Visual assessments of Postural Orientation Errors using ensembles of Deep Neural Networks

Filip Kronström

Master's Thesis
April 2021

LUND UNIVERSITY

Faculty of Engineering
Centre for Mathematical Sciences
Mathematical Statistics

CENTRUM SCIENTIARUM MATHEMATICARUM

April 2021

Mathematical Statistics
Centre for Mathematical Sciences
Lund University
Box 118, SE-221 00 Lund, Sweden

http://www.maths.lth.se/

Contents

Abstract

Injuries to the Anterior Cruciate Ligament (ACL) are severe and common among the physically active young to middle aged population. After suffering from such an injury, the patient typically face a lengthy rehabilitation process. Usually, it takes 1-2 years before an injured knee returns to pre-injury performance, if that is ever achieved. The risk of re-injury is high and is increased by early return to sports. One measure which has been suggested as an indicator of the increased risk of re-injury, and hence could work as an indicator of when to return to normal activity, is altered postural orientation. The postural orientation describes the positions of different body parts in relation to each other and the surroundings. Assessment of this is a time consuming task requiring human experts trained to find such alterations. This thesis propose a method to automate this task by analyzing videos recorded with a regular video camera, e.g. a mobile phone.

The proposed method uses well established deep learning techniques, in this case HRNet with DARK-pose, to extract body part positions from each video frame. Deep learning based models are trained in a supervised fashion to classify the sequences of extracted keypoints. Models trained to perform according to different metrics were combined in ensembles classifying the quality of the postural orientation on an ordinal scale from 0 (Good), via 1 (Fair), to 2 (Poor).

We evaluated the method on four different segment-specific Postural Orientation Errors (POEs) when the patient performed a single leg squat. The different POEs were trunk, pelvis, femoral valgus, and Knee Medial-to-Foot Position (KMFP). For femoral valgus and trunk a classification accuracy of 82.3% and 80.0%, respectively, was achieved. The corresponding number for KMFP was 90.3%, but this data was heavily imbalanced. The pelvis was the most difficult to analyze resulting in an accuracy of 73.3%.

The most important contribution of this thesis is to provide a foundation and a number of insights of what is needed before introducing a method like this for clinical use.

Populärvetenskaplig sammanfattning

Skador på det främre korsbandet (ACL) är allvarliga och vanliga bland fysiskt aktiva och ofta unga personer. Efter att ha drabbats av en sådan skada står patienten inför en lång och svår rehabiliteringsprocess. Vanligtvis tar det 1-2 år innan ett skadat knä återgår till den nivå det hade innan skadan, om denna någonsin nås. Risken för nya skador är hög och höjs ytterligare av att återvända till idrott för tidigt. Ett mått som föreslagits visa på förhöjd skaderisk, och alltså skulle kunna fungera som en indikation på när det är möjligt att återgå till normal aktivitet, är postural orientering. Denna orientering beskriver förmågan att upprätthålla kroppsdelars positioner i förhållande till varandra och omgivningen. Bedömning av detta är en tidskrävande uppgift som kräver tränade experter. Denna uppsats föreslår en metod för att automatisera denna process baserat på filmer inspelade med vanliga videokameror, till exempel mobiltelefoner.

Den föreslagna metoden använder etablerade djupinlärningstekniker, HRNet med DARK-pose, för att hitta kroppsdelar för varje bildruta i filmerna. Andra djupinlärningsmodeller tränas för att analysera sekvenserna som beskriver kroppsdelarnas positioner. Flera modeller kombineras för klassificera kvalitén på rörelserna på en skala från 0 (Bra), via 1 (Ganska bra), till 2 (Dålig).

Vi utvärderade metoden för fyra olika segmentspecifika avvikelser i postural orientering, kallade Postural Orientation Errors (POEs). De POEs som utvärderades relaterade till positionen av bål, höft, lår och knä. För lår och bål uppnåddes en klassificerings-noggrannhet på 82.3% respektive 80.0%. Motsvarande siffra för knäet var 90.3%, men denna data var kraftigt obalanserad. Höften var svårast att analysera och en noggrannhet på 73.3% uppnåddes.

De viktigaste bidragen från denna uppsats är att visa att detta är ett problem där maskininlärning kan användas, utvecklingen av en grund att bygga vidare på samt ett antal insikter kring vad som behövs för att använda systemet kliniskt.

Acknowledgements

The computations were enabled by resources provided by the Swedish National Infrastructure for Computing (SNIC) at Chalmers Centre for Computational Science and Engineering (C3SE) partially funded by the Swedish Research Council through grant agreement no. 2018-05973.

Regarding thank yous I would like to begin by giving a big one to Eva Ageberg and Mark Creaby for your ideas and insights. Secondly I would like to thank Jenny Älmqvist Nae for introducing this field to me, for explaining concepts which were very alien to me six months ago, and for assessing so many videos. Finally I would like to give a big thank you to Andreas Jakobsson for your enthusiasm and ideas throughout the project, for finding so many missing commas in this text, and for being a source of inspiration.

Acronyms

ACL Anterior Cruciate Ligament

AI Artificial Intelligence

CNN Convolutional Neural Network

COCO Microsoft Common Objects in Context dataset

CORAL COnsistent RAnk Logits

COTE Collective Of Transformation-based Ensembles

DARK Distribution-Aware coordinate Representation of Key-point

DNN Deep Neural Network

GAP Global Average Pooling

GPU Graphics Processing Unit

Grad-CAM Gradient-weighted Class Activation Mapping

HIVE-COTE Hierarchical Vote Collective of Transformation-based Ensembles

HPE Human Pose Estimation

HRNet High-Resolution Net

KMFP Knee Medial-to-Foot Position

POE Postural Orientation Error

ReLU Rectified Linear Unit

SLS Single Leg Squat

SOTA State of the Art

TSC Time Series Classification

UCR University of California, Riverside

XCM Explainable Convolutional Neural Network for Multivariate Time Series Classification

Chapter 1

Introduction

Among the physically active, young to middle aged population injuries to, e.g., knee or hip are common. In a study by Thorborg et al., 49% of the questioned sub-elite football players in Denmark reported they had issues with hip and/or groin pain during the previous season [61]. Among serious injuries, rupture of the Anterior Cruciate Ligament (ACL) is one of the more severe and common and the incidence rate among athletes is about 1.5 per 10000 athlete-exposure. The injury rate for women is 1.7 times higher than it is for men [44]. Treatment of such injuries is a debated subject and whether surgery yields better results is difficult to say [32, 43]. Independently of the treatment, the patient needs to undergo a rehabilitation process, potentially for as long as 2 years [47]. Apart from the rehabilitation, an injury may also lead to long term physical impairments, such as joint instability [2] and increased risk of knee osteoarthritis (OA) [40], as well as increased risk of depression [14] and re-injury [51].

The ACL, seen in Figure 1.0.1, is one of the ligaments connecting the femur (thigh bone) with the tibia (shin bone) and is a key structure for providing stability in the knee [17]. Injuries to the ACL commonly occurs without any direct contact with e.g. other athletes. Instead a typical injury mechanism is a sudden change of direction or velocity while the knee bears weight [68].

1.1 Postural Orientation Errors

The ability to uphold the alignment of body segments, both in relation to each other and the surroundings, is called postural orientation [24]. Altered postural orientation, for example Postural Orientation Errors (POEs), has been seen to increase the risk of suffering new injuries [23]. Hence this is suggested to be an important measure during rehabilitation and before returning to sports. 3D motion capture systems are considered to be the ”gold standard” for measurement of postural orientation. This is, however, an expensive and time consuming procedure, requiring specific resource in terms of laboratories and experts to perform the measurements. A potential alternative, more suitable for clinical use, is visual assessments of 2D measurements (i.e., videos) [4].

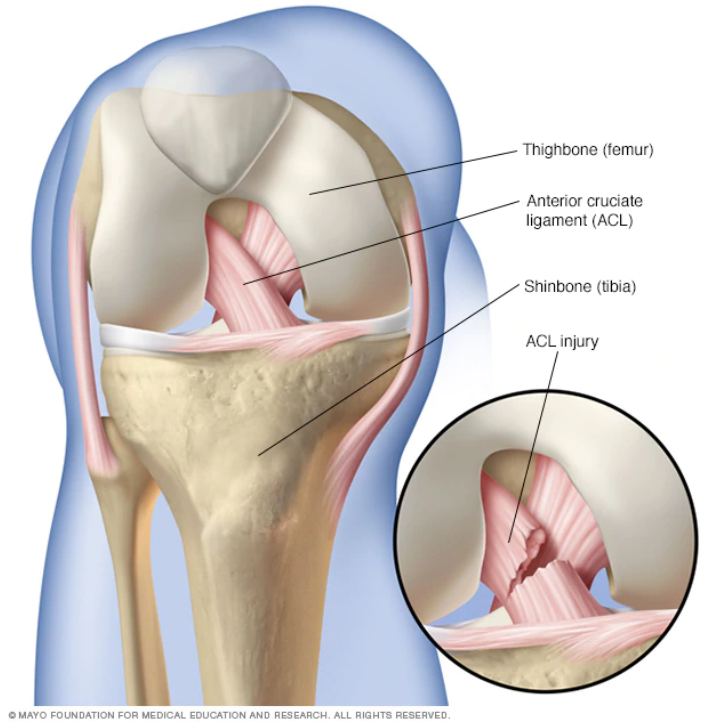


Figure 1.0.1: Illustration of the position of the ACL in the knee. Image from [13].

There is not one established method for visual assessments of postural orientation, but Nae et al. presented a test battery where up to six segment specific POEs were assessed for five different functional tasks. Each POE was scored on an ordinal scale from 0 to 2 [45, 46]. A detailed explanation of the POE-task combinations can be found in Appendix A. This scoring system is the foundation of this thesis, but, in this work, we only study the deviation of trunk-, deviation of pelvis-, femoral valgus-, and KMFP-POEs for the Single Leg Squat (SLS) task. The segment of the body assessed for each of these POEs are shown in Figure 1.1.1 and the criteria for the assessments are presented in Table 1.1.1. A description of the SLS task can be found below. When a patient is assessed four-five repetitions are performed, each being scored according to Table 1.1.1. A total segment-specific score is then calculated as the median of these repetitions.

Single-leg squat, SLS

The subject performs a squat standing on one leg to a knee angle of approximately 60° before returning to extension. The exercise is repeated five times and the entire movement is used to assess the POEs [4].

Although visual assessments based on 2D videos are a time and resource efficient way of evaluating POEs compared to 3D motion capture systems, it is still a time consuming method. Automating such assessments could hopefully improve the quality of the rehabilitation, both by allowing the physiotherapist to spend more time with the patient, but also by helping to direct the focus of the training to a specific segment in need of improvement. It could also give more accurate and less biased results compared to human assessments.

