Prediction of pain duration and pain intensity from patellofemoral pain maps using deep learning
- Worksheet

Biomedical Engineering and Informatics
Aalborg University, 1. semester
Written by 17gr7402
Supervisors: Shellie Boudreau and Lasse Riis Østergaard

Contents

Chapter 1

Introduction

Patellofemoral pain (PFP) syndrome is a painful musculoskeletal condition that is presented as pain behind or around the patella [2, 3]. PFP syndrome affects 6-7% of adolescents, of whom two thirds are highly physically active [4]. Additionally the prevalence is more than twice as high for females than males [4, 5]. PFP syndrome is typically present over a longer period of time where a high number of individuals experience a recurrent or chronic pain [6]. Chronic pain may be maintained by the phenomenon central sensitization, which may result in widespread pain over time. Furthermore, PFP syndrome may lead to osteoarthritis [5, 7]. PFP is often described as diffuse knee pain, that can be hard for individuals to explain and localize [6]. Despite the fact that individuals feel pain in the knee, there is no structural changes in the knee such as significant chondral damage. There is no definitive clinical test to diagnose PFP syndrome, and thereby is often diagnosed based on exclusion criterias [5], to which PFP syndrome is also described as an orthopaedic enigma, and is one of the most challenging pathologies to manage [8]. To assist diagnosis of PFP syndrome, pain maps may be used as a helpful tool for the individuals to communicate their pain by drawing pain areas on a body outline [9].

A study by Boudreau et al. [10] indicates, through the use of pain maps, that there is a correlation between the size of the pain (number of pain pixels) and the pain duration as well as pain intensity for individuals with PFP longer than five years.[10] However, it is unknown whether pain duration has an influence on morphology of the pain and location, as well as whether morphology of pain and location have an influence on pain intensity.

The relation between pain maps and pain duration or pain intensity may be complex, because the perceived PFP is subjective, and considered as multifactorial [11]. Additionally the study by Boudreau et al. [10] did not find a fully correlation between 35 pain maps and pain duration or pain intensity for individuals with a pain duration below 5 years. To investigate the potential nonlinear correlation, a deep learning method was used, which is a method that previously has not been applied on this type of data.

The goals of this study is to explore how accurate a deep learning model can classify pain maps according to pain duration or pain intensity. It is assumed that pain duration is a better predictor than pain intensity, because of the subjectivity of pain, and its possibility to be affected by multidimensional factors. The pain maps are encoded into multiple representations to investigate whether morphology and location are correlated to pain duration or pain intensity.

It is assumed that a deep learning model will perform better with more features, thus a combination of morphology and location of the pain constitute a representation. The representations are referred to as morphology-, location- and combined-representation.

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

Furthermore, a secondary aim was to compare the performance accuracy with different pain map representations (morphology-, location- and combined-representation), when predicting pain duration or pain intensity.

Chapter 2

Background

This chapter presents the background knowledge that improves 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. Furthermore, the chapter is essential for getting a basic understanding of some properties and optimizers in the deep learning 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 the gliding joint is formed. The structure of the knee is illustrated in figure 2.1.[12]

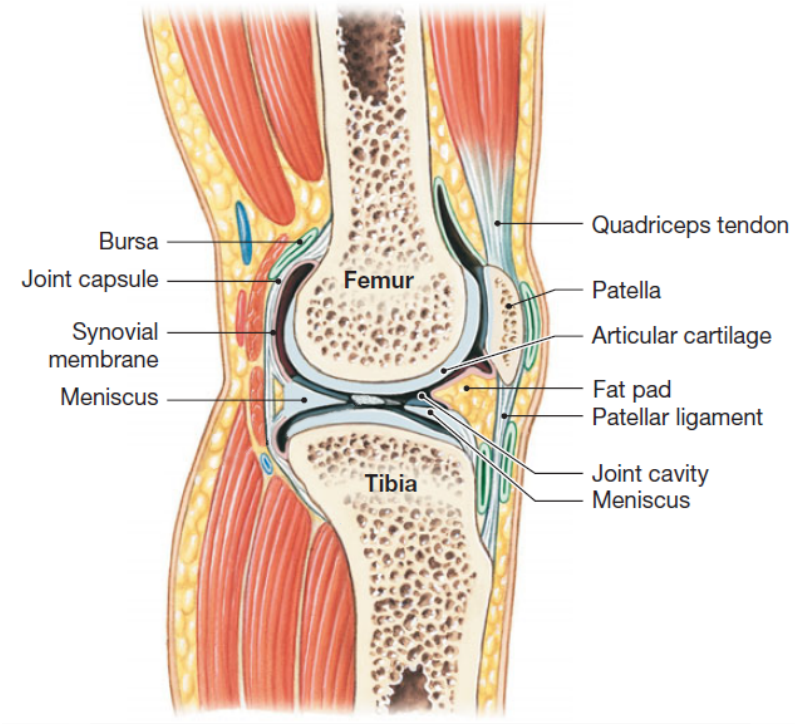


Figure 2.1: *The anatomy of the knee. Edited [12].*

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.[12]
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.[12]
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.[12]

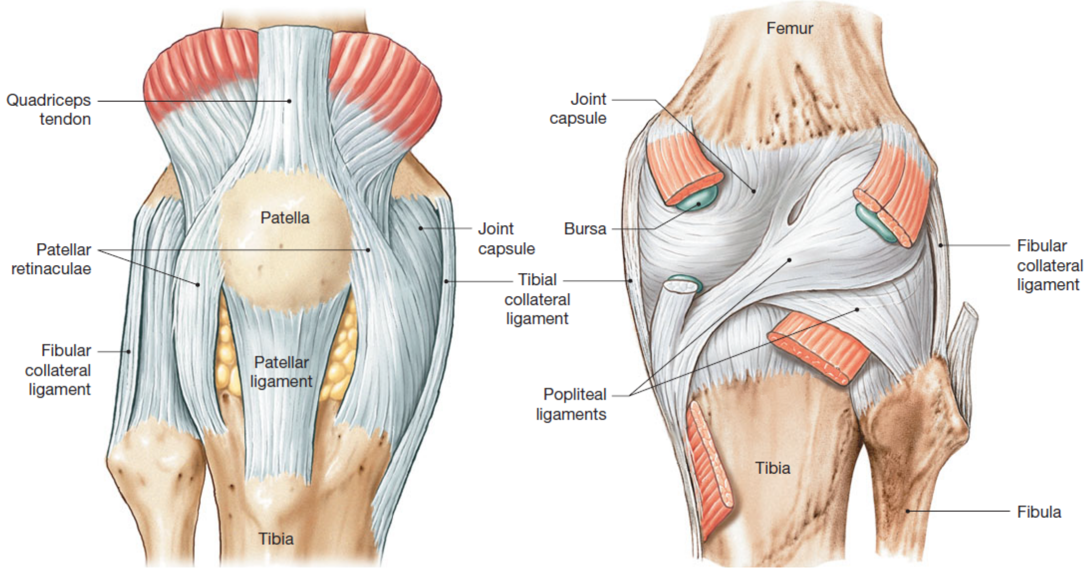


Figure 2.2: *The anatomy of the knee focusing on the ligaments. Edited [12].*

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.[12]

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" [13, 14].
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.[14]
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. Neuropathic pain occurs central from the nervous system. This pain can be caused by illness or physical damage.[15]

Furthermore, pain can be divided into three categories: acute pain (less than three months), persistent or chronic pain and cancer pain.[16] Additionally, the pain can be divided into qualities, which is shown in figure 2.3.

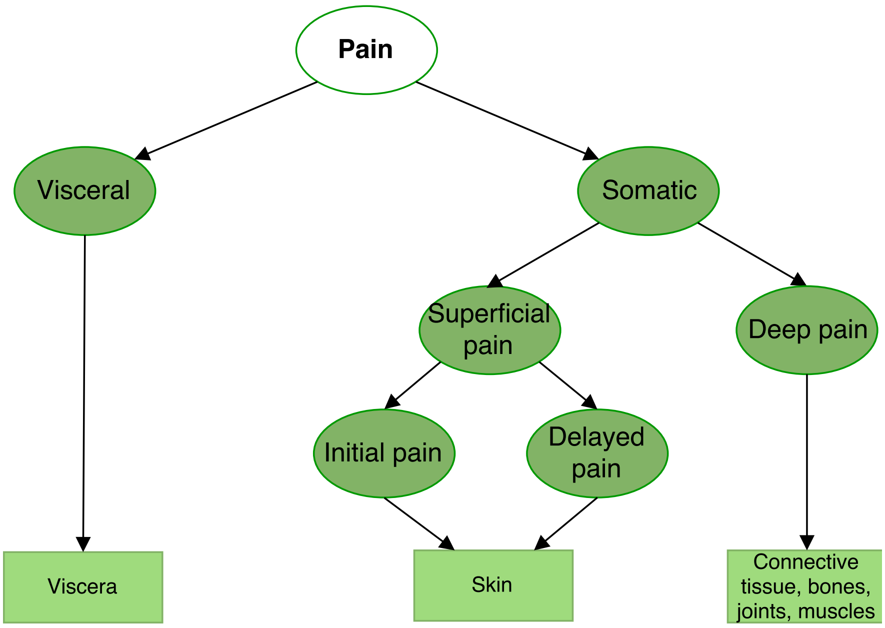


Figure 2.3: *Model of pain qualities. Ovals with green background represent qualities of pain. The rectangles show where the pain originates. Edited [15].*

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 dull or diffuse. 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 localize than the initial pain and lasts longer.[14, 15] 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.[14, 15]

2.3 Patellofemoral pain syndrome

Patellofemoral pain (PFP) syndrome is a painful musculoskeletal condition [2, 3], which presents as pain behind or around the patella. The PFP is often known as anterior knee pain or 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.[3, 7, 10, 17] However, this pain is not caused by previous trauma [7]. Knee pain is not the only symptom of PFP syndrome, the individuals often complain about knee stiffness, patellofemoral crepitus, swelling knee and having trouble with common daily activities [7, 12]. The individuals may limit or stop physical activity because of the pain, and that can lead to an increase in weight [5, 7].

Physiologically PFP syndrome 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.[12]

PFP is mostly prevalent in adolescents and younger adults who are physically active, but it can affect people of all ages and activity levels [2, 7, 17]. Additionally, females are affected about more than twice as often as males [5]. Furthermore, the PFP syndrome may persist for up to 20 years, and thereby categorised as a chronic pain, and may lead to osteoarthritis [5, 7].

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 [5]. 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 [18].

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.[7, 17] Therefore, the diagnosis PFP syndrome is often based on exclusion [5]. After the diagnosis, evidence based treatments may reduce pain and improve function that allows individuals to maintain physical activity [17]. The aetiology of PFP syndrome still remains unclear [3].

Central sensitization

Central sensitization (CS) may give an evidence-based explanation of PFP, which is an unexplained chronic musculoskeletal pain [19, 20]. CS in not completely biological, but may further depend on psychosocial factors [21].

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.[22, 23, 21]

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 [21, 22, 23].

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.[24]

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 there is McMurray test which tests for meniscal tear.[25] Illustrations of the tests are shown in figure 2.4.

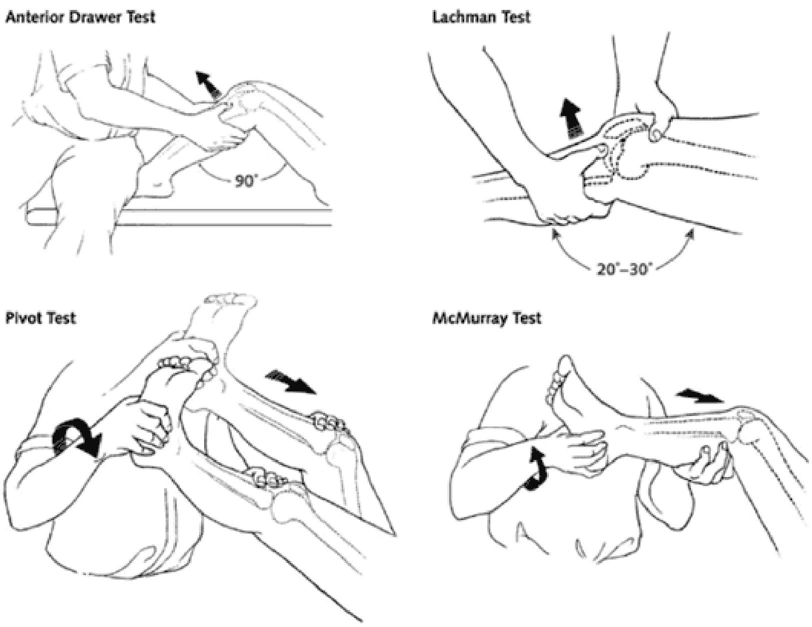


Figure 2.4: *Clinical examination with provocative tests: Anterior Drawer Test, Lachman Test, Pivot Test and McMurray Test.[25]*

In addition to clinical tests there is some paraclinical tests such as X-ray and MRI, but as mentioned PFP syndrome does not show any structural changes in the knee [5], which makes it difficult for healthcare personnel to treat the individuals.

