
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
1.1 Primary aim	2
1.2 Secondary aim	2
Chapter 2 Background	3
2.1 Anatomy of the Knee	3
2.2 Pain	5
2.3 Identify and interpret pain	6
2.3.1 Identify cause of pain	6
2.3.2 Pain interpretation	7
2.4 Knee regions	8
2.5 Machine learning	10
2.5.1 Deep Learning	10
2.5.2 Back-propagation	11
2.5.3 Core NN layers	14
Chapter 3 Methodology	16
3.1 Data	16
3.1.1 Software application: Navigate Pain	17
3.1.2 Data representations	17
3.2 Pre-analysis	18
3.2.1 Classification of data	18
3.2.2 Distribution and threshold	20
3.2.3 Simple regression models	22
3.3 Pre-processing	23
3.3.1 Morphology-representation	23
3.3.2 Regions-representation	24
3.3.3 Superimposed-representation	25
3.4 Programs	25
Chapter 4 Description of models	26
4.1 General for the three models	26
4.1.1 Data-handling in python	26
4.1.2 Applied optimization techniques	26
4.1.3 Training of the networks	27
4.2 Vector image model	28
4.3 Raw binary representation model	28
4.4 Combined representation model	30
Chapter 5 Results	31

Bibliography	32
Appendix A Appendix	35
A.1 Appendix I	35

Chapter 1

Introduction

Patellofemoral pain syndrome (PFPS) is a painful musculoskeletal condition that is presented as pain behind or around patella [1, 2]. PFPS affects 6-7 % of adolescents, of whom two thirds are highly physically active [?]. Additionally the prevalence is more than twice as high for females than males.[3?]. PFPS may be present over longer periods of time where a high number of individuals experience a recurrent or chronic pain [?] and may also lead to osteoarthritis [3, 4].

Patellofemoral pain (PFP) is often described as diffuse knee pain, that can be hard to explain and localize [?]. Despite the fact that patients feel pain in the knee, there is not any structural changes in the knee such as significant chondral damage or increased Q-angle. There is no definitive clinical test to diagnose PFPS and it is thereby often diagnosed on exclusion criteria [3] to which PFPS is also described as an orthopaedic enigma, and is one of the most challenging pathologies to manage [?]. 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 [5]. A study shows that through the use of pain maps it is possible to find a correlation between the symptom duration and the size and morphology of pain area [6]. Another method to measure pain is by using visual analog scale (VAS), that scores pain between no-pain to the worst pain imaginable [7]. However it is a known problem that chronic pain is considered as a multidimensional pain, because the perceived pain of an individual is influenced by biomedical, psychosocial and behavioral factors [?]

Since PFP is associated with a lack of knowledge, and it has been shown that there is a correlation between pain maps and duration and pain intensity, it is interesting to investigate if pain maps can be used to classify and predict PFP related information.

A method that has not been found used in this context before is a deep learning. The deep learning method is chosen for this study because it is a state of the art method, that has shown greater performance in specific computation fields, compared to other machine learning methods [8]. Furthermore the method is chosen because of its ability to find a non-linear connection between input and output data [8], which is found relevant for this study mainly based on the fact that PFP is subjective and may be affected by the multidimensionality of chronic pain.

The goals of this project is to explore how accurate a deep learning model can classify symptom duration and pain intensity associated to PFP pain maps using a limited dataset. Because the prevalence is more than twice as high for females than males, the gender is included as an input parameter for the model. Furthermore morphology of the pain maps is considered to be relevant, based on the indication that morphology and size of pain area increase with prolonged symptom duration. To investigate the influence of morphology three types of pain map representations are created: a binary representation, a simplified representation based

on knee regions and a combined representation that contains binary representations divided into knee regions.

1.1 Primary aim

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

It is hypothesized that classification performance of the deep learning model is higher when using pain maps and gender to predict symptom duration than pain intensity.

1.2 Secondary aim

The further aim of this project is to investigate if multiple pain map representations affect the deep learning model classification performance.

It is hypothesized that different data representations of pain maps affect the performance accuracy of a deep learning model as related to the classification of symptom duration and pain intensity.

Chapter 2

Background

This chapter encompasses 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 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. The gliding joint is formed between the patella and femur. The structure of the knee is illustrated in figure 2.1.[9]

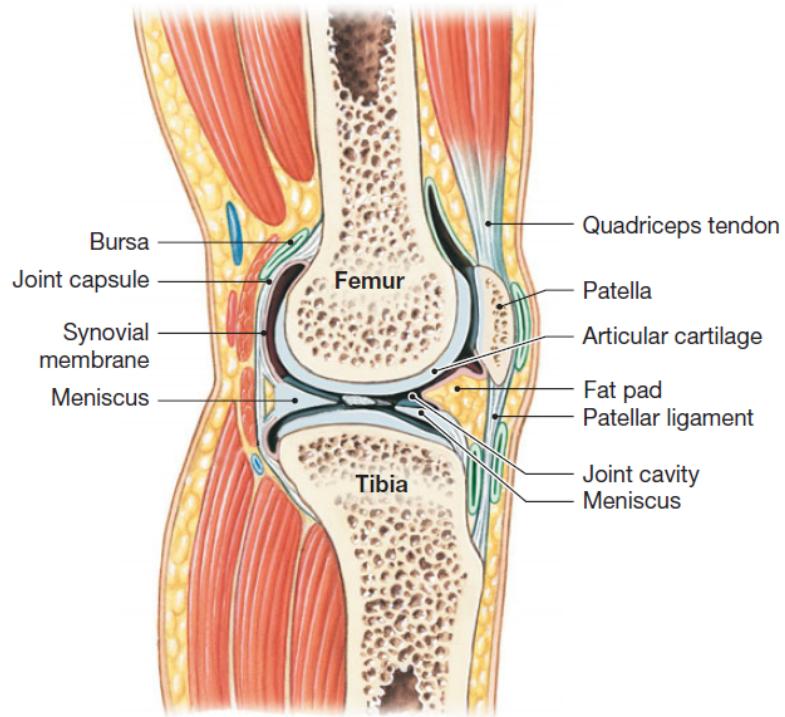


Figure 2.1: The figure illustrates the anatomy of the knee. Edited from [9].

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.[9]

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.[9]

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 is shown in figure 2.2.[9]

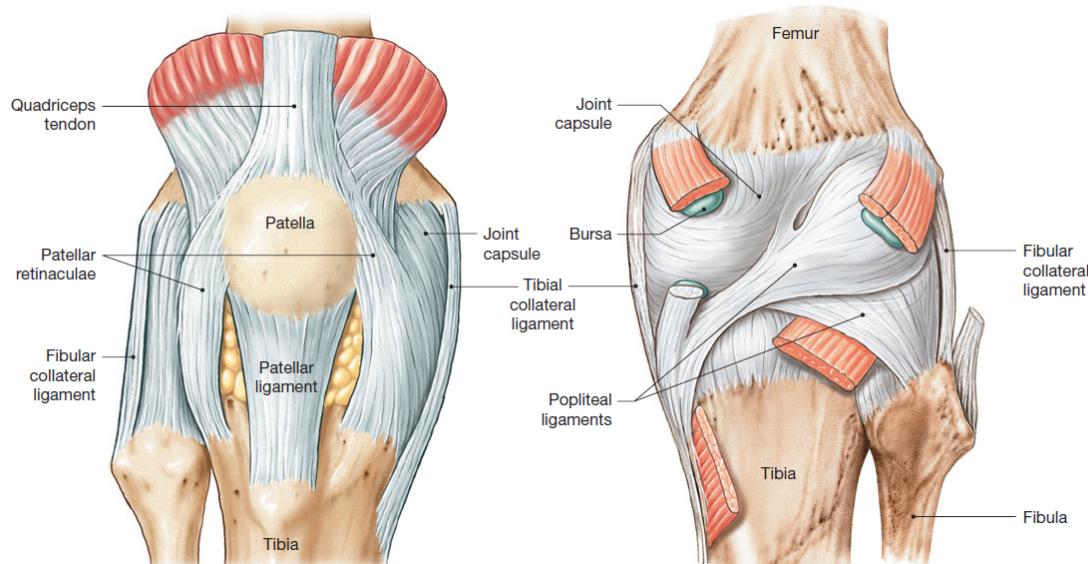


Figure 2.2: The figure illustrates the anatomy of the knee with focus on the ligaments. Edited from [9].

The ligaments patellar retinaculæ 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. They reduce the movement, anterior and posterior.[9]

As previously mentioned the gliding joint is formed between the patella and femur, so that during knee movement patella is gliding up and down at the femoral condyle. A condition

associated with incorrect movement of the patella, is patellofemoral pain syndrome (PFPS), that occurs when the patella moves outside of its ordinary track, which for instance can be movement in lateral direction.[9]

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” [10, 11].