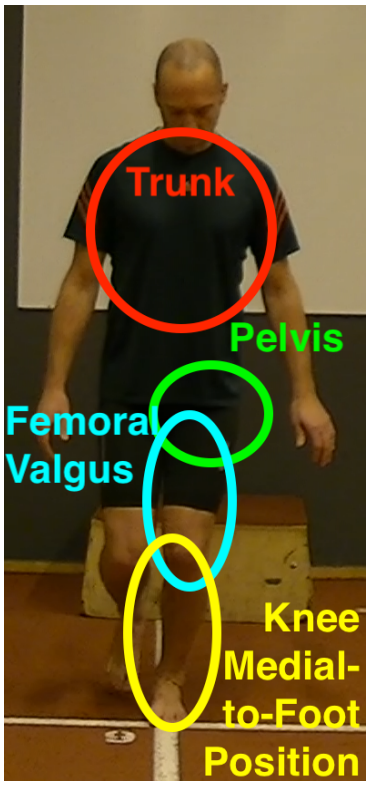


Figure 1.1.1: Segments of the body assessed for the different POEs.

Table 1.1.1: Descriptions for the visual assessment of segment specific POEs evaluated in this thesis. Table taken from [4].

Segment-specific POEs	**Scoring of 0: Good (no POE)**	**Scoring of 1: Fair (minor POE)**	**Scoring of 2: Poor (major POE)**
Deviation of trunk in any plane	The absence of a trunk position into forward lean, lateral lean and/or rotation indicates no POE	A slight position of the trunk into forward lean, lateral lean and/or rotation indicates minor POE	A clear position of the trunk into forward lean, lateral lean and/or rotation indicates major POE
Deviation of pelvis in any plane	The absence of pelvis into lateral deviation, pelvic tilt and/or rotation of pelvis respectively indicates no POE	A slight position of the pelvis into lateral deviation, pelvic tilt and/or rotation of pelvis respectively indicates minor POE	A clear position of the pelvis into lateral deviation, pelvic tilt and/or rotation of pelvis respectively indicates major POE
Femoral valgus	The absence of femoral valgus indicates no POE	A slight position of femoral valgus indicates minor POE	A clear position of femoral valgus indicates major POE
Knee Medial-to-Foot Position	Mid-point of patella is in line with or lateral to the second toe	Mid-point of patella is placed medial to the second toe	Mid-point of patella is clearly placed medial to the big toe

1.2 Related work

A study by the National Board of Health and Welfare in Sweden from 2019 show that the use of Artificial Intelligence (AI) systems in the Swedish healthcare is still fairly limited. At the date of the report such systems were used for 59 applications, of which 27 were based on some machine learning approach. The learning based methods were most commonly used for diagnosis and decision support, mainly based on different types of image analysis [58]. Within the research community the situation is different, here machine learning applied in a medical field is very widespread, for instance in medical imaging fields such as radiology and ultrasound, but also electrocardiograms (EKGs), anamnesis, and mental health. The introduction of wearable devices and cheaper IoT devices are deemed to be important factors for the digitalization of healthcare [62].

Machine learning has also been used in the field of physiotherapy and rehabilitation. Kianifar et al. estimated joint poses from IMU data using extended Kalman filters. Based on these joint positions they classified the quality of SLS movements using Support Vector Machines, Linear Multinomial Logistic Regression, and Decision Trees [30]. Liao et al. used combinations of Gaussian Mixture Models and Deep Neural Networks (DNNs) to calculate quality scores for movements during rehabilitation based on 3D joint positions captured with a Microsoft Kinect [36].

1.3 Focus of this work

This work should be seen as a proof of concept showing that these kinds of assessments can be automated, allowing physiotherapists to help patients in a more efficient way. Due to limitations in time and labeled data only four POEs for the SLS task was evaluated. After some initial studies it was decided to use 2D body joint positions, i.e., 3D reconstruction from the videos was not evaluated.

1.4 Thesis organization

After this first chapter, where the background and rationale for this thesis has been presented, some deep learning background is introduced in Chapter 2. Chapters 3 and 4 present methods for estimating human body joints and classifying time series, both important for this work. The methods developed as part of this thesis is provided in Chapter 5. Finally, the results are presented and discussed in Chapter 6. Chapter 7 summarizes the work and suggests improvements for the future.

Chapter 2

Background - Deep learning

A supervised machine learning problem can be described as finding a mapping between some input and output data, e.g. an image and a category, based on labeled input-output combinations. The idea with such methods is that a mapping found for the available data also should represent unseen data of the same type, i.e. it should generalize. To be able to get a measure of this generalization the available data is commonly divided into two parts, training data and test data. The training data is used to find the mapping and the test data is used to evaluate how well it performs on unseen data [8].

This chapter gives a brief introduction to a special type of machine learning called deep learning, which forms the basis of this work. The evaluation metrics used are also presented along with the problem of explainability in deep learning and a method to achieve ordinal classification.

2.1 Evaluation metrics

To be able to evaluate and compare models some evaluation metrics are needed. Table 2.1.1 shows four common classification metrics and the way they are calculated from the quantities defined in Definition 1.

Definition 1. **TP, True positives:** Correctly classified positive samples
FP, False positives: Incorrectly classified positive samples
TN, True negatives: Correctly classified negative samples
FN, False negatives: Incorrectly classified negative samples

As can be seen in Table 2.1.1, the F1 score is the harmonic mean of the precision and the recall. For problems with imbalanced data this might give a better idea of the actual performance compared to the accuracy. Precision, recall, and F1 are for binary classifications. When used in a multiclass setting they are computed in a one-vs-all fashion for all classes and then combined in some way. In this work they are macro averaged, i.e. calculated as the average of all scores. The macro F1 is calculated from

Table 2.1.1: Evaluation metrics using quantities in Definition 1.

Precision	$\frac{\mathrm{TP}}{\mathrm{TP+FP}}$
Recall	$\frac{\mathrm{TP}}{\mathrm{TP+FN}}$
F1 score	$2\frac{\mathrm{Precision}\cdot\mathrm{Recall}}{\mathrm{Precision+Recall}}$
Accuracy	$\frac{\mathrm{TP+TN}}{\mathrm{TP+FP+TN+FN}}$

the individual F1 scores according to

$$F1_{mac} = \frac{1}{N}\sum_{i=1}^{N} F1_i. \tag{2.1}$$

Another way to present the result of a classification task is using the confusion matrix. This is a matrix where the columns corresponds to the predicted classes and the rows to the correct classes. Hence, this metric shows what kind of errors the model makes. An example of a confusion matrix is shown in Figure 2.1.1. All entries on the main diagonal are correctly classified; in this example it can, for instance, be seen how 43 1s are predicted as 0s.

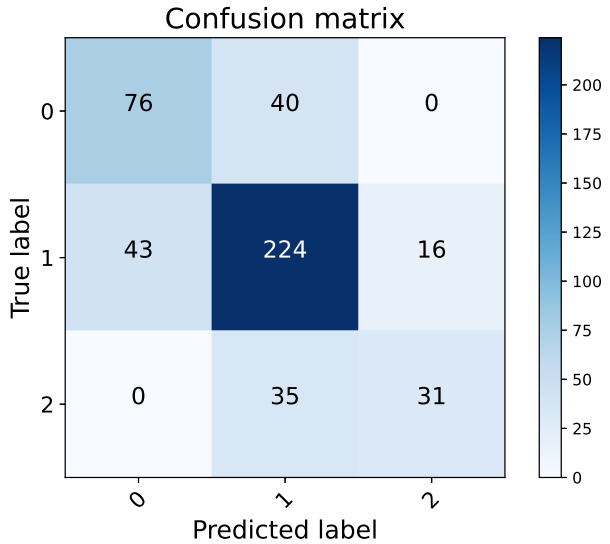


Figure 2.1.1: An example of a confusion matrix. The entries on the diagonal are correctly classified, while the position of an off-diagonal entry shows what kind of error has been made. The true class is given by the row and the predicted class by the column.

2.2 Historical background of deep learning

In 1943, McCulloch and Pitts [41] presented a mathematical model of a neuron which at the time had limited capabilities (e.g. it did not learn), but lay the foundations for much of what today is considered to be deep learning. Ivakhnenko and Lapa [28] introduced what would later be called deep learning with the first multi-layered network in 1965. The

first convolutional network was introduced by Fukushima in 1980 [20]. A few years later, in 1989, LeCun et al. [34] showed it possible to train such networks with backpropagation and illustrated their effectiveness for computer vision problems. In 2009 Raina et al. [53] suggested that DNNs could efficiently be trained on Graphics Processing Units (GPUs). Krizhevsky et al. [33] used this when they with AlexNet proved it possible to train deeper networks which also greatly outperformed models of the time at computer vision tasks. Since then deep learning based methods has been adopted in various fields, such as computer vision, natural language processing, and even autonomous vehicles [48].

2.3 Deep Neural Networks

DNNs are combinations of linear and non-linear functions trained to approximate some other, potentially very complicated, function. The output of the network is formed as $f(x) = f_n \circ f_{n-1} \circ ... \circ f_1 \circ f_0(x)$ resulting in the layer terminology since the output from one function is passed as input to the subsequent one [21].

Below the building blocks used in our work are briefly explained.

Dense layer

The dense, or fully connected, layer is the basic model for a feedforward network. The outputs of such a layer is formed as linear combinations of the inputs and bias terms. Usually a non-linear activation function is applied to this to be able to capture more general behaviors, resulting in the output

$$y_i = h\Big(\sum_{j=1}^{D} w_{ij}x_j + b_i\Big) \tag{2.2}$$

where $h(\cdot)$ is a, possibly non-linear, activation function. x_j, $j \in \{1, ..., D\}$ are the inputs to the layer, w_{ij} and b_i are the weights and biases learned during training [8]. A network with two dense layers is shown in Figure 2.3.1.

Convolutional layers

Convolutional layers have proved successful for feature extraction from for instance time series or images. A reason for this is that they are equivariant to translation, meaning that patterns in a time series will be recognized in the same way no matter at which time steps they occur. The 1D convolution operation can be expressed as (2.3).

$$(x * w)(t) = \sum_{a=-\infty}^{\infty} x(a)w(t-a) \tag{2.3}$$

where x is the input and w is the kernel or filter which consist of the trainable parameters. As the kernel size is not affected by the input size the convolutional layer can be applied to

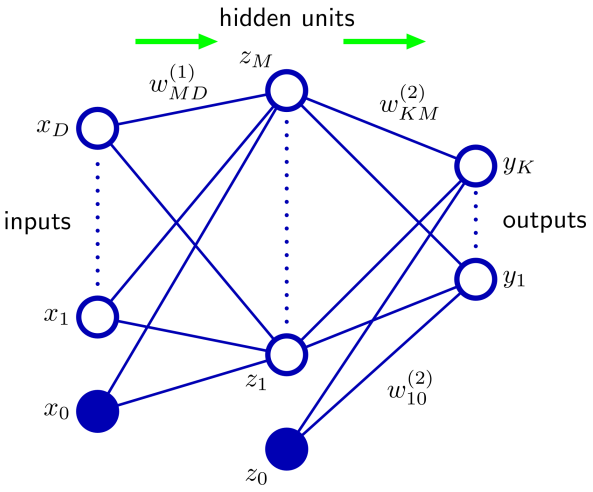


Figure 2.3.1: Feedforward neural network with two densely connected layers. Each line corresponds to one trainable parameter. Here, x_0 and z_0 can be seen as ones added to the inputs introducing the bias terms [8].