2.4.2 Pain interpretation

Pain is experienced and perceived subjectively [13, 24] and is dependent on personality and character [15], which is why it is important to measure the pain from the individual's perspective. Additionally, there may be a difference in how females and males reports pain intensity, where females reports more intense and frequent pain [26].
One of the most commonly used methods to measure pain intensity is Visual Analogue Scale (VAS) [27]. 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.[28] An illustration of a VAS is shown in figure 2.5.

Figure 2.5: *Visual Analogue Scale (VAS). Edited [28].*

Additionally, questionnaires are 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.[29] The KOOS questionnaire can be seen in Appendix A.

Pain mapping

Individuals indicate PFP by 'placing both hands over their knees', apparently because of 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 [30], 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.[31]

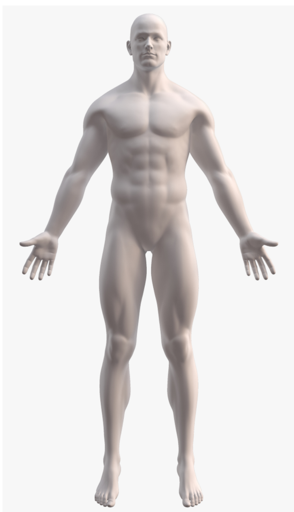


Figure 2.6: *An anterior body outline for pain drawing taken from the software application Navigate Pain.*

Pain mapping are commonly used in clinical practice [31], 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.[9] According to a study by Schott [31], 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 [31].

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 diffuse pain, which is hard to localize.
To distinguish between different pain locations, 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.

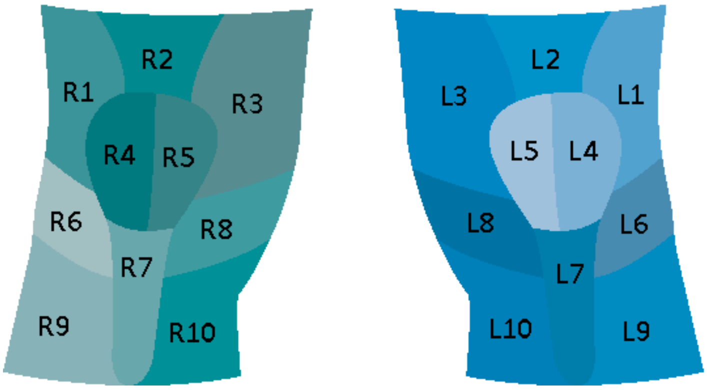


Figure 2.7: *The location of the left and right knees, where each knee is split into ten regions. Edited [32].*

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.[32]
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.[32]

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 [33]. 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.[34] 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.[35]

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.[33, 36] 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 [36]. 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.[33] An example of a neural network structure can be seen in figure 2.8.

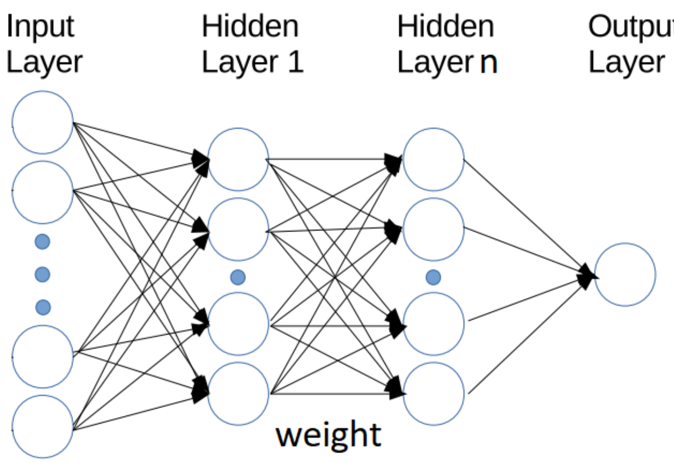


Figure 2.8: *Example of the structure of a neural network. Edited [37].*

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 [38]. 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.[36] 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.[33]

Learning scenarios

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

Supervised learning is the most common way of training in deep 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.[33] Overall supervised learning may be described as teaching the network how to associate a given input to a specific output [39], and is mostly associated with classification, regression, and ranking problems [40].

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 [40]. 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 [39].

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.[41, 42] 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 correct output. This makes back-propagation closely related to supervised learning, to which back-propagation is the most general method used.[42]

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.[38, 43] 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 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.[33, 39, 42] This learning process continues until optimal weights with minimum error is reached.[38]

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 a node can be considered as active or not [39]. The activation function is a nonlinear function which gives neural network nonlinear capabilities [41].
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 are shown in figure 2.9. Symmetric hyperbolic tangent is used more often because it converges faster then standard logistic function.[41]

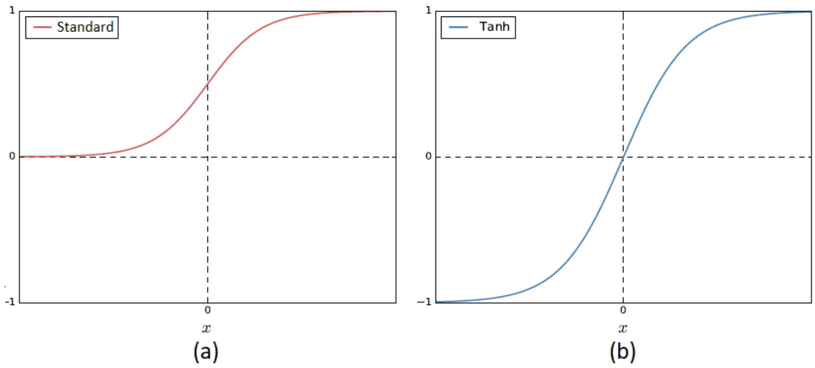


Figure 2.9: *(a) Standard logistic function and (b) hyperbolic tangent. Edited [41].*

Another activation function is rectified linear unit (ReLU), 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[39]. In modern neural networks, ReLU is recommended to use as a default activation function and could be defined as $g(x) = max\{0, x\}$. ReLU is shown in figure 2.10.

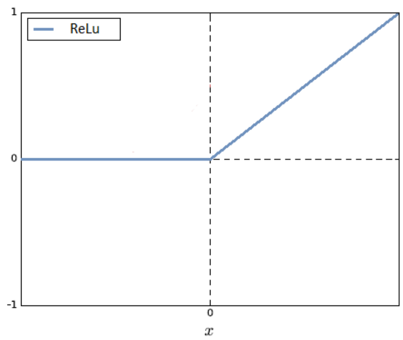


Figure 2.10: *Rectified linear activation function. Edited [39].*

Learning curves

Learning curves are used to observe the error on the training, validation and testing sets, which is increased because of randomly initialized weights and biases. The error is presented over time, which in deep learning is expressed as the number of epochs. Epochs defines the complete pass through a dataset. During the beginning of training, the training error of the network will typically be relatively high, but during training the error decreases monotonically, as the weights are adjusted in the network.[42] An illustration of how the error values are affected during training can be seen in figure 2.11.

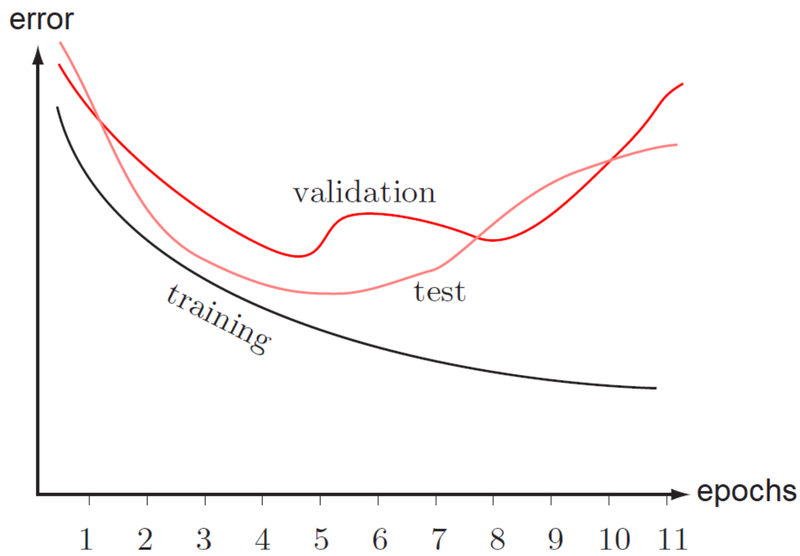


Figure 2.11: *Illustration of how training (black), validation (red), and test (orange) error is affected by the increase in epochs. Edited [42].*

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. Therefore, validation error can 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.[42]

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.[44] 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.[45]

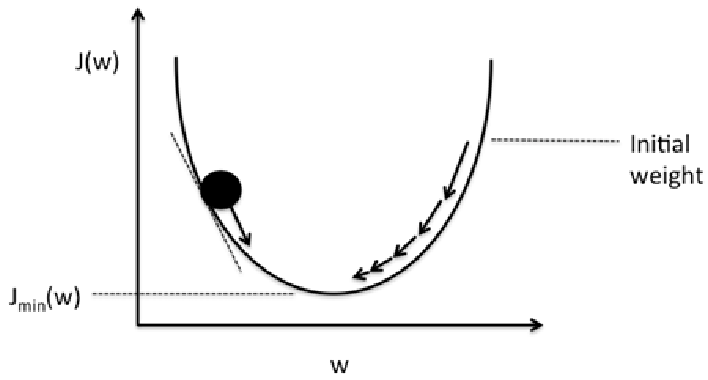


Figure 2.12: *Illustration of gradient descent working principle, where J(w) is the loss function, $J_{min}(w)$ is final approximation to the local minimum of J(w), w is value of the parameter. The arrows indicate the step direction, i.e. the negative gradient.[45]*

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 loss 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.[44]

Stochastic gradient descent (SGD) performs a parameter update for each training example and label. Therefore, it is 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.[46]

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.[44]

Additionally, there can be a few challenges while using gradient descent as an optimizer. It is difficult to pick a proper learning rate so few gradient descent optimization algorithms are invented. The most widely used methods are momentum, Adagrad, Adadelta and Adam.[44]

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.[44, 47]

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.[44]

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.[44]

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.[48]

A study by Patacchiola and Cangelosi [49] showed the evaluation of the performance on different optimizers on a dataset containing 21,977 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.

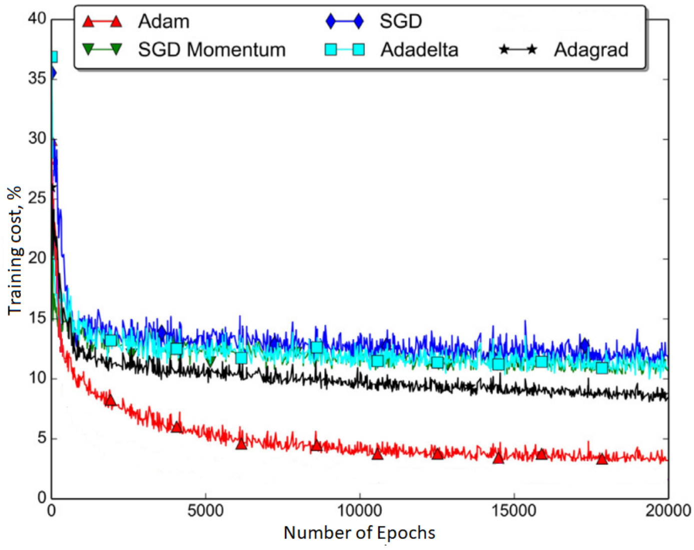


Figure 2.13: *Comparison of the convergence speed between different optimizers used to train architecture on ALFW dataset. The loss values are the mean of the five attempts.[49]*

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 [39], mainly used to restrict a model's expressiveness in order to prevent it from overfitting, underfitting, and generalization. A 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 [50].

