a branch of machine learning. The main difference between the use of machine learning and deep learning, is that deep learning is more suitable for handling raw data forms. Instead, a machine learning system often needs a feature extractor, that will generate a feature vector from the data that can be used as an input. Deep learning is based on different techniques that makes it able to handle raw data, mainly because of its structure.[25, 28] This allows the system to automatically detect necessary representations needed for classification and detection.

Neural network is a structure of deep learning which consists of different layers, that can be divided into input- and output-layers, with one or more hidden layers in between [28]. The key aspect of these layers is that the features are not defined by programmers, but they are found and learned from data using a general-purpose learning procedure.[25] An example of a neural network structure can be seen in figure 2.8.

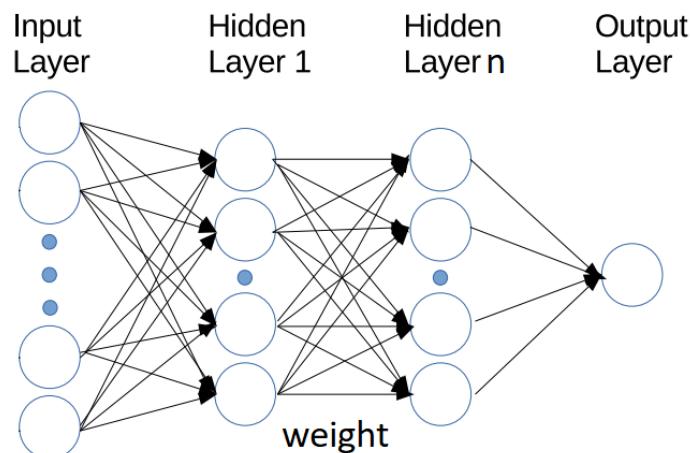


Figure 2.8: Example of the structure of a neural network. Edited [29].

The different layers consist of a series of nodes, where each node is connected by weights to one or several other nodes in different layers. The weights are interconnection between two layers and they work as a set of coefficients, defining an image feature[30]. In the input-layer the nodes receive data. The second layer will then receive the output from the previous layer, and this process continues through the layers until the output-layer is reached.[28] An example of how the hidden layers may affect an image in one type of neural network can be explained

in the following. Firstly, the system detects minor changes like edges. Secondly, the edges are compared and put together to make up different kind of shapes. In the third hidden layer, it will be further combined to make up an object that can be identified.[25]

Learning scenarios

Each neural network has to be trained and validated before used on real test data. There are different approaches for training a neural network, where the two main learning scenarios are supervised and unsupervised learning.

Supervised learning is the most common way of training in machine learning. When applying this learning method the neural network is trained with input data that has a corresponding label. The network calculates an output through the forward pass, where the data is simply passed through the network. This output may then be compared to the label, and used to evaluate the performance of the system. As a result of the evaluation, the network may learn from the data by doing a backward pass through the network, also known as back-propagation.[25] Overall supervised learning may be described as teaching the network how to associate a given input to a specific output [31], and is mostly associated with classification, regression, and ranking problems [32].

Unsupervised learning is when training is performed with data that has no output label. Instead of learning associations between input and output, the network organizes the data by searching for common characteristics [32]. An example of an unsupervised learning algorithm is k-mean clustering, where the unlabeled dataset goes through a classification, and splits data into clusters that are near each other [31].

2.6.2 Back-propagation

Back-propagation is a popular learning algorithm in neural networks, that is based on gradient descent, and used because of its simplicity and computationally efficiency.[33, 34] It is the learning process where the weights of a neural network are adjusted in order to reduce the error calculated between the calculated output of the network and the expected output. This makes back-propagation closely related to supervised learning, to which back-propagation is the most general method used.[34]

The basic concept is that gradients can be computed efficiently by propagation from the output to the input in order to minimize the overall output error as much as possible during the learning stage. This algorithm process is divided in two main stages: forward and backward. In the first process (forward), the back-propagation architecture is described as the inputs and weights multiplication of separate node summed with additional coefficient called bias.[30, 35]. When a neural network is initialized the weight may be set with a random value, meaning that the neural network may perform very poorly through the first iterations of the training. During the second backward part, a loss is calculated based on a loss function for every input that passes through the network, that may be used as a part of back-propagation to make the adjustments on the weights to reduce the loss. As training progresses, the loss should decrease as a result of the weight adjustments, and improve the performance of the neural network.[25, 31, 34] This learning process continues until optimal weights with minimum error is reached.[30]

Activation functions

Deep learning networks are structures of hidden layers, and this requires to choose the activation functions for computing the hidden layer values and to decide whether node could be considered as active or not [31]. The activation function is a nonlinear function which give neural network nonlinear capabilities [33].

Sigmoid activation function is one of the most common activation functions which monotonically approaches at some finite values as $\pm\infty$. Most common examples would be the standard logistic function $g(x) = 1/(1 + e^{-x})$ and hyperbolic tangent $g(x) = \tanh(x)$ which is shown in figure 2.9. Symmetric hyperbolic tangent is used more often because it converges faster than standard logistic function.[33]

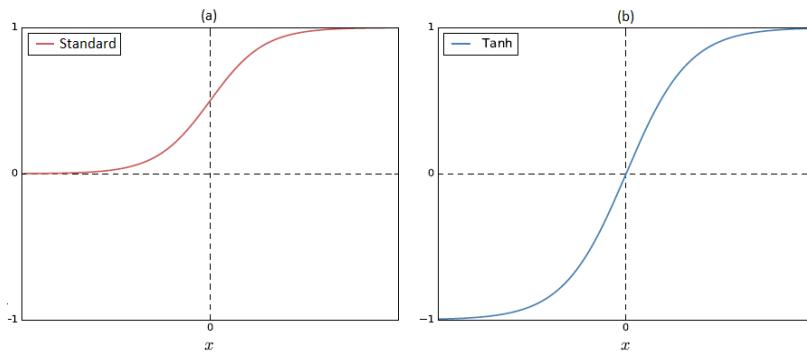


Figure 2.9: (a) Standard logistic function and (b) hyperbolic tangent. Edited [33].

Another activation function is rectified linear unit, which transforms the linear output to nonlinear function. However, the function still remains nearly linear, which means it could be easily optimized with gradient descent based methods[31]. In modern neural networks, ReLu is recommended to use as a default activation function and could be defined as $g(x) = \max\{0, x\}$ and shown in figure 2.10

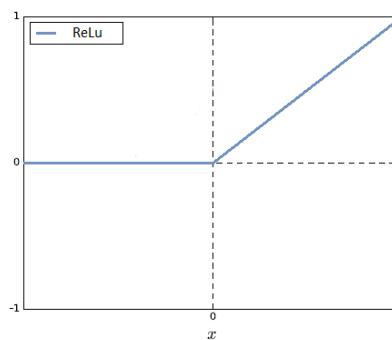


Figure 2.10: Rectified linear activation function. Function $g(x)$ is equal to zero when x is less than zero and $g(x)$ is equal to x when x is above or equal to zero. Edited [31].

Learning curves

Learning curves is used to observe the error on the training, validation and testing sets, which is increased because of randomly initialized weights and biases [34]. The error is presented over time, which in deep learning is expressed as the number of epochs. During the beginning of training, the training error of network will typically be relatively high, but during training the

error decreases monotonically, as the weights are adjusted in the network [34]. An illustration of how the error values are affected during training can be seen in figure 2.11.

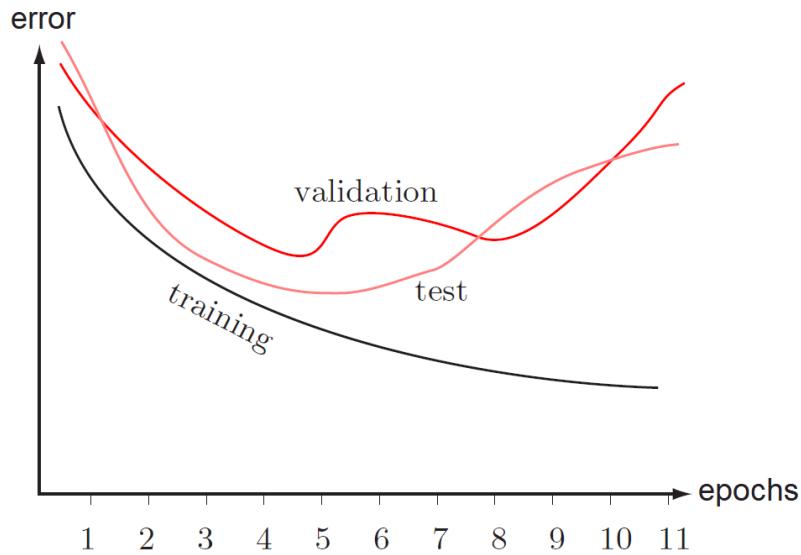


Figure 2.11: Illustration of how training (black), validation (red), and test (orange) error is affected by the increase in epochs. Edited [34].

From the figure it can be seen how the error value of the validation, can be used to evaluate the network. Near the fifth epoch the validation and the test error starts to rise, indicating that the network is overfitting to the training data, thereby decreasing the generalization abilities. Validation error can therefore be used as stop criterion for when the training is optimal, and prevent overfitting. Typically the validation and test error will always be higher than the training error, which is also seen in figure 2.11.[34]

Gradient Descent

Gradient descent is one of the most common techniques for optimizing neural networks. It is a way to minimize the loss function by updating the parameters, like weights, in the opposite direction of the gradient of the objective function.[36] The principle of the gradient descent could be explained as a "ball climbing down a hill" until a local minimum is reached as shown in figure 2.12. At each step, the opposite direction of the gradient is taken and the step size is determined by the value of the learning rate together with the slope of the gradient until the convergence is reached. Convergence means that oscillations of the value are small enough to call it the minimum value.[37]

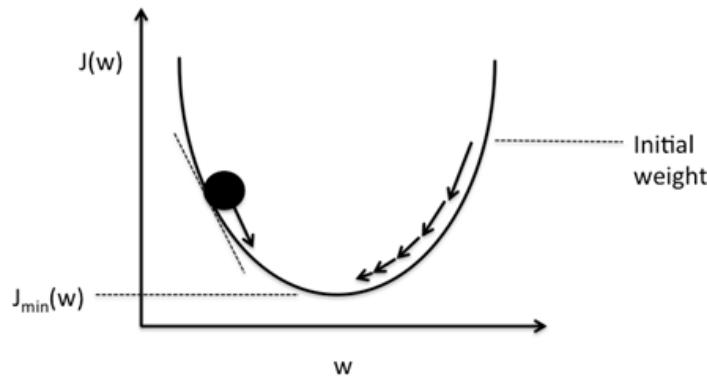


Figure 2.12: Illustration of gradient descent working principle, where $J(w)$ is the loss function, $J_{min}(w)$ is final approximation to the local minimum of $J(w)$, w is value of the parameter. The arrows indicate the step direction, i.e. the negative gradient.[37]

There are three variants of gradient descent: batch gradient descent, stochastic gradient descent, and mini-batch gradient descent. They differ on the amount of data used to compute the gradient of the loss function. Depending on which of the gradient descent variants is used, the trade-off between the accuracy and the runtime could be seen.

Batch gradient descent computes the gradient of the cost function with regards to the parameters for the entire training dataset. Batch gradient descent has the significant deficiency, it takes a single step for one pass over the training set, meaning the larger dataset, the slower algorithm updates the weights and the longer it will take to reach global minimum.[36]

Stochastic gradient descent (SGD) performs a parameter update for each training example and label. It is therefore much faster and it also performs frequent updates with a high variance causing loss function to fluctuate. These fluctuations enable it to jump to new potentially better local minima, but it may complicate the convergence to reach the exact minimum because of overshooting.[38]

Mini-batch gradient descent performs the parameters update for every mini-batch of training examples, specified by a batch size. By that, the variance of the parameter updates are reduced leading to more stable convergence and fast performance.[36]

Additionally, there could be few challenges while using gradient descent as an optimizer. It is difficult to pick a proper learning rate so few gradient descent optimization algorithms were invented. The most widely used methods are momentum, adagrad, adadelta and adam.[36]

Momentum is a method for accelerating SGD in a relevant direction and for reduction of oscillations. As a result, faster convergence is obtained but there is a risk of overshooting the minimum value.[39, 36]

Adagrad is an algorithm for gradient descent optimization which adapts the learning rate to the parameters. It performs larger updates for frequent and smaller for infrequent features. It has one weakness if the learning rate shrinks too much, the algorithm is no longer able to adapt.[36]

Adadelta is an adaptive learning rate method. Different to Adagrad, in Adadelta learning rate is monotonically decreasing. Using this optimizer, there is no manual tuning of the parameters of optimizer meaning that it can be applied in a variety of situations.[36]

Adam stands for Adaptive Moment Estimation. It is the most used method for computing adaptive learning rate and updating the parameters. This optimizer calculates the learning

rate for each parameter and stores momentum changes separately. This helps to reach the convergence faster with a decent learning speed.[40]

A study Patacchiola and Cangelosi showed the evaluation of the performance on different optimizers on a dataset containing 21977 female and male head pose images. The result is shown in figure 2.13, where the Adam optimizer had the fastest convergence rate and it reached the lowest loss values.