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 [12]. Neuropathic pain occurs central from the nervous system. This pain can be caused by illness or physical damage.

Furthermore, pain can be divided into three categories: acute pain (less than three months), persistent or chronic pain and cancer pain.[13] Additionally, the sense of pain can be divided into some qualities, which is shown in figure 2.3.

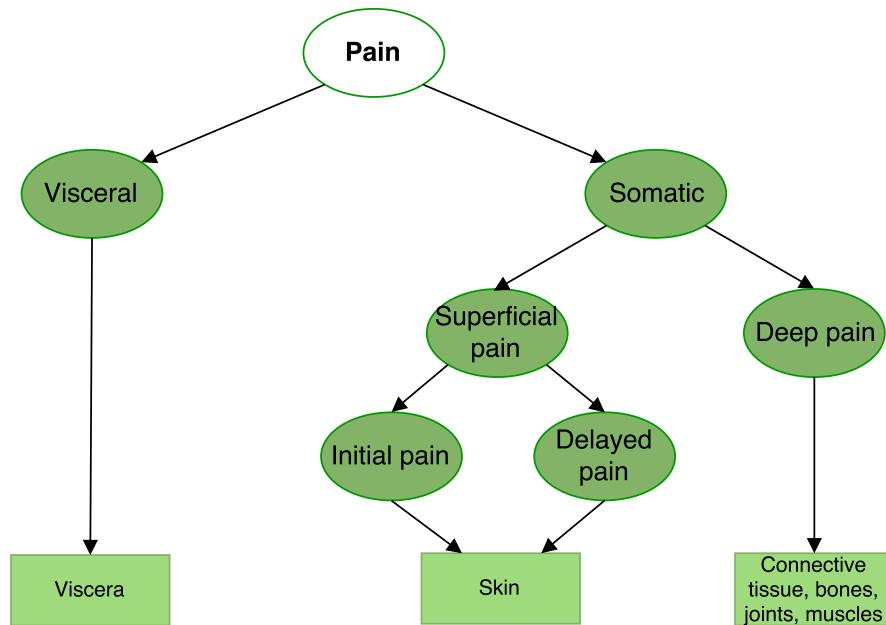


Figure 2.3: Model of pain qualities. Ovals with green background represent qualities of pain. The rectangles show where the pain occurred. Edited from [12].

Pain can be divided into two qualities; visceral and somatic pain. Examples of visceral pain include pain associated with gallstones 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 this pain is characterised as sharp and localizable. The delayed pain, also known as the second pain, is

sensed as 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.[11, 12] The other 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.[11, 12]

Since the aetiology of PFPS still remains unclear [2], it is hard to place this type of pain in addition to nociceptive and neuropathic pain. But PFPS can be classified as deep pain and acute or chronic pain. Since the PFPS is often longer than six month it is described as a chronic deep pain.

2.3 Identify and interpret pain

There are many ways to identify and interpret pain. To identify pain and find some physical damage that causes the pain can objective methods be used. Subjective methods is used to interpret pain for collecting knowledge of the subjects pain intensity, behavior and how it is experienced.[14]

2.3.1 Identify cause of pain

A objective pain measurement is often used when a subject 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 test for meniscal tear.[15] Illustrations of the tests are shown in figure 2.4.

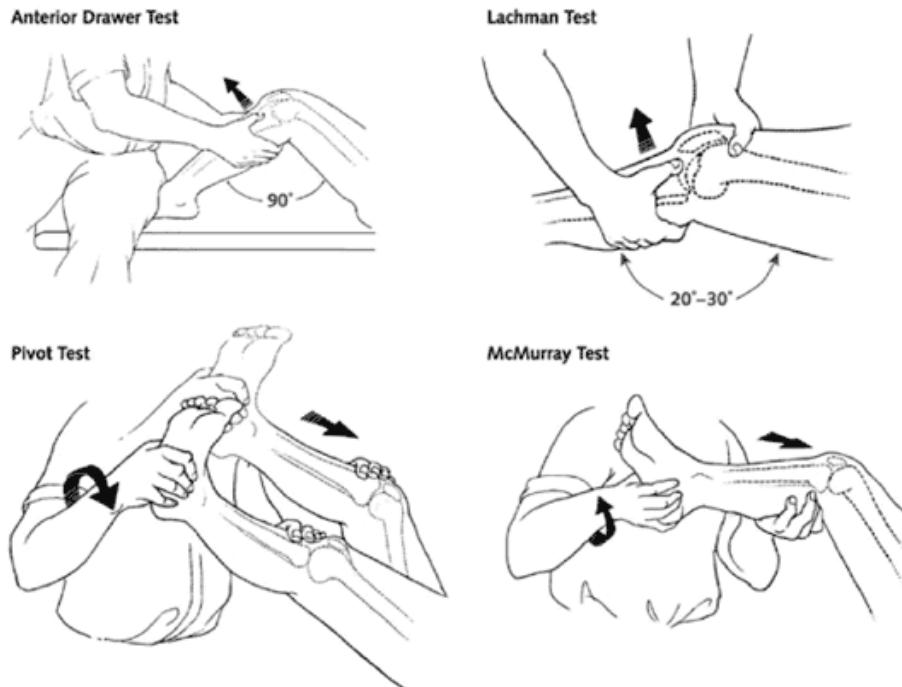


Figure 2.4: Clinical examination with provocative tests; Anterior Drawer Test, Lachman Test, Pivot Test and McMurray Test.[15]

In addition to clinical tests there is some paraclinical tests such as X-ray and MRI, but PFPS does not show any structural changes in the knee, like increased Q-angle or significant chondral damage [3], which makes it difficult for healthcare personnel to treat the subjects.

2.3.2 Pain interpretation

Pain is experienced and perceived subjectively [10, 14] and is dependent on personality and character [12], which is why it is important to measure the pain from the subject's perspective. One of the most commonly method used to measure pain intensity is Visual Analogue Scale (VAS) [16]. VAS is often used in clinical and research settings, where the subjects mark their pain on a scale from no-pain to the worst pain they can imagine.[7] A illustration of a VAS is shown in figure 2.5.



Figure 2.5: Visual Analogue Scale (VAS). Edited from [7].

Additionally to mark pain on a scala is questionnaires used to define subjects pain. An example on a questionnaire is Knee injury and Osteoarthritis Outcome Score (KOOS), which contains questions about symptoms, stiffness, pain, function daily living, function, sports and recreational activities and quality of life. When the subjects 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.[17] The questionnaire can be seen in Appendix A.1.

Since PFPS is describe as a diffuse pain, where subjects indicate their pain by 'placing both hands over their knees', is it hard for subject precise communicate their pain. Thereby is pain mapping a method for subjects to better indicate and communicate their pain.

Pain mapping

Pain mapping is a technique, that Harold Palmer introduced in 1949 [18], which is used to transfer a patient's perceived pain into an objective graph or map by drawing the pain area. Pain drawings can be made by the patients who draw their pain areas on a body outline. Pain drawing can also be made by observers who observe the patients and then draw from the signs the patients are showing. An example of a body outline is shown in figure 2.6. Sometimes a 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.[19]

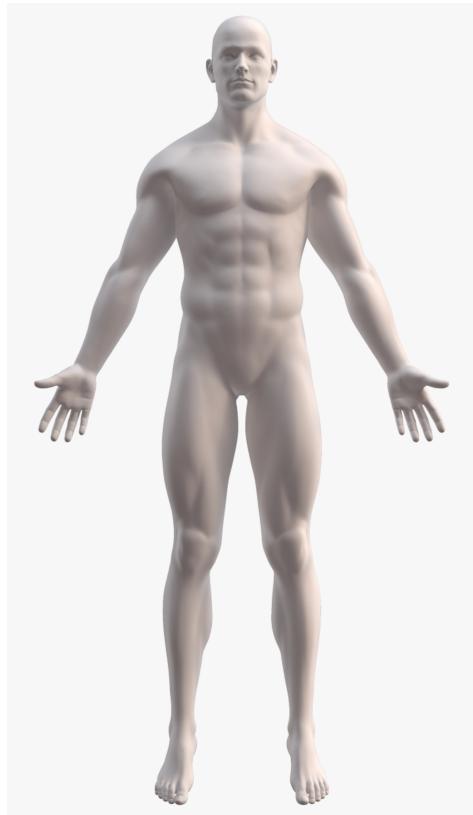


Figure 2.6: The figure illustrates an anterior body outline for pain drawing. The figure is a screenshot from the application Navigate Pain.

Pain mapping are commonly used in clinical practice [19], and can be useful for patients when they try to describe their pain. Pain maps may also be helpful in diagnosing patients and follow-ups during or after treatment to get an indicator of the patient's response to the treatment.[5] 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.[19]

2.4 Knee regions

Patients with PFPS often describe the knee pain as a diffuse pain, and when looking at pain drawing samples from multiple patients it is also evident that there is a high variability in the distribution of pain patterns across different areas of the knee.