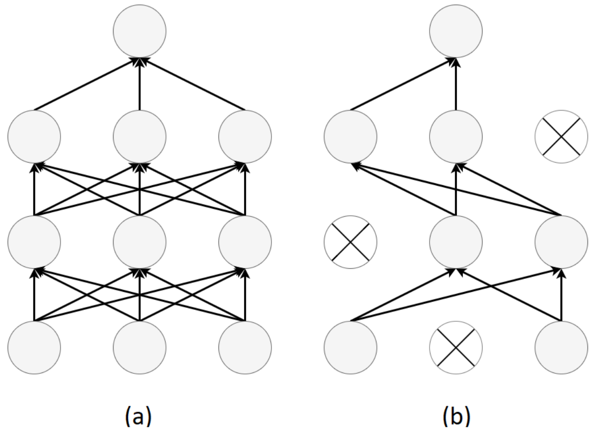


Figure 2.14: *Dropout effect: (a) illustrates a fully connected network without dropout, and (b) shows a network where dropout is enabled on the first three layers. Edited [51].*

This reduces co-adaptation, where nodes compute the same features, to where this may increase the generalization capabilities for a neural network. A study by Srivastava et al. [51] 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.[51]

Initializers

In neural networks starting values of hyperparameters can have a significant effect on the training process [41]. Hyperparameters often define many different values that can be adjusted to control the behaviour of the algorithm, to which some parameters may affect the runtime and computational cost when training the model [39]. Weights, biases initializers could be used in order to take the values randomly and exclude the symmetry between nodes [52].

2.6.4 Core Neural Network layers

Convolutional, pooling, and fully connected layers could be called core neural network layers. In architecture of the network these layers have a specific place and functions [33, 39].

Convolutional Neural Networks

Convolutional Neural Networks (CNNs) is a type of special neural network for processing data with a grid-like topology [39]. 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.[37, 43]
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 as feature map [39, 43]. A complete convolutional layer consists of several feature maps, so multiple features can be extracted at each location in the image [43].

Pooling layer

Convolution layers are typically followed by pooling layers [33, 39]. 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.[43] 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.[39, 43] 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 [39, 53]. A pooling layer may be defined simply by its window (kernel) size, paddling size and a stride length, where stride length is the number of values the window jumps as shown in figure 2.15.[53]

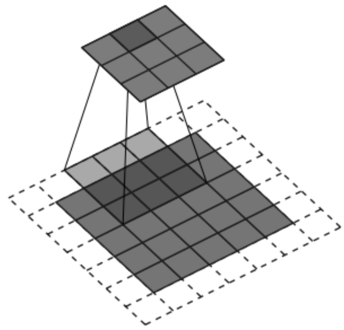


Figure 2.15: *Pooling with 3x3 kernel over a 5x5 input using paddling and strides (i.e., input = 5, kernel = 3, stide = 1, paddling size = 1) [53].*

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 [39]. Fully connected layers have full connections to all activations in previous layers. A typical architecture containing core neural network layers for character recognition of the images, called LeNet-5 [43], can be seen in figure 2.16.

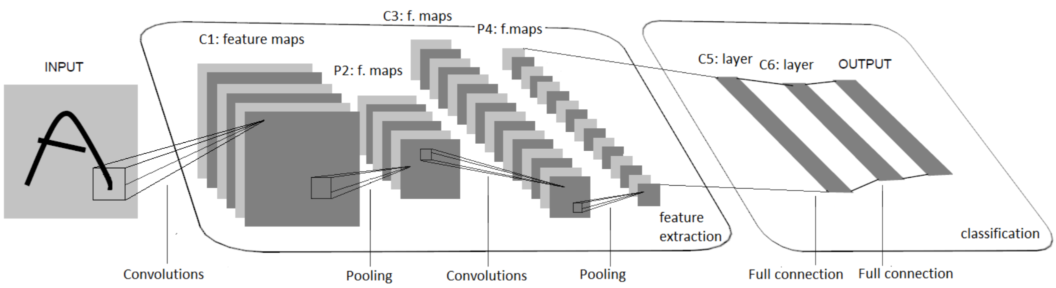


Figure 2.16: *Architecture of LeNet-5, a CNN for character recognition. Each plane is a feature map and the size of the feature map differs throughout the layers. The model consists of two convolutional, two pooling and two fully connected layers [43].*

Chapter 3

Methodology

This chapter creates an understanding of the available data, how it is analysed and prepared before including it in the deep learning models. Additionally, the program Navigate Pain and programs used to develop the deep learning models are described. Furthermore, the architecture of the models is presented.

3.1 Data description

Pain maps used in this study were collected beforehand from an on-going FOXH trial which is conducted in collaboration with Danish and Australian universities. The pain maps were drawn by individuals with PFP syndrome 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 map with bilateral pain is shown in figure 3.1.

Figure 3.1: *An example of a pain drawing from individual with PFP. 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, pain 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 pain 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 on left knees was mirrored and thereby only visualized on the right knee. This resulted in 333 pain maps with associated pain duration, and 319 pain maps with associated pain intensity.

3.1.1 Software application: Navigate Pain

Navigate Pain is a software application that is used to visualize 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 was developed at Aalborg University, and a commercial web application is available at Aglance Solutions (Denmark).[54] Figure 3.2 illustrates the process using the application.

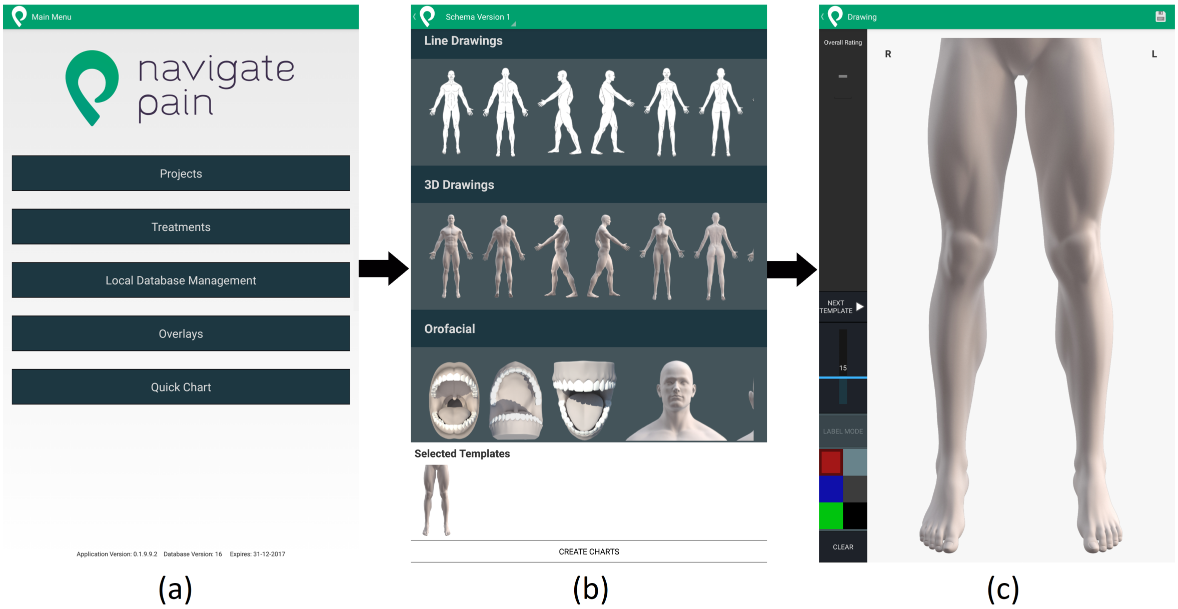


Figure 3.2: The process for making a pain map with Navigate Pain. (a) shows the main screen, (b) categories of body outlines, and (c) body outline for lower extremities.

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, and 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 of the screen the seleceted 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 Pain map representations

It is presumed that different representations of the pain maps affect the performance accuracy of the deep learning models, hence different pain map representations were created. A study by Boudreau et al. [10] found a partial correlation between a prolonged pain duration and the size of the pain area. It was shown that the pain area increased for individuals that had a pain duration above five years. Likewise, pain intensity had a correlation with the size of pain area for individuals with a pain duration above 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.[10]

It is unknown whether the morphology or location of PFP influence either pain duration or pain intensity, to which two pain map representations reflecting morphology or location of the pain were created. A combination of the two pain map representations was created to achieve a third pain map representation which both included the morphology of the pain and the location, which is defined using the knee regions. The three pain map representations are referred to as morphology-, location-, and combined-representation. Furthermore, gender may be considered as an important parameter to use as an input, because of the difference in how females and males report their pain [26]. Additionally, there is an imbalance in prevalence[4].

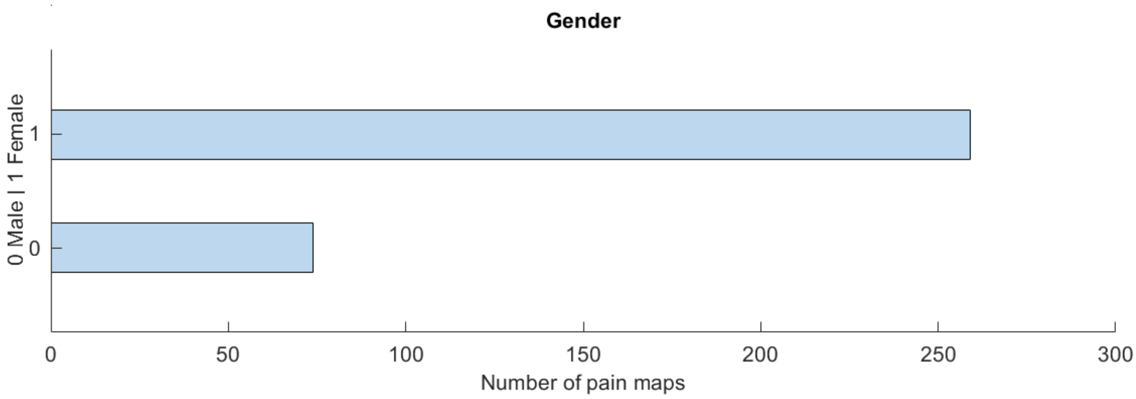


Figure 3.3: *Histogram of the distribution of gender.*

According to the given pain maps the distribution is higher for females than males. The females constitute 259 of the 333 individuals, and males constitute 74 individuals.

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 pain 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 considered active according to the amount of pain. Simple linear regressions are made to investigate whether pain duration or pain intensity have a linear relation to the size of pain as well as the number of active pain regions.

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 pain intensity intervals. To help define the class intervals histograms were created.
A histogram of the pain duration associated with the pain maps is illustrated in figure 3.4.

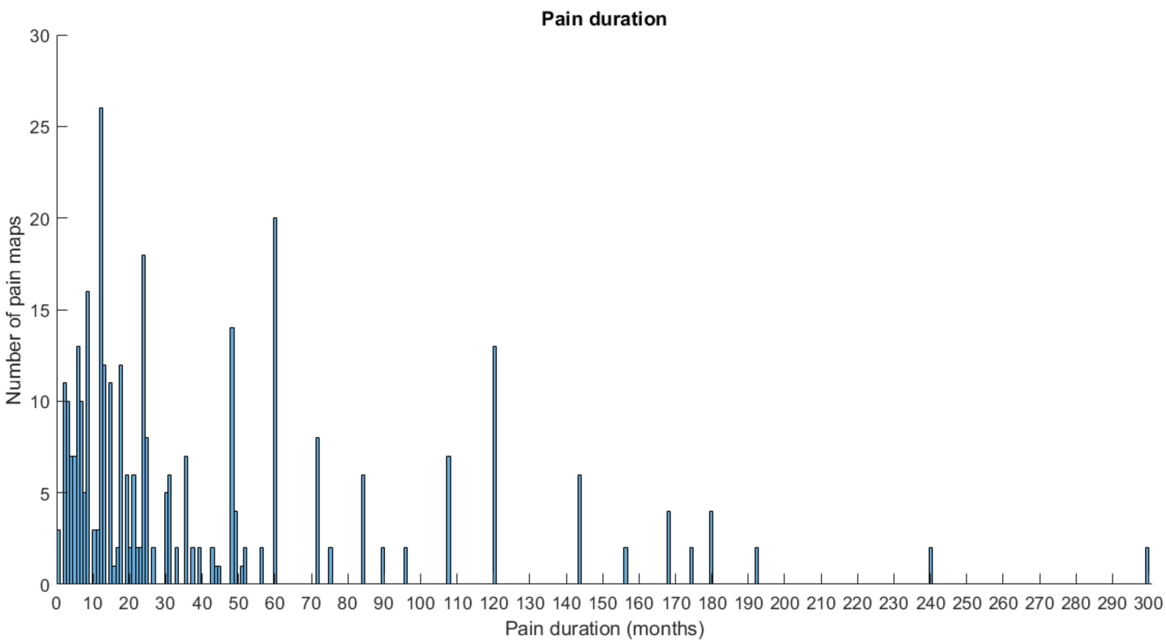


Figure 3.4: *A histogram of the pain duration according to the number of pain maps.*

The pain duration is divided into some classes which the models should classify pain maps according to. The pain duration and pain intensity were divided into two extremes to investigate the models' performance accuracy when classifying according to these intervals. It was chosen to divide the outputs in two classes consisting of the extremes, because it was assumed that the two intervals were more dissimilar, compared to intervals with closer values, and thereby easier to distinguish. 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).