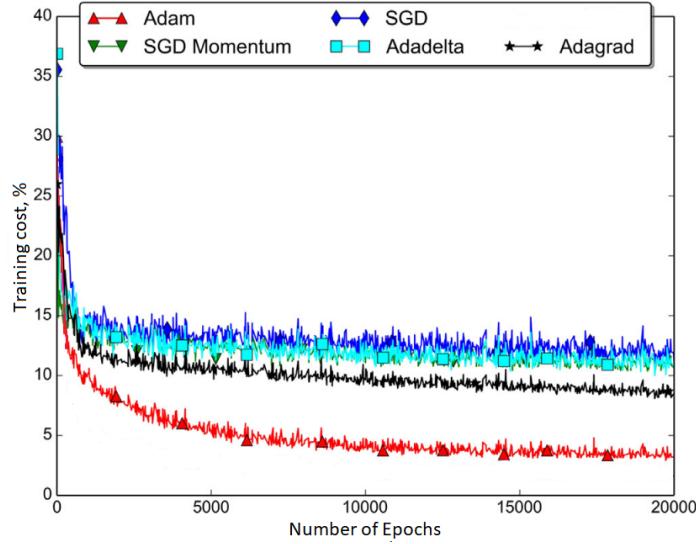


Figure 2.13: Comparison of the convergence speed between different optimizers used to train architecture on AFLW dataset. The loss values are the mean of the five attempts.[41]

However, the results are not always similar. All of the optimizers perform differently depending on the problem and parametrization, which in the majority of the cases is the most challenging part. This leads to the conclusion that there are not a winning optimizer and it has to be chosen based on every problem.[42]

2.6.3 Regularization

Many strategies are used in deep learning to reduce the test or training error, they are known as collectivity of regularization[31] , mainly used to restrict a models expressiveness in order to prevent it from overfitting, underfitting, generalization. Few of these strategies will be presented in this section.

Dropout

Dropout is a regulation technique used to reduce overfitting of the neural network. In a network the dropout is applied to the individual layers, and works by randomly dropping different nodes temporarily in the given layer during training. An illustration of the principle of dropout can be seen in figure 2.14. This parameter is specified with a percentage, which defines the fraction of the nodes that drop [43].

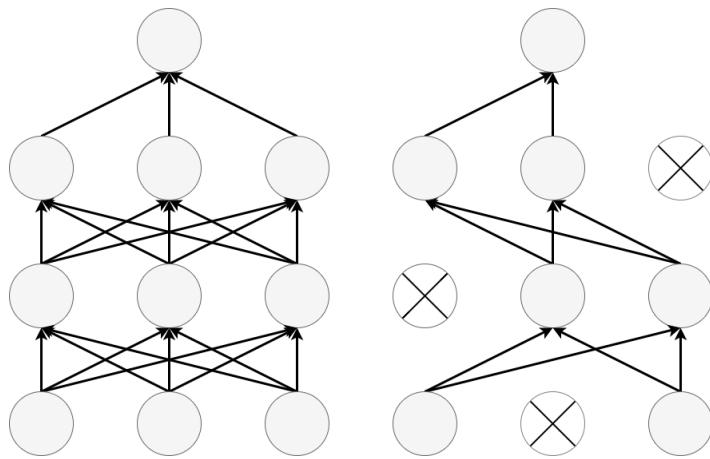


Figure 2.14: Dropout effect: The figure on the left illustrates a fully connected network, without dropout, and the right shows a network where dropout is enabled on the first three layers. Edited [44].

This reduces co-adaptation, where nodes compute the same features, to where this may increase the generalization capabilities for a neural network. A study Srivastava et al. has tested the use of dropout in different neural networks, and indicates that the most optimal range of dropout is 20% of the nodes in the visible layers, and 50% in the hidden layers.[44]

Initializers

In neural networks starting values of hyperparameters can have a significant effect on the training process [33]. Hyperparameters often define many different values that can be adjusted, to control the behavior of the algorithm, to which some parameters may affect the runtime and computational cost when training the model [31]. Weights, biases initializers could be used in order to take the values randomly and exclude the symmetry between nodes [?].

2.6.4 Core Neural Network layers

Convolutional, pooling and fully connected layers could be called as core neural network layers, where convolutional and pooling layers also act as hidden layers. In architecture of the network these layers have specific place and functions [? 31].

Convolutional Neural Networks

Convolutional Neural Networks (CNNs) is a type of special neural network for processing data with a grid-like topology [31]. CNNs perform highly in several tasks, including digit recognition, image classification and face recognition. The key aspect of CNNs is to automatically learn a complex pattern by extracting visual features from the pixel-level content.[29, 35]

The purpose of the convolutional layer is to recognize the features in the input by taking e.g. an image and scan it, then split it up into the feature maps. The terminology regarding the output of a convolutional layer can be referred to a feature map [31, 35]. A complete convolutional layer consists of several feature maps, so multiple features can be extracted at each location in the image [35].

Pooling layer

As previously mentioned convolution are typically followed by pooling [25, 31]. It consists of the same number of feature maps as the previous convolutional layer had. Each feature map is used as a new input in a pooling layer. Depending on the network's depth, the convolutional and pooling layers alternate until the last pooling layer is reached.[35] Pooling can be used to reduce the size of the dataset, which may increase computation speed, because the amount of data passed to the next layer is smaller. By pooling the input, a smaller representation is given, that still contains the relevant features.[31, 35] The pooling process can be defined as a window that passes over e.g. a feature map from convolution, where a value within the window is extracted. One type of pooling layer is max pooling that takes the maximum value within the window [31?]. A pooling layer may be defined simply by its window (Kernel) size, padding size and a stride length, where stride length is the number of values the window jumps as shown in figure 2.15.[?]

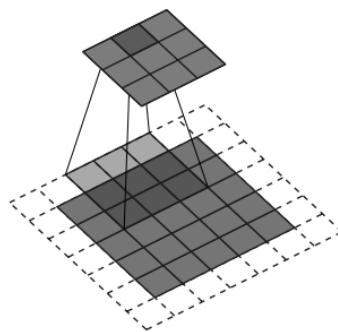


Figure 2.15: Illustration of pooling with 3×3 kernel over a 5×5 input using padding and strides (i.e., input = 5, kernel = 3, stride = 1, padding size = 1) [?].

Fully-connected layers

The combination of convolutional and pooling layers defines the part of the network which performs feature extraction while the classification part is made by fully connected layers [31]. Fully connected layers have full connections to all activations in previous layers. The further connections to the output layer is made compared the final values in fully connected layer to specific value of the class in the output. no decent source A typical architecture containing core neural network layers for character recognition of the images, called LeNet-5, can be seen in figure 2.16.

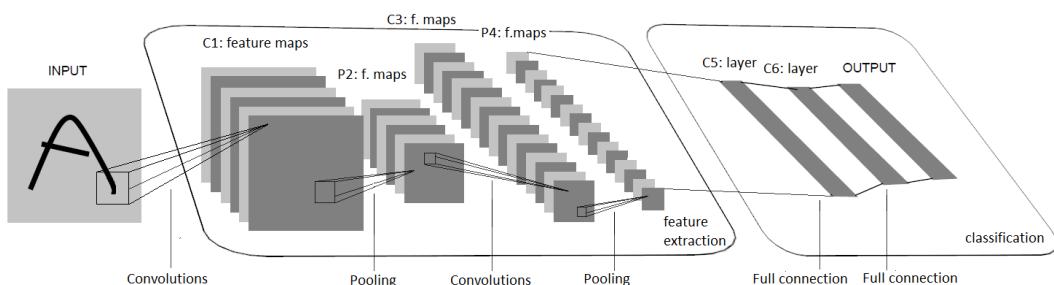


Figure 2.16: Architecture of LeNet-5, a CNN for character recognition. Each plane is a feature map and the size of the feature map differs throughout the layers. The model consists of two convolutional, two pooling and two fully connected layers [35].

Chapter 3

Methodology

This chapter creates an understanding of the available data, how it is analysed and prepared before including it in deep learning models. Additionally, the program Navigate Pain and programs used to development of the deep learning models are described. Furthermore, the implementation of the deep learning model is presented consisting of hardware, software and architecture of used neural network.

3.1 Data description

Data used in this project were collected beforehand from an on-going FOXH trial which is conducted in collaboration with Danish and Australian universities. The data consists of pain maps which were drawn by individuals with PFPS through the use of an application, Navigate Pain, in a clinical setting. Navigate pain is further described in section 3.1.1. The pain maps are both from individuals with uni- and bilateral PFP. An example of a pain drawing with bilateral pain is shown in figure 3.1.



Figure 3.1: An example of a pain drawing from individual with PFPS. The red markings indicate the area of pain perceived by the individuals. In this case the PFP is bilateral (on both knees).

In addition to the pain maps a corresponding dataset was available. This contained information regarding the individuals in terms of i.a. age, gender, pain duration, and intensity and the most prominent knee for pain. However, not all of the information was present for all

individuals. Before using the data in the deep learning models, a manual data handling was necessary. This incorporated matching the given pain maps and associated ID regarding the individuals. Furthermore, specific information like gender, pain duration, and intensity were collected.

To create more pain maps a split body approach was used, which contained splitting pain maps in two legs, whereafter the pain was mirrored and only visualized on the right knee. This resulted in 333 pain maps with gender and pain duration, and 319 pain maps with gender and pain intensity.

3.1.1 Software application: Navigate Pain

Navigate Pain is a software application that is used to visualise the location, shape and spatial distribution of pain from individuals to healthcare personnel. The application permits individuals to draw their pain into a body outline. Navigate Pain android was developed at Aalborg University, and a commercial web application is available at Aglance Solutions (Denmark).[45] Figure 3.2 illustrates the process using the application.

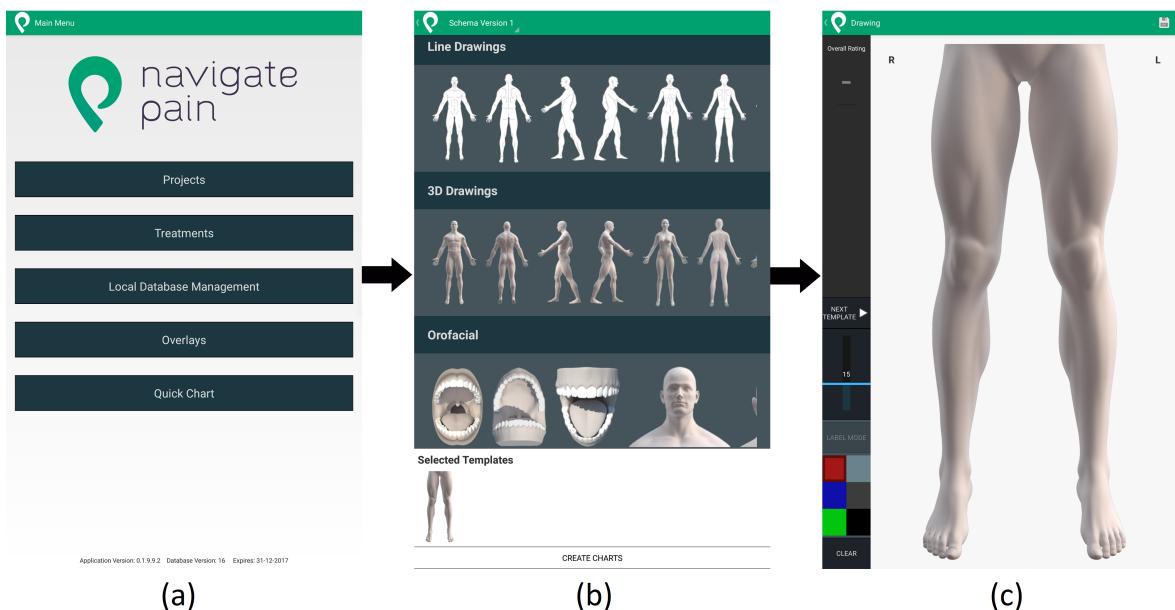


Figure 3.2: The process for making a pain map with Navigate Pain. (a) shows the main screen, (b) categories of body outlines and (c) body outline for lower extremities.

In figure 3.2(a) is the main screen. By clicking on "Projects", a folder with individuals is created. For each individual information like name, age, height is saved. Before the individual can draw their pain areas, the body outline has to be chosen, which is illustrated in figure 3.2(b). The body outlines are divided into five categories: Line Drawings, 3D Drawings, Orofacial, Special Zooms and Knee Pain. In the bottom the selected templates are shown. When clicking on "create charts" the screen in figure 3.2(c) is shown. Here it is possible to draw the pain areas with different colors and line thickness, which can be seen in the left side of the screen. Afterwards, the pain map can be saved.

3.1.2 Data representations

It is presumed that different representations of the pain maps affect the performance accuracy of a deep learning models, hence different data representations were created. A study by Boudreau et al. found a correlation between a prolonged pain duration and the size of the pain area. It was shown that the pain area increased for individuals that have a pain duration for longer than five years compared to those with a pain duration below five years. Likewise, pain intensity had a correlation with the size of pain area for individuals with a pain duration for more than five years. Furthermore, the shape of the pain developed from a U-shape to an O-shape for individuals with a pain duration above five years.[9] However it is unknown whether the morphology or location of the pain influence either pain duration or intensity to which two data representations reflecting morphology or locations of the pain are created. A combination of the two data representations is created to achieve a third data representation which both include the morphology of the pain and the different active knee regions. Furthermore, gender may be considered as an important parameter to use as an input, because of the difference prevalence in PFPS. The distribution of gender in the available data is investigated by creating a histogram, which is shown in figure 3.3.