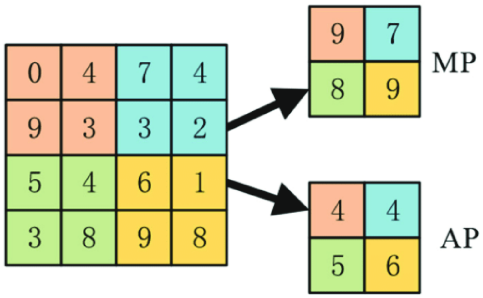
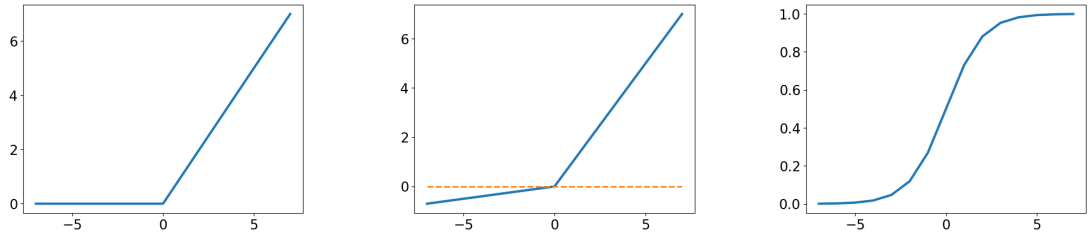


Figure 2.3.2: Illustration of max and average pooling with pooling size 2×2 and stride 2×2. Image from [66].

inputs of different size, which is not possible with, for instance, the fully connected layer. When applied to images the convolution is performed in two dimensions. [21].

Pooling layers

Pooling layers are used to reduce the dimensionality of feature maps. Common types of poolings are the max and the average pooling methods. Traditional max pooling represents nearby numbers by it maximum value while average pooling uses their average. This type of max pooling has proved efficient together with convolutional layers for computer vision tasks. Figure 2.3.2 illustrates how the pooling works. It can also be performed globally, i.e. on the entire feature map, which can be a way of handling differently sized data. For a Time Series Classification (TSC) problem, it is for instance possible to use size agnostic convolutional layers as feature extractors followed by a Global Average Pooling (GAP) layer resulting in a fixed size of the data to be classified [12].



(a) ReLU **(b)** Leaky-ReLU **(c)** Sigmoid

Figure 2.3.3: Different activation functions, note that the slope of leaky ReLU for negative numbers is exaggerated for visualization purposes.

Activation functions

The activation functions in a neural network has two main tasks. The first one is to introduce non-linearity to an otherwise linear model. The function $h(\cdot)$ in (2.2) is an example of such an activation function. A common such function is Rectified Linear Unit (ReLU), $h(z) = \max\{0, z\}$. Benefits with ReLU is that it in its active region ($z > 0$) does not have a suppressing effect on the gradient and it is easily computable. A drawback, however, is that the gradient is zero in its inactive region ($z < 0$) meaning gradient based training methods does not work here. An alternative to avoid this issue is the Leaky-ReLU given by $h(z) = \max\{0.01z, z\}$. ReLU and Leaky-ReLU are shown in Figure 2.3.3a and 2.3.3b, respectively. Activation functions are also used for the output of the network, e.g. to obtain outputs representing probabilities. The sigmoid function, $h(z) = 1/(1 + \exp(-z))$, shown in Figure 2.3.3c, can be used for this. The sigmoid function will saturate the output between 0 and 1. However, if the model has several outputs, e.g., representing the probabilities of the input belonging to different classes, the total probability will not sum to 1. In this case the softmax function, $h(z)_i = \exp(z_i)/\sum_{j=1}^{K} \exp(z_j)$, can be used instead [21].

2.4 Training

During training of a network a loss function, $\mathcal{L}$, which describes the desired behavior, is evaluated on the training data. To improve the performance of the model its parameters are changed to minimize this loss. In deep learning problems, this optimization is usually performed with some gradient descent inspired method,

$$\boldsymbol{W}_{k+1} = \boldsymbol{W}_k - \alpha \boldsymbol{D}, \tag{2.4}$$

where $\boldsymbol{W}_k$ denotes model parameters at iteration k, α the learning rate or step size, and $\boldsymbol{D}$ the parameter update direction, e.g. $\frac{\partial \mathcal{L}}{\partial \boldsymbol{W}}$ or a weighted average of earlier gradients.This means that the parameters are updated in the direction which reduces the loss the most. With a large training data set, the computation of the gradient quickly becomes expensive. A remedy for this has been to use stochastic or mini-batch

gradient descent methods. Such algorithms use one or a few data points from the training set to estimate the gradient for each parameter update. Algorithms common today often use momentum, where previous gradients affect the parameter update direction, and adaptive learning rates (step size of parameter update), allowing different learning rate for different parameters [21]. One example of such a method is the Adam optimizer [31].

The gradients of the loss with respect to the model parameters are calculated using the back-propagation algorithm [55] which recursively uses the chain rule,

$$\frac{dz}{dx} = \frac{dz}{dy}\frac{dy}{dx}, \tag{2.5}$$

to propagate the loss gradient through the network. For a network where $f_0, f_1, \dots, f_n$ denotes the outputs of the $n+1$ layers, with corresponding layer parameters $\boldsymbol{w}_0, \boldsymbol{w}_1, \dots, \boldsymbol{w}_n$ and loss function $\mathcal{L}$, the gradient is calculated by first performing a forward pass of input $\boldsymbol{x}$. This allows for computation of the the gradient w.r.t. the output of the final layer, f_n, either analytically or using automatic differentiation. As both the structure and the parameters of the layers are known this can be used to calculate the gradient w.r.t. the parameters in that layer, $\boldsymbol{w}_n$, as well as the output of the previous layer, f_{n-1}. By applying

$$\frac{\partial \mathcal{L}}{\partial f_k} = \frac{\partial \mathcal{L}}{\partial f_{k+1}}\frac{\partial f_{k+1}}{\partial f_k} \tag{2.6}$$

recursively, the gradient is propagated through the network and from this

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{w}_k} = \frac{\partial \mathcal{L}}{\partial f_k}\frac{\partial f_k}{\partial \boldsymbol{w}_k} \tag{2.7}$$

gives the gradients needed for the optimization.

Loss functions

For a classification problem with K mutually exclusive classes the categorical cross-entropy is commonly used. With this loss the labels are one-hot encoded meaning that each label is represented by K binary variables, i.e. $y_n \in \mathbb{Z}_2^K$. Each variable represents a class and $y_n^{(k)} = 1$ for the k corresponding to the class of the label and 0 otherwise. The final layer of the model has K outputs with softmax activation. The loss to be minimized is [8]

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{W}) = -\sum_{n=1}^{N}\sum_{k=1}^{K} \lambda^{(k)} y_i^{(k)} \log \widehat{y}_n^{(k)}(x_n, \boldsymbol{W}) \tag{2.8}$$

where $y_n^{(k)}$ denotes the correct binary label of class k for data point n in the training set, $\hat{y}_n^{(k)}$ the corresponding prediction from the model, and $\lambda^{(k)}$ weight for class.

The categorical cross-entropy will aim to maximize the predicted probability for the correct class. However, incorrect probabilities have no direct effect on the loss. To be able to affect what kind of errors the model makes in its predictions a modification of this loss can be used. This modified loss, here referred to as confusion-entropy, introduces a matrix, U, which can be seen as a target confusion matrix distribution. Entries in U rewards predictions at the corresponding positions in the confusion matrix, including possibly incorrect classifications. The confusion-entropy loss is [1]

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{W}, U) = -\sum_{i=1}^{K}\sum_{j=1}^{K} u_{ij} \log \sum_{n=1}^{N} y_n^{(i)} \hat{y}_n^{(j)}(x_n, \boldsymbol{W}). \tag{2.9}$$

2.5 Explainability

Much of the recent progress in the deep learning space is inherently incomprehensible for us humans, due to its black-box nature and the size of the models [16]. However, explainability is important at many stages of the development of an AI-system. When the systems performance is at sub-human levels, it simplifies for human experts to improve it. When the system achieves similar results to those of human experts, it can help enforce trust to the system. Finally, in a scenario where the AI outperforms humans, it can help us get a better understanding of the problem [56]. With these methods playing a bigger role in fields such as healthcare the importance of explainable decisions also grows from a legal and ethical perspective [5].

Gradient-weighted Class Activation Mapping (Grad-CAM)

Although most deep learning models are not interpretable, there are post-hoc methods which tries to explain decisions. Selvaraju et al. [56] suggested one such method, called Gradient-weighted Class Activation Mapping (Grad-CAM), where an activation map is calculated which shows what parts of the data is important for the decisions. This method is typically applied to the final convolutional layer ahead of, e.g., GAP layer. Let y_c be the output corresponding to class c and A be the feature map, of height H, width W, and with F filters, from which the activation should be calculated. The Grad-CAM activation, M_{GC}, is then calculated as

$$\begin{aligned} w_k^c &= \frac{1}{H \times W} \sum_{i=1}^{H}\sum_{j=1}^{W} \frac{\partial y_c}{\partial A_{ij}^k} \\ M_{GC} &= ReLU(\sum_{k=1}^{F} w_k^c A^k). \end{aligned} \tag{2.10}$$

With time series inputs, the resulting activation map, $M_{GC} \in \mathbb{R}^{H \times W}$, gives importance values for each time step if the network does not alter the time dimension of the data.

2.6 Consistent Rank Logits (CORAL)

Categorical data with a natural ordering are considered to be ordinal, examples of such data are the response to some medical treatment (e.g. poor, fair, good) [3] or the age of a person [9].

When classifying ordinal data, it is desirable to exploit the fact that the categories are ordered [3]. An ordinal classification problem, or ordinal regression as it is also referred to, can be formulated as assigning labels, $y \in \mathcal{Y} = \{\mathcal{C}_0, \mathcal{C}_1, ..., \mathcal{C}_{K-1}\}$, to inputs $\boldsymbol{x}$, where the classes $\mathcal{C}_0 \prec \mathcal{C}_1 \prec ... \prec \mathcal{C}_{K-1}$ according to some ordering relation [9].

Li and Lin [35] presented a method for ordinal regression where the combined result of $K-1$ binary classifiers for K classes were used. Each classifier checked whether the rank of the sample class was larger than rank $r_k \in \{r_1, ... r_{K-1}\}$. Niu et al. [50] developed this further using a multi-output Convolutional Neural Network (CNN) as $K-1$ binary classifiers, called OR-CNN. The classifiers share all weights except the ones in the output layer. This method achieved State of the Art (SOTA) performance on datasets where age was estimated based on facial images. However, consistency was not guaranteed in the predictions, e.g. sometimes simultaneously predicting an age under 20 and over 30. Cao et al. [9] addressed this issue with COnsistent RAnk Logits (CORAL) which is an architecture-agnostic method that can extend any neural network based classifier. Similarly to OR-CNN, CORAL uses $K-1$ binary classifiers, here however sharing all weights parameters apart from the biases in the output layer. Instead of representing the labels as one-hot encodings they are now formed as $K-1$ binary labels, i.e. $y_n \in \mathbb{Z}_2^{K-1}$, where $y_n^{(k)} = 1$ if the rank of the class is greater than r_k and 0 otherwise. The loss function to minimize is

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{W}, \boldsymbol{b}) = -\sum_{n=1}^{N} \sum_{k=1}^{K-1} \lambda^{(k)} [\log(\sigma(g(\boldsymbol{x}_n, \boldsymbol{W}) + b_k)) y_n^{(k)} + \log(1 - \sigma(g(\boldsymbol{x}_n, \boldsymbol{W}) + b_k))(1 - y_n^{(k)})], \tag{2.11}$$

where $\boldsymbol{W}$ denotes all model parameters except biases of final layer, $\boldsymbol{b}$ the bias weights of final layer, $\lambda^{(k)}$ the loss weight for rank k, $g(\boldsymbol{x}_n, \boldsymbol{W})$ the output of penultimate layer, $\sigma(z)$ the logistic sigmoid function, and $\sigma(g(\boldsymbol{x}_n, \boldsymbol{W}) + b_k)$ predicted output of the binary classifier.