In the associated 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.

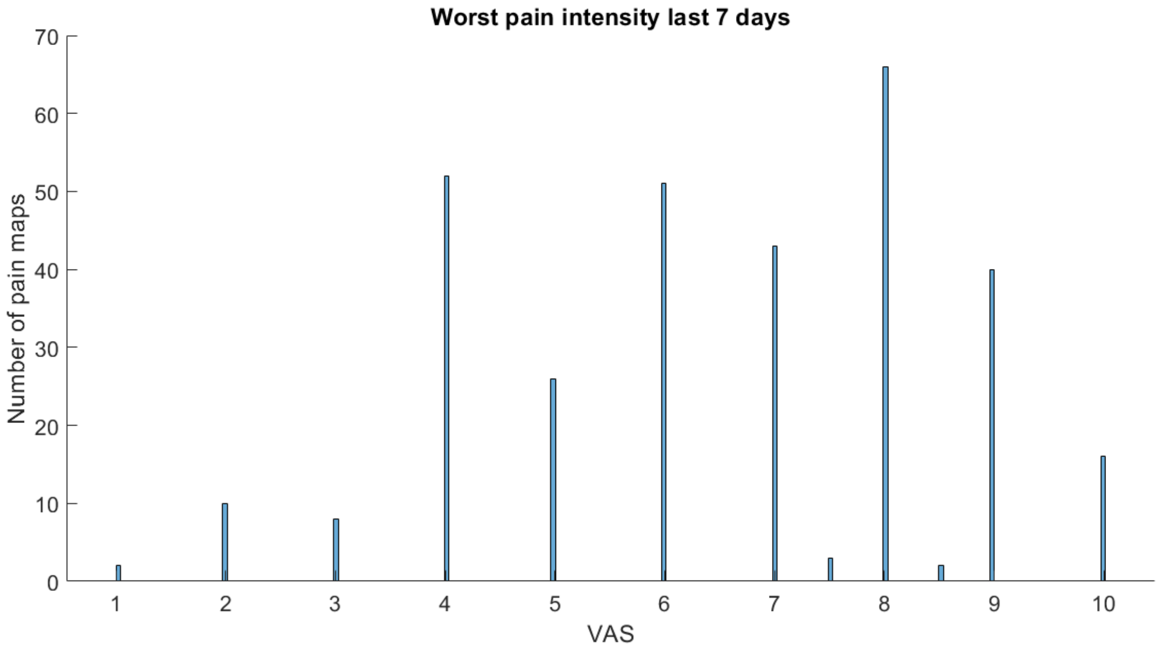


Figure 3.5: *A histogram of the worst pain intensity in the last seven days according to the number of pain maps.*

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).

3.2.2 Threshold selection

In relation to the pain map representation that contains information about the location of the pain divided into knee 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 areas 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% are explored. A 0% threshold is 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

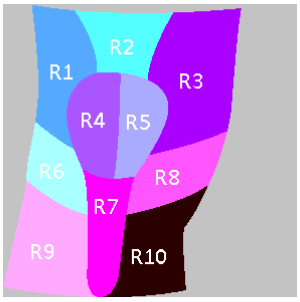


Figure 3.6: *Knee regions colored to easier distinguish.*

An example of pain maps and related 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 3.7(a) no threshold, 3.7(b) 5% threshold, 3.7(c) 10% threshold and 3.7(d) 15% threshold. Figure 3.7(e) shows a bar chart that indicates how many and which active regions there are according to the threshold values.

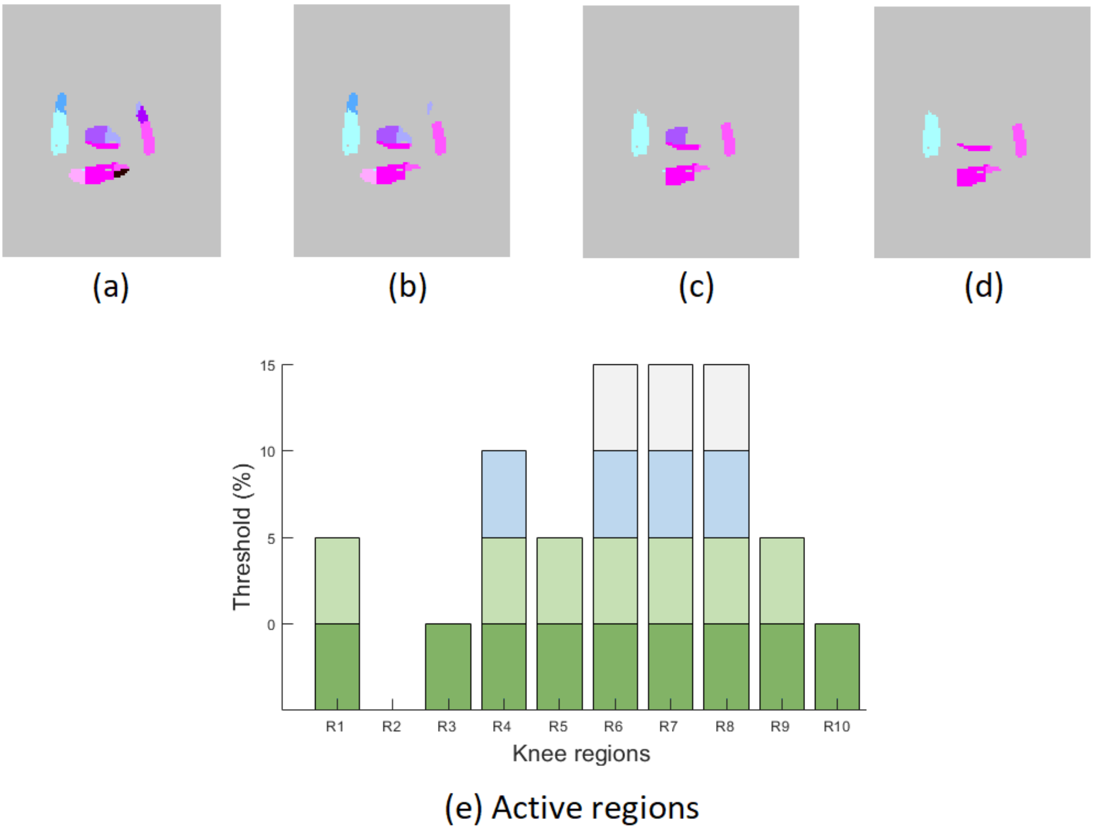


Figure 3.7: *The active pain regions when the threshold is (a) 0%, (b) 5%, (c) 10% and (d) 15%. (e) is the bar chart that indicates how many and which regions that are considered active.*

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 (purple) and R10 (black) are very small and thereby the first regions to be discarded when the threshold is increased by 5%, which is shown in figure 3.7(b). By comparing figure 3.7(a) and 3.7(b) minor changes according to the missing regions can be seen, compared to figure

3.7(c) and 3.7(d) where greater areas disappear after increasing the threshold to 10% and 15%. Based on an 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 pain 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 complexity of pain maps, linear regressions on the pain map representations and output, pain duration or pain 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 pain regions. If these simple features have a linear correlation to the pain duration or pain intensity, it may not be significant to investigate morphology or location as features in the deep learning models.
To investigate if pain size has a linear correlation to pain duration, a linear regression i created, which is shown in figure 3.8.

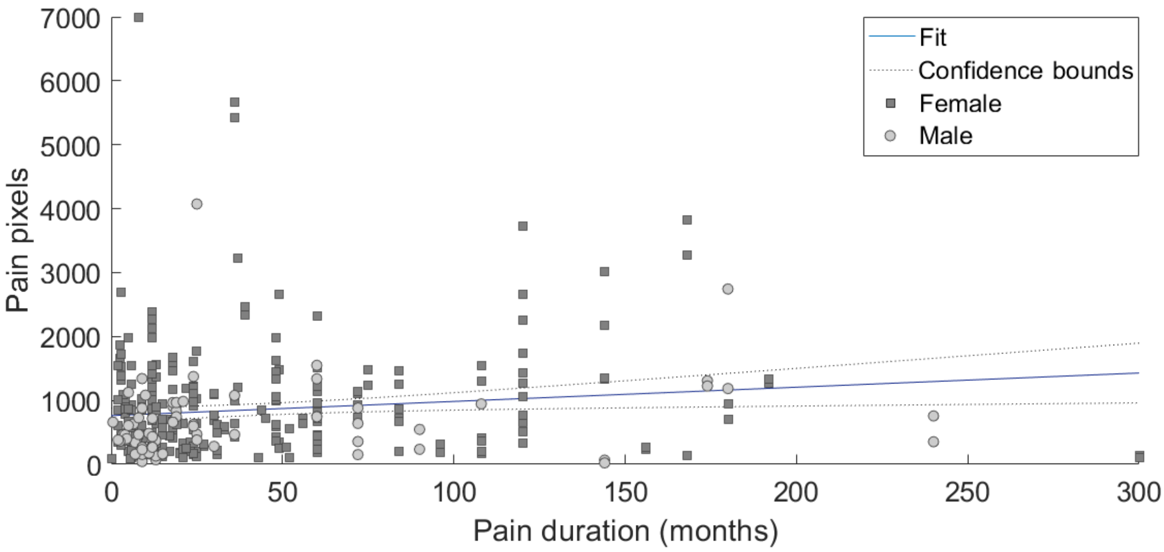


Figure 3.8: *A linear regression of the the number of pain pixels and pain duration.*

As a result of the linear regression model is $R^2 = 0.046$, 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.

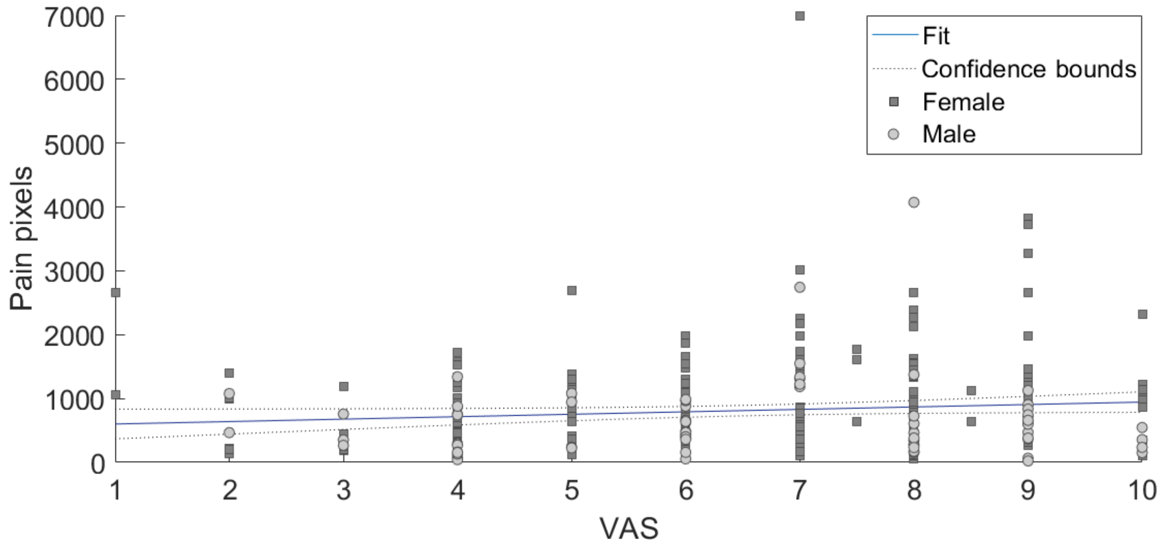


Figure 3.9: *A linear regression of the number of pain pixels and pain intensity stated in VAS.*

The result in this linear regression gives $R^2 = 0.0117$, 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 pain 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.