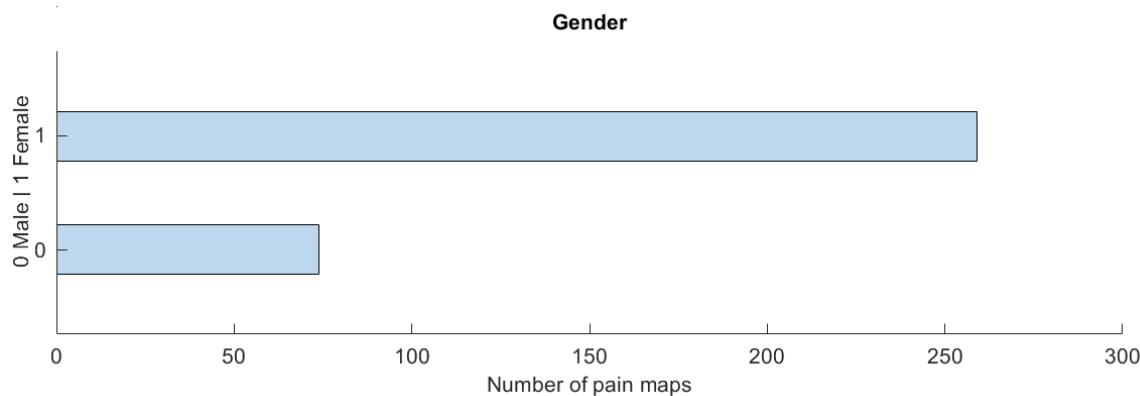


Figure 3.3: Histogram of the distribution of gender.

According to the given data the distribution is higher for females than males. The females constitute 156 of the 207 individuals, and males constituted 51 individuals.

Since the pain duration of PFP seems to affect the size and morphology of the pain area, it is chosen to classify the three data representations in proportion to pain duration. Likewise it is chosen to classify according to pain intensity because of the influence from the size of pain area. The three data representations are referred to as morphology-, regions- and combined-representation.

3.2 Pre-analysis

The pain maps and associated pain duration as well as pain intensity are analysed to get an overview of the data. The data is analysed in MatLab, where the distribution of the outputs, pain duration and intensity, are investigated whereafter intervals used for classification in the deep learning models are decided. Furthermore, different threshold values are analysed according to five pain maps to select the threshold which should define when a region is active. Simple linear regressions are made to investigate whether pain duration or intensity have a linear relation to the size of pain.

3.2.1 Classification of data

The deep learning models should classify the input, pain maps and gender, in different intervals in relation to pain duration or intensity intervals. To help define the class intervals of both pain duration and intensity, histograms were created.

A histogram of the pain duration associated with the pain maps is illustrated in figure 3.4.

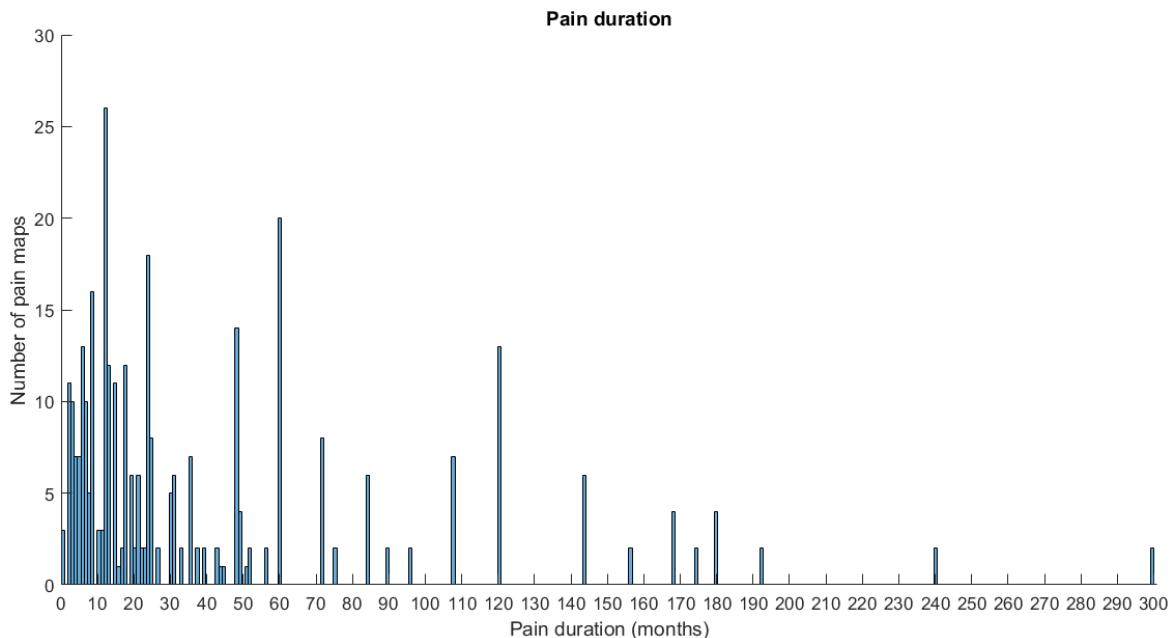


Figure 3.4: A histogram of the pain duration according to the number of pain maps.

The pain duration is divided into some classes which the models should classify in addition to. To test the models are the data firstly divided into two extremes, since it is assumed that if the models predict badly with the extremes, the models will not predict better with multiple classifications of pain duration. By considering the amount of data and the distribution shown in the histogram class intervals were chosen to be 0 to 12 months ($n=144$), and 36 to 300 months ($n=122$). Afterwards, the interval between is used as a third classification.

In the appurtenant data to the pain map the individuals have stated their pain intensity as the worst pain in the last 24 hours and the last seven days. It is not assumably that the individuals have performed any PFP provoked activity in the last 24 hours before drawing their pain, therefore it is chosen to use the worst pain intensity in the last seven days to

get a more average value for the worst pain intensity. To illustrate the distribution of the individuals' worst pain intensity in the last seven days a histogram is created which can be seen in figure 3.5.

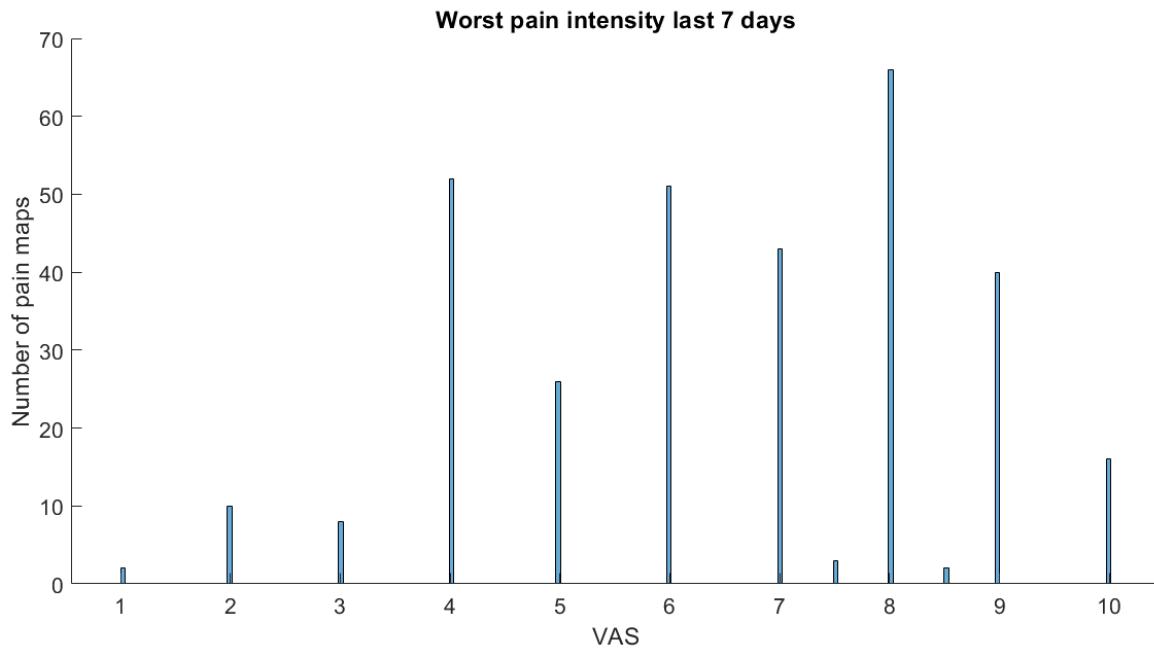


Figure 3.5: A histogram of the worst pain intensity in the last seven days according to the number of pain maps.

Likewise to the pain duration classification, the worst pain intensity is divided into the extremes, which are chosen to be intervals 1 to 4 ($n=72$) and 8 to 10 ($n=124$). The third classification constitute the interval between the extremes.

3.2.2 Threshold selection

In relation to the data representation that contains information about the active pain regions, it is necessary to find a threshold that decides when a knee region contains enough pain pixels to be considered active. A threshold is required to increase the confidence of an active pain region by avoiding minimal contributions e.g. small pain areas in the associated regions. Simultaneously the threshold may not be too large so that potential pain regions will not be incorporated. The threshold to indicate active pain regions is decided based on an analysis, where threshold values of 5, 10 and 15%. A 0% threshold was used as a reference. The analysis of the threshold is tested on five random pain maps to get a general impression of the data. To better distinguish the regions in figure 2.7 different colors are used as shown in figure 3.6

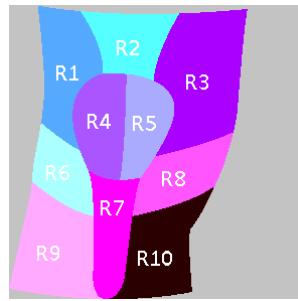


Figure 3.6: Knee regions colored to easier distinguish.

An example of pain maps and appurtenant bar chart are illustrated in figure 3.7. The pain maps are colored in the same colors as figure 3.6 to indicate which regions are affected by pain according to 3.6(a) no threshold, 3.6(b) 5% threshold, 3.6(c) 10% threshold and 3.6(d) 15% threshold. Figure (e) shows a bar chart that indicates how many and which active regions there are according to the threshold values.

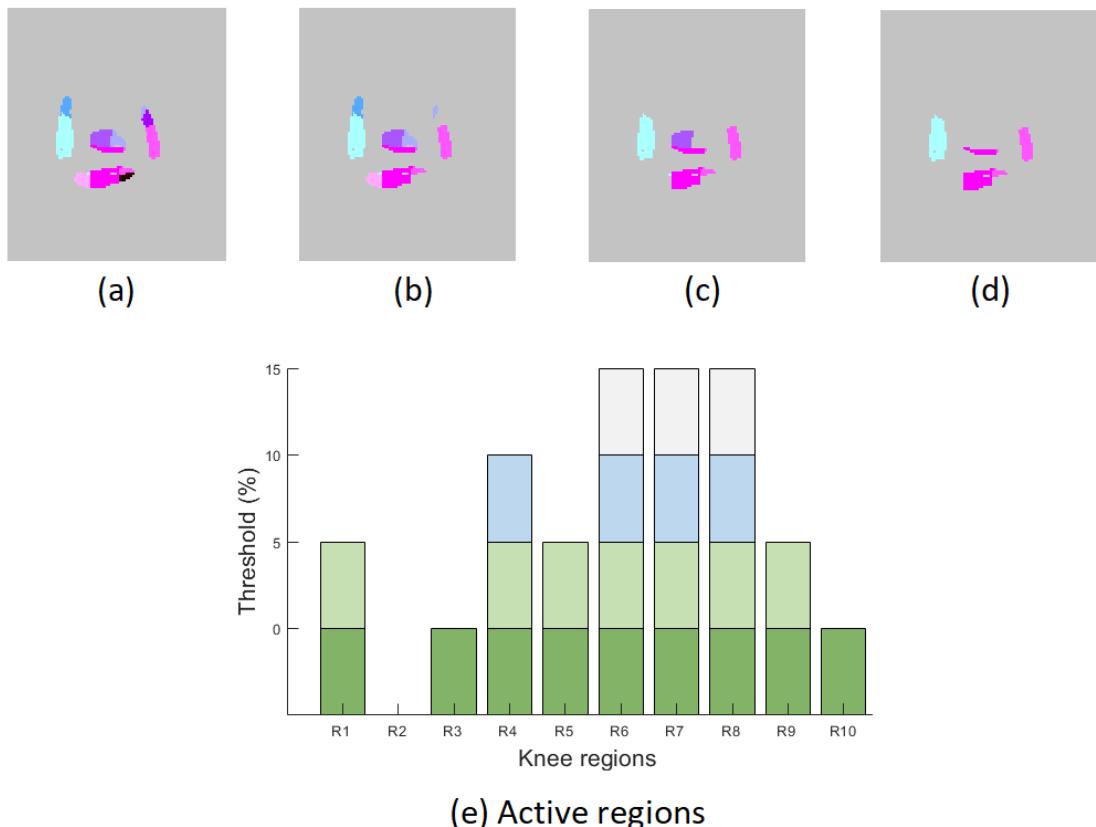


Figure 3.7: The active knee regions when the threshold is (a) 0%, (b) 5%, (c) 10% and (d) 15%. (e) is the bar chart that indicates how many and which knee regions that are considered active.