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.

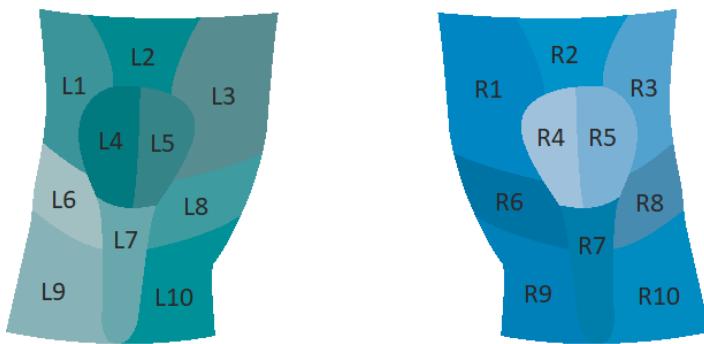


Figure 2.7: The regions of the left and right knees, where each knee is split into ten regions.

The divisions is inspired by Photographic Knee Pain Map (PKPM) which is designed to categories location of knee pain, diagnostic and research purposes. PKPM represent both knees that makes it possible to identify unilateral and bilateral pain.[?]

The regions are based on the anatomic structures according to the areas where subjects often indicate pain. There is 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 tendor is region 7 and the two last regions, 9 and 10, are tibia lateral and medial.[?]

2.5 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 [8]. Machine learning is a method that uses inductive inference in order to identify rules in a dataset from given input and output [20]. If the computer learns this feature, it can be used to make intelligent decisions and predict specific outcomes.[20] It 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. This is simply easier than trying to manually write a piece of code that anticipate different scenarios from different input types.[21]

2.5.1 Deep Learning

Deep learning is a branch of machine learning. The main difference between the use of machine learning and deep learning, is that machine learning is not suitable for handling raw data form. 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 for the machine learning system. Deep learning is based on different techniques that makes it able to handle that data in its raw form, mainly because of its structure.[8, 22] Because of this the system will automatically detect the 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 a input-layer and an output-layer, with one or more hidden layers in between [22]. The key aspect of these layers is that the features are not defined by programmers, but they are found and learned from raw data using a general-purpose learning procedure.[8] An example of a neural network structure can be seen in figure 2.8.

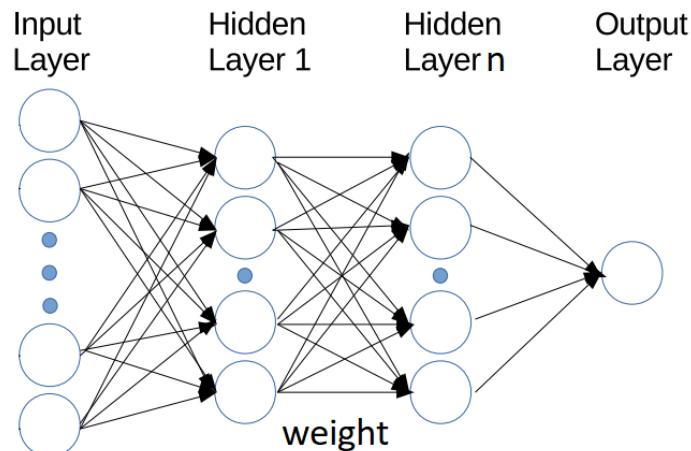


Figure 2.8: Example of the neural network with possible layers[23].

The different layers consist of a series of nodes, where each node is connected by weights to one or several other nodes from a different layer. 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.[22] An example of how the hidden layers affect an image can be explained as follows: 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.[8]

Learning scenarios

There are different approaches for training a neural network, where the three 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 backpropagation. [8] Overall supervised learning may be described as teaching the network how to associate a given input to a specific output [?], and is mostly associated with classification, regression, and ranking problems [24].

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 [24]. 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 [25].

2.5.2 Back-propagation

Backpropagation is a popular learning algorithm in neural networks, that is based on gradient descent, and used because of its simplicity and computationally efficiency. [26, 27] It's the (learning) process where the weights of a neural network are adjusted in order to reduce the error calculated between the output of the network and the expected output. This by definition makes backpropagation closely related to supervised learning, as written in ??, to which backpropagation is the most general method used.[27] 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. Based on an objective function a loss is calculated for every input that passes through the network, this is used by backpropagation to make the adjustments on the weight to reduce this loss. As training progresses the loss should decrease as a result of the weight adjustments, and improve the performance of the neural network. [8, 27, 25] This learning process continues until optimal weights with minimum error is reached.[28]

The basic idea behind it 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 each node (separate input) summed with additional coefficients called biases.[28?]

MAYBE TALK ABOUT ACTIVATION FUNCTIONS IN RELATION TO BACKPROPAGATION.

A PROBLEM OFTEN RELATED TO OPTIMIZATION IS THE LARGER THE NETWORK THE HARDER IT IS TO OPTIMIZE, BUT IN ANOTHER WAY IF THE

NETWORK IS SMALL AND SIMPLE THE EASIER IT IS TO OPTIMIZE.

Gradient Decent

Gradient descent is one of the most common technique for optimizing neural networks. It is a way to minimize the objective function (loss function) by updating the parameters in the opposite direction of the gradient of the objective function.[?] The principle of the gradient descent could be explained as a "ball climbing down a hill" until the (local) minimum is reached is it could be seen in Figure 2.9. 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.[?]

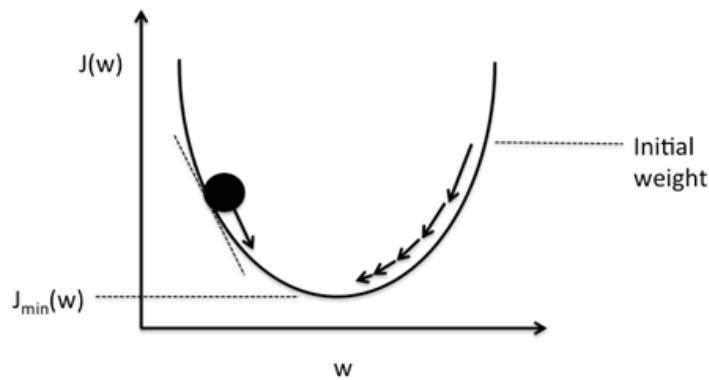


Figure 2.9: Schematic of 1-dimensional gradient descent working principle, where $J(w)$ – the loss function, $J_{\min}(w)$ – final approximation to the (local) minimum of $J(w)$, w - value of the parameter. The arrows indicate the step direction, i.e. the negative gradient.[?].

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 cost function. Depending on which of the gradient descent variant is used, the trade-off between the accuracy and the time 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. [quota] In cases like these, stochastic gradient descent is being used more commonly.

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. On the other hand, it complicated convergence to the exact minimum it keeps overshooting.[?]

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

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. Most popular optimizers are described below.

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.[?]

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.[?]

Adadelta is an updated version of Adagrad, but here the learning rate is monotonically decreasing. According to the source, with this optimizer, there is no need to tune the parameters of optimizer meaning that it can be applied in a variety of situations.[?]

Adam stands for Adaptive Moment Estimation. It is mostly 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 very fast with a decent learning speed.[?]

As the example, the article of Patacchiola (2016) shows the evaluation of the performance on different optimizers on ALFW image dataset containing 21977 male and female head pose images. As Figure 2.10 shows the Adam optimizer had the fastest convergence rate and it reached the lowest loss values.

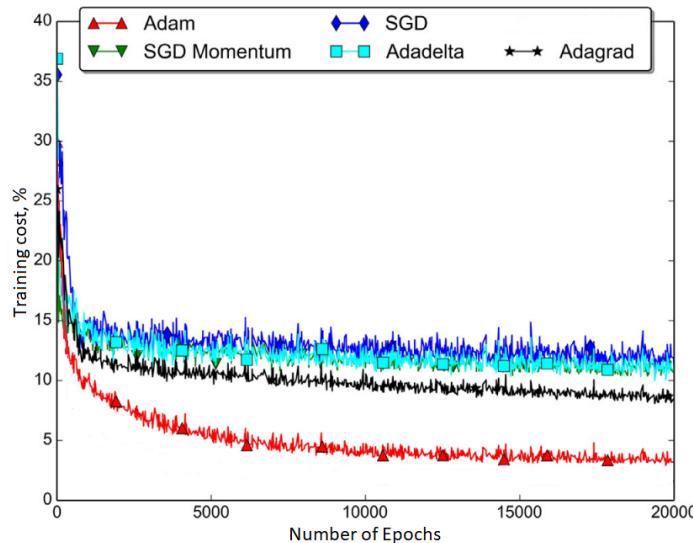


Figure 2.10: Comparison of the convergence speed between different optimizers used to train architecture on ALFW dataset. The loss values are the mean of the five attempts [?].