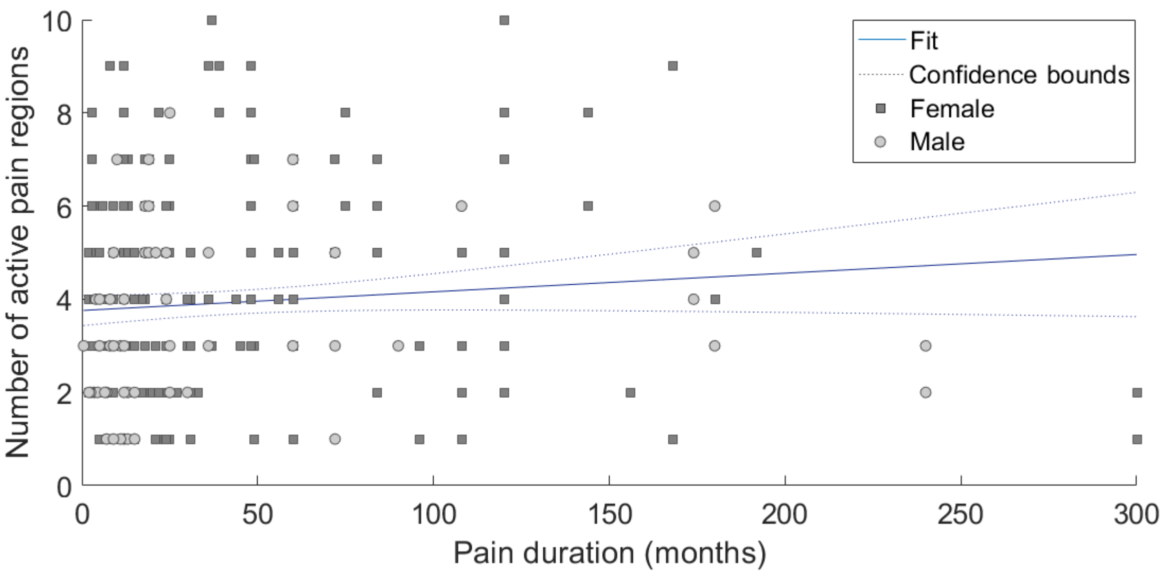


Figure 3.10: *A linear regression of the number of active pain regions and pain duration.*

The result in this linear regression between the number of active pain regions and pain duration gives $R^2 = 0.0357$, 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.

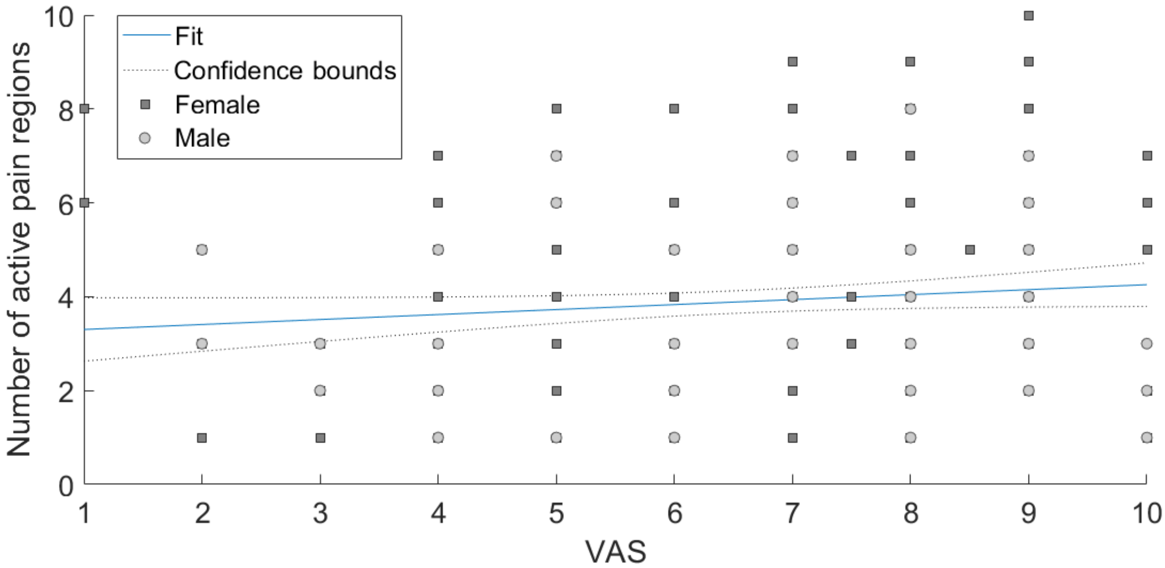


Figure 3.11: *A linear regression of the number active pain regions and pain intensity.*

The result of the linear regression between the number of active pain regions and pain intensity is $R^2 = 0.00833$, which once again indicates nonlinearity.
Based on the four linear regressions, 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 pain intensity. Hence a deep learning model may find patterns in the pain maps in relation to either pain duration or pain intensity.

3.3 Pre-processing

The data is pre-processed in MatLab to prepare the three different pain map representations. The three pain map representations are morphology-, location-, and combined-representation, which are described in section 3.1.2. Common for the representations is that the pain maps are imported as image-matrices whereafter the matrices are resized (pixelsize: $252 \times 118 = 29,736$), since the given pain maps were collected at different resolutions (screen sizes). Furthermore, the matrices are cropped to sort out unnecessary data. Before the pain maps are 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 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 pain intensity, is added as a column vector. The following sections describe the pre-processing of the individual pain map representations.

3.3.1 Morphology-representation

The first representation of the pain maps 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.

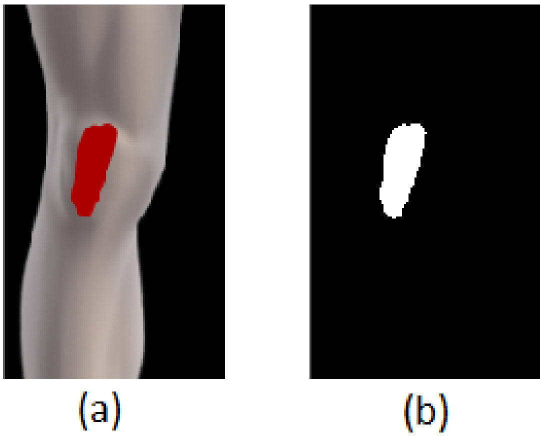


Figure 3.12: *(a) Original pain map and (b) image consisting of a binary matrix where white color represents the pain pixels.*

An illustration of the morphology-representation is created to convey how the pain maps are 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 pain 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 individual where the two last columns indicate the corresponding gender (blue column vector) and either pain duration or pain intensity (green column vector). The image-vectors have a length equal to the number of pixels in the pain maps.*

3.3.2 Location-representation

The second representation of the pain maps is a matrix consisting of vectors with 10 values which indicate location of pain in relation to the knee regions. Figure 2.7 is cropped, and converted into a matrix consisting of 10 values, which represent each knee region on the right knee. 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.

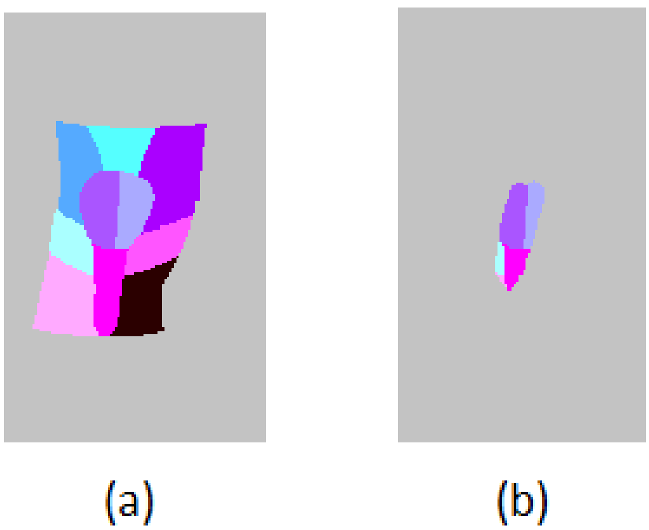


Figure 3.14: *(a) Knee regions and (b) an example of a pain map with pain in the specific regions.*

After superimposing the two matrices, knee regions and pain pixels, the number of pain pixels in each active pain 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. Location-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 pain maps is a matrix consisting the morphology of pain divided into the knee regions. In this representation the superimposed matrix from the location-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 [55]. 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).

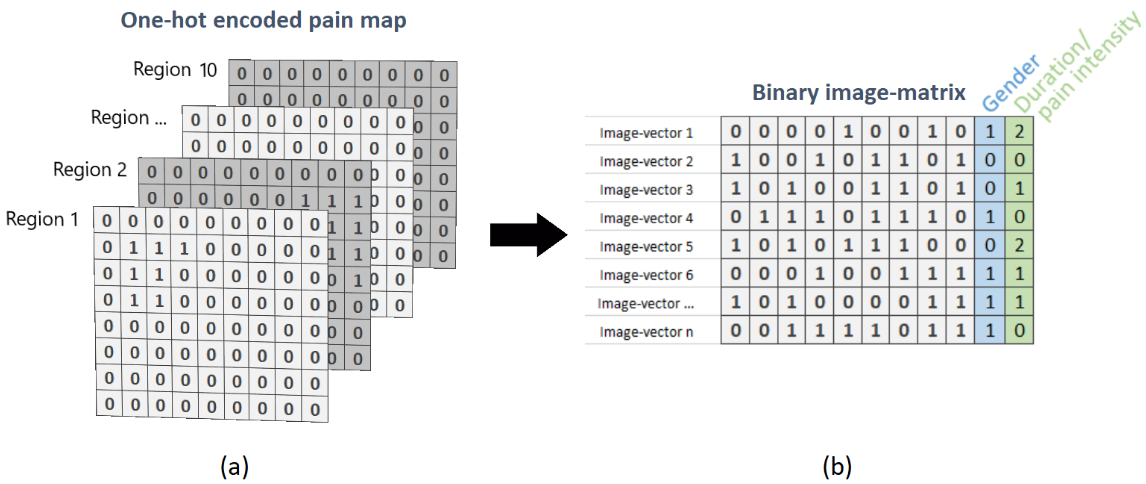


__Figure 3.15:__ (a) An illustration of a one-hot encoded pain map and (b) shows the image-vectors in one assembled matrix with gender and either pain duration or pain intensity.

3.4 Neural network implementation and models

Building neural network for classification of pain maps was trial and error process. In this project, the deep learning method is used to classify the pain maps and gender according to determined outputs. The data is a set of 2D-images combined with gender and the outputs are pain intensity and duration. For classification purpose, the 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. The models are trained, and tested on a single computer with GPU and ran on Python programming language with Keras library. Specifications of hardware and software are described in detail in subsection 3.4.1.

3.4.1 Software and hardware

The neural networks 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 the development of deep learning application in python, different libraries are available, where some of the popular is the Theano and TensorFlow libraries [56]. The Tensorflow v1.3.0 library was chosen for this study. An additional 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 [50]. This was deemed suitable given that no previous experience with the neural network was 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 developed on a laptop with 4x 'Intel® Core™ i7' CPU's and one GPU type 'Geforce GTX 970M' with specifications listed in table 3.1.

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 [57, 58]*

3.4.2 Design choices and model architecture

This section presents the different models, their architecture, and implementation used in this study. For classification of each pain map 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 consists of 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 alternate in order to extract relevant features from the image, as described in section subsection 2.6.4. The fully connected part works as classification, where computed feature maps get 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 subsets, from which training data was used to regulate and optimize the model. Supervised learning was used for training the models. The common input for all of the models was gender, along with the different pain map representations.

Morphology-representation models

The models for the morphology-representation consists of convolution, max pooling, and fully connected layers. The models are essentially made up of two parts, where one is the extraction of image features, and another being a classification part. For the extraction of image features a series of three convolutional and max-pooling layer are stacked after each other, to which the first layer is specified to receive the main input of the shape $1 \times 118 \times 252$ where 1 reflects the fact that the pain map representation is binary. The classification part of the network is the same architecture as that of the model for the location-representation and thereby consists of fully connected layers. Between these two parts of the model, a secondary input is added specifying the gender related to the given pain map. Before the pain maps, features reach the fully connected layers it is flattened from the shape of a matrix to that of a single row in order to merge the extracted 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. The architecture of the morphology representation model is shown in figure 3.16.