It can be shown that

$$b_1 \geq b_2 \geq ... \geq b_{K-1}. \tag{2.12}$$

The proof can be found in [9] and from this and the shared weights it follows that

$$\widehat{P}(y_n > r_1) \geq \widehat{P}(y_n > r_2) \geq ... \geq \widehat{P}(y_n > r_{K-1}) \tag{2.13}$$

since the only thing that differs between the predictions is the bias. The probabilities for the individual classes are computed from this as

$$\begin{aligned} \widehat{P}(\mathcal{C}_0) &= 1 - \widehat{P}(y_n > r_1) \\ \widehat{P}(\mathcal{C}_1) &= \widehat{P}(y_n > r_1) - \widehat{P}(y_n > r_2) \\ &\vdots \\ \widehat{P}(\mathcal{C}_{K-1}) &= \widehat{P}(y_n > r_{K-1}). \end{aligned} \tag{2.14}$$

Chapter 3

Related work - Human Pose Estimation

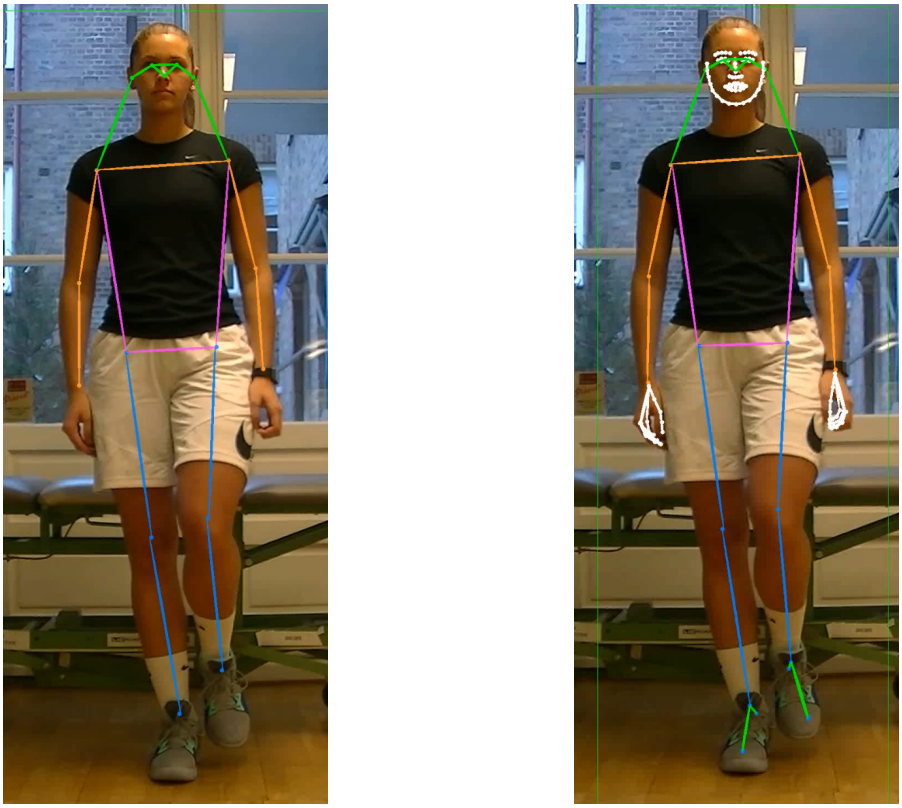
Human Pose Estimation (HPE) is a well explored problem which, like many other computer vision tasks, has developed rapidly in recent years. The reasons behind this progress can mainly be explained by two factors. Firstly, the emergence of computing power discussed in Section 2.2, allowing for more expressive deep learning models. Secondly, several datasets with images labeled with human body joints has been made available [10]. These datasets not only provide data, but also introduces competition in the research community, making it possible to compare the results of different approaches.

3.1 Datasets

In this work, models trained on the Microsoft Common Objects in Context dataset (COCO) dataset are used. COCO consists of 328k images containing 91 different object types. The images comes from Google, Bing, and Flickr image search and are mainly hand annotated through Amazon Mechanical Turk. The interesting part of the dataset for this work is the one with human poses. In total there are 250k instances of people labeled with joint locations [37]. The joints, 17 per person, in the dataset can be seen in Figure 3.1.1a. Along with the datasets containing body keypoints mentioned above there are also datasets with dense keypoints for specific bodyparts, such as hands or face. COCO-wholebody [29] is an attempt to combine these two types of datasets by extending COCO with dense keypoints at hands, feet, and faces. The resulting 133 joints can be seen in Figure 3.1.1b.

3.2 Background - Human pose estimation

The HPE problem has been explored since long before the most recent deep learning era. Pictoral Structures were introduced by Fishler and Elschlager in the 1970s. This



(a) Regular COCO. **(b)** COCO-wholebody.

Figure 3.1.1: Keypoints for the two COCO datasets.

meant identifying individual parts or features in images modeled with pair-wise spring-like connections [19]. After the progress of deep learning reviewed in Section 2.2, Toshev and Szegedy [63] presented DeepPose, the first HPE method based on DNNs, in 2014. Today's HPE methods are generally categorized as top-down or bottom-up approaches. This has to do with how they handle multiple persons. Bottom-up models starts by finding all keypoints for all persons in an image and then match them together to form persons. Top-down models on the other hand starts by finding bounding boxes for all individuals and then identifies keypoints for one person at a time. The sequential nature of the top-down methods and the fact that two models are needed means that bottom-up models scale better with the number of persons to analyze. However, top-down models tends to be more accurate [11].

3.3 Pose estimation models

Below the HPE models used in our work are presented. As we are interested in single person recognition, the model used is of top-down type.

3.3.1 High-Resolution Net (HRNet)

Sun et al. presented the High-Resolution Net (HRNet) [59] architecture in 2019, initially for HPE, but also for other computer vision tasks such as semantic segmentation and

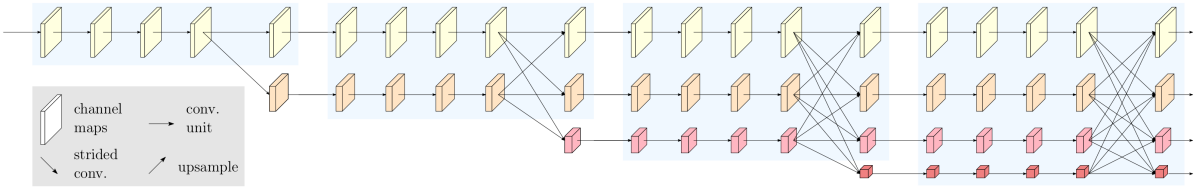


Figure 3.3.1: Network architecture for HRNet. The top row shows high resolution representations with fewer number of feature maps. Each step downwards reduces the resolution with a factor of two while the number of feature maps are doubled [65].

object detection. Such problems had traditionally been solved using networks built on high-to-low resolution convolutions with increasing numbers of feature maps (e.g ResNet [22], VGGNet [57]). The classification task was solved in the low-resolution space and then transformed back to form the high-resolution representation needed for e.g. the HPE. Sun et al's proposed architecture preserves a high resolution representation throughout the network. It does so while also producing low-resolution/high dimensional representations suitable for classification.

The network architecture is shown in Figure 3.3.1 and consists of four stages (blue blocks in depth direction in the figure) with convolutional layers. After each stage a new low-resolution representation is created by performing strided convolutions. At these instances, the existing representations also exchange information by either nearest neighbor upsampling or strided convolutions. The K estimated keypoints are represented as heatmaps, $\{\mathbf{H}_1, \ldots, \mathbf{H}_K\}$, indicating the locations. These heatmaps are formed from the last high-resolution feature map (top right in Figure 3.3.1). Corresponding ground truth heatmaps are generated by applying 2D Gaussians to the correct keypoint locations and the model is trained by minimizing the mean squared error between these [59]. Although a high resolution heatmap is desirable as it gives smaller quantization errors, the computational cost increases quadratically with the size [69]. Hence, the performance of the model can be improved by extracting the region of interest from the input image. This can for instance be done using an object detection model trained to find humans.

Object detectors usually work by first producing a large number of regions of interest in the image which are then classified to either belong to some object class or the background. Faster R-CNN by Ren et al. [54] is an example of such a detector where these steps are performed by a single CNN. The model outputs bounding boxes and class scores for the objects in the image deemed not to be part of the background.

3.3.2 Distribution-Aware coordinate Representation of Key-point (DARK)

As discussed above a high resolution heatmap should result in higher accuracy, but is computationally expensive. Zhang et al. propose a method they call Distribution-Aware coordinate Representation of Key-point (DARK) [69] to reduce the quantization error by i) analyzing the distributions of the predicted heatmaps, and ii) creating the training heatmaps in a slightly new fashion.

The actual keypoint location is found at the maximal activation of the heatmap. Since it is smaller than the actual image this turns into a sub-pixel localisation problem. Newel et al. [49] empirically found that good results may be found using a weighted average between the two highest activations,

$$\boldsymbol{p} = \boldsymbol{m} + \frac{1}{4}\frac{\boldsymbol{s}-\boldsymbol{m}}{\|\boldsymbol{s}-\boldsymbol{m}\|_2}, \tag{3.1}$$

where $\boldsymbol{p}$ denotes the predicted maximum, $\boldsymbol{m}$ the highest activation, and $\boldsymbol{s}$ the second highest activation.

This has been the de facto standard heatmap decoding, but Zhang et al. suggests using the fact that the heatmaps used for training usually are created as 2D Gaussian distributions, i.e. that the heatmaps can be expressed as

$$\mathcal{G}(\boldsymbol{x};\boldsymbol{\mu},\Sigma) = \frac{1}{2\pi \mid \Sigma \mid^{\frac{1}{2}}} \exp\Big(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^\top \Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\Big) \tag{3.2}$$

where $\boldsymbol{\mu}$ denotes the maximum of the heatmap, $\boldsymbol{x}$ the pixel location, and Σ a diagonal covariance matrix.

By Taylor expanding the logarithm of (3.2) in the point $\boldsymbol{m}$, i.e. the point with the highest sampled activation, an expression for $\boldsymbol{\mu}$ is obtained as

$$\boldsymbol{\mu} = \boldsymbol{m} - \Big(\mathcal{D}''(\boldsymbol{m})\Big)^{-1}\mathcal{D}'(\boldsymbol{m}) \tag{3.3}$$

where $\mathcal{D}(\boldsymbol{x}) = -\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^\top\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu})$, i.e., the non constant term in the logarithm of $\mathcal{G}$ in (3.2).