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 winning optimizer and it has to be chosen based on every problem.[?]

Learning curves

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 [27]. An illustration of how the error values are affected during training can be seen in Figure 2.11.

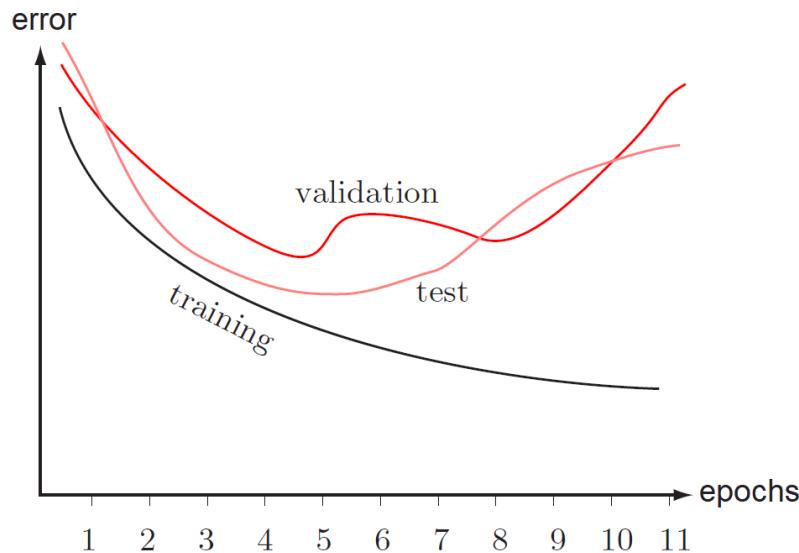


Figure 2.11: Illustration of how training (black), validation (red), and test (orange) error is affected by the increase in epochs. Edited from [27].

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. [27]

2.5.3 Core NN layers

Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are a type of special neural network for processing data with a grid-like topology [25]. The architecture of a typical CNN's are almost always combined with another layer type called pooling [25, 8].

Convolutional neural networks (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 model by extracting visual features from the pixel-level content.[23]“

The terminology regarding the output of a convolutional layer can be referred to as a feature map [25]

The way that convolutional networks work is by taking e.g. an image and scan it and split it up into a kind of feature map.

INSERT IMAGE HERE

Additional explanation of the example in start of deep learning (How it first detects edges and then shape and then objects.)

Pooling

As previously mentioned convolution are typically followed by pooling [8?]. Pooling can be used to reduce the size of the dataset, which in turn can 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 overall information, in which the output does not change. [25] 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 maxpooling that takes the maximum value within the window [25]. A pooling layer may be defined simply by its window size and a stride length, where stride length is the number of values the window jumps [?].

Chapter 3

Methodology

*** OPS! IKKE RIGTIGT! *** This chapter creates an understanding of the given data and the different programs respectively the program where pain maps are created and the program for development of the neural networks.

3.1 Data

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 PFP through the use of an application Navigate Pain in a clinical setting. In addition to the pain maps an appurtenant dataset was available. This contained information regarding the individuals in terms of i.a. age, gender, symptom duration, pain intensity and the most prominent knee for pain. An example of a pain drawing can be seen in figure 3.1.



Figure 3.1: Pain drawings of the lower extremities. The red markings indicate the area of pain perceived by the individual subject. In this case the PFP is bilateral (on both knees).

Before using the data in the deep learning model a manual data handling was necessary. This

incorporate matching the given pain maps and appertaining ID regarding the individuals, which resulted in 217 pain maps. Furthermore specific information like gender, symptom duration and pain intensity is collected. The number of pain maps and associated information, gender and symptom duration, was 205. Additionally, there were 197 pain maps with associated information, 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 patient to healthcare personnel. The application permits individuals to draw their pain into a body outline with different colors and line thickness. Navigate Pain android was developed at Aalborg University and a commercial web application is available at Aglance Solutions (Denmark). [29] Figure 3.2 illustrate the process using the application.

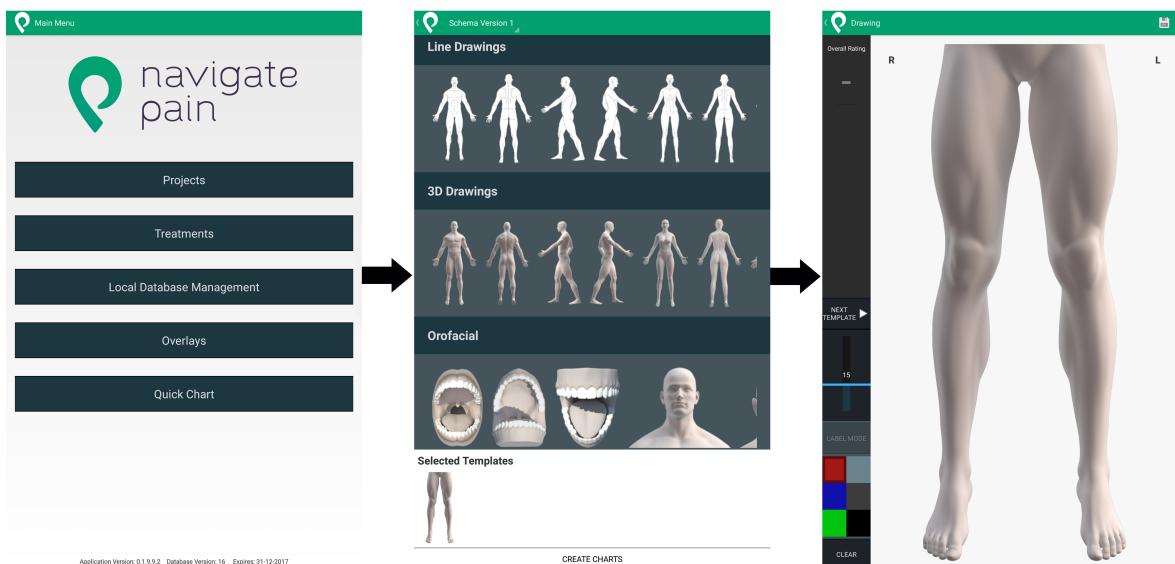


Figure 3.2: The figure illustrates the process for making a pain map with Navigate Pain. There is three screenshots of the application.

The left screen in figure 3.2 is the main screen. By clicking on "Project" a folder with subjects is created. From each subject information like name, age, height is saved. Before the subject can draw their pain areas, the body outline has to be chosen, which illustrates the screen in the middle. The body outlines is divided into five categories: Line Drawings, 3D Drawings, Orofacial, Special Zooms and Knee Pain. In the bottom the selected templates is shown. When clicking on "CREATE CHARTS" the right screen 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 data representations of pain maps affect the performance accuracy of a deep learning model, which is why different data representations are created. A study found a correlation between a prolonged symptom duration and the size of the pain area. It was shown that the pain area increased for individuals that have a symptom duration

for more than five years compared to those with a symptom duration below five years. Likewise pain intensity had a correlation with the size of pain area for symptom duration above five years. Furthermore, the shape of the pain developed from a U-shape to an O-shape for individuals with a symptom duration above five years.[6] Based on this study the morphology is considered to be relevant to investigate, which is why morphology is one of the data representations.

The PFP is often described as diffuse pain and therefore difficult to describe and localise [?]. To accommodate this is it chosen to divide the pain into different knee regions, which may indicate whether a specific region of the knee influence the PFP. This is converted to a simplified data representation that indicate active knee regions. A combination of the two data representation is combined to create a third data representation which both include the morphology of the pain and the different knee regions. Gender is included as an input parameter in the three data representations, because it is shown that the prevalence is more than twice as high for females than males [3?].

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

3.2 Pre-analysis

The pain maps and associated symptom duration and pain intensity are analysed to get an overview of the data. The data is analysed in MatLab, where the distribution of the outputs, symptom duration and pain intensity, are investigated whereafter the classifications to the neural network models are decided. Furthermore the distribution of gender is compared to the literature which states that the prevalence is higher for females than males. To select the threshold for data representation according to the active pain regions are the pain areas in different pain maps analysed. Simple linear regressions of pain area and either symptom duration or pain intensity are made to get a reference to the neural network models.

3.2.1 Classification of data

The neural networks models have to classify the data representations into categories. To find these are histograms of the outputs created.

A histogram of the symptom duration associated with the pain maps is illustrated in figure 3.3.