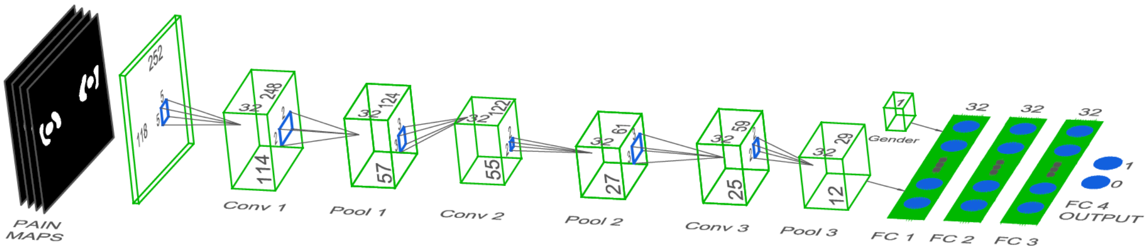


Figure 3.16: *Architecture of the neural network models.*

Location-representation models

Given the location-representations a simplicity, in the form of an 11 element row-vector that contains both the pain maps information along with a gender value as described in section 3.3.2, the architecture of the model is also relatively simple, compared to the other models. For this representation, the model consists of four fully-connected layers where the last of the layers is the output layer, where the number of outputs matches the number of classes. It was chosen not to use convolution layers, like in the morphology- and combined-representation, because it was believed that there would be no greater benefit given how convolution works, and the size of the location-representation. An overview of the architecture of the model for the location-representation can be seen in figure 3.17.

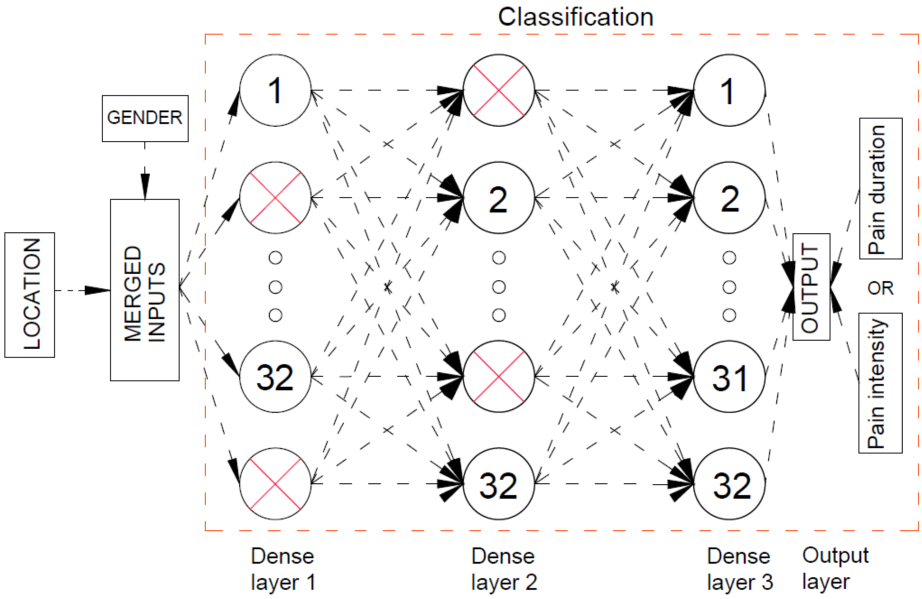


Figure 3.17: *Architecture of the location-representation model.*

Combined-representation models

The architecture of this model is nearly identical to that of the morphology-representation model, as seen in figure 3.16. The main difference can only be seen in the dimension of input for the pain map representation, where the input shape is $10 \times 118 \times 252$ to which 10 is the number of image layers per pain map. This is the result of the one-hot encoding done to the images as described in section 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 to ensure that the pain maps separated into a training, validation and test subsets randomly. The different subsets were respectively made up of 75%, 10% and 15% of the available pain maps. The data was separated to clearly distinguish between dataset and their purpose, and thereby prevent the same data from being used for training and e.g testing since this would give a wrong estimation of the model's performance. The training and validation subsets were used to train and optimize the models, to find the optimal weights with the back-propagation algorithm [41]. The test subset was for final testing of the model, to get an estimation of the ability to generalize "unseen" pain maps. Additionally, the models for the morphology- and combined-representation, the data was reshaped from a row vector, back into a matrix to retain their 2D structure of the pain map. The true classification labels were one-hot encoded to make the number of classes compatible according to the number of outputs in the output layer.

3.5 Optimization-process of the models

To reduce overfitting and improve general performance of the model, different methods were implemented and test. A grid search method was used to find initial hyperparameters for the models. For this method, a set of different hyperparameters were defined and tested using a

10-fold cross-validation, where the hyperparameters that resulted in the highest accuracy were chosen, and can be seen in appendix D. Learning rate, type of initializer, number of nodes, batch size and number of epochs were tested using this method. An additional manual optimization was then performed, where all the changes to the hyperparameters were based on the behavior of the validation set during training of the deep learning models. The development of the learning process was visually evaluated by plotting the accuracy and loss of both the training and validation subset.

According to number of the epoch when the models start to overfit, changes to learning rate, epochs, batch size, momentum, and a number of nodes were made if needed. The examples of the accuracy, loss plots, and overfitting is shown Fig. **??**.

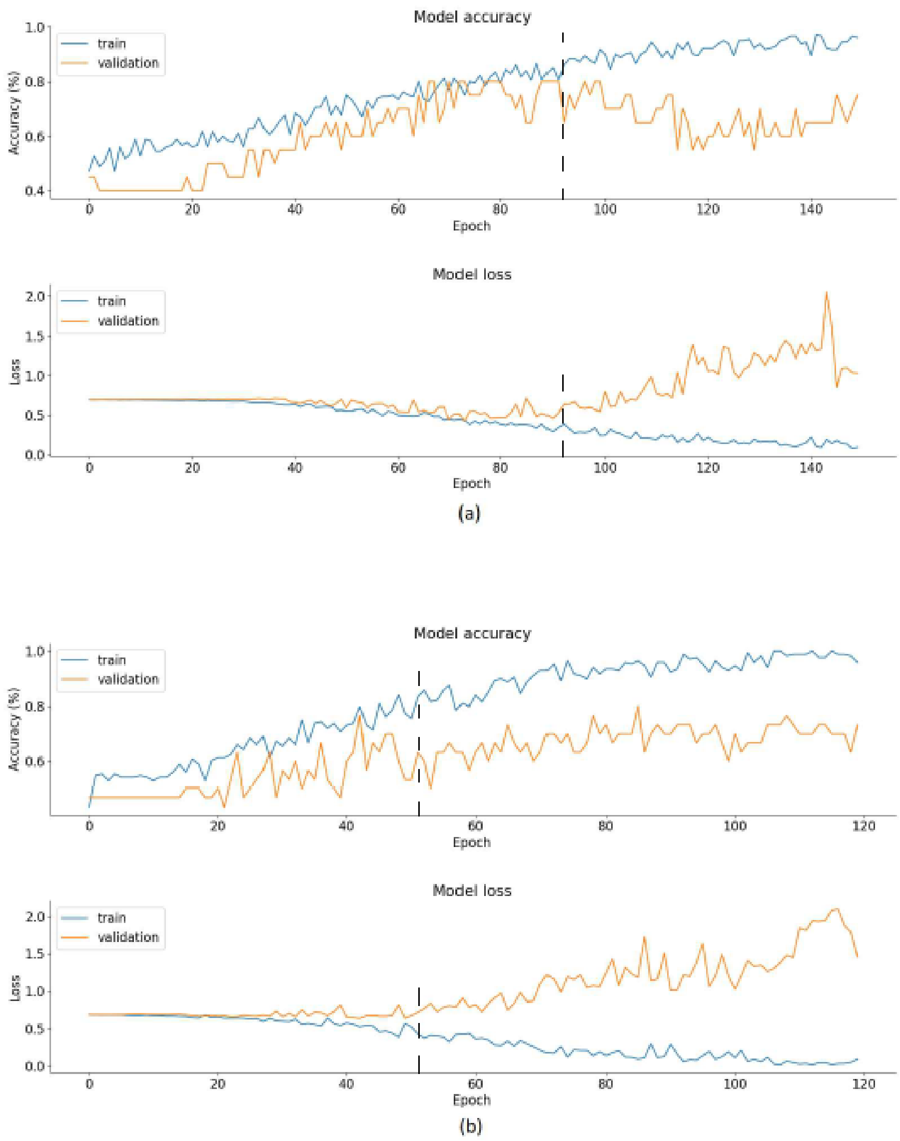


Figure 3.18: *Plots of the models accuracy, loss, and overfitting on training and validation datasets: (a) model of combined-representation according to the duration, (b) model of the morphology-representation according to the duration*

Activation function

For all models the activation function in all layers except the output layer was the ReLU function. This was chosen as this is seen as the most common activation function in more modern neural networks. The reason for its popularity is that it prevents the problem known as the vanishing or exploding gradient, typically seen when using sigmoid activation function in the hidden layers.[39] All output layer activation functions were chosen as sigmoid, because it was stated as the typical activation function for output classification problem [42].

Dropout

Dropout was implemented in the models due to the benefits described in section 2.6.3, where it is written that dropout should be used to reduce overfitting of the models. A dropout was then implemented within the hidden fully connected layers of all models, except the output layer, and set to randomly drop 50% of the nodes.

Bibliography

[1] KOOS. KOOS, 2016. URL http://www.koos.nu/index.html.

[2] 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.

[3] T.O. Smith, B.T. Drew, and Et.al. Knee orthoses for treating patellofemoral pain syndrome (review). 2015. doi: 10.1002/14651858.CD010513.pub2.

[4] 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.

[5] Amir Haim, Moshe Yaniv, and Et.al. Patellofemoral Pain Syndrome. *Knee Surg sports traumatol arthrose*, 451:223–228, 2006. ISSN 0009-921X. doi: 10.1007/s00167-013-2759-6. URL http://content.wkhealth.com/linkback/openurl?sid=WKPTLP:landingpage{&}an=00003086-200610000-00041.

[6] 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.

[7] Kay M. Crossley, Michael J. Callaghan, and Et.al. Patellofemoral pain. 2016. doi: 10.1136/bjsports-2015-h3939rep.

[8] Scott F Dye. Patellofemoral Pain Current Concepts: An Overview. *Sports Medicine and Arthroscopy Review*, 2001.

[9] Shellie A. Boudreau, Susanne Badsberg, and Et.al. Digital pain drawings: Assessing Touch-Screen Technology and 3D Body Schemas. 2016. doi: 10.1097/AJP.0000000000000230.

[10] 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.

[11] E. J. Dansie and D. C. Turk. Assessment of patients with chronic pain. 2013. doi: 10.1093/bja/aet124.

[12] Frederic H. Martini, Judi L. Nath, and Et.al. *Anatomy & Physology*. 2012.

[13] IASP. IASP Taxonomy, 2012.

[14] 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.

[15] R. F Schmidt. Nociception and Pain. In *Fundamentals of Sensory Physiology*. Springer Science & Business Media, 2 edition, 2013. ISBN 9783662011287.

[16] Emma Briggs. Understanding the experience and physiology of pain. 2010.

[17] Kay M. Crossley, Michael J. Callaghan, and Et.al. Patellofemoral pain. 2015. doi: 10.1136/bmj.h3939.

[18] Katherine Stabenow Dahab. Q angle. *Enccyclopedia of Sports Medicine*, 2011. doi: 10.4135/9781412961165.n415.

[19] Jo Nijs, Paul van Wilgen, and Et.al. How to explain central sensitization to patients with 'unexplained' chronic musculoskeletal pain: Practice guidelines. *Elsevier*, 2011. doi: 10.1016/j.math.2011.04.005.

[20] Dagfinn Matre and Stein Knardahl. 'Central sensitization' in chronic neck/shoulder pain. *Scandinavian journal of pain*, 2012. doi: 10.1016/j.sjpain.2012.04.003.

[21] Muhammed B. Yunus. Role of central sensitization in symptoms beyond muscle pain, and the evaluation of a patient with widespread pain. *Elsevier*, 2007. doi: 10.1016/j.berh.2017.03.006.

[22] Haavard Oesteraas, Tom Arild Torsensen, and Et.al. Treatment of chronic bilateral knee pain without objective clinical findings: Central sensitization. A case report. *Advances in Physiotherapy*, (12:2):95–99, 2010. doi: 10.3109/14038191003642385.

[23] Tom G. Mayer, Randy Neblett, and Et.al. The development and psychometric validation of the central sensitization inventory. *World Institute of Pain*, 2011. doi: 10.1111/j.1533-2500.2011.00493.x.

[24] Jarred Younger and Sean Mackey. Pain outcomes: A brief review of instruments and techniques. 2009.

[25] Kanishka M. Ghosh and David J. Deehan. Soft tissue knee injuries. In Mark Wilkinson, editor, *Orthopaedic surgery: lower limb*. Elsevier, 2010.

[26] Christoph Pieh, Jurgen Altmeppen, and Et.al. Gender differences in outcomes of a multimodal pain management program. *Elsevier*, 2012. doi: 10.1016/j.pain.2011.10.016.

[27] 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.