According to figure 3.7(a) with no threshold, and (e) it is shown that the knee has nine active regions when no threshold is applied. In proportion to the active regions, pain regions R3 and R10 are very small and thereby the first regions are discarded when the threshold is increased by 5%, which is shown in figure (b). By comparing figure (a) and (b) minor changes according to the missing regions can be seen, compared to figure (c) and (d) where greater

areas disappears after increasing the threshold to 10 and 15%. Based on analysis of the five pain maps and bar charts, in figure 3.7 and appendix B, a threshold on 5% is chosen to avoid including minor pain areas, like region R10, as active knee regions, and to avoid discarding too many and large areas, like regions R4 and R5.

3.2.3 Simple regression models

To verify the assumption of nonlinear data, linear regressions on the data representations and output, pain duration or intensity, are created. Other features in the morphology- and location-representation are respectively the size of the pain (number of pain pixels) and the number of active regions. If these simple features have a linear correlation to the pain duration or intensity, it may not be significant to investigate morphology and location as features in the deep learning models.

To investigate if pain size has a linear correlation to pain duration, a linear regression is created, which is shown in figure ??.

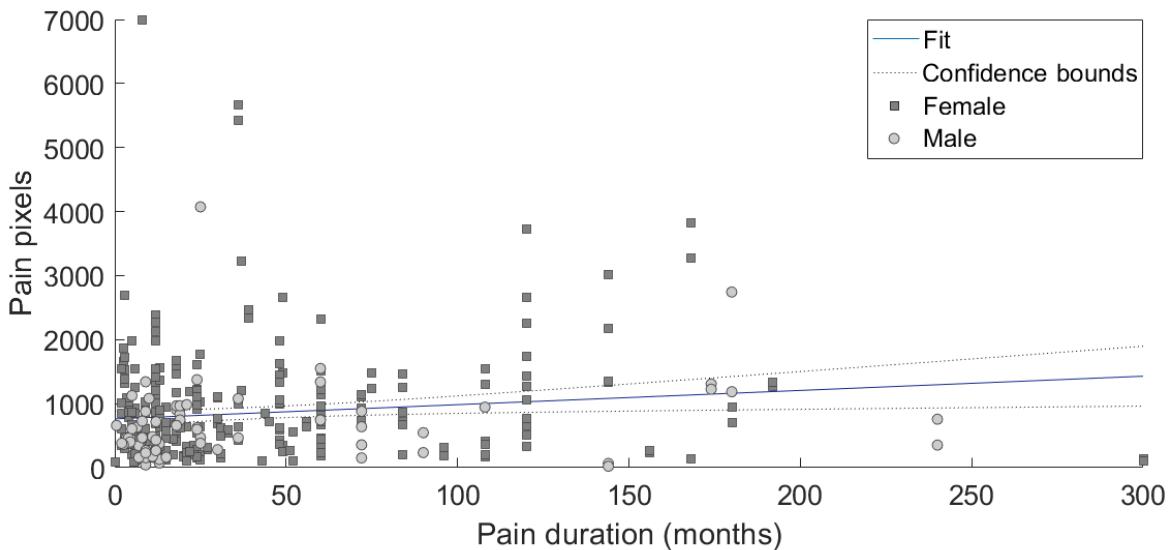


Figure 3.8: A linear regression of the the number of pain pixels and pain duration.

As a result of the linear regression model was $R^2 = 0.046$, which indicates that there is no linear correlation between the number of pain pixels and pain duration.

A linear regression model of the number of pain pixels and pain intensity is made, which is shown in figure 3.9.

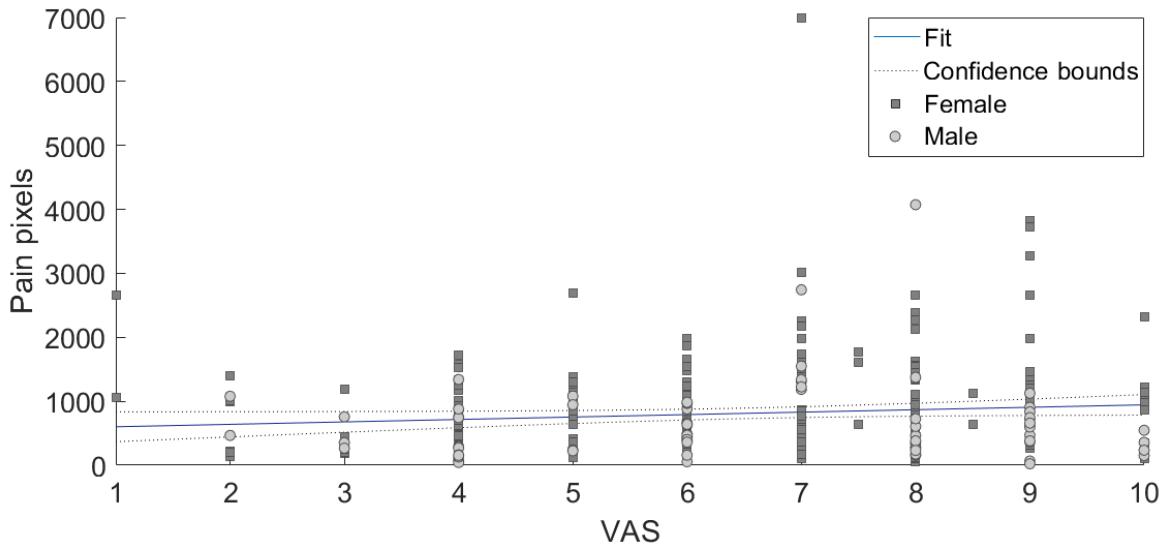


Figure 3.9: A linear regression of the number of pain pixels and pain intensity stated in VAS.

The result in this linear regression was $R^2 = 0.0117$, which indicates that there is no any linear correlation to be found between the number of pain pixels and pain intensity.

These linear regression models are not very suitable when trying to find a correlation between the number of pain pixels and pain duration or intensity. However, they can be compared to the performance of the deep learning models.

A linear regression between the number of active pain regions with a threshold on 5% and pain duration is shown in figure 3.10.

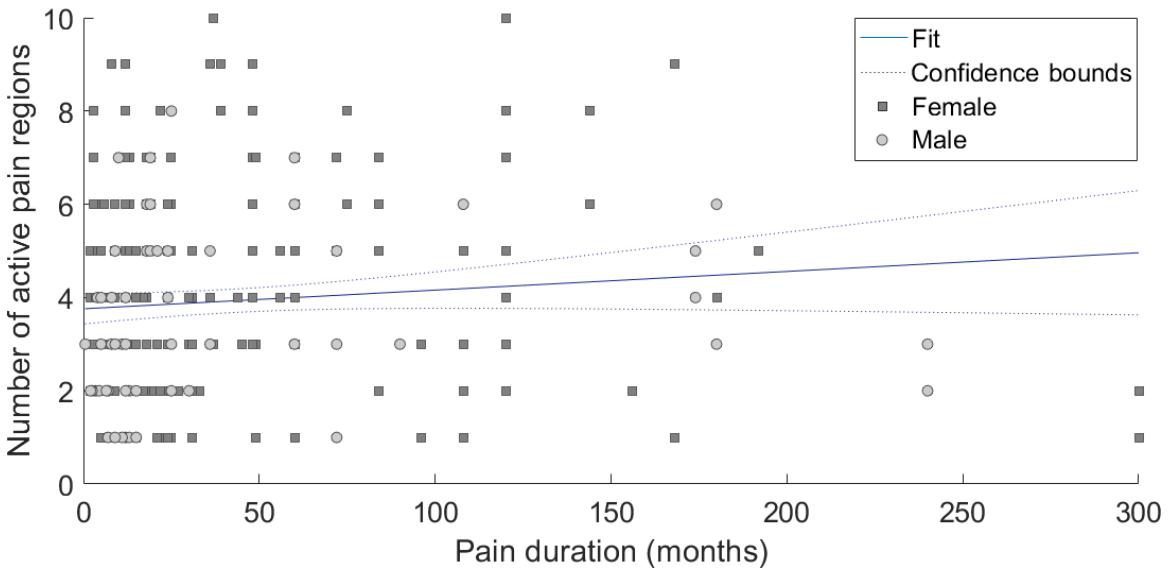


Figure 3.10: A linear regression of the number of active pain regions and pain duration.

The result in this linear regression between the number of active pain regions and pain duration

was $R^2 = 0.0357$, which indicates that there is no any linear correlation to be found. Thereto, a linear correlation between the number of active pain regions and pain intensity is shown in figure 3.11.

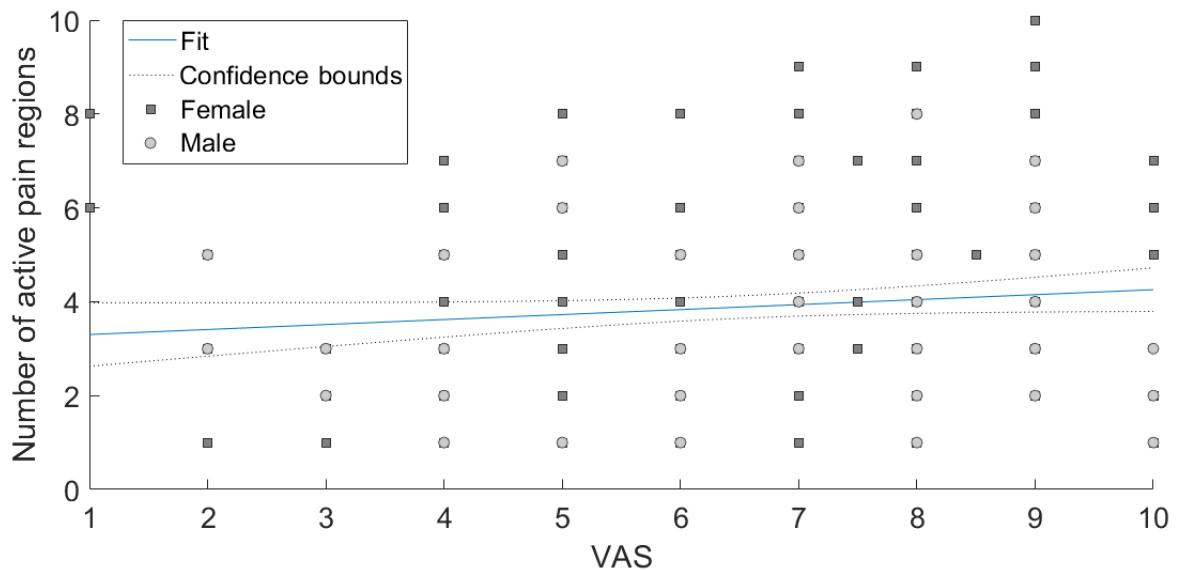


Figure 3.11: A linear regression of the number active pain regions and pain intensity.

The result of the linear regression between the number of active pain regions and pain intensity was $R^2 = 0.00833$, which once again indicates nonlinearity.

Based on the four linear regression models, it is assumed that a single feature, number of pain pixels or number of active pain regions with a 5% threshold, may not have a simple correlation with the outputs, pain duration or intensity. Hence a deep learning model may find patterns in the pain maps in relation to either pain duration or intensity.

3.3 Pre-processing

The data is pre-processed in MatLab to prepare the three different data representations. The three data representations are morphology, regions and combined, which are described in section 3.1.2. Common for the data representation is that the pain maps are imported as image-matrices whereafter the matrices are resized, since the given pain maps was collected at different resolutions (screen sizes). Furthermore, the matrices are cropped to sort out unnecessary data like the areas inferior and superior to the knee. Before the data is used as an input in the deep learning models, each matrix which represent an image, is converted into a vector whereafter they are assembled in one matrix for each data representation. To get additional information associated with the pain maps, gender is added by including a column vector to the three matrices. In addition to the input, a classification label is added. The label, which is either pain duration or intensity, is added as a column vector. The following sections describe the pre-processing of the individual data representations.

3.3.1 Morphology-representation

The first representation of data is a binary matrix of the original pain maps. Firstly, the image of the original pain map is gray-scaled to get a one-dimensional matrix instead of a three-dimensional RGB-matrix. This matrix is then converted into a matrix consisting of zeroes and ones, where the pain pixels are defined with a value of one. An original pain map and a pain map consisting of a binary matrix is shown in figure 3.12.

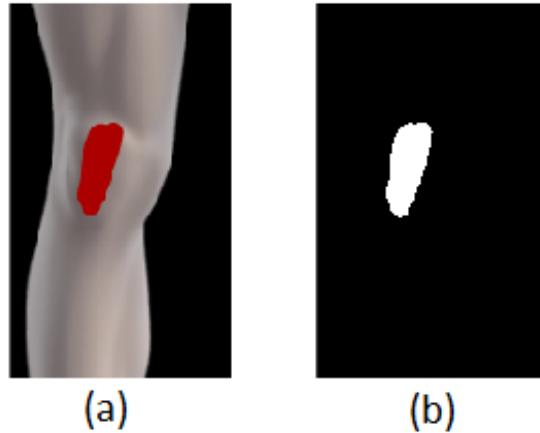


Figure 3.12: (a) Original pain map and (b) image consisting of a binary matrix where white color represents the pain pixels.