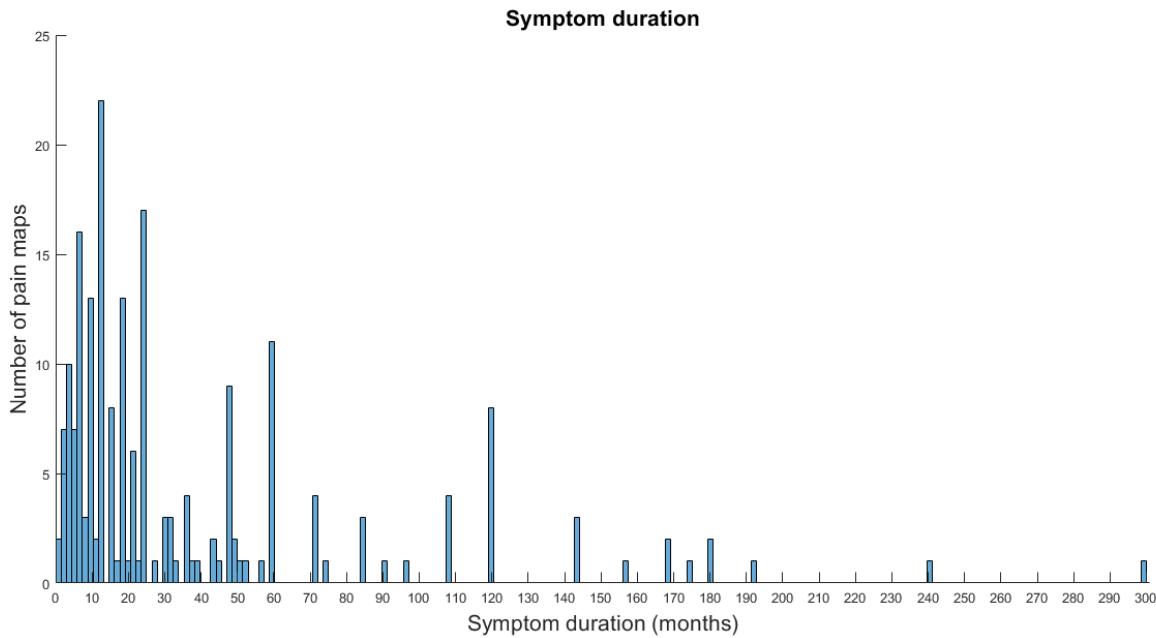


Figure 3.3: A histogram of the symptom duration.

SOMETHING hvor mange klasser hvor de skal splittes

In the appurtenant data to the pain map the individuals have stated their pain intensity like the worst pain in the last 24 hours and the last 7 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 7 days to get a more average worst pain intensity. To explore the difference between the individuals' stated pain intensity in the last 7 days is a histogram created which can be seen in figure 3.4.

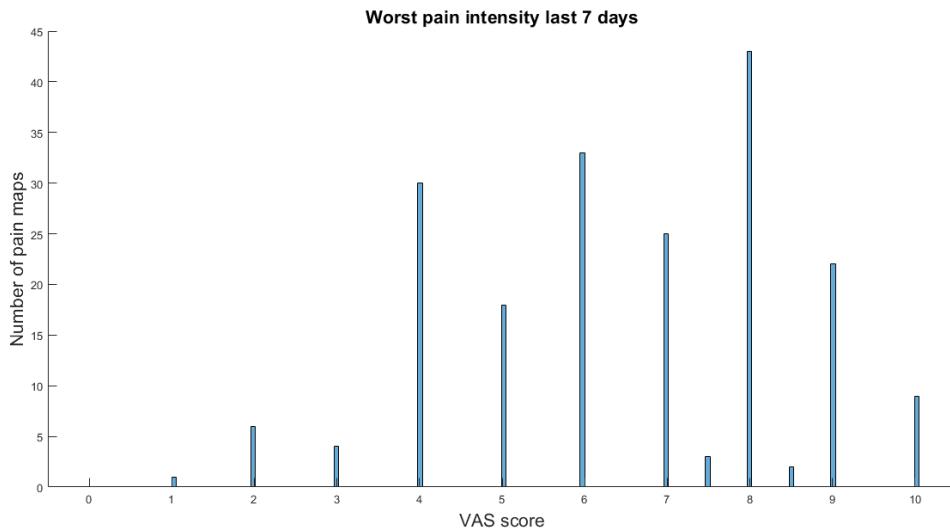


Figure 3.4: Histograms of the pain intensity the last 7 days.

The worst pain intensity is divided into some classes which the models should classify in addition to. To test the models are the data firstly divided into the extremes, since it is assumed that if the models predict badly with the extremes, the models would not predict

better with multiple classification of the pain intensity. The extremes is chosen to be intervals 1 to 4 and 8 to 10 by which the last classification constitute the interval between.

3.2.2 Distribution and threshold

Gender is an interesting parameter to use as an input, because the prevalence is more than twice as high for females than males. Thereto perceived pain is subjective and depends on the individual's character and personality. The distribution of gender is investigated by creating a histogram, which is shown in figure 3.5.

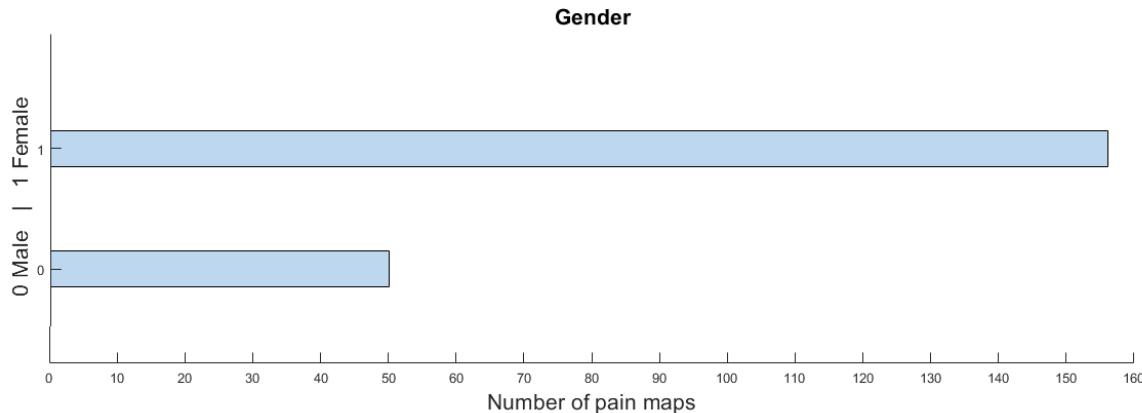


Figure 3.5: Histograms of the distribution/division of gender.

According to the given data the prevalence is higher for females than males. The females constitute 156 of the 206 individuals.

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. Simultaneous may the threshold not be too large so potential pain regions will not be incorporated. The threshold to indicate active pain regions is decided based on an analysis, where threshold values of 0, 5, 10 and 15 percent are tested. The analysis of the threshold is tested on five random pain maps to get a general impression of the data. To better illustrate the division of the pain regions are the regions in figure 2.7 colored in different colors that are easier to distinguish, which is shown in figure 3.6.

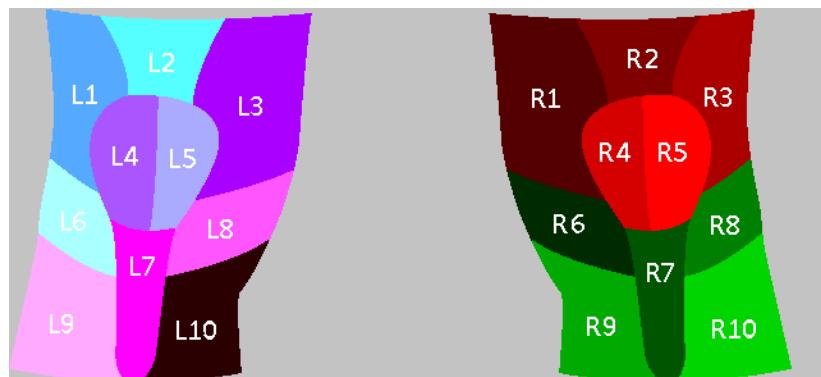


Figure 3.6: Knee regions colored in colors that are easier to distinguish.

An example of pain maps and appurtenant bar chart are illustrated in figure 3.7. The pain maps, figure ??(a-d), are likewise colored in the same colors as figure 3.6 to indicate which regions that are affected according to 0, 5, 10 and 15 percent threshold. The last figure (e) is the bar chart that indicates how many and which active regions there are according to the threshold values.