[28] Mathias Haefeli and Achim Elfering. Pain assessment. 2005. doi: 10.1007/s00586-005-1044-x.

[29] Ewa M. Roos and L. Stefan Lohmander. The knee injury and osteoarthritis outcome score (KOOS): from joint injury to osteoarthritis. *Bio Med Central*, 2003.

[30] 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.

[31] Geoffrey D. Schott. The cartography of pain: The evolving contribution of pain maps. 2010. doi: 10.1016/j.ejpain.2009.12.005.

[32] D. W. Elson, S. Jones, and Et.al. The photographic knee pain map: Locating knee pain with an instrument developed for diagnostic, communcation and research purposes. 2010. doi: 10.1016/j.knee.2010.08.012.

[33] Yann LeCun, Yoshua Bengio, and Et.al. 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`.

[34] Mads Nielsen. *Den digitale revolution – fortællinger fra datalogiens verden*. Datalogisk Institut, 2010. ISBN 978-87-981270-5-5.

[35] 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`.

[36] 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`.

[37] Jacopo Acquarelli, Twan van Laarhoven, and Et.al. 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`.

[38] Alaa Ali Hameed, Bekir Karlik, and Et.al. 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{%}3Dihub`.

[39] Ian Goodfellow, Yoshua Bengio, and Et.al. *Deep Learning*. MIT Press, 2016. URL `http://www.deeplearningbook.org`.

[40] Mohri Mehryar, Rostamizadeh Afshin, and Et.al. *Foundations of Machine Learning*. 2012. ISBN 9780262018258. URL `https://ebookcentral.proquest.com/lib/aalborguniv-ebooks/reader.action?docID=3339482{&}ppg=17{#}`.

[41] 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`.

[42] Richard Duda, Peter Hart, and Et.al. *Pattern Classification*. Second edi edition, 2000. ISBN 9780471056690.

[43] Yann LeCun, Léon Bottou, and Et.al. 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`.

[44] 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`.

[45] Sebastian Raschka. Single-Layer Neural Networks and Gradient Descent. 2016. URL http://sebastianraschka.com/Articles/2015{_}singlelayer{_}neurons.html.

[46] Sixin Zhang, Anna Choromanska, and Et.al. Deep learning with Elastic Averaging SGD. 2014. ISSN 10495258. URL http://arxiv.org/abs/1412.6651.

[47] 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.

[48] 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.

[49] 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.

[50] François Chollet. Keras Documentation, 2015. URL https://keras.io.

[51] Nitish Srivastava, Geoffrey Hinton, and El.al. 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.

[52] Karl Gustafson. Assigning Initial Weights in Feedforward Neural Networks. 1998. URL https://www.sciencedirect.com/science/article/pii/S1474667017419380.

[53] Vincent Dumoulin and Francesco Visin. A guide to convolution arithmetic for deep learning. 2016. ISSN 16113349. doi: 10.1051/0004-6361/201527329. URL http://arxiv.org/abs/1603.07285.

[54] Aglance Solutions. Visual insight for clinical reasoning – Navigate Pain, 2015. URL http://www.navigatepain.com/.

[55] David Money Harris and Sarah L. Harris. Sequential Logic Design. In *Digital design and computer architecture*. Elsevier, 2012. ISBN 9780123978165.

[56] 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.

[57] Intel. Processor specs. URL https://ark.intel.com/products/88967/Intel-Core-i7-6700HQ-Processor-6M-Cache-up-to-3{_}50-GHz.

[58] Nvidia. GPU specs. URL https://www.geforce.com/hardware/notebook-gpus/geforce-gtx-970m/specifications.

Appendix A

Knee injury and Ostearthritis Outcome Score (KOOS) [1]

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 <u>one</u> 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
☐	☐	☐	☐	☐

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

Never	Rarely	Sometimes	Often	Always
☐	☐	☐	☐	☐

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

Never	Rarely	Sometimes	Often	Always
☐	☐	☐	☐	☐

S4. Can you straighten your knee fully?

Always	Often	Sometimes	Rarely	Never
☐	☐	☐	☐	☐

S5. Can you bend your knee fully?

Always	Often	Sometimes	Rarely	Never
☐	☐	☐	☐	☐

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
☐	☐	☐	☐	☐

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

None	Mild	Moderate	Severe	Extreme
☐	☐	☐	☐	☐

Pain

P1. How often do you experience knee pain?

Never	Monthly	Weekly	Daily	Always
☐	☐	☐	☐	☐

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
☐	☐	☐	☐	☐

P3. Straightening knee fully

None	Mild	Moderate	Severe	Extreme
☐	☐	☐	☐	☐

P4. Bending knee fully

None	Mild	Moderate	Severe	Extreme
☐	☐	☐	☐	☐

P5. Walking on flat surface

None	Mild	Moderate	Severe	Extreme
☐	☐	☐	☐	☐

P6. Going up or down stairs

None	Mild	Moderate	Severe	Extreme
☐	☐	☐	☐	☐

P7. At night while in bed

None	Mild	Moderate	Severe	Extreme
☐	☐	☐	☐	☐

P8. Sitting or lying

None	Mild	Moderate	Severe	Extreme
☐	☐	☐	☐	☐

P9. Standing upright

None	Mild	Moderate	Severe	Extreme
☐	☐	☐	☐	☐

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
☐	☐	☐	☐	☐

A2. Ascending stairs

None	Mild	Moderate	Severe	Extreme
☐	☐	☐	☐	☐

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	Mild	Moderate	Severe	Extreme
☐	☐	☐	☐	☐

A4. Standing

None	Mild	Moderate	Severe	Extreme
☐	☐	☐	☐	☐

A5. Bending to floor/pick up an object

None	Mild	Moderate	Severe	Extreme
☐	☐	☐	☐	☐

A6. Walking on flat surface

None	Mild	Moderate	Severe	Extreme
☐	☐	☐	☐	☐

A7. Getting in/out of car

None	Mild	Moderate	Severe	Extreme
☐	☐	☐	☐	☐

A8. Going shopping

None	Mild	Moderate	Severe	Extreme
☐	☐	☐	☐	☐

A9. Putting on socks/stockings

None	Mild	Moderate	Severe	Extreme
☐	☐	☐	☐	☐

A10. Rising from bed

None	Mild	Moderate	Severe	Extreme
☐	☐	☐	☐	☐

A11. Taking off socks/stockings

None	Mild	Moderate	Severe	Extreme
☐	☐	☐	☐	☐

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

None	Mild	Moderate	Severe	Extreme
☐	☐	☐	☐	☐

A13. Getting in/out of bath

None	Mild	Moderate	Severe	Extreme
☐	☐	☐	☐	☐

A14. Sitting

None	Mild	Moderate	Severe	Extreme
☐	☐	☐	☐	☐

A15. Getting on/off toilet

None	Mild	Moderate	Severe	Extreme
☐	☐	☐	☐	☐

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
☐	☐	☐	☐	☐

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

None	Mild	Moderate	Severe	Extreme
☐	☐	☐	☐	☐

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
☐	☐	☐	☐	☐

SP2. Running

None	Mild	Moderate	Severe	Extreme
☐	☐	☐	☐	☐

SP3. Jumping

None	Mild	Moderate	Severe	Extreme
☐	☐	☐	☐	☐

SP4. Twisting/pivoting on your injured knee

None	Mild	Moderate	Severe	Extreme
☐	☐	☐	☐	☐

SP5. Kneeling

None	Mild	Moderate	Severe	Extreme
☐	☐	☐	☐	☐

Quality of Life

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

Never	Monthly	Weekly	Daily	Constantly
☐	☐	☐	☐	☐

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

Not at all	Mildly	Moderately	Severely	Totally
☐	☐	☐	☐	☐

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

Not at all	Mildly	Moderately	Severely	Extremely
☐	☐	☐	☐	☐

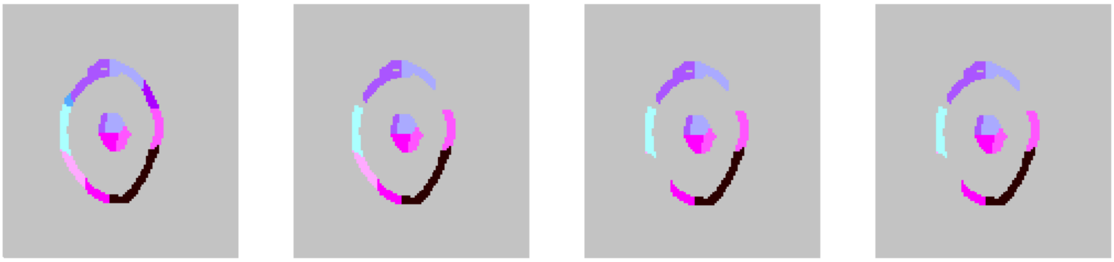
Q4. In general, how much difficulty do you have with your knee?

None	Mild	Moderate	Severe	Extreme
☐	☐	☐	☐	☐

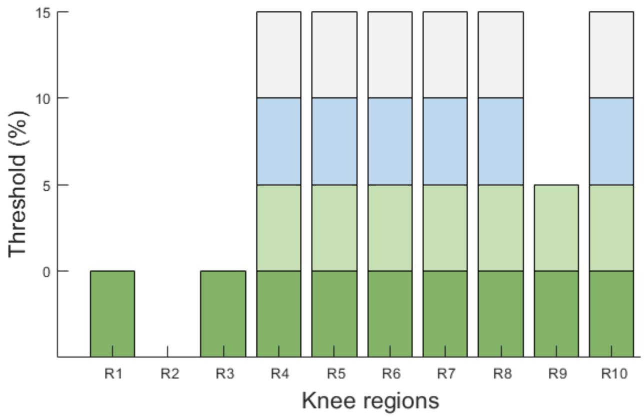
Thank you very much for completing all the questions in this questionnaire.

Appendix B

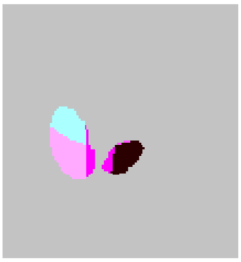
Threshold analysis



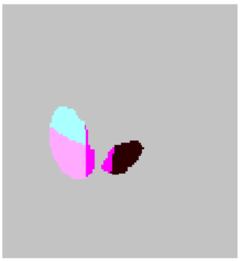
(a) (b) (c) (d)



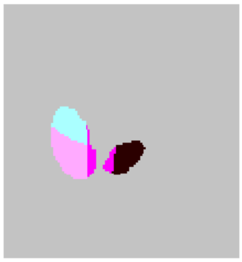
(e) Active regions



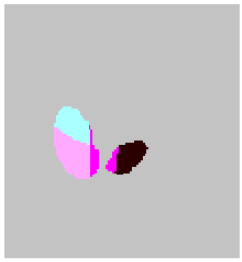
(a)



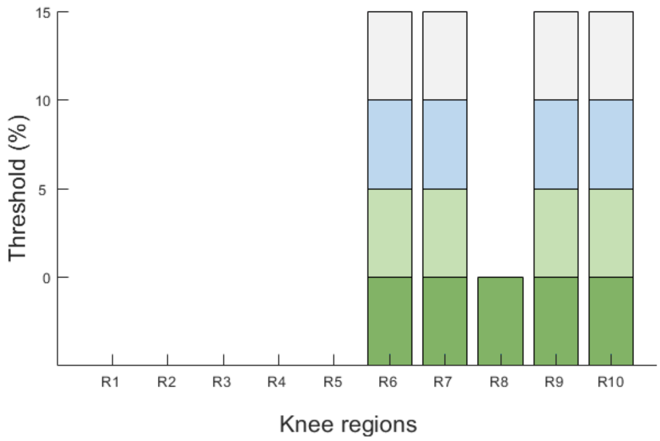
(b)



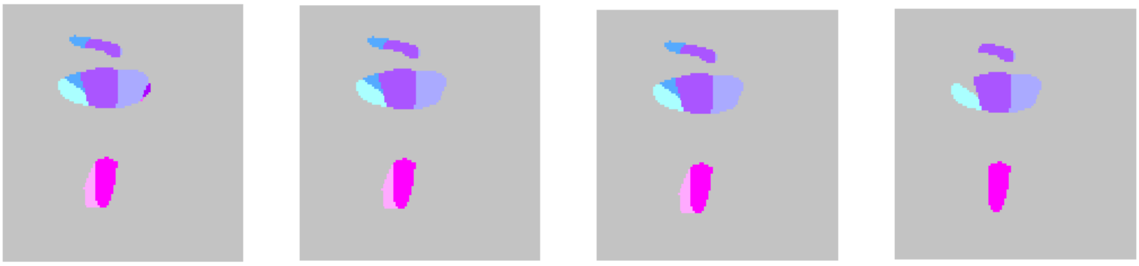
(c)