The derivatives $\mathcal{D}'(\boldsymbol{m})$ and $\mathcal{D}''(\boldsymbol{m})$ are efficiently estimated from the heatmap. As this approach strongly assumes a Gaussian structure, it is proposed to modulate the heatmap before estimating the maximal activation. This is done by performing a convolution with a Gaussian kernel with the same covariance as the one used for the training data.

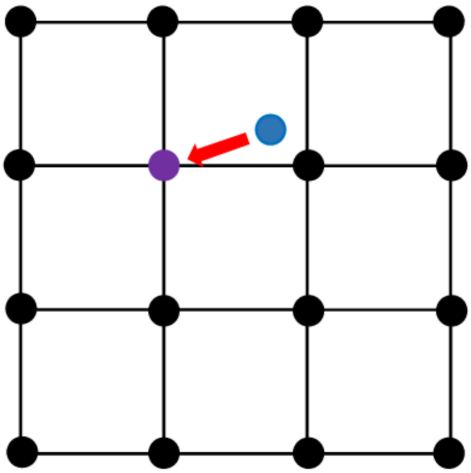


Figure 3.3.2: Quantization error due to off-grid keypoint location. Correct location (blue) represented by on-grid coordinate [69].

The second improvement suggested by Zhang et al. concerns the creation of the training heatmaps. Traditionally, these have been created from the quantized keypoint locations, resulting in a slightly biased heatmap. In Figure 3.3.2, this would correspond to having the peak activation in e.g. the purple dot. By instead using the non-quantized location, an unbiased heatmap is obtained. This would correspond to having the peak off-grid, in the blue dot in Figure 3.3.2. Combining this with the method described above using derivatives to predict the location of the maximum seems to improve the accuracy of pose estimators.

The DARK method is model agnostic and can be used with any pose estimator as it is just a slightly different way of generating the training heatmaps and analyzing the output heatmaps.

Chapter 4

Related work - Time Series Classification

4.1 Background - Time series classification

Time series are sequences of data ordered in some dimension, e.g. time. They can be univariate, i.e. containing one variable, or multivariate and they can be used to describe any ordered, discrete data, for instance information extracted from videos.

Definition 2. *A univariate time series of length n, with ordered indices*

$$X = [x_1, x_2, \dots, x_n]^\top$$

Definition 3. *A multivariate time series of length n, with M channels*

$$\boldsymbol{X} = [X_1, \dots, X_M]$$

The TSC task is about finding a function, $f : \mathbb{R}^{n \times M} \rightarrow \mathbb{R}$, that assigns one label to each, possibly multivariate, time series. The problem bears strong resemblance with that of image classification, but with the two spatial dimensions replaced by one temporal dimension. Despite this the use of end-to-end deep learning models is not as dominant in the TSC community [26]. Instead it is common to manually extract features deemed suitable for classification. Similarily to the fields of computer vision various datasets has emerged recently. This has been important for the development of TSC as it allows for fair comparison between methods. One of the most widely used dataset collections today is the University of California, Riverside (UCR) archive [15] containing 85 different time series datasets.

Traditionally a nearest neighbor method together with dynamic time warping has been common for classification [7]. Simply put, this means that a time series during classification is compared to the training data and assigned the class of the most similar

time series. Lines and Bagnall suggested a method where an ensemble of 11 nearest neighbor classifiers with different similarity measures [39] which yielded SOTA results. Bagnall et al. [6] developed the idea of ensemble based classifiers with Collective Of Transformation-based Ensembles (COTE), where 35 different classifiers using different transforms were used. Lines et al. [38] extended COTE further with two new classifiers resulting in Hierarchical Vote Collective of Transformation-based Ensembles (HIVE-COTE). One drawback with HIVE-COTE is the computational intensity, both during training and test time. Training time is large partly due to one of the transforms used is the Shapelet Transform with a time complexity of $O(n^2l^4)$, n being the number of time series and l the length of them. Due to the nature of the nearest neighbor algorithm, the result of the 37 classifiers during test time needs to be compared to the corresponding result for each time series in the training set, causing this method to be impractical for real-time use [26].

In 2016 Zheng et al. [70] presented a neural network model based on convolutional layers for the classification task. Wang et al. [67] developed these ideas and presented models with performance close to that of COTE on the UCR archive. The development of neural network based classification has since then continued and below two architectures which are either used or have inspired our work are presented.

4.2 Deep learning architectures

The last few years the number of proposed neural network based time series classifiers has increased drastically. Below the two most influential architectures for our work are presented.

4.2.1 InceptionTime

InceptionTime, presented by Fawaz et al. [27], is, as the name suggests, inspired by Inception [60] which is an architecture successful in computer vision tasks. It is comprised of several stacked Inception modules consisting of differently sized convolutions as well as pooling layers. To reduce the number of parameters in the network 1×1 convolutions are often used as a dimensionality reduction. One such module can be seen in Figure 4.2.1a. The architecture of Fawaz et al. is similar, but with only one temporal dimension instead of two spatial dimensions. The number of filters in the convolutions and the length of these filters are hyperparameters to be tuned. All convolutions use the same number of filters to simplify the tuning and the same length parameter is also used throughout the network. The length of the three parallel filters is given by $0.5^k * \text{filter\_length}$, $k = [0, 1, 2]$. Thus the length parameter in Figure 4.2.1a is 40. As with the computer vision task the dimensionality is reduced, here through a bottleneck of size m. The bottleneck is achieved by convolutions with m filters of length 1. The InceptionTime module is shown in Figure 4.2.1b.

Figure 4.2.2 shows how stacked modules makes up the InceptionTime architecture. Residual connections are used to decrease the risk of vanishing gradients once the network

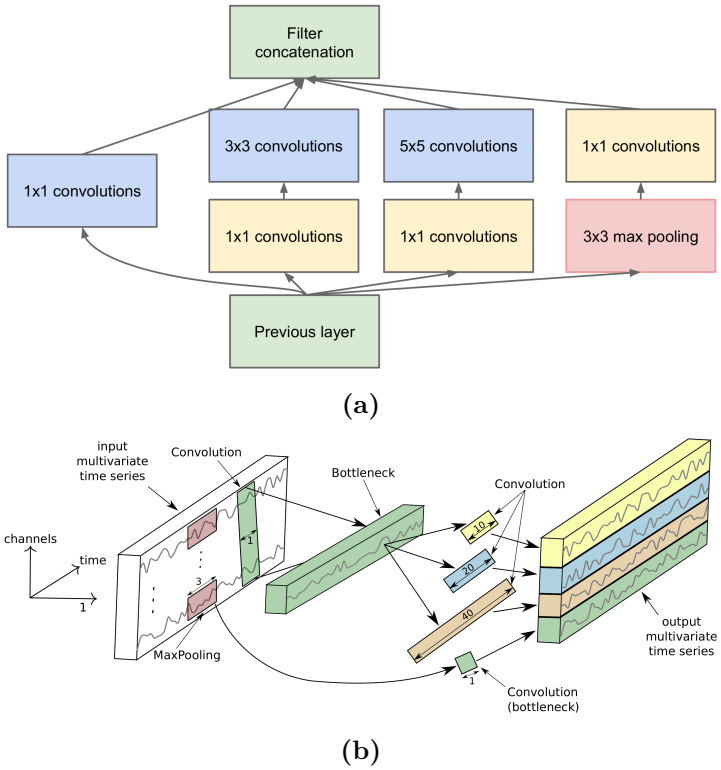


Figure 4.2.1: Inception modules for computer vision (a) with dimensionality reduction ahead of the 3×3 and 5×5 convolutions and InceptionTime module for TSC (b), here illustrated with a bottleneck size of 1. Figures from [60] and [27] respectively.

becomes deeper, as suggested by He et al. [22]. The InceptionTime modules are followed by a GAP layer which averages each time series over its time dimension. The classification is performed by fully connected layers with softmax activations.

Many deep learning based time series classifiers experiences a significant variance in their accuracy, especially when evaluated on the UCR archive with rather small training sets [25]. To overcome this Fawaz et al. suggest the use of an ensemble of identical InceptionTime networks, but with randomly initialized weights before training. The ensemble's output is then calculated as the average of the outputs of the individual models. Such an ensemble of five models achieves a performance similar to that of HIVE-COTE on the UCR archive.

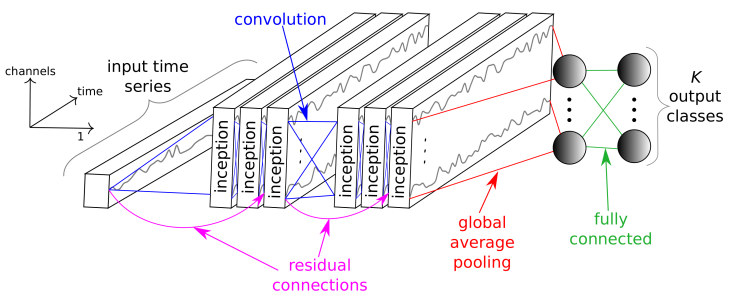


Figure 4.2.2: InceptionTime architecture for TSC [27].

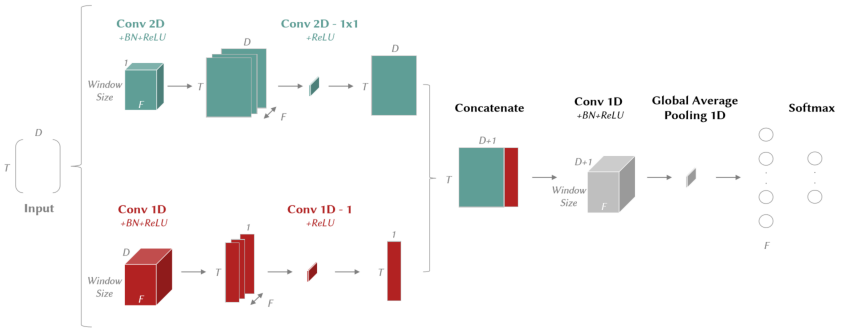


Figure 4.2.3: The XCM architecture with BN - Batch Normalization, D - number of input channels, F - number of filters, T - length of time series [18].

4.2.2 Explainable Convolutional Neural Network for Multivariate Time Series Classification (XCM)

As discussed in Section 2.5 explainability is desirable, but not inherent in most black-box deep learning models. Fauvel et al. [18] propose an architecture, Explainable Convolutional Neural Network for Multivariate Time Series Classification (XCM), which allows for tracking which time steps and which inputs are important for the classification decision. By using 2D convolutions with kernels of size $ks \times 1$, where ks is the kernel size hyperparameter, the convolution is only performed in the time dimension and the input channels are kept separated throughout the feature extraction. Through dimensionality reduction from a 1×1 2D convolution a single feature map for each input is produced. From this, the importance of input channels and time steps can be traced using Grad-CAM, described in Section 2.5. In parallel to the channel specific features Fauvel et al. also suggests using 1D convolutions over all channels resulting in a combined feature map along the time dimension. The XCM architecture is depicted in Figure 4.2.3.

Chapter 5

Methods

5.1 Overview

In this chapter, the system for assessing POEs will be presented. This system is naturally divided into two parts where firstly the videos are analyzed. The first subsystem extracts body part coordinates of the subjects. This information is then passed to the second subsystem where it is used to calculate a score according to [45]. The data used is presented in Section 5.2 and the two subsystems are described in Sections 5.3 and 5.4, respectively.