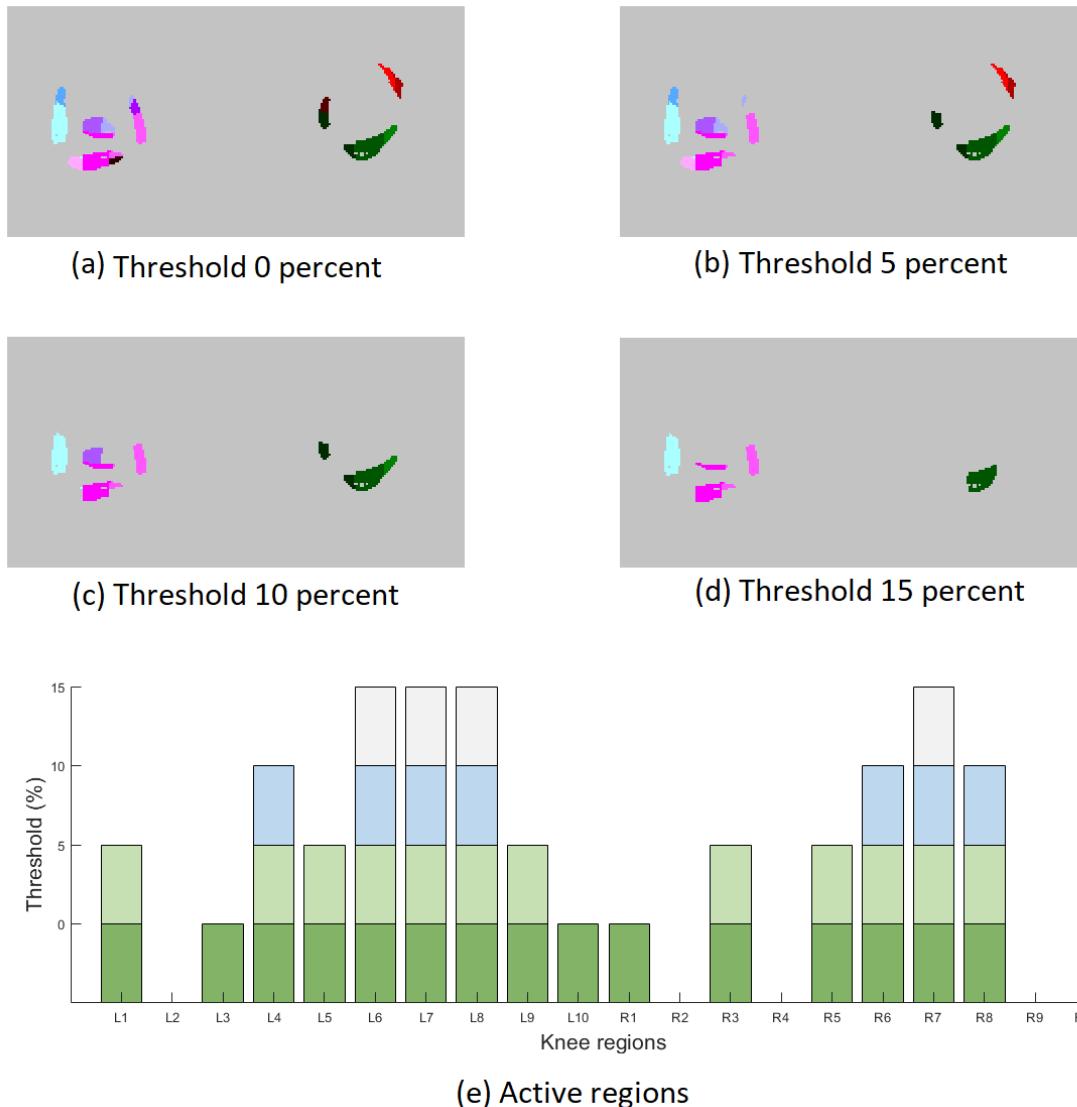


Figure 3.7: The active knee regions when the threshold is (a) 0 percent, (b) 5 percent, (c) 10 percent and (d) 15 percent. (e) is the bar chart that indicates how many and which knee regions that are considered active.

According to figure 3.7 (a) and (e) it is shown that the knee to the left has nine active regions and the knee to the right has six active knee regions, when the threshold is zero. In proportion to the active regions are region L3, L10 and R1 very small and are thereby the first regions to be discarded when the threshold is increased by five percent, which is shown in figure (b). By comparison figure (a) and (b) are there minor changes according to the missing regions, compared to figure (c) and (d) where greater areas disappears after increasing the threshold to 10 and 15 percent.

Based on analysis of the five pain maps and bar charts, figure ?? and appendix ??, is a

threshold on five percent chosen to avoid including minor pain areas, like L10, as active knee regions, and to avoid discarding too many and large areas, like R3 and R5.

3.2.3 Simple regression models

To test whether there is a linear correlation between the size of the pain area and the symptom duration and pain intensity, are simple regression models made. If the size can predict the symptom duration and pain intensity, it may not be significant to investigate pain morphology and location.

In figure 3.8 is a linear regression fit of symptom duration and size of pain area illustrated.

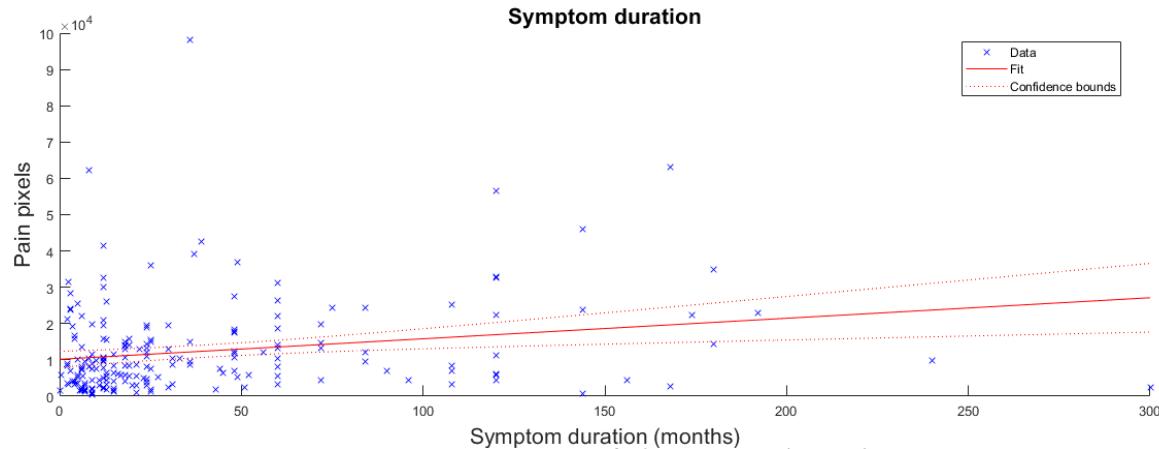


Figure 3.8: A simple linear regression fit of symptom duration and the size of pain area.

The linear regression model of symptom duration has an R-squared value of 0.046, which is close to zero and therefore the model is not a good fit for the data. A linear regression model of pain intensity and the size of pain area is also made, which is illustrated in figure 3.9.

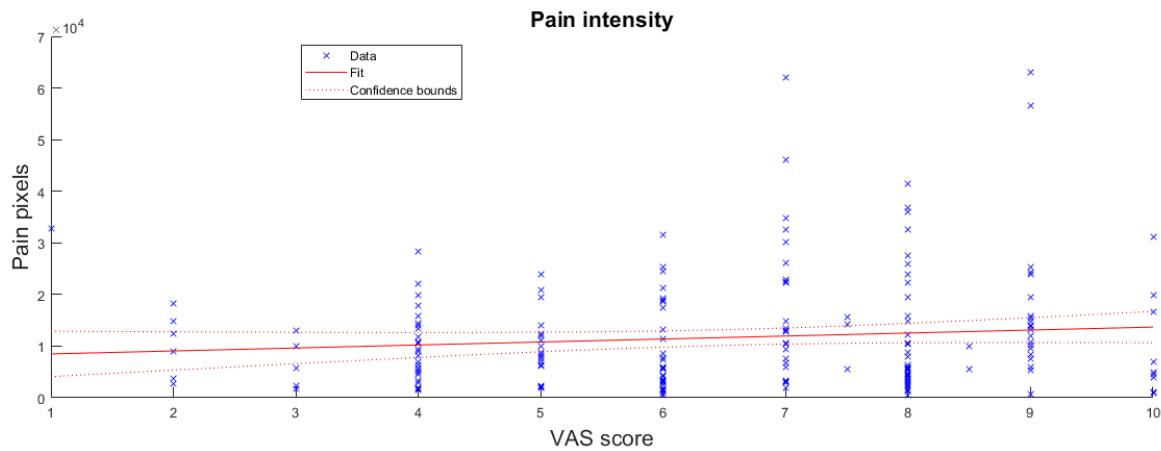


Figure 3.9: A simple linear regression fit of pain intensity and the size of pain area.

The R-squared value of this model is 0.0117, so the linear model is not a good fit on the data. These linear regression models are not very suitable when trying to predict symptom duration or pain intensity from the size of pain area. However they can be compared to the performance of the neural network models.