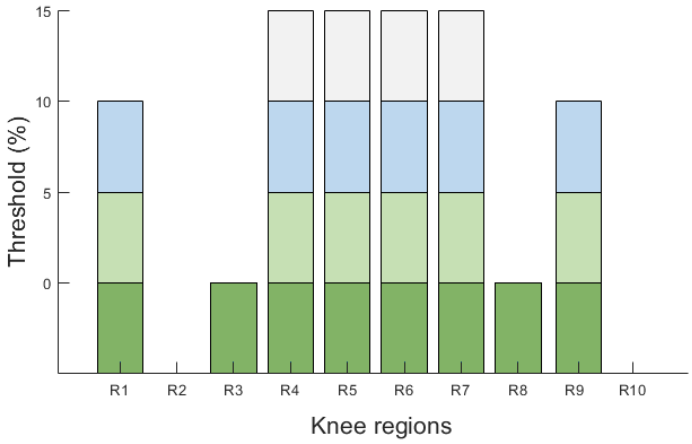
(d)



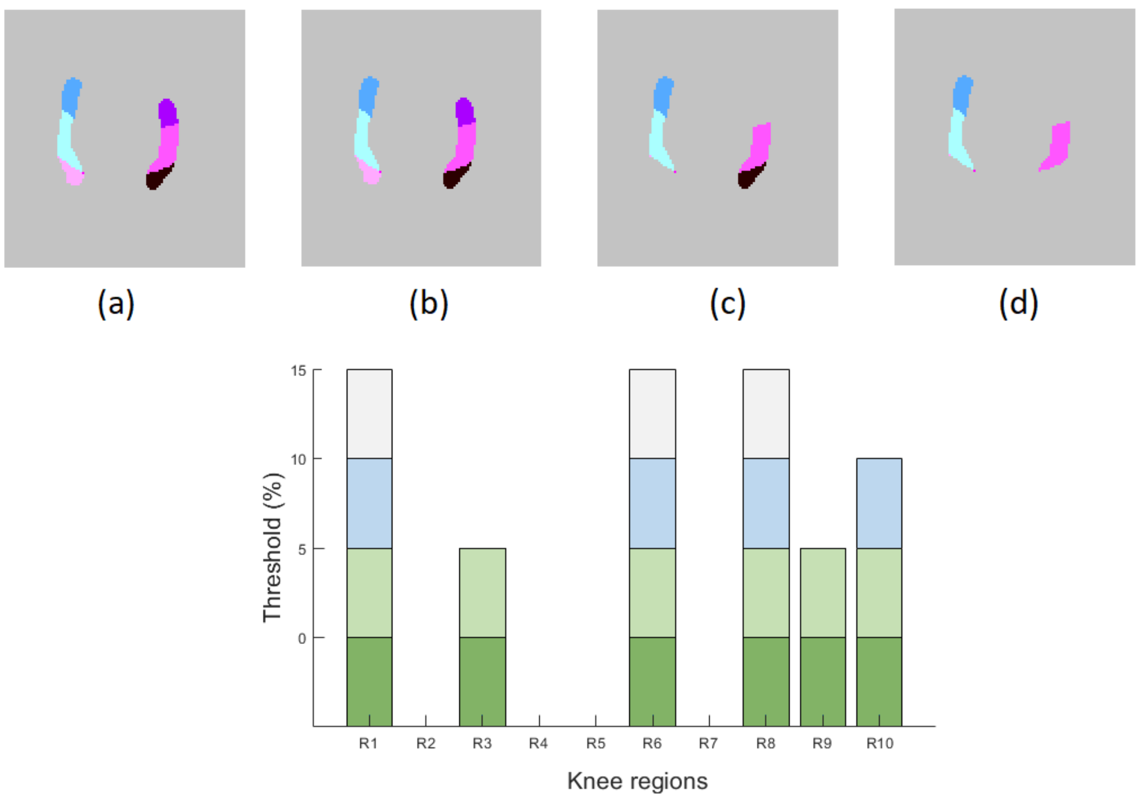
(e) Active regions



(a) (b) (c) (d)



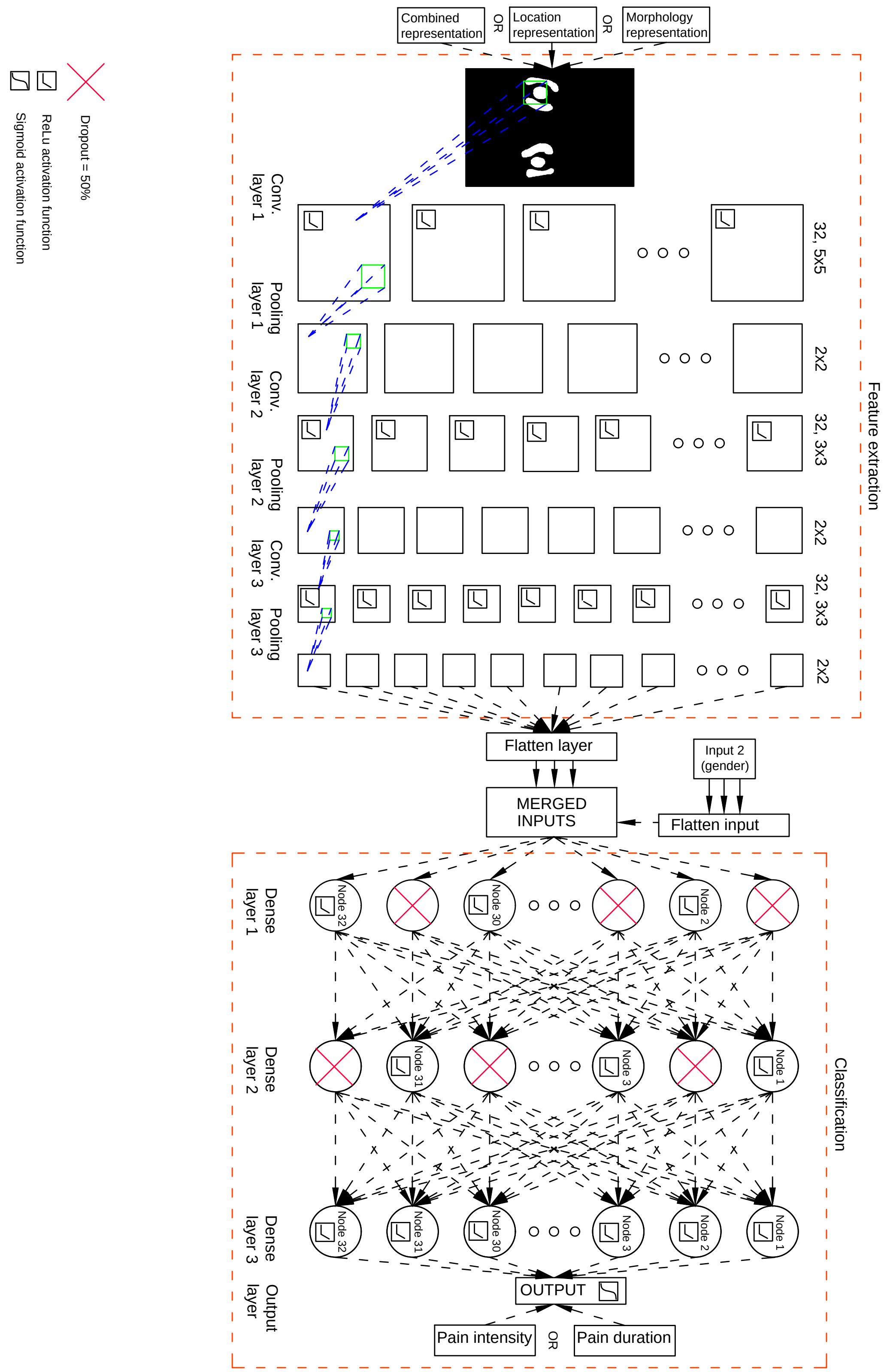
(e) Active regions



(e) Active regions

Appendix C

Architecture of the deep learning models



Dropout = 50%
ReLu activation function
Sigmoid activation function
Combined representation
OR
Location representation
OR
Morphology representation
Feature extraction
Conv. layer 1
Pooling layer 1
Conv. layer 2
Pooling layer 2
Conv. layer 3
Pooling layer 3
32, 5x5
2x2
32, 3x3
2x2
32, 3x3
2x2
Flatten layer
Input 2 (gender)
Flatten input
MERGED INPUTS
Classification
Dense layer 1
Dense layer 2
Dense layer 3
Output layer
Node 32
Node 31
Node 30
Node 3
Node 2
Node 1
OUTPUT
Pain intensity
OR
Pain duration

Appendix D

Optimization parameters

Morphology-representation						
	Pain duration			Pain intensity		
	Avg. Accuracy	± SD	Chosen parameter	Avg Accuracy	± SD	Chosen parameter
Learning rate:						
0.1	58.50	10.50		68.07	9.27	X
0.01	60.50	7.22	X	60.24	10.10	
0.001	56.00	8.00		64.46	9.20	
Kernal inintializer:						
normal	45.00	7.75		62.65	9.74	
glorot_normal	48.50	9.50		62.65	9.74	X
glorot_uniform	58.50	5.50	X	62.04	7.97	
Neurons:						
16	54.50	13.87		64.46	8.61	X
32	60.50	8.20	X	62.05	8.89	
64	60.50	2.69		60.24	9.29	
Epochs, Batch size:						
100, 10	XX	XX		64.46	8.19	
100, 20	XX	XX		62.65	9.74	
100, 30	XX	XX		62.65	9.74	
120, 10	XX	XX		65.06	8.81	
120, 20	XX	XX		62.65	9.74	
120, 30	XX	XX		62.65	9.74	
140, 10	XX	XX		69.88	9.49	X
140, 20	XX	XX		59.64	9.54	
140, 30	XX	XX		62.65	9.74	

Table D.1: *Results from the grid search for the morphology-representation*

Location-representation

	Pain duration			Pain intensity		
	Avg. Accuracy	± SD	Chosen parameter	Avg Accuracy	± SD	Chosen parameter
Learning rate:						
0.1	56.08	13.35		63.92	12.37	
0.01	56.08	13.35	X	63.92	12.37	X
0.001	56.08	13.35		63.92	12.37	
Kernal inintializer:						
normal	56.08	13.35		63.92	12.37	
glorot_normal	56.08	13.35		63.92	12.37	
glorot_uniform	56.08	13.35	X	63.92	12.37	X
Neurons:						
8	56.08	13.35		63.92	12.37	
16	56.08	13.35	X	63.92	12.37	X
32	56.08	13.35		63.92	12.37	
Epochs, Batch size:						
100, 10	56.08	13.35		63.92	12.37	
100, 20	56.08	13.35		63.92	12.37	
100, 30	56.08	13.35		63.92	12.37	
120, 10	56.08	13.35		63.92	12.37	
120, 20	56.08	13.35	X	63.92	12.37	X
120, 30	56.08	13.35		63.92	12.37	
140, 10	52.12	11.02		63.92	12.37	
140, 20	56.08	13.35		63.92	12.37	
140, 30	56.08	13.35		63.92	12.37	

Table D.2: *Results from the grid search for the location-representation*

Combined-representation

	Pain duration			Pain intensity		
	Avg. Accuracy	± SD	Chosen parameter	Avg Accuracy	± SD	Chosen parameter
Learning rate:						
0.1	66.16	8.52	X	60.98	14.67	
0.01	52.02	9.79		60.98	15.13	
0.001	54.04	10.67		62.80	14.35	X
Kernal inintializer:						
normal	62.63	8.45		61.59	15.35	
glorot_normal	63.63	6.45		61.59	15.35	
glorot_uniform	70.20	8.49	X	61.59	15.35	X
Neurons:						
8	67.17	6.49	X	61.59	15.35	X
16	60.10	10.03		61.59	15.35	
32	62.12	10.92		61.59	15.35	
Epochs, Batch size:						
100, 10	61.11	4.85		61.59	15.35	
100, 20	58.08	8.81		61.59	15.35	
100, 30	61.61	4.32		61.59	15.35	
120, 10	63.13	5.48		61.59	15.35	
120, 20	62.12	10.57		61.59	15.35	X
120, 30	68.68	6.90	X	61.59	15.35	
140, 10	60.61	10.30		61.59	15.35	
140, 20	64.65	8.75		61.59	15.35	
140, 30	64.65	8.34		61.59	15.35	

Table D.3: *Results from the grid search for the combined-representation*