5.2 Data

The data available is in the form of videos each containing one subject, recorded from a frontal view. As discussed in Chapter 1, the SLS task and the trunk, pelvis, femoral valgus, and KMFP POEs are evaluated. Each video contains four-five repetitions and for each repetition the POEs above have been scored according to Table 1.1.1. Along with the POE scores a certainty score, describing the confidence of the physiotherapist assessing the videos, is provided. This is between 0 (certain) and 2 (uncertain) and when above 0 the uncertainty direction is provided as well. This certainty score is only available for the combined (median) score for all repetitions, i.e. one certainty score per video.

In total there are labeled videos from 103 different subjects and for some of the subjects there are videos of the task performed with both right and left leg. The number of labeled repetitions per video and POE varies slightly. This variation is due to different factors such as all subjects not performing five repetitions, or that the entire movement was not captured in the video and was thereby excluded by us. The total amount of data for the different POEs is summarized in Table 5.2.1. From this data 22 of the repetition sequences (110 repetitions), from 22 different subjects, were put aside as a test set. The test set was chosen from the set of repetition sequences ensuring no data from the same sequence was in both the training and test data.

Table 5.2.1: Data available for the different POEs.

POE	Number of unique subjects	Number of repetition sequences	Number of repetitions
Trunk	103	105	520
Pelvis	103	105	519
Femoral Valgus	103	107	530
KMFP	103	107	530

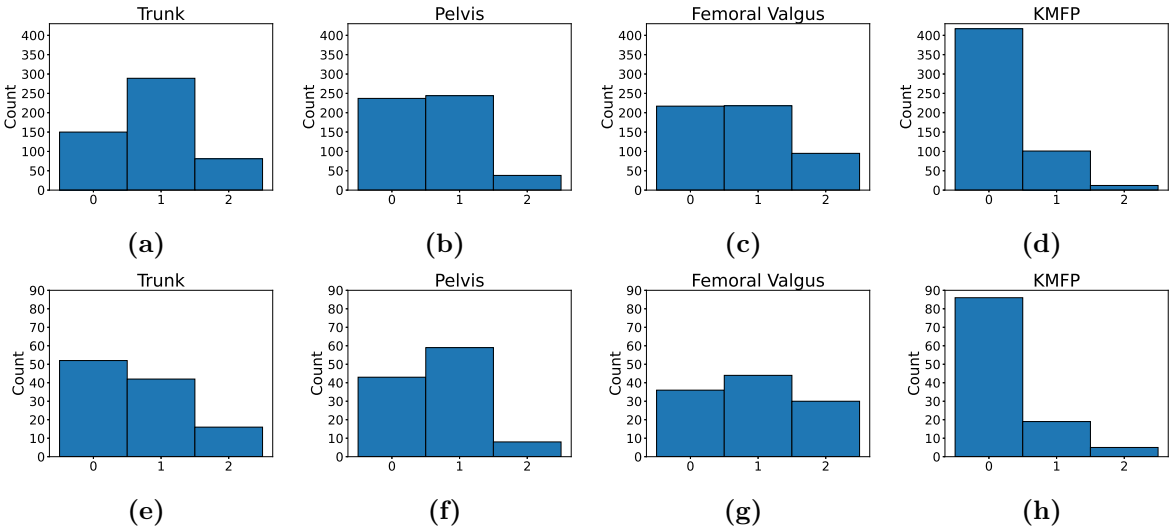


Figure 5.2.1: Distributions of the labels for the available data. (a)-(d) shows distribution over all available data and (e)-(h) shows the distribution in the test set.

The label distributions can be seen in Figure 5.2.1 and it can be noted that for all POEs there is a slight imbalance with fewer repetitions classified as Poor. The imbalance for KMFP is very clear with about 80% of the data classified as Good. In the same figure the label distribution for the test set is shown as well. For the trunk POE it is slightly different from its overall distribution while similar for the other sets.

5.3 Body part localization

The pose estimation is built around the open-source toolbox MMPose [42] from MMLab. Each frame is considered to be an independent image and is analyzed with a HRNet model using the DARK extension trained on the COCO-wholebody dataset[1]. Both the model and the dataset is described in Section 3. The extended wholebody dataset is used since it, along with the ankle positions, also estimates the positions of the toes and heels which contain valuable information.

To get comparable results some of the videos were rotated and flipped before inferring the keypoints. This was needed since the videos were recorded in different orientations and the actions were performed with different legs. The rotations were based on the

[1]The model used can be found here: `https://mmpose.readthedocs.io/en/latest/top_down_models.html`.

orientation of the subject (position of head w.r.t. the feet) in the first frame to have it standing up in the y-direction. Videos where the squats were performed with the left leg were flipped around the y-axis to be able to use the same model for the left and right leg in a more efficient manner.

A bounding box for the subject is found using a Faster R-CNN model trained on the COCO dataset[2]. The content of this bounding box is resized to match the input size of the HPE model used, 384×288 pixels in our case. Each video analyzed results in sequences of x- and y-coordinates for all the keypoints in the dataset used to train the model.

5.4 Classification

5.4.1 Preprocessing and creation of dataset

Before assessing the POEs based on the body part positions, a number of preprocessing steps are conducted. Firstly the data is resampled as the videos are recorded with a number of different frame rates ranging from 25 to 60 Hz. The resampling is performed using linear interpolation to a new sample frequency of 25Hz. This data is then low pass filtered through a fourth order Butterworth filter with a cutoff frequency of 5Hz.

While the POE assessment is performed on a per repetition basis, the body part coordinates are extracted on a per video or repetition sequence basis. Hence, the sequences corresponding to the entire video is split up in the individual repetitions. This splitting is presented in Algorithm 1 and is based on finding the edges of the peaks in specific position data. For the SLS task, the y-coordinate of the right shoulder is used. The number of points extracted for each repetition depends on the width of the peak. The length of the observed repetitions varies from about 1 to 8 seconds. For practical reasons, such as handling of data and training performance[3], it is desirable to save the data as multidimensional arrays with the same dimensions. Two different ways of solving this problem is evaluated, namely i) padding the sequences and use maskings for the padded samples in the models, and ii) alternate the sample frequency to thereby achieve sequences of the same length.

Finally, the data is normalized. All coordinates are moved to put the mean position of the first five right hip-samples in the origin and are scaled to set the distance between the right shoulder and right hip to one, according to

[2]The model used can be found here: `https://github.com/open-mmlab/mmdetection/tree/master/configs/faster_rcnn`.

[3]All data in one batch must have the same size. Hence, to be able to train with a batch size larger than 1, which usually improves training performance [21], all data in the same batch needs to have the same dimensions.

Algorithm 1: Extraction of repetitions from sequences

```
right_edges, left_edges = find_edges(sequence);
for peak, right, current_left, next_left in peaks, right_edges, left_edges do
    split_index = mean(right, next_left);
    start = max(current_left - extra_points, 0);
    end = min(right + extra_points, split_index);
    repetition = normalize_length(sequence[start:end]);
    sequence = sequence[end:];
end
```

$$
\begin{aligned}
(x,y)_i &= (x,y)_i - \overline{(x,y)}_{rh} \\
(x,y)_i &= \frac{(x,y)_i}{\|\overline{(x,y)}_{rs}\|_2} \qquad ,\forall i
\end{aligned}
\tag{5.1}
$$

where $\overline{(x,y)}_i$ denotes the mean over first five samples for body part i, rh the right hip, rs the right shoulder, and i the different bodyparts.

After these preprocessing steps, a dataset with inputs $\in \mathbb{R}^{N\times T\times F}$ and corresponding one-hot labels $\in \mathbb{Z}_3^N$ are created. The inputs consists of N multivariate time series of length T with F channels. These channels are a subset of the extracted x- and y-coordinates as well as angles and differences between keypoints. Which features are used and how they are chosen are presented in Section 5.4.3.

5.4.2 Classifiers

For the modeling, we used ensembles of different deep learning based model architectures. The reasoning behind this was based on the results of Fawaz et al. [25], suggesting that the output a deep learning model trained on a limited amount of data will vary based on the initial parameter values. By averaging the result over several models this variance will be reduced. Another reason for using an ensemble is, as can be seen in e.g. [6, 38], that the combined result of many specialized models can be better than that of one more general model.

All models used have been modified to handle the padded input data discussed in Section 5.4.1. This is done by adding masking layers setting the padded samples to zero throughout the networks, as illustrated in Figure 5.4.1. This reduces the impact of the padded samples to something similar to the padding performed in convolutional layers to keep the size of the feature map intact. The same mask indicates which time steps should be ignored in the GAP layer.

The models eventually used were InceptionTime (Section 4.2.1) with different loss functions as well as an architecture designed by us, inspired by XCM (Section 4.2.2) and InceptionTime. We call this model X-InceptionTime and it is presented below.

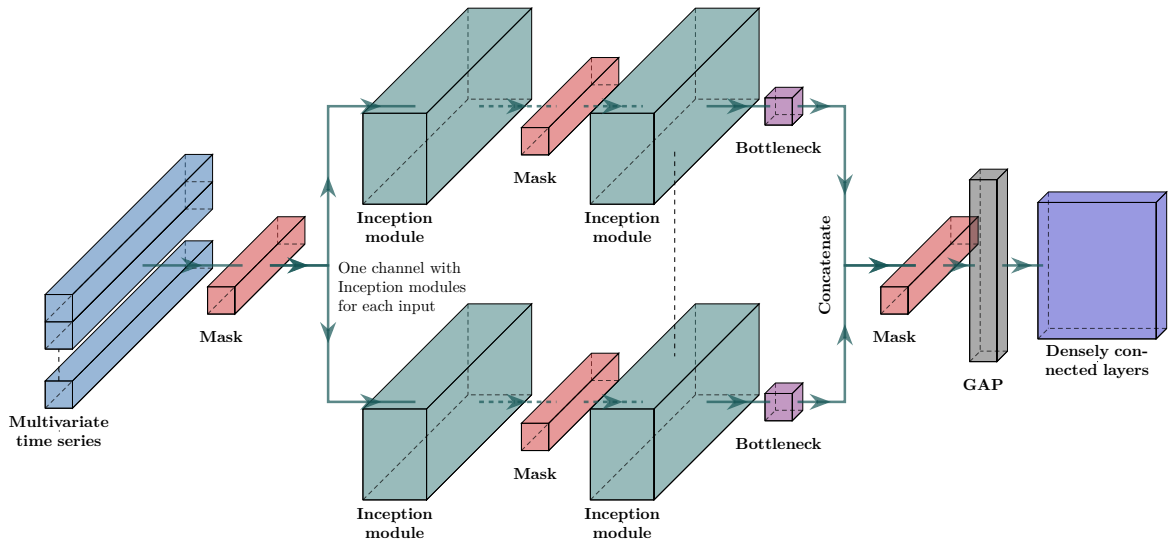


Figure 5.4.1: The X-InceptionTime architecture developed in this work.