3.3 Pre-processing

The data is pre-processed in MatLab to prepare it to the three different neural network models. Each model has an appurtenant data representation which are prepared in three different ways. The three data representations are morphology, regions and superimposed morphology and region. Common for the data representation is that the pain maps are imported as image-matrices whereafter the matrices are resized, since the given data 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 neural network models, the image-matrices are converted into vectors whereafter they are assembled in one matrix for each data representation. To get additional information associated with the pain maps, is gender added by including a column vector to the three matrices. In addition to the input, the neural network models need an output to train the models. The output, which is either symptom duration or pain intensity, is likewise 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 symbolized with ones. An original pain map and a pain map consisting of a binary matrix is shown in figure 3.10.

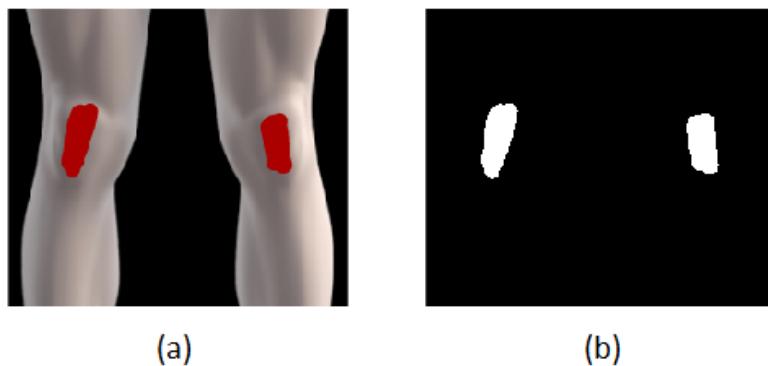


Figure 3.10: (a) Original pain map and (b) image consisting of a binary matrix where white color represents the pain pixels.

An illustration of this data representation is created to convey how the data is assembled and transferred to the model. The illustration is shown in figure 3.11.

	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 3.11: An illustration of the matrix of the morphology data representation. The matrix consist of image-vectors for each subjects where the two last column indicate the appurtenant gender and either duration or pain intensity. The image-vectors has a length equal to the number of pixel 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. The knee regions shown in figure 2.7 are converted into a matrix consisting of 20 values, which represent each knee regions. This matrix is superimposed to the binary image of the pain map, which results in a matrix with pain represented in each knee region. In figure 3.12 are the knee regions and the pain associated with the regions illustrated.

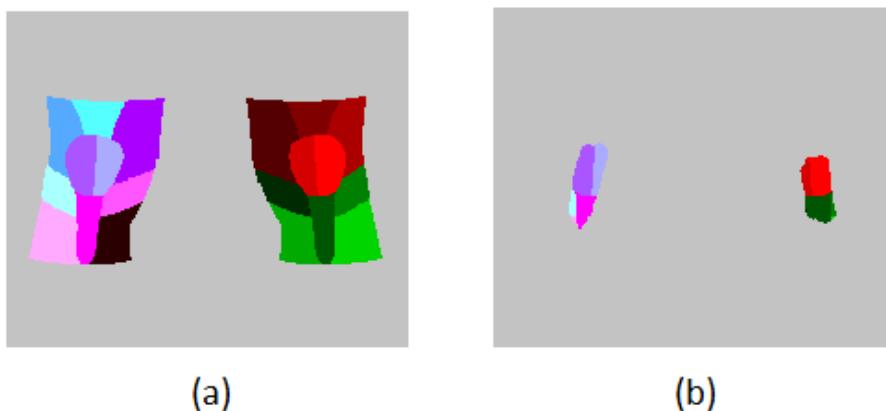


Figure 3.12: (a) Knee regions and (b) pain in the specific regions.

After superimposing the two matrices, knee regions and pain, the number of 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 15 % pain are excluded. WHY 15%. As a result a vector with 20 values is created. This data representation is implemented the same way as the first representation, figure 3.11. The only difference is that the length of the

image-vectors respond to the 20 regions, and therefore are there only 20 values in this data representation.

3.3.3 Superimposed-representation

The third representation of the data is a matrix consisting of individuals' pain divided into the knee regions. In this representation the superimposed matrix from the second data representation is used. Since the data representation should reflect the morphology of the pain and divide the pain into the different knee regions is one-hot encoding used. One-hot encoding is a way to separate categorical data into binary data [30]. This means that the 20 values for each knee region do not have a correlation. After one-hot encoding the superimposed matrix consists of 20 layers where each layer represents a knee region. An illustration of this data representation is shown in figure 3.13.

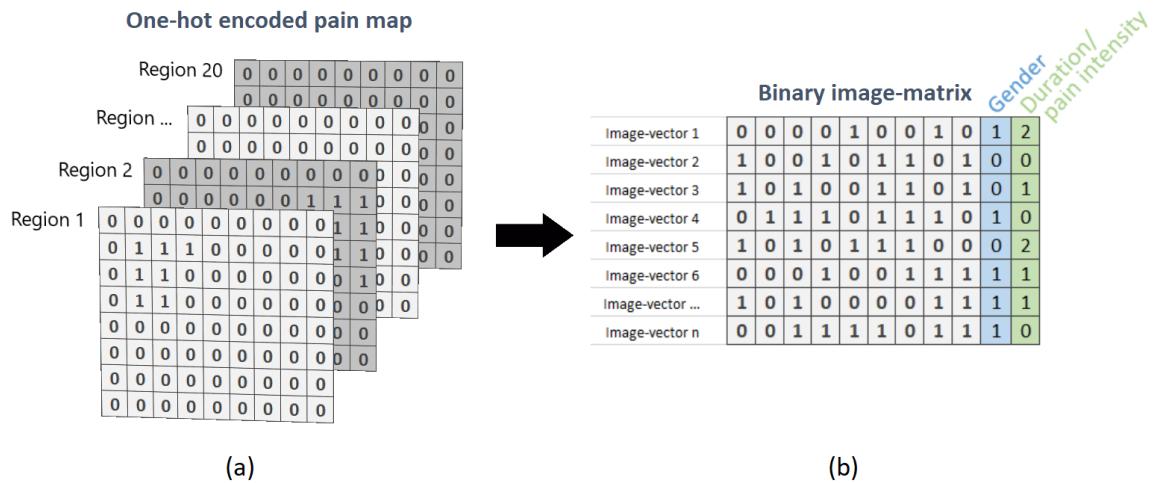


Figure 3.13: (a) illustrates the one-hot encoded pain map and (b) shows the images-vectors in one assembled matrix with gender and either symptom duration or pain intensity.

3.4 Programs

In this project it is chosen to use Python v3.6.3 for development of the neural network. Python is an object-oriented and general-purpose programming and scripting language. Python is among other things used for programming websites, mobile applications, desktop GUI's, but also used for machine learning programming. When developing a machine learning application, there are different libraries that can be used, where some of the most popular is the Theano and the TensorFlow libraries.[31]

In this project the TensorFlow v1.3.0 library has been used. TensorFlow is an open source library for development of machine learning applications, that has been released by Google [31].

maybe something about keras... if we are gonna use it. Keras is a high-level neural network library, that runs on top of either TensorFlow or Theano. Keras is a simplified version of the two libraries, which makes it easier to program in Python, but still allows for building complex models.[31]

Chapter 4

Description of models

4.1 General for the three models

For all the models the input and hidden layers uses the ReLu activation, since this is the most common used in neural network. [25]

and the sigmoid for the output layer.

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 valid source, but so far it's the only one that i have found.)

4.1.1 Data-handling in python

The preprocessed data save as a .mat file which is loaded into python. Before splitting the data into different sub-sets, it is shuffled to ensure generalization through randomization. The data is then split into a training-set and a test-set, and respectively makes out 85 % and 15 % of the preprocessed data. **NEED A SOURCE OR A ARGUMENT FOR WE SPLIT THE DATA LIKE THIS.** The test-set will only be used to evaluate the generalization of the model, and therefore won't contain data that's been used during training [27]. By keeping the test-set separate will act as new unseen data, for the model.

The training-set is additionally split into another training-set and a validation-set. Here the new training-set makes out 90 % of the sub-set and the validation set is the remaining 10 %. **WE NEED A SOURCE/OR A GOOD ARGUMENT FOR THIS, SAME AS ABOVE** The validation set will be used to estimate the generalization error during training [27].

4.1.2 Applied optimization techniques

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

Dropout

Dropout is a technique that can be used to prevent or reduce overfitting of a neural network. In a network the dropout can be applied to the individual layers, and works by randomly drop/"turn off" different nodes temporarily in the given layer during training. An illustration of the principle of dropout can be seen in Figure 4.1.