An illustration of this morphology-representation is created to convey how the data is assembled and transferred to the model. The illustration is shown in figure 3.13, where a matrix containing image-vectors for all the pain maps and appurtenant gender and either pain duration or intensity.

	Binary image-matrix												Gender Duration/ pain intensity
Image-vector 1	0	0	0	0	1	0	0	1	0	1	2		
Image-vector 2	1	0	0	1	0	1	1	0	1	0	0		
Image-vector 3	1	0	1	0	0	1	1	0	1	0	1		
Image-vector 4	0	1	1	1	0	1	1	1	0	1	0		
Image-vector 5	1	0	1	0	1	1	1	0	0	0	2		
Image-vector 6	0	0	0	1	0	0	1	1	1	1	1		
Image-vector ...	1	0	1	0	0	0	0	1	1	1	1		
Image-vector n	0	0	1	1	1	1	0	1	1	1	0		

Figure 3.13: An illustration of the matrix of the morphology-representation. The matrix consists of image-vectors for each individuals where the two last columns indicate the corresponding gender (blue column vector) and either pain duration or intensity (green column vector). The image-vectors have a length equal to the number of pixels in the pain maps.

3.3.2 Regions-representation

The second representation of the data is a matrix consisting of vectors with 20 values which indicate pain in relation to the knee regions. An image of the knee regions as shown in figure 2.7 are converted into a matrix consisting of 10 values, which represent each knee regions. This matrix is superimposed on the binary image of the pain map, which results in a matrix with pain pixels represented in each knee region. An illustration of the knee regions and an example of a pain map with the pain divided into regions are shown in figure 3.14.

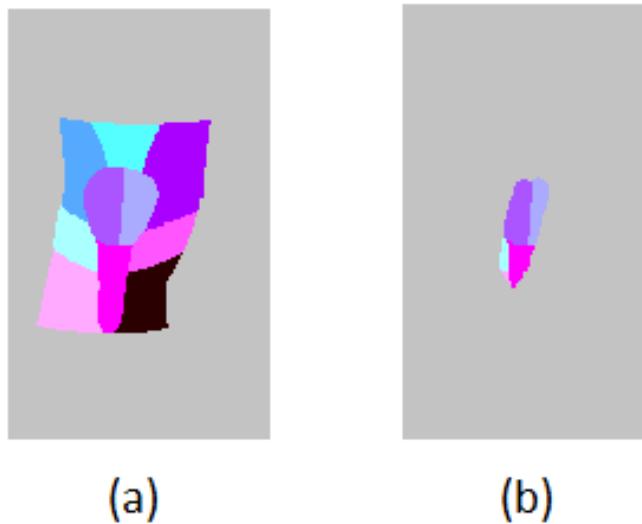


Figure 3.14: (a) Knee regions and (b) an example of a pain map with pain in the specific regions.

After superimposing the two matrices, knee regions and pain pixels, the number of pain pixels in each active knee region is found. This number is compared to the total number of pixels that are in each knee region, so knee regions with less than 5% pain are excluded. The threshold on 5% is chosen based on the analysis in section 3.2.2. As a result a vector with 10 values is created with zeros and ones, where one represent an active pain region. Region-representation is imported in the deep learning models the same way as the morphology-representation, which is illustrated in figure 3.13. The only difference is that the length of the image-vectors respond to the 10 regions, and therefore there are only 10 values for each pain map.

3.3.3 Combined-representation

The third representation of the data is a matrix consisting the morphology of pain divided into the knee regions. In this representation the superimposed matrix from the region-representation is used. Additionally one-hot encoding is used to divide the pain into different knee regions. One-hot encoding is a way to separate categorical data into binary data [46]. This means that the 10 values do not have a correlation when analysed in the deep learning model. After one-hot encoding, the superimposed matrix consists of 10 layers where each layer represents a knee region, which is illustrated in figure 3.15(a), and afterwards converted to image-vector with gender and the output as illustrated in figure 3.15(b).

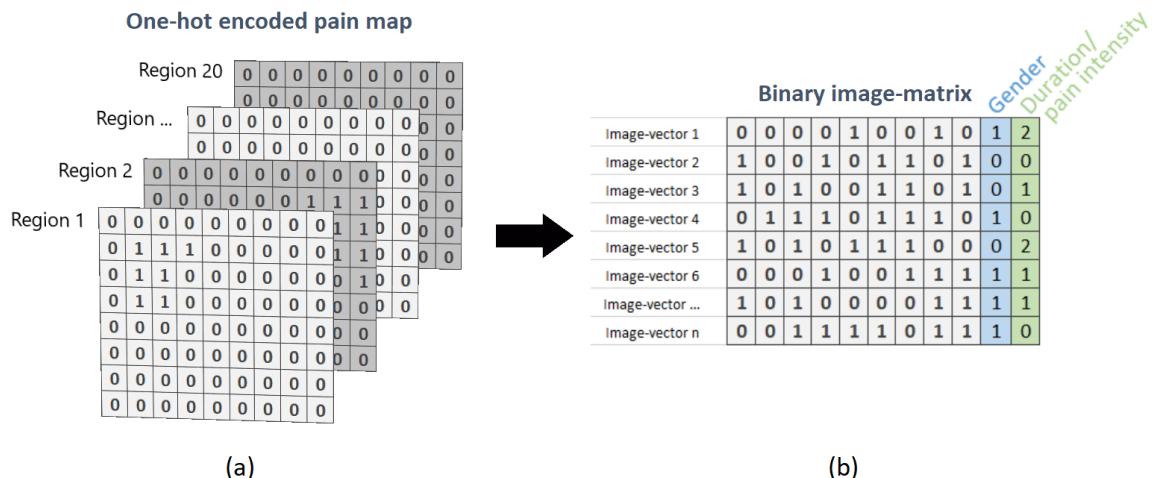


Figure 3.15: (a) An illustration of a one-hot encoded pain map and (b) shows the image-vectors in one assembled matrix with gender and either pain duration or intensity.

3.4 Neural network implementation and models

Building neural network for classifying pain maps was try and error process. In this project the deep learning method is used to classify the pain maps and gender by determined outputs. The data is a set of 2D-images combined with gender and the outputs are pain intensity and duration. For classification purpose supervised convolutional neural network is used followed by fully-connected layers at the end. This architecture of the model is chosen to the interest of morphology and location of the pain. as there was no "golden recipe" for most optimized The models are trained and tested on a single computer with GPU and ran on Python programming language with Keras library and Tensorboard tool for visualizing results. Specifications of hardware and software are described in detail in section Software and hardware.

3.4.1 Software and hardware

The neural network developed for this study, was programmed using Python v3.6.3. Python is an object-oriented and general-purpose programming and scripting language, that may be used for e.g. programming websites, mobile applications, but also for machine learning programming applications. For development of deep learning application in python, different libraries are available, where some of the popular are the Theano and TensorFlow libraries.[47], where the Tensorflow v1.3.0 library was used in this study. An additionally library, Keras v2.0.8, was imported, which runs on top of either Tensorflow or Theano, and is a high-level neural networks application programming interface (API). Keras is a simplified version of the two libraries, which allows for fast and easier prototyping of neural network [?]. This was deemed suitable given that no previous experience with neural network were available during this study.

Utilization of graphics processing unit (GPU) computation was also implemented using CUDA drivers and cnDNN communication libraries, which allowed for faster runtimes, then through the use of the central processing unit (CPU).

The neural network was created on a laptop with 4x 'Intel® Core™ i7' CPU's and one GPU of type 'Geforce GTX 970M' with specifications listed in table ??.

CPU specifications	Value
Cores	4
Clock speed	2.6 GHz
Cache	6 MB SmartCache
Memory bandwidth	34.1 GB/s
GPU specification	Value
CUDA cores	1280
Base Clock speed	924 MHz
Memory	6144 MB
Memory bandwidth	120 GB/s

Table 3.1: Specifications of CPU and GPU [? ?]

3.4.2 Data handling and design choices of the models

This section presents the different models, their architecture and implementation used in this study. For classification of each representation a corresponding model was made. The classifier models for *morphology*- and *combined*- representation were operating on a pixel level by learning features of the pain charts, whilst the *location* representation operated from the 10 element location vector. The architecture of the models consist two main parts, a convolution part and a fully-connected part, except the *location* model that only consist of the latter. Convolution works as feature extraction of the pain maps, where convolutional and pooling layers alternates in order to extract relevant features out of the image, as described in subsection 2.6.4. The fully-connected part works as classification, where computed feature maps gets weighted and classified to a particular class in the output layer. A higher generalization performance of the classifiers, was investigated through the use of regulation and optimization methods. The available data was separated into a training and test subset, from which training data was used to regulate and optimize the model. Supervised learning was used for training the three models. The common input for all of the model were gender, along with the different image representation

Morphology representation model

The pain map representation of morphology works as input of the model, where the input layer is a convolution layer. This layer is set-up to receive a input shape of the dimension of the pain map, that is defined during re-scaling of the pain maps in 3.3.1. Gender works as secondary input in the second section of the model, along with the pain maps features extracted through the convolution layers. Before the pain maps features reach the fully connected layers it is flattened from a matrix to a single row in order to merge the features with gender. The merged data passes through fully connected layers and reaches the output layer where it is given a percentage value according to which class it fits the most. Morphology model architecture is shown in figure 3.16.

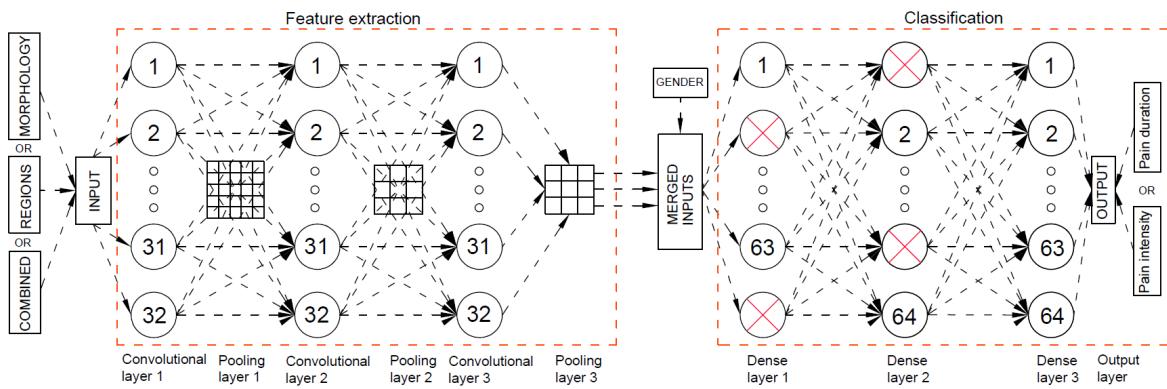


Figure 3.16: Architecture of the neural network model used for this study.

Locations representation model

The architecture for this model only contains fully connected layer, since the data representation only contains 21 element vector that reflects the active pain regions and gender as described in 3.3.2. It was evaluated that there would not be any benefits in making the model more complex by adding of convolution, based on the information available from vector, since the level of detail in relation to morphology was very simple. The architecture of the model is illustrated in 3.17.

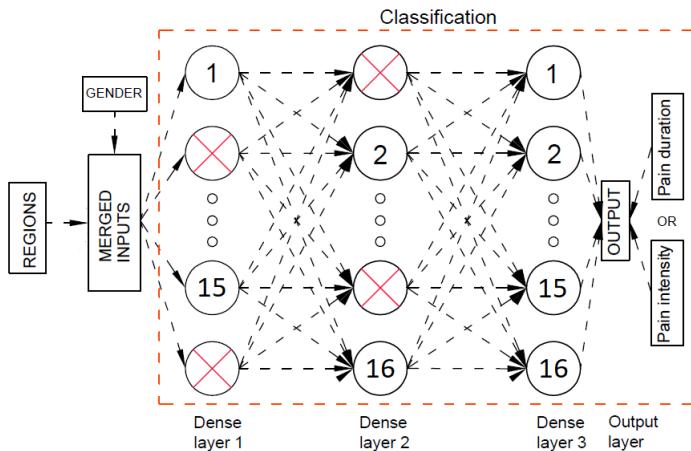


Figure 3.17: Architecture of the regions-representation model.

Combined representation model

The architecture of this model is nearly identical to that of the morphology representation model. The main difference can only be seen in the input layer for the pain map representation, where the input shape is altered to contain 10 image layers pr pain chart instead, which is also illustrated in Figure 3.16. This is the result of the one-hot encoding done to the images as described in 3.3.3.

3.4.3 Data handling in python

Pre-processed data is loaded from a *.mat* file into python. Data within the file was pre-shuffled and split into a training and test subsets, as described in ???. The training subset

was made up 85% of the available data and test subset the remaining 15%. Data was pre-shuffled to ensure that the training and test subset were separated at random. Furthermore the data was separated to prevent data used for training to be used for testing, to which this would give a false estimation of performance. The training subset was used to train and optimize the models, to find the optimal weights with the backpropagation algorithm [33]. The test subset was used to evaluate the generalization of the model, after training and optimization. Morphology, and combined representations, were reshaped from a row vector, back into a matrix to retain their 2D structure of the pain map. Classification labels used for during supervised learning was one-hot encoded, so the number of label values were compatible to the number of outputs in the models. This was a result of the loss function *categorical_crossentropy* used for the models, that tries to reduce loss between the categories.