X-InceptionTime

The idea with this model was to combine the explainability of XCM with the inception module from InceptionTime. This was done by separating the inputs and having individual inception modules for each input channel as can be seen in Figure 5.4.1. After the final module (the depth can be seen as a hyperparamtere and needs to be tuned) a bottleneck of size one is applied reducing the dimensionality of each input channel back to $T\times 1$. The features for the individual inputs are concatenated resulting in a feature map of size $T\times F$ where each input feature is only affected by that input. This makes it possible to use Grad-CAM to get a measure of the importance of each time step for each input.

The Grad-CAM method is slightly modified and simplified for this model compared to what is presented in Section 2.5, mainly due to the one dimensional data. Consider the final feature map A consisting of the concatenated feature maps from the separate input channels. As mentioned above $A \in \mathbb{R}^{T\times F}$ and the aim is to find importance values for each time step in each input. With the same notations as in (2.10), i.e. A_i^k corresponds to the activation of input k at time step i, the Grad-CAM, M_c^k, for class c and input k can be calculated as follows

$$\begin{aligned} w_k^c &= \frac{1}{T}\sum_{i=1}^{T}\frac{\partial y_c}{\partial A_i^k} \\ M_c^k &= w_k^c A^k. \end{aligned} \tag{5.2}$$

Compared to (2.10) the *ReLU* activation has been removed. This means that the importance values also contain information about which features suggesting this sample belongs to another class than c.

Along with the effect of the time steps it is also possible, thanks to the GAP layer, to get a measure of the importance of each input. The importance value, α_k^c, for input k describes how much effect this input has on the classification decision. It is given by applying the Grad-CAM method to the output of the GAP layer, B. From the feature map, A, this importance weight is calculated according to

$$\begin{aligned} B^k &= \frac{1}{T}\sum_{i=1}^{T} A_i^k \\ w_k^c &= \frac{\partial y_c}{\partial B^k} \\ \alpha_k^c &= w_k^c B^k. \end{aligned} \tag{5.3}$$

Ensembles

The ensembles used consists of multiple models whose outputs are linearly combined to form the ensemble output. As discussed above, this allows the ensemble to benefit from models optimized to perform well according to different metrics. In Section 5.4.4, we will present how k-fold cross-validation was used for training and validation. This was also used for design of the ensembles.

The idea with the ensemble was to combine models performing across all classes with models with high precision for the individual classes. The models with good overall performance was in general using the CORAL activation and loss. The other models were trained with the confusion entropy loss with a target confusion matrix, from which the u_{ij} in (2.9) comes, aiming at achieving high precision for one class and ignoring the other predictions. Examples of matrices used to find high precision models are

$$\begin{bmatrix} 0.6 & 0.05 & 0.05 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix} \quad \text{(5.4a)} \qquad \begin{bmatrix} 1 & 0 & 0 \\ 0.3 & 0.5 & 0.3 \\ 0 & 0 & 1 \end{bmatrix} \quad \text{(5.4b)} \qquad \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0.1 & 0.1 & 0.4 \end{bmatrix}. \quad \text{(5.4c)}$$

A higher value can be seen as a reward for predictions turning up at that position in the confusion matrix and clearly the sum of the columns are not equal, meaning that the models will be biased towards certain predictions. Eq. (5.4a) is used for class 0 and the ones in the lower right corner in this example means that correct prediction of a 1 will give the same reward as incorrectly predicting it as a 2. The idea behind the lower reward for correct prediction of 0 together with the small rewards for predicting it as a 1 or 2 is to classify uncertain examples as 1s or 2s. The idea for class 2, in (5.4c) is the same, but opposite. As the underlying scoring scale is ordinal, the same approach was not suitable for class 1. Instead a target matrix like (5.4b) was used. An alternative way of achieving high precision models was to tune the weight parameters λ in the CORAL loss, (2.11), to emphasize one of the rank predictions over the other. Due to the significant class imbalance for the KMFP POE the goal for class 1 and 2 was

not to achieve high precision. Instead models slightly biased towards these classes were used. This was achieved by using confusion entropy losses with target matrices shown in Appendix B.4.

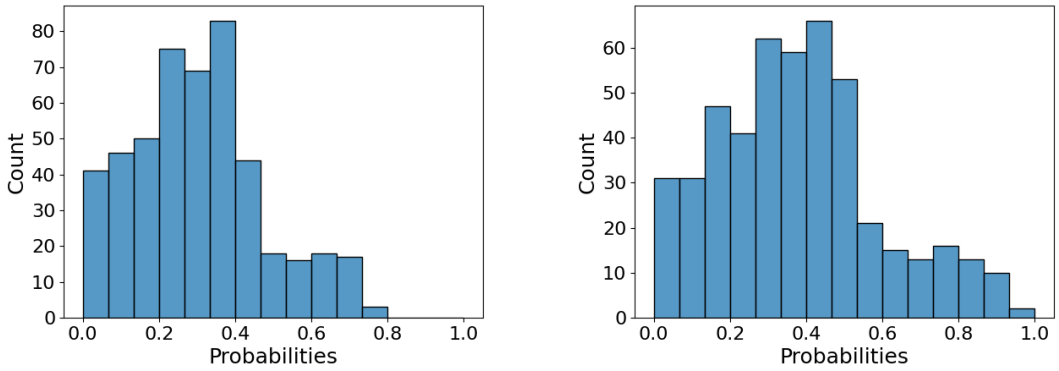
For the ensemble output the outputs of the individual models were combined as a weighted sum, initially according to

$$\bar{y}^{(c)} = \sum_{i=1}^{E} w_i^{(c)} \hat{y}_i^{(c)}, \tag{5.5a}$$

$$1 = \sum_{i=1}^{E} w_i^{(c)}, \quad \forall c, \tag{5.5b}$$

where E denotes the number of ensembles, $\bar{y}$ the ensemble output, and $\hat{y}_i$ the output of the i-th model.

The models used were determined based on the performance during the cross-validation and the weights were chosen uniformly for all models except the high precision ones. For these the weights instead were set to zero for the classes it was not trained for. By studying the predicted output probabilities for the validation data it could be seen that they in some cases were slightly biased away from class 1. The histogram of the predicted class 1 probabilities for the Femoral Valgus data is shown in Figure 5.4.2a. The main reason for this bias is the CORAL classifier which tends to result in slightly lower predictions for intermediary classes. To prevent this from resulting in skewed combined scores (see Section 5.4.5) the weights for this class were increased slightly, resulting in (5.5b) summing to something greater than one for class 1. The final ensemble weights can be seen in Appendix B. The weights for all models in the ensemble were multiplied with the same factor, resulting in the adjusted histogram in Figure 5.4.2b. This adjustment shifts all probabilities, but will have a greater effect for predictions with already high probability.



(a) **(b)**

Figure 5.4.2: Output probabilities for class 1 (Femoral Valgus) from the unadjusted (a) and adjusted (b) ensemble suggesting some bias away from this label.

The ensembles used are presented in Table 5.4.1. The suffixes -coral and -conf-X indicates

Table 5.4.1: Models forming the ensembles. * indicates data length normalized to 100 samples. No star means that original sample frequency (25Hz) was kept and the input data was padded to the same length. If neither coral nor conf is stated cross entropy loss is used. IT shows InceptionTime modules were used. The identifier TX, PX, FX, KX is used in Appendix B where more information, such as ensemble and training weights, can be found.

Trunk	**Pelvis**	**Femoral Valgus**	**KMFP**
T1, IT-coral*	P1, IT-coral*	F1, IT-coral*	K1, IT
T2, IT-coral	P2, IT-coral	F2, X-IT-coral*	K2, X-IT
T3, X-IT-conf-0*	P3, IT-conf-0*	F3, X-IT-conf-0*	K3, X-IT-conf-0
T4, IT-conf-1*	P4, IT-conf-1	F4, X-IT-conf-1	K4, IT-conf-1*
T5, X-IT-coral-2*	P5, X-IT-coral-2*	F5, X-IT-conf-2*	K5, X-IT-conf-2

CORAL classifiers and models trained with the confusion entropy loss, respectively. The X tells which was the high precision class. This is also used to indicate CORAL models trained to achieve high precision. The depth, filter length, and number of filters for all X-InceptionTime models were 1, 31, and 32. The corresponding parameters for the InceptionTime models were 2, 31, and 128. All models use Leaky-ReLU activations throughout the network. More detailed model descriptions can be found in Appendix B along with its training and ensemble weights.

5.4.3 Input selection

To select which inputs to use when classifying the different POEs the importance weight introduced in (5.3) was used. This was done by iteratively training models and evaluating

$$W_k = \text{mean\_folds}\Big(\text{normalize}(\sum_{i=1}^{N} |w_{ik}^{c_i}|)\Big), \tag{5.6}$$

where mean_folds(z) denotes the average of z over all folds, normalize(z) $= \frac{z}{\|z\|_2}$, and $c_i = \text{argmax}(\widehat{y}_i)$, i.e., the predicted class.

For each set of inputs evaluated the features corresponding to the lowest W_k were removed until the performance of the model dropped significantly. This method has its drawback, the most notable being that only input features suitable for this architecture will be found. This model does not consider interaction between different features directly as the inputs are kept separate throughout the network, hence features important through such interactions might not be deemed important. This was somewhat circumvented by validating the reduced input features on other model architectures as well, allowing this methodical way of finding suitable inputs. The inputs used for the different POEs are presented in Table 5.4.2.

Table 5.4.2: Inputs to the models classifying the different POEs. If the task was performed with the left leg the video has been mirrored, as described in Section 5.3.

Trunk	**Pelvis**	**Femoral Valgus**	**KMFP**
Left shoulder - x	Right shoulder - x	Right shoulder - x	Left shoulder - y
Right shoulder - x	Right shoulder - y	Right hip - x	Right hip - y
Right shoulder - y	Right hip - x	Right knee - y	Angle: right ankle and toes
Left hip - x	Right hip - y	Angle: right knee and ankle	Difference: right hip and knee - x
Left hip - y	Left hip - y		Difference: right knee and ankle - x
Right hip - x	Difference: right hip and knee - x		
Difference: right hip and knee - x	Difference: right knee and toes - x		

5.4.4 Training

We used 10-fold cross-validation on the training set for the development of the models and ensembles. The amount of data used for training ranges from 74-77 sequences of repetitions and 369-380 repetitions for the different folds and POEs. Like the test set, the folds were created such that no repetitions from the same subject were in both the test and validation set. For all models a batch size of 32, a learning rate of 5×10^{-3}, the Adam optimizer, and early stopping was used. The learning rate was also reduced by a factor of 0.85 every 50th epoch. The training was performed on Nvidia Tesla T4 and V100 GPUs.

5.4.5 Combined score

The eventual score in the method proposed by Nae et al. [45] is the combined assessments of all the repetitions. This score is formed as the median of the repetition scores. We propose to do this by averaging the ensemble outputs instead and chose the class with highest average probability over all repetitions. This means that a repetition classified with some uncertainty will affect the combined score less than a certain one. By evaluating the distributions for the ensemble outputs (Appendix C) it is also possible to introduce thresholds for both the combined score and the repetition score. For the repetition predictions the probability was set to 0 if the prediction was below 0.2. The validation data also suggests that the combined probability should work as an indicator of the certainty of the prediction. It could also be used to introduce a threshold ignoring predictions below a certain confidence. Studying the histograms, based on the validation sets in the cross-validation, in Appendix C, a threshold of 0.4 seems to give a reasonable trade-of between ignored correct and incorrect samples.