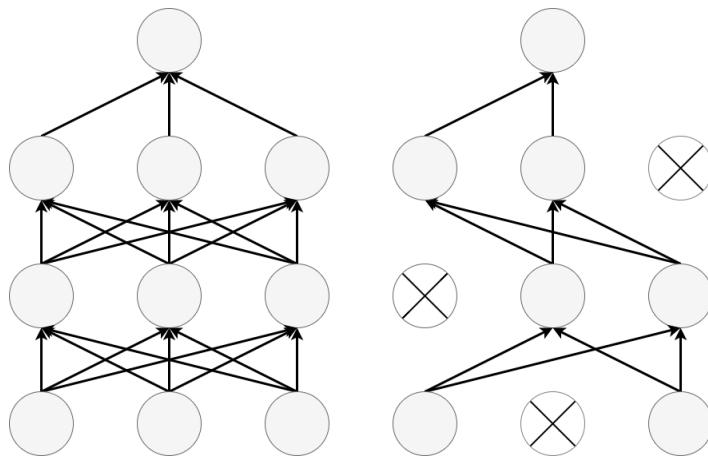


Figure 4.1: Illustration of dropout effect: The figure on the left illustrates a fully connected network, without dropout, and the right shows a network where dropout is enabled on the first three layers.

This reduces co-adaptation, where nodes compute the same features, to where this may increase the generalisation capabilities for a neural network.[32] A study, 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 [32]. When implementing dropout to the models, it is defined in the given layers, and specified with a float value between 0 and 1, which defines the fraction of the nodes that drop [33].

Kernel-initializer 'Maybe'

This is a must i think because this relates to the reproducible aspect.

Learning rate 'Maybe'

In order to determine the most optimal Learning rate for the adam optimizer a several different learning rate were tested. A must because we have done this.

Manufacturing data 'Maybe'

I don't think this is gonna be a reality

Batch training

Save computational cost, update parameters when a batch has been trained.

Grid testing 'Maybe'

MAYBE SOMETHING ABOUT GRID TESTING if we can make it reproductive OBS can only use images as an input

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 described in REF!!!.

These inputs trained and then compared against their respective category label. The models classify data into three different classes e.g. duration interval of 0-12 months, 18-30 months and 36 months and above. Because of this multi classification, the output labels are changed from integer representation, 0 (0-12 months), 1 (16-30 months) and 2 (and 36 months and above) into a one-hot encoded representation [0 , 0 , 1], [0 , 1 , 0] and [1 , 0 , 0] respectively. This is done by using keras utility function called `to_categorical`.

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 either as 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. [27] 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. [27]

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's evaluated that there wouldn't 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 4.2.

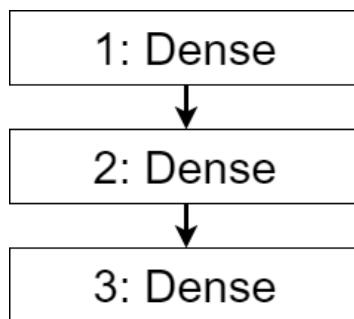


Figure 4.2: Arcitechture of the neural network using knee region representation.

The model consists of three layers, where the input and output layer...

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 [8]. 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 4.3. Convolution layer are implemented for this pain map representation, because of their ability to extract morphology features from images, as written in ???

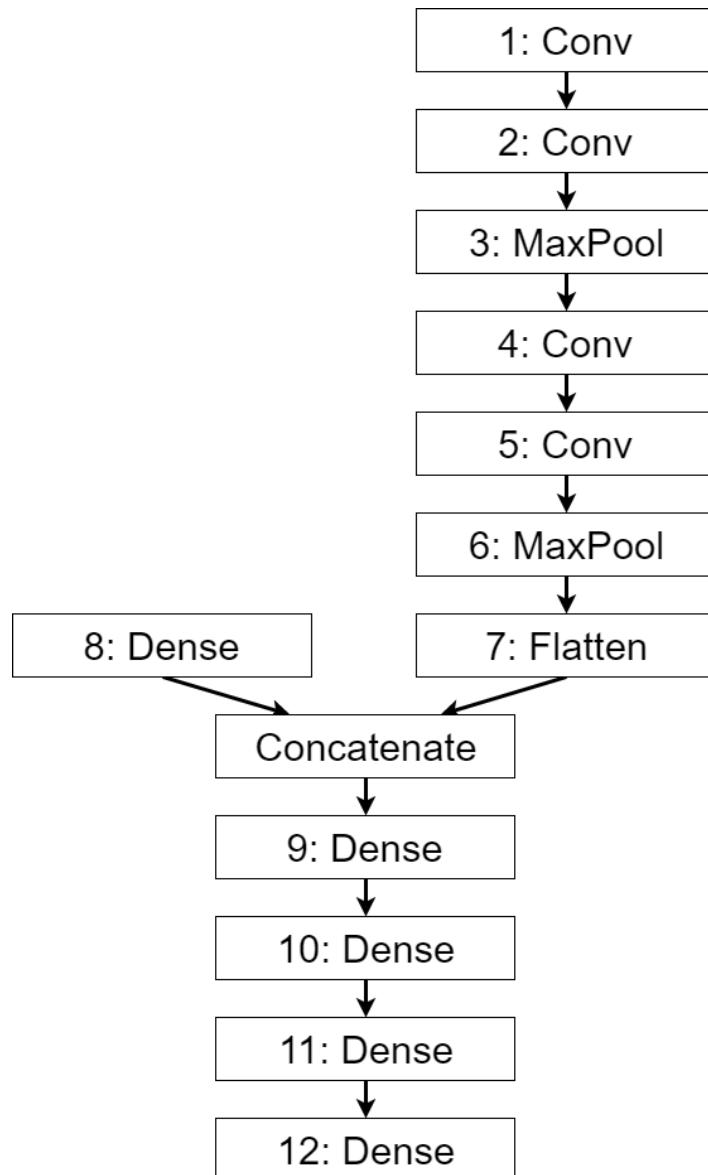


Figure 4.3: Architecture for the neural network model using binary pain map representation.
REMEMBER TO INCLUDE THE INPUT IN THE FIGURE

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-scaling 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 ??.

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] Wolf Petersen, Andree Ellermann, and Et.al. Patellofemoral Pain Syndrome. *Clinical Orthopaedics and Related Research*, 2013. doi: 10.1097/01.blo.0000229284.45485.6c.
- [4] Kay M. Crossley, Michael J. Callaghan, and Et.al. Patellofemoral pain. 2016. doi: 10.1136/bjsports-2015-h3939rep.
- [5] 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.
- [6] Shellie A. Boudreau and E. N. et.al. Kamavuako. 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.
- [7] Mathias Haefeli and Achim Elfering. Pain assessment. 2005. doi: 10.1007/s00586-005-1044-x.
- [8] 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>.
- [9] Frederic H. et. al. Martini. *Anatomy & Physiology*. 2012.
- [10] IASP. IASP Taxonomy, 2012.
- [11] 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.
- [12] R. F Schmidt. Nociception and Pain. In *Fundamentals of Sensory Physiology*. Springer Science & Business Media, 2 edition, 2013. ISBN 9783662011287.
- [13] Emma Briggs. Understanding the experience and physiology of pain. 2010.
- [14] Jarred Younger and Sean Mackey. Pain outcomes: A brief review of instruments and techniques. 2009.
- [15] Kanishka M. Ghosh and David J. Deehan. Soft tissue knee injuries. In Mark Wilkinson, editor, *Orthopaedic surgery: lower limb*. Elsevier, 2010.
- [16] 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.

- [17] Ewa M. Roos and L. Stefan Lohmander. The knee injury and osteoarthritis outcome score (KOOS): from joint injury to osteoarthritis. *Bio Med Central*, 2003.
- [18] 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.
- [19] Geoffrey D. Schott. The cartography of pain: The evolving contribution of pain maps. 2010. doi: 10.1016/j.ejpain.2009.12.005.
- [20] Mads Nielsen. *Den digitale revolution – fortællinger fra datalogiens verden*. Datalogisk Institut, 2010. ISBN 978-87-981270-5-5.
- [21] 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>.
- [22] 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>.
- [23] 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>.
- [24] Mohri Mehryar, Rostamizadeh Afshin, and Talwalka Ameet. *Foundations of Machine Learning*. 2012. ISBN 9780262018258. URL [#](https://ebookcentral.proquest.com/lib/aalborguniv-ebooks/reader.action?docID=3339482&ppg=17).
- [25] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. URL <http://www.deeplearningbook.org>.
- [26] 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.
- [27] Richard Duda, Peter Hart, and David Stork. *Pattern Classification*. Second edi edition, 2000. ISBN 9780471056690.
- [28] 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>.
- [29] Aglance Solutions. Visual insight for clinical reasoning – Navigate Pain, 2015. URL <http://www.navigatepain.com/>.
- [30] David Money Harris and Sarah L. Harris. Sequential Logic Design. In *Digital design and computer architecture*. Elsevier, 2012. ISBN 9780123978165.

- [31] 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>.
- [32] 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>.
- [33] François Chollet. Keras Documentation, 2015. URL <https://keras.io>.

Appendix A

Appendix

A.1 Appendix I