3.5 Optimization

To reduce overfitting and improve general performance of the model, different methods was tested and implemented. The performance of the models was evaluated from an average accuracy, sensitivity and specificity, that was calculated from a 10-fold cross validation on the training subset. THIS IS AS FAR AS I'VE GOTTEN A grid search method was used to find optimal hyperparameters for the models. This allowed to run the models through different parameters, from which the parameters the resulted in the highest performance was used. by cross-validating them 5 times and evaluating the mean values with SD. Activation function, dropout, optimizer, learning rate, type of initializer, number of neurons, batch size and number of epochs were tested using this method. All used optimization techniques are presented in this section and table with used values is presented in FIGURE or RESULTS section (x,x)

Activation function

During this study different activation functions, such as ReLu, Softmax, Sigmoid were tested. Results were compared using grid-search and ReLu was picked as activation function for the models. Activation function was implemented in convolutional and fully-connected layers. Sigmoid activator was used for final output layer, because (NEED TO EXPLAIN).

Dropout

Dropout is implemented in the models due to the benefits described in section 2.6.3. Dropout of three different values were tested on grid-search and 0.5 were picked as it scored the highest. This regulator is only used within the hidden fully-connected layers, where it is defined to randomly drop 50 % of the nodes. Dropout is shown in figure 3.17 and marked as red crosses.

Optimizer

Adam, RMSprop, Adagrad and SGD, presented in section 2.6.2, were tested and compared. SGD scored the highest and was implemented while compiling the model.

Learning rate

In order to determine the most optimal *learning_rate* for the Adam and SGD optimizers several different values were tested. Default value 0,001 performed higher than values of 0,01 and 0,0001. To fit the model this parameter was set to 0,001 meaning that the convergence of gradient descent is reached slower but with more accurate *minima*.

Batch size and number of epochs

Three different values of batch size and number of epochs were chosen based on computing power. Grid search on batch size were tested on 5, 10, 15 values, while number of epochs were 15, 25, 35. On every data representations these hyperparameters varied. Batch size and number of epochs were used during the fit process.

Weight and biases initializing

Weights and biases initialization was performed, since it affects the performance of the model. A grid search of weight initializer using *uniform*, *lecun_uniform*, *normal*, *glorot_normal* and *glorot_uniform* were tested and results revealed that *glorot_normal* performed the best by 3,2 % compared to the second best. Results can be seen in ???. Weight initializer was used in convolutional and fully-connected layers. Biases were initialized to zeros values

Number of neurons

Number of neurons were chosen based on the best performance comparing 16, 32, and 64 neurons per hidden layer. 32 in convolutionals and 64 neurons in fully-connected layers scored the highest in terms of accuracy. These values were used in in the model within all data representations.

3.5.1 Training and validation

The training was performed using mini-batches of size 10 with SGD optimizer with default learning rate, explained in section 2.6.2. Data was split into train(75%), validation (15%), and test (10%) sets, and the models were regularized using dropout (50%). Number of neurons in hidden layers were decided to be used as 32 for convolutional and 64 for fully-connected layers. All networks were using ReLu activation function, except output layers, which were using Sigmoid activator instead, described in Section The model weight parameters were initialized at random values using *glorot_normal*, biases parameters were initialized to zero values.

Stratified m-folds cross-validation

As a result of binary output, 10-fold cross-validation method were used for training the data and evaluating the performance of the models. Folds were separated 10 times randomly in relation to the distribution of the classes. Stratified m-folding prevents the fold to contain the data of one class. Accuracy, sensitivity and specificity were calculated after each fold and the mean values of these parameters were generated at the end of the training. Results are given in section

Error and accuracy graphs)

During each training and validating epoch, the error and accuracy is calculated and presented as graph at the end. According the error graph, the overfitting or underfitting could be determined. Model could be optimized based on these results with technique described in section 2.6.2.

3.5.2 Testing

The testing or prediction refers the generalization performance from the given classification question and is the main job for the neural network. Separated 10% dataset were used for testing how accurate the training model can perform using unknown data. By feeding new data into the network, testing process begins. At each time-step classifier predicts the probability for every possible class, and selects with the highest probability as it prediction, giving it as a percentage result. Predictions were presented as a confusion matrix in section results providing true positive and negative together with false positive and negative results.

Chapter 4

Description of models

4.1 Data handling and design choices of the models

This chapter presents how the pre-processed data is loaded and prepared for training of the models in Python, as well as the different architectures of the models, and features related to their implementation. Three separate models have been made fitting to each of the data representations described in section 3.3.

4.1.1 Data handling in python

The pre-processed data is loaded from a *.mat* file into python. Data within the file is pre-shuffled and split into a training and test subsets, as described in section 3.3. In Python the training subset is further divided into a training and validation set. The training set is used to find the "optimal" weights with the back-propagation algorithm [33]. In this project, training set makes the biggest percentage amount of the data up to 75%. The validation set is used for estimating the generalization error of the model, and to examine how adjustments to hyperparameters effect the model performance [34]. Hyperparameters often define many different values that can be adjusted, to control the behavior of the algorithm, to which some parameters may affect the runtime and computational cost when training the model [31]. The validation set makes up 10% of the available data for the different models. The test subset will only be used to evaluate the generalization of the model, and therefore will not contain data that has been used during the training [34]. By keeping the test subset separate it will act as new unseen data, for the model. It contains 15% of the data.

For the binary, and combined pain map representation, the images were reshaped from a row vector, back into a matrix to retain their 2D structure.

Furthermore the classification labels were one-hot encoded, so they were compatible to the number of outputs in the models. This is a result of the loss function *categorical_crossentropy* used for the models, that tries to reduce loss between the categories. This demands that the class label has the same dimension as the number of outputs in the model [43].

4.1.2 Optimization techniques

To try to reduce overfitting and improve generalization of the neural network different techniques are applied network models.

One method used to automatically chose hyperparameters is used, and is known as grid search [31]. By using this method a listing of different hyperparameters were tested on a model, that allowed for choosing the parameter that give the highest improvements in performance in relation to the validation loss and accuracy.

Grid search is mainly practical when there are only a few hyperparameters that needs to be tested, because of the exponentially computational cost follows [31].

Optimizer used for the models

The Adam has been implemented in the three models, based on the findings of comparing it to the performance results with stochastic gradient descent optimizer. Adams optimizers showed a higher accuracy and significant higher training speed. Comparison results can be seen in ??

Kernel-initializer

Kernel-initialization was performed, since it affects the performance of the model. A grid search of kernel-initializer using *uniform*, *lecun_uniform*, *normal*, *glorot_normal* and *glorot_uniform* were tested and results revealed that *glorot_normal* performed the best by 3,2 % compared to the second best. Results can be seen in ??

Learning rate

In order to determine the most optimal *learning_rate* for the adam optimizer a several different values were tested. To fit the model this parameter was set to *0,0001* (lower than default value) meaning that the convergence of gradient descent is reached slower but with more accurate *minima*. It showed the

Batch training

Initial experimentation of the batch size showed little flexibility, because of the available computation power for training of the model. The batch size could not exceed a value of four, by reason of the relative high image resolution (912 x 2315). As a result of this it was chosen reduce the resolution (233 x 251). Resizing of the images, are described in 3.3.

Furthermore, the grid search of batch parameters were performed with values of 5, 10, 15, 20, and the highest accuracy with the value 5 were chosen

Cross-validation

Because the amount of data for this project is limited it is chosen to implement m-fold cross validation, where the training data is divided into m number of subsets. Each of the subsets can function a either a validation set or as a part of a training set e.g. if a classifier is trained m times, then each time a different subset will be used as a validation set, and the rest is used for training. [34] Because of the property of cross validation, it can be used as a way of investigate a general accuracy since all data is included during training, but may not be beneficial for every kind of problem. [34]

Dropout

Dropout is implemented in the models due to the benefits described in section 2.6.3. The dropout is only used within the hidden fully-connected layers, where it's defined to randomly drop 50 % of the nodes.

Activation function

ReLU activation function is used as controlling unit for nodes activation. It defines when the node is considered as active. In the model ReLU is implemented in convolutional and fully-connected layers.

Sigmoid is implemented only in the output layer, because it

It has something to with the fact that the output layer performs the classification and needs sigmoid is more sutible for this task then ReLu...

4.1.3 Training of the networks

Supervised learning is used for training in all the models. The generic input for all of the model is gender, along with the different image representation, which are presented in /refto data. These inputs trained and then compared against their respective category label. WE CAN MENTION HERE THE TYPES OF DATA AND TYPES OF MODELS IF THEY ARE NOT PRESENTED LATER

Temp-placeholder: The process of making the neural network model has been a trial and error process, because there is not an actual “cookbook” for developing NN (This statement is from a not vaild source, but so far it's the only one that i have found.) [31]

4.2 Vector image model

The architecture for this model only contains fully connected layer, since the data representation only contains 21 element vector that reflects the active pain regions and gender as described in ???. It is evaluated that there would not be any gains in making the model more complex e.g. adding of convolution, based on the information available from vector, since the level of detail in relation to morphology is very simple. The architecture of the model is illustrated in Figure 4.1.

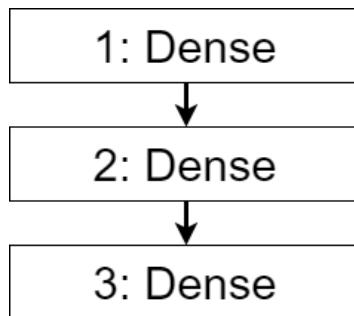


Figure 4.1: Arcitechture of the neural network using knee region representation.

The model consists of three layers, where the input and output layer..(WE ADD MORE)

4.3 Raw binary representation model

The architecture of this model is based on the typical structure of a convolutional network, where the first layers alternate between convolutional layers and max pooling layers [25]. This defines the first part of the model. The following layers consists of three fully connected layers, and output layer, and defined the second part of the model. An overview of the architecture

is shown in Figure 4.2. Convolution layer are implemented for this pain map representation, because of their ability to extract morphology features from images, as written in ??

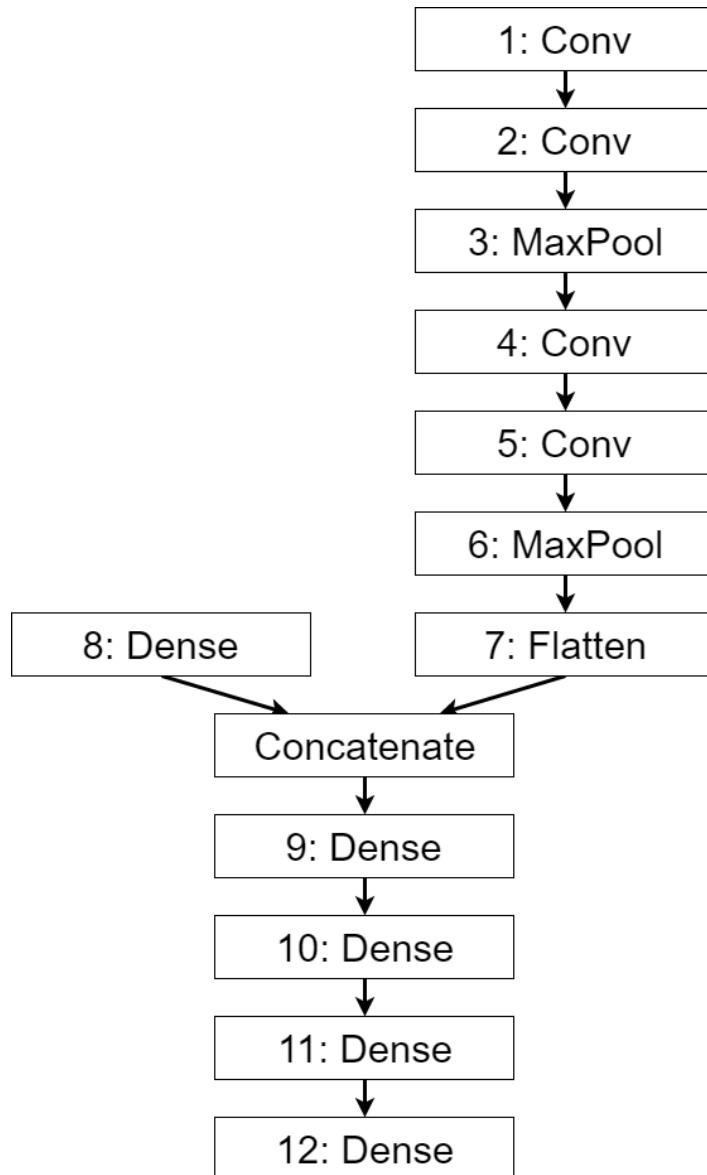


Figure 4.2: Architecture for the neural network model using binary pain map representation.
REMEMBER TO INCLUDE THE INPUT IN THE FIGURE

Rather than feeding both pain map representation and gender into the same input layer, they are separated and used as input at two different locations. The binary pain map representation works as input in first part of the model, where the input layer is a convolution layer. This layer is setup to receive a input shape to that of the dimension of the pain map, that is defined during re-scaleing of the pain maps in ?. Gender works as secondary input in the second section of the model, along with the pain maps features extracted through the convolution layers. Before the pain maps features reach the fully connected part of the network it is flattened from a matrix to a single row in order to merge the features with gender. The merged data passes the fully connected layers and reaches the output layer where it is given a percentage value according to which class it fits the most. The second part of the model

resembles the simple representation model, described in 4.2

THIS NEEDS TO BE REWRITTEN: The reason for separating gender and binary images is given as separate inputs is because of that there is no benefit in feeding gender through several convolutional layers, since these layer are use for looking at the shapes of the pain. The reason for using gender as input this far into the model, is a result of the way that convolution works

4.4 Combined representation model

The architecture of this model is nearly identical to that of the binary representation model as described in 4.3. The main difference can only be seen in the input layer for the pain map representation, where the input shape is altered to contain 20 layers per pain maps instead of one. This is the result of the one hot encoding done to the images as described in subsection 3.3.3.

Chapter 5

Results

Bibliography

- [1] Liam R. MacLachlan, Natalie J. Collins, and Et.al. The psychological features of patellofemoral pain: a systematic review. 2017. doi: 10.1136/bjsports-2016-096705.
- [2] T.O. Smith, B.T. Drew, and Et.al. Knee orthoses for treating patellofemoral pain syndrome (review). 2015. doi: 10.1002/14651858.CD010513.pub2.
- [3] M. S. Rathleff, B. Vicenzino, and Et.al. Patellofemoral Pain in Adolescents and Adulthood: same same, but different? 2015. doi: 10.1007/s40279-015-0364-1.
- [4] Wolf Petersen, Andree Ellermann, and Et.al. Patellofemoral Pain Syndrome. *Clinical Orthopaedics and Related Research*, 2013. doi: 10.1097/01.blo.0000229284.45485.6c.
- [5] Erik Witvrouw, Michael J. Callaghan, and Et.al. Patellofemoral Pain: consensus statement from the 3rd International Patellofemoral Pain Research Retreat held in Vancouver, September 2013. 2014. doi: 10.1136/bjsports-2014-093450.
- [6] Kay M. Crossley, Michael J. Callaghan, and Et.al. Patellofemoral pain. 2016. doi: 10.1136/bjsports-2015-h3939rep.
- [7] Scott F Dye. Patellofemoral Pain Current Concepts: An Overview. *Sports Medicine and Arthroscopy Review*, 2001.
- [8] Shellie A. Boudreau and Susanne et.al Badsberg. Digital pain drawings: Assessing Touch-Screen Technology and 3D Body Schemas. 2016. doi: 10.1097/AJP.0000000000000230.
- [9] Shellie A. Boudreau, E. N. Kamavuako, and Et.al. Distribution and symmetrical patellofemoral pain patterns as revealed by high-resolution 3D body mapping: a cross-sectional study. 2017. doi: 10.1186/s12891-017-1521-5.
- [10] Frederic H. et. al. Martini. *Anatomy & Physiology*. 2012.
- [11] IASP. IASP Taxonomy, 2012.
- [12] R. F Schmidt. Nociception and Pain. In *Human Physiology*. Springer Berlin Heidelberg, 1989. ISBN 978-3-642-73831-9. doi: 10.1007/978-3-642-73831-9_10.
- [13] R. F Schmidt. Nociception and Pain. In *Fundamentals of Sensory Physiology*. Springer Science & Business Media, 2 edition, 2013. ISBN 9783662011287.
- [14] Emma Briggs. Understanding the experience and physiology of pain. 2010.
- [15] Kay M. Crossley, Michael J. Callaghan, and Et.al. Patellofemoral pain. 2015. doi: 10.1136/bmj.h3939.
- [16] Katherine Stabenow Dahab. Q angle. *Encyclopedia of Sports Medicine*, 2011. doi: 10.4135/9781412961165.n415.

- [17] Jarred Younger and Sean Mackey. Pain outcomes: A brief review of instruments and techniques. 2009.
- [18] Kanishka M. Ghosh and David J. Deehan. Soft tissue knee injuries. In Mark Wilkinson, editor, *Orthopaedic surgery: lower limb*. Elsevier, 2010.
- [19] Maria Ferreira Valente, José Ribeiro Pais, and Et. Al. Validity of four pain intensity rating scales. 2011. doi: 10.1016/j.pain.2011.07.005.
- [20] Mathias Haefeli and Achim Elfering. Pain assessment. 2005. doi: 10.1007/s00586-005-1044-x.
- [21] Ewa M. Roos and L. Stefan Lohmander. The knee injury and osteoarthritis outcome score (KOOS): from joint injury to osteoarthritis. *Bio Med Central*, 2003.
- [22] Marie Grunnesjö. The course of pain drawings during a 10-week treatment period in patients with acute and sub-acute low back pain. 2006. doi: 10.1186/1471-2474-7-65.
- [23] Geoffrey D. Schott. The cartography of pain: The evolving contribution of pain maps. 2010. doi: 10.1016/j.ejpain.2009.12.005.
- [24] D. W. Elson, S. Jones, and Et.al. The photographic knee pain map: Locating knee pain with an instrument developed for diagnostic, communication and research purposes. 2010. doi: 10.1016/j.knee.2010.08.012.
- [25] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep Learning. *Nature Insight Review*, pages 436–444, 2015. doi: 10.1038/nature14539. URL <https://www.nature.com/nature/journal/v521/n7553/pdf/nature14539.pdf>.
- [26] Mads Nielsen. *Den digitale revolution – fortællinger fra datalogiens verden*. Datalogisk Institut, 2010. ISBN 978-87-981270-5-5.
- [27] M. I. Jordan and T. M. Mitchell. Machine learning: Trends, perspectives, and prospects. *Science*, 349(6245):255–260, 2015. ISSN 0036-8075. doi: 10.1126/science.aaa8415. URL <http://www.sciencemag.org/cgi/doi/10.1126/science.aaa8415>.
- [28] Jürgen Schmidhuber. Deep Learning in Neural Networks: An Overview. *Neural Networks*, 61:85–117, 2015. ISSN 18792782. doi: 10.1016/j.neunet.2014.09.003. URL <http://www.sciencedirect.com/science/article/pii/S0893608014002135>.
- [29] Jacopo Acquarelli, Twan van Laarhoven, Jan Gerretzen, Thanh N. Tran, Lutgarde M.C. Buydens, and Elena Marchiori. Convolutional neural networks for vibrational spectroscopic data analysis. *Analytica Chimica Acta*, 954:22–31, 2017. ISSN 18734324. doi: 10.1016/j.aca.2016.12.010. URL <http://linkinghub.elsevier.com/retrieve/pii/S0003267016314842>.
- [30] Alaa Ali Hameed, Bekir Karlik, and Mohammad Shukri Salman. Back-propagation algorithm with variable adaptive momentum. *Knowledge-Based Systems*, 114:79–87, 2016. ISSN 09507051. doi: 10.1016/j.knosys.2016.10.001. URL <http://www.sciencedirect.com/science/article/pii/S0950705116303811?via%23Dihub>.

- [31] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. URL <http://www.deeplearningbook.org>.
- [32] Mohri Mehryar, Rostamizadeh Afshin, and Talwalkar Ameet. *Foundations of Machine Learning*. 2012. ISBN 9780262018258. URL [#](https://ebookcentral.proquest.com/lib/aalborguniv-ebooks/reader.action?docID=3339482&ppg=17).
- [33] Yoshua Bengio. *Neural Networks: Tricks of the Trade: Second Edition*. 2012. ISBN 978-3-642-35289-8. doi: 10.1007/978-3-642-35289-8_26. URL http://dx.doi.org/10.1007/978-3-642-35289-8{_}26.
- [34] Richard Duda, Peter Hart, and David Stork. *Pattern Classification*. Second edi edition, 2000. ISBN 9780471056690.
- [35] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2323, 1998. ISSN 00189219. doi: 10.1109/5.726791. URL <http://yann.lecun.com/exdb/publis/pdf/lecun-01a.pdf>.
- [36] Sebastian Ruder. An overview of gradient descent optimization algorithms. 2016. ISSN 0006341X. doi: 10.1111/j.0006-341X.1999.00591.x. URL <http://arxiv.org/abs/1609.04747>.
- [37] Sebastian Raschka. Single-Layer Neural Networks and Gradient Descent. 2016. URL http://sebastianraschka.com/Articles/2015{_}singlelayer{_}neurons.html.
- [38] Sixin Zhang, Anna Choromanska, and Yann LeCun. Deep learning with Elastic Averaging SGD. 2014. ISSN 10495258. URL <http://arxiv.org/abs/1412.6651>.
- [39] Ning Qian. On the Momentum Term in Gradient Descent Learning Algorithms The Momentum Term in Gradient Descent. *Neural Networks: The Official Journal of the International Neural Network Society*, 5213(12(1)):145–151, 1999. URL <https://www.sciencedirect.com/science/article/pii/S0893608098001166>#aep-bibliography-id16.
- [40] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. *International Conference on Learning Representations*, 2015. ISSN 09252312. doi: <http://doi.acm.org.ezproxy.lib.ucf.edu/10.1145/1830483.1830503>. URL <http://arxiv.org/abs/1412.6980>.
- [41] M. Patacchiola and A. Cangelosi. Head pose estimation in the wild using Convolutional Neural Networks and adaptive gradient methods. *Pattern Recognition*, 71, 2017. ISSN 00313203. doi: 10.1016/j.patcog.2017.06.009. URL https://ac.els-cdn.com/S0031320317302327/1-s2.0-S0031320317302327-main.pdf?{_}tid=a09e6e8a-d356-11e7-9b28-00000aacb360&{&}acdnat=1511775647{_}f91c81864b46ddf0becfc12de21bce48.
- [42] Int8. Optimization techniques comparison in Julia: SGD, Momentum, Adagrad, Adadelta, Adam. *Www*, 2016. URL <http://int8.io/comparison-of-optimization-techniques-stochastic-gradient-descent-momentum-adagrad-and>

- [43] François Chollet. Keras Documentation, 2015. URL <https://keras.io>.
- [44] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014. ISSN 15337928. doi: 10.1214/12-AOS1000. URL <https://dl.acm.org/citation.cfm?id=2670313&CFID=818407627&CFTOKEN=74532044>.
- [45] Aglance Solutions. Visual insight for clinical reasoning – Navigate Pain, 2015. URL <http://www.navigatepain.com/>.
- [46] David Money Harris and Sarah L. Harris. Sequential Logic Design. In *Digital design and computer architecture*. Elsevier, 2012. ISBN 9780123978165.
- [47] Manohar Swamynathan. *Mastering Machine Learning with Python in Six Steps*. 2017. ISBN 978-1-4842-2865-4. doi: 10.1007/978-1-4842-2866-1. URL <http://link.springer.com/10.1007/978-1-4842-2866-1>.

Appendix A

Knee injury and Osteoarthritis Outcome Score (KOOS) [?]

KOOS KNEE SURVEY

Today's date: ____ / ____ / ____ Date of birth: ____ / ____ / ____

Name: _____

INSTRUCTIONS: This survey asks for your view about your knee. This information will help us keep track of how you feel about your knee and how well you are able to perform your usual activities.

Answer every question by ticking the appropriate box, only one box for each question. If you are unsure about how to answer a question, please give the best answer you can.

Symptoms

These questions should be answered thinking of your knee symptoms during the **last week**.

S1. Do you have swelling in your knee?

Never	Rarely	Sometimes	Often	Always
<input type="checkbox"/>				

S2. Do you feel grinding, hear clicking or any other type of noise when your knee moves?

Never	Rarely	Sometimes	Often	Always
<input type="checkbox"/>				

S3. Does your knee catch or hang up when moving?

Never	Rarely	Sometimes	Often	Always
<input type="checkbox"/>				

S4. Can you straighten your knee fully?

Always	Often	Sometimes	Rarely	Never
<input type="checkbox"/>				

S5. Can you bend your knee fully?

Always	Often	Sometimes	Rarely	Never
<input type="checkbox"/>				

Stiffness

The following questions concern the amount of joint stiffness you have experienced during the **last week** in your knee. Stiffness is a sensation of restriction or slowness in the ease with which you move your knee joint.

S6. How severe is your knee joint stiffness after first wakening in the morning?

None	Mild	Moderate	Severe	Extreme
<input type="checkbox"/>				

S7. How severe is your knee stiffness after sitting, lying or resting **later in the day**?

None	Mild	Moderate	Severe	Extreme
<input type="checkbox"/>				

Knee injury and Osteoarthritis Outcome Score (KOOS), English version LK1.0

Pain

P1. How often do you experience knee pain?

Never	Monthly	Weekly	Daily	Always
<input type="checkbox"/>				

What amount of knee pain have you experienced the **last week** during the following activities?

P2. Twisting/pivoting on your knee

None	Mild	Moderate	Severe	Extreme
<input type="checkbox"/>				

P3. Straightening knee fully

None	Mild	Moderate	Severe	Extreme
<input type="checkbox"/>				

P4. Bending knee fully

None	Mild	Moderate	Severe	Extreme
<input type="checkbox"/>				

P5. Walking on flat surface

None	Mild	Moderate	Severe	Extreme
<input type="checkbox"/>				

P6. Going up or down stairs

None	Mild	Moderate	Severe	Extreme
<input type="checkbox"/>				

P7. At night while in bed

None	Mild	Moderate	Severe	Extreme
<input type="checkbox"/>				

P8. Sitting or lying

None	Mild	Moderate	Severe	Extreme
<input type="checkbox"/>				

P9. Standing upright

None	Mild	Moderate	Severe	Extreme
<input type="checkbox"/>				

Function, daily living

The following questions concern your physical function. By this we mean your ability to move around and to look after yourself. For each of the following activities please indicate the degree of difficulty you have experienced in the **last week** due to your knee.

A1. Descending stairs

None	Mild	Moderate	Severe	Extreme
<input type="checkbox"/>				

A2. Ascending stairs

None	Mild	Moderate	Severe	Extreme
<input type="checkbox"/>				

Knee injury and Osteoarthritis Outcome Score (KOOS), English version LK1.0

For each of the following activities please indicate the degree of difficulty you have experienced in the **last week** due to your knee.

A3. Rising from sitting

None <input type="checkbox"/>	Mild <input type="checkbox"/>	Moderate <input type="checkbox"/>	Severe <input type="checkbox"/>	Extreme <input type="checkbox"/>
----------------------------------	----------------------------------	--------------------------------------	------------------------------------	-------------------------------------

A4. Standing

None <input type="checkbox"/>	Mild <input type="checkbox"/>	Moderate <input type="checkbox"/>	Severe <input type="checkbox"/>	Extreme <input type="checkbox"/>
----------------------------------	----------------------------------	--------------------------------------	------------------------------------	-------------------------------------

A5. Bending to floor/pick up an object

None <input type="checkbox"/>	Mild <input type="checkbox"/>	Moderate <input type="checkbox"/>	Severe <input type="checkbox"/>	Extreme <input type="checkbox"/>
----------------------------------	----------------------------------	--------------------------------------	------------------------------------	-------------------------------------

A6. Walking on flat surface

None <input type="checkbox"/>	Mild <input type="checkbox"/>	Moderate <input type="checkbox"/>	Severe <input type="checkbox"/>	Extreme <input type="checkbox"/>
----------------------------------	----------------------------------	--------------------------------------	------------------------------------	-------------------------------------

A7. Getting in/out of car

None <input type="checkbox"/>	Mild <input type="checkbox"/>	Moderate <input type="checkbox"/>	Severe <input type="checkbox"/>	Extreme <input type="checkbox"/>
----------------------------------	----------------------------------	--------------------------------------	------------------------------------	-------------------------------------

A8. Going shopping

None <input type="checkbox"/>	Mild <input type="checkbox"/>	Moderate <input type="checkbox"/>	Severe <input type="checkbox"/>	Extreme <input type="checkbox"/>
----------------------------------	----------------------------------	--------------------------------------	------------------------------------	-------------------------------------

A9. Putting on socks/stockings

None <input type="checkbox"/>	Mild <input type="checkbox"/>	Moderate <input type="checkbox"/>	Severe <input type="checkbox"/>	Extreme <input type="checkbox"/>
----------------------------------	----------------------------------	--------------------------------------	------------------------------------	-------------------------------------

A10. Rising from bed

None <input type="checkbox"/>	Mild <input type="checkbox"/>	Moderate <input type="checkbox"/>	Severe <input type="checkbox"/>	Extreme <input type="checkbox"/>
----------------------------------	----------------------------------	--------------------------------------	------------------------------------	-------------------------------------

A11. Taking off socks/stockings

None <input type="checkbox"/>	Mild <input type="checkbox"/>	Moderate <input type="checkbox"/>	Severe <input type="checkbox"/>	Extreme <input type="checkbox"/>
----------------------------------	----------------------------------	--------------------------------------	------------------------------------	-------------------------------------

A12. Lying in bed (turning over, maintaining knee position)

None <input type="checkbox"/>	Mild <input type="checkbox"/>	Moderate <input type="checkbox"/>	Severe <input type="checkbox"/>	Extreme <input type="checkbox"/>
----------------------------------	----------------------------------	--------------------------------------	------------------------------------	-------------------------------------

A13. Getting in/out of bath

None <input type="checkbox"/>	Mild <input type="checkbox"/>	Moderate <input type="checkbox"/>	Severe <input type="checkbox"/>	Extreme <input type="checkbox"/>
----------------------------------	----------------------------------	--------------------------------------	------------------------------------	-------------------------------------

A14. Sitting

None <input type="checkbox"/>	Mild <input type="checkbox"/>	Moderate <input type="checkbox"/>	Severe <input type="checkbox"/>	Extreme <input type="checkbox"/>
----------------------------------	----------------------------------	--------------------------------------	------------------------------------	-------------------------------------

A15. Getting on/off toilet

None <input type="checkbox"/>	Mild <input type="checkbox"/>	Moderate <input type="checkbox"/>	Severe <input type="checkbox"/>	Extreme <input type="checkbox"/>
----------------------------------	----------------------------------	--------------------------------------	------------------------------------	-------------------------------------

Knee injury and Osteoarthritis Outcome Score (KOOS), English version LK1.0

For each of the following activities please indicate the degree of difficulty you have experienced in the **last week** due to your knee.

A16. Heavy domestic duties (moving heavy boxes, scrubbing floors, etc)

None	Mild	Moderate	Severe	Extreme
<input type="checkbox"/>				

A17. Light domestic duties (cooking, dusting, etc)

None	Mild	Moderate	Severe	Extreme
<input type="checkbox"/>				

Function, sports and recreational activities

The following questions concern your physical function when being active on a higher level. The questions should be answered thinking of what degree of difficulty you have experienced during the **last week** due to your knee.

SP1. Squatting

None	Mild	Moderate	Severe	Extreme
<input type="checkbox"/>				

SP2. Running

None	Mild	Moderate	Severe	Extreme
<input type="checkbox"/>				

SP3. Jumping

None	Mild	Moderate	Severe	Extreme
<input type="checkbox"/>				

SP4. Twisting/pivoting on your injured knee

None	Mild	Moderate	Severe	Extreme
<input type="checkbox"/>				

SP5. Kneeling

None	Mild	Moderate	Severe	Extreme
<input type="checkbox"/>				

Quality of Life

Q1. How often are you aware of your knee problem?

Never	Monthly	Weekly	Daily	Constantly
<input type="checkbox"/>				

Q2. Have you modified your life style to avoid potentially damaging activities to your knee?

Not at all	Mildly	Moderately	Severely	Totally
<input type="checkbox"/>				

Q3. How much are you troubled with lack of confidence in your knee?

Not at all	Mildly	Moderately	Severely	Extremely
<input type="checkbox"/>				

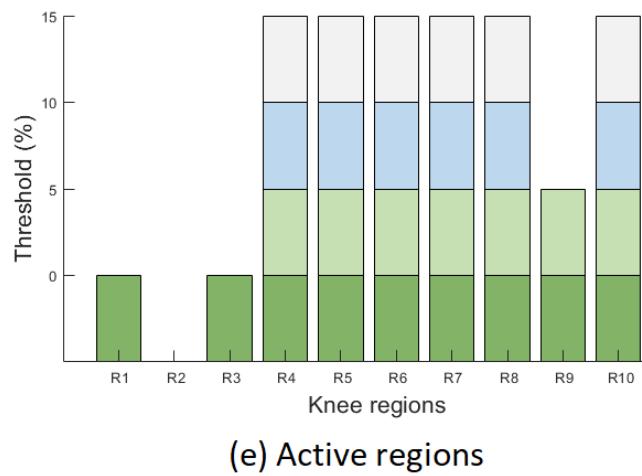
Q4. In general, how much difficulty do you have with your knee?

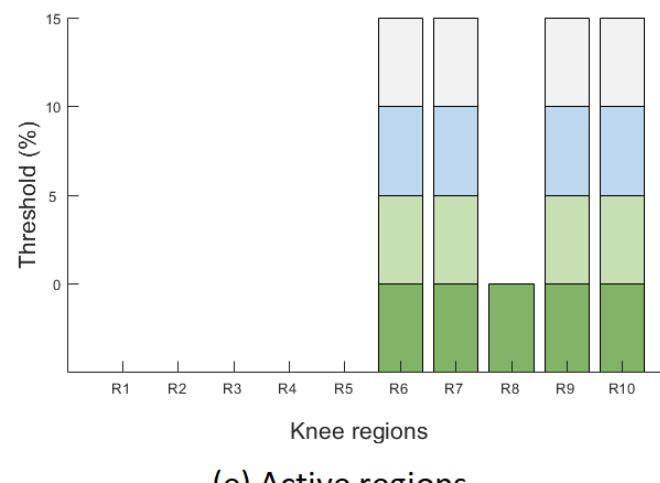
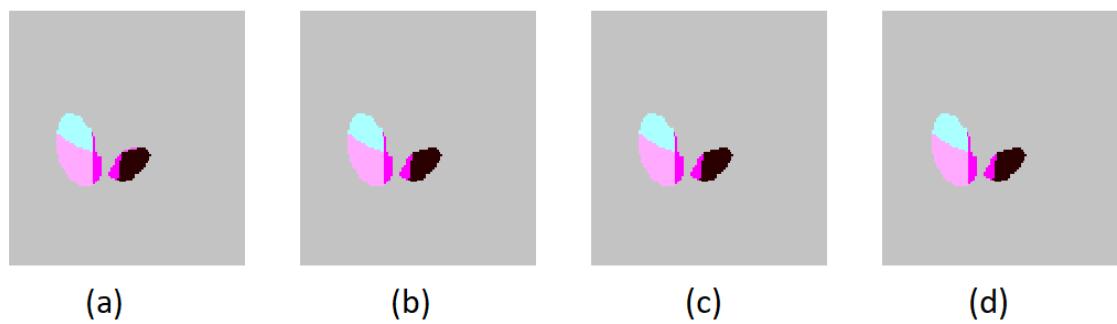
None	Mild	Moderate	Severe	Extreme
<input type="checkbox"/>				

Thank you very much for completing all the questions in this questionnaire.

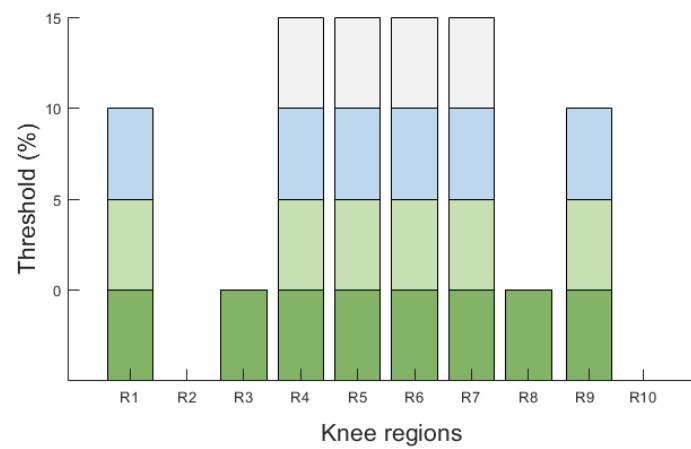
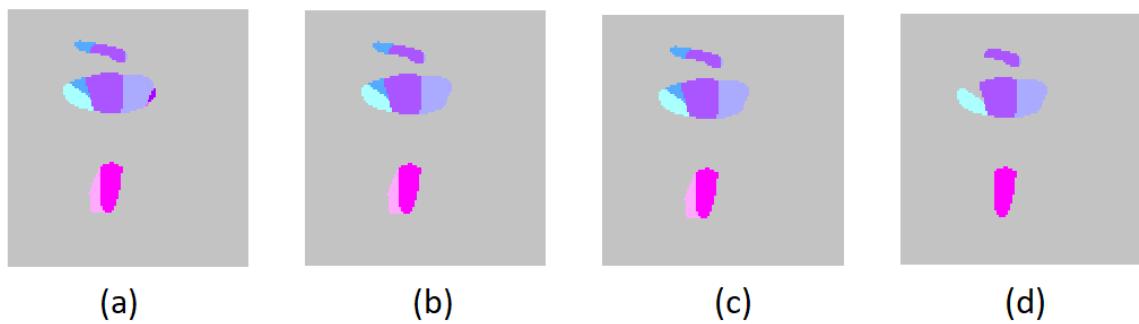
Appendix B

Threshold analysis





(e) Active regions



(e) Active regions

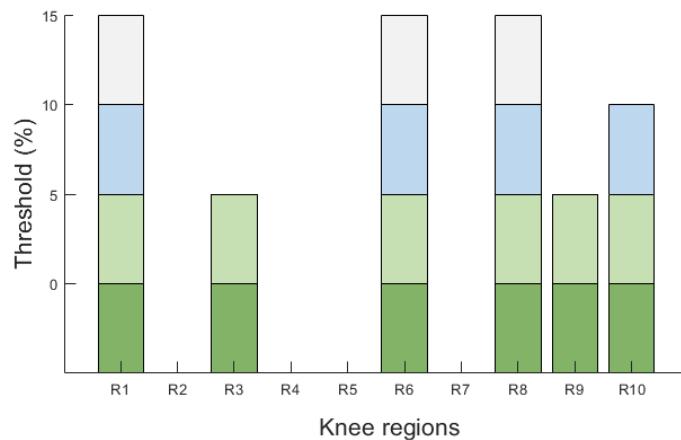


(a)

(b)

(c)

(d)



(e) Active regions