5.4.6 Baseline method

The models presented above are compared to baseline methods using a large number of features combined with support vector machines and Fisher's linear discriminants.

The features are extracted using the default options of tsfresh [64], i.e., no tuning of parameters or choice of features based on this specific problem has been performed. Examples of features used are number of peaks, mean values, derivatives of different orders, and variances.

Chapter 6

Results and Discussion

6.1 Body part localization

The pose estimation was reliable as long as body parts were not occluded. Figure 6.1.1 shows a sequence of frames from which it is clear that all body parts except the right foot are found accurately. This figure also suggests this video was not recorded from straight ahead. This is a problem as the plane with the the coordinates will be slightly rotated, resulting in distorted body part positions.

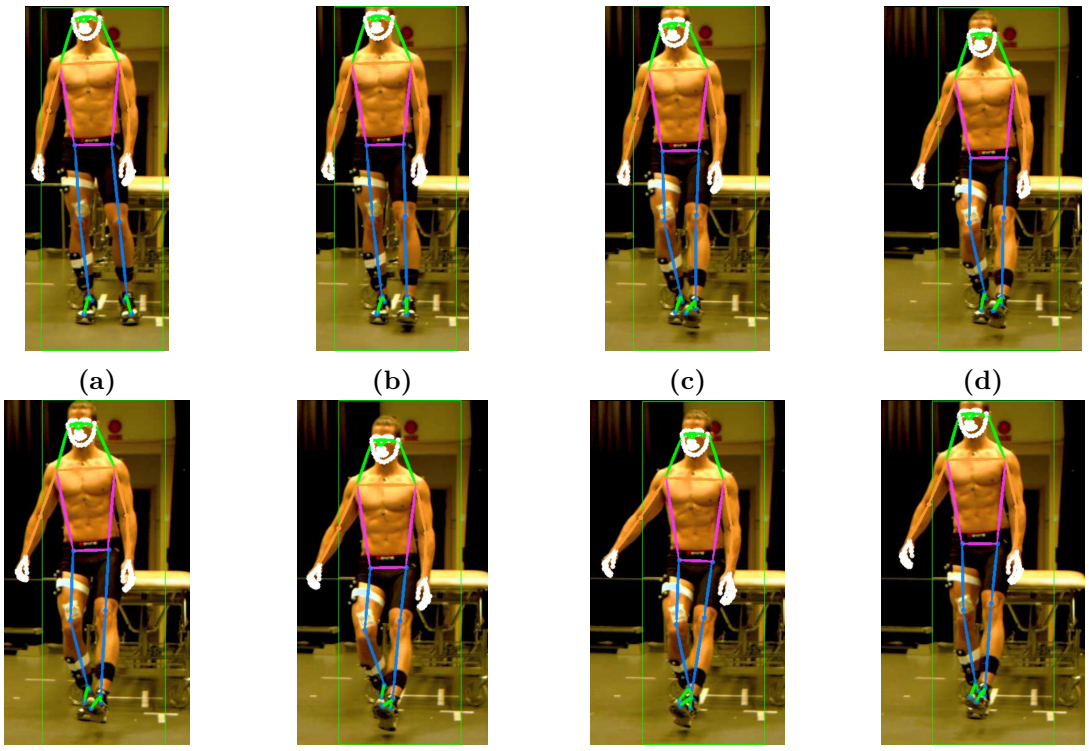


Figure 6.1.1: Frames from video where the right foot is occluded (c)-(h).

6.2 Classification

The figures and results presented in this section are generated by models and ensembles trained according to the cross validation strategy described in Section 5.4.4 and evaluated on the test set. This results in 10 sets of models trained on slightly different data.

6.2.1 Baseline results

The repetition accuracy and F1 scores for the baseline methods presented in Section 5.4.6 are shown in Table 6.2.1.

Table 6.2.1: Baseline results for the different POEs for the classification of the individual repetitions. The results are the mean from the 10 folds ± the corresponding standard deviations. The F1 score is macro averaged.

	Trunk		**Pelvis**		**Femoral Valgus**		**KMFP**	
	SVM	LDA	SVM	LDA	SVM	LDA	SVM	LDA
Acc(%)	35.3±5.2	45.2±5.4	52.3±4.3	46.2±3.9	38.6±3.6	42.1±3.5	78.7±5.8	53.7±6.8
F1(%)	17.3±1.9	41.5±5.5	35.5±3.1	37.9±3.0	29.6±2.6	40.6±3.9	36.7±2.4	47.1±9.2

6.2.2 Trunk

Figure 6.2.1a shows the confusion matrix summarizing the classifications of the individual repetitions made by the ensemble presented in Table 5.4.1 and Appendix B. Each entry in the matrix is the mean along with the standard deviation for models from the 10 folds. Figures 6.2.1b and 6.2.1c shows the corresponding matrices for the combined scores, with and without the threshold suggested in Section 5.4.5. Figure 6.2.1d shows how the samples ignored due to the threshold were classified. These results are also summarized as accuracy, F1 scores, precision, and recall in Table 6.2.2. This table also shows the model performance for the data with different label certainty, specified by the physiotherapist labeling the data. Histograms for these metrics are shown in Figure 6.2.2 along with the corresponding metrics for the individual models in the ensemble (high precision models are not shown as they only predict one label).

Table 6.2.2: Results of the ensemble for the trunk POE. Rep., Comb., and Thresh. represents the results for the repetitions, combinations, and combinations with thresholds, respectively. The Certainties columns show the results making up the Comb. column, but for the certainty levels of the expert labeling the data. These ranges from certain (0) to uncertain (2), the variable n shows how many datapoints each category contains. All results are the mean from the 10 folds ± the corresponding standard deviations. F1, recall, and precision are macro averaged.

	Rep.	**Comb.**	**Thresh.**	**Certainties**		
				0(n=15)	1(n=6)	2(n=1)
Accuracy (%)	73.7±4.5	75.0±7.9	**80.0±7.8**	81.3±7.1	56.7±15.2	textbf90.0±30.0
F1 score (%)	73.1±3.4	75.0±5.8	**79.9±8.9**	80.4±7.1	25.8±6.3	textbf90.0±30.0
Recall (%)	73.3±3.7	76.7±3.8	**81.6±7.2**	**81.1±5.0**	28.0±11.0	30.0±10.0
Precision (%)	76.7±4.1	78.7±5.2	**83.0±6.8**	**84.9±5.9**	28.9±6.8	30.0±10.0

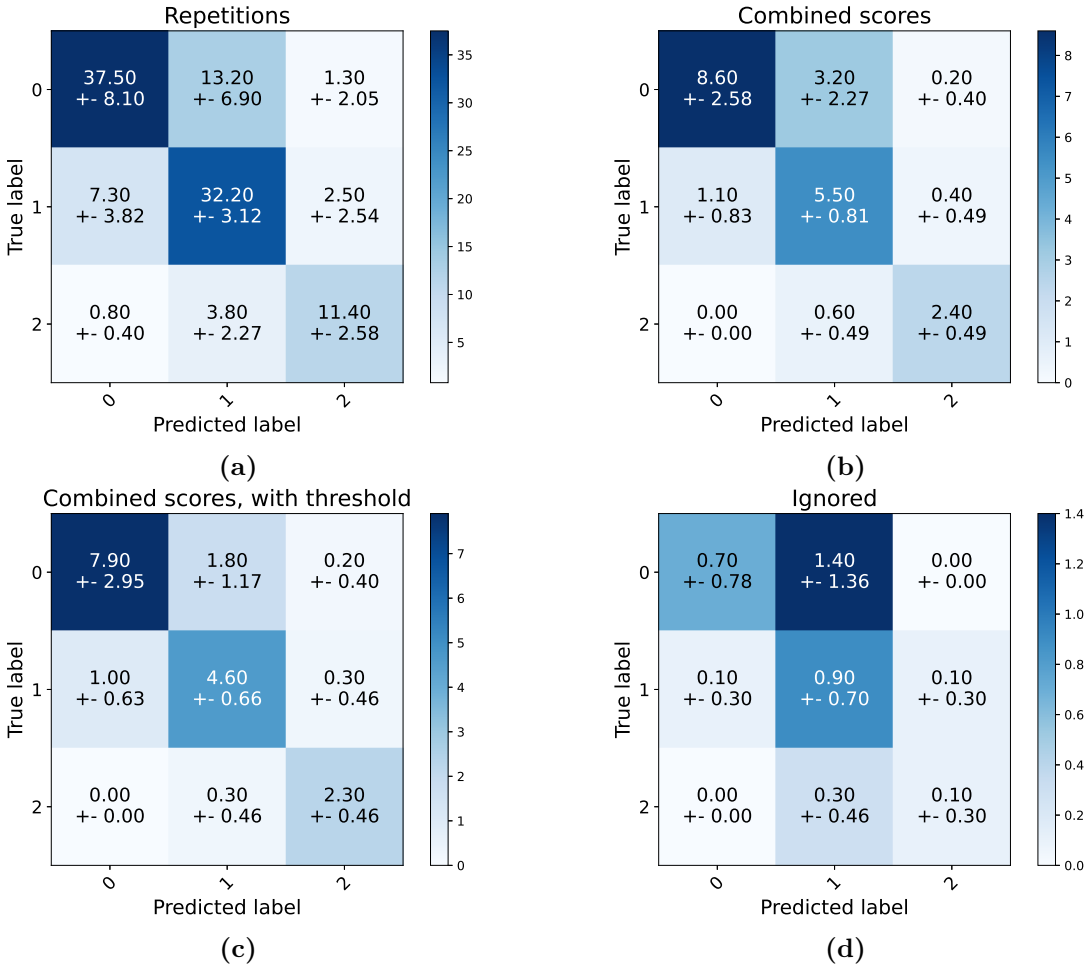


Figure 6.2.1: Confusion matrices for the trunk classification on the test set. Classification of the individual repetitions is shown in (a), the combined score for the sequences of 5 repetitions is shown in (b). (c) shows the combined score with the threshold suggested in Section 5.4.5, i.e. all scores with a predicted probability higher than 0.4. The scores ignored due to this threshold are shown in (d). The entries in the matrices show the mean and standard deviation of the 10 ensembles trained in the cross validation.

Based on what is presented in Figures 6.2.1 and 6.2.2 as well as Table 6.2.2, it seems like the performance of the classifier is enhanced by the measures taken. Table 6.2.4 shows this is true with different confidence. Neither the combined score nor the use of an ensemble can be said to improve the performance significantly, but the combined score for the five repetitions is not primarily done to improve the performance, instead this is the way the scoring system is designed. Regarding the ensemble it might be difficult to say how big of an improvement it is based on these metrics, but it reduces the variability in the results. By introducing the threshold, ignoring predictions with a predicted probability lower than 0.4, an average of 3.6 sequences are overlooked. Of these 1.7 were correctly classified and 1.9 were incorrect.

From Figure 6.2.1 and Table 6.2.3 it is clear that the majority of misclassifications are between the classes 0 and 1. This is a natural effect of there being more 0s and 1s than 2s in the test set. Also, asking the experts working with this assessment system these classes are generally the ones difficult to tell apart. Another sign that the model to some extent aligns with the assessments made by the human is the better performance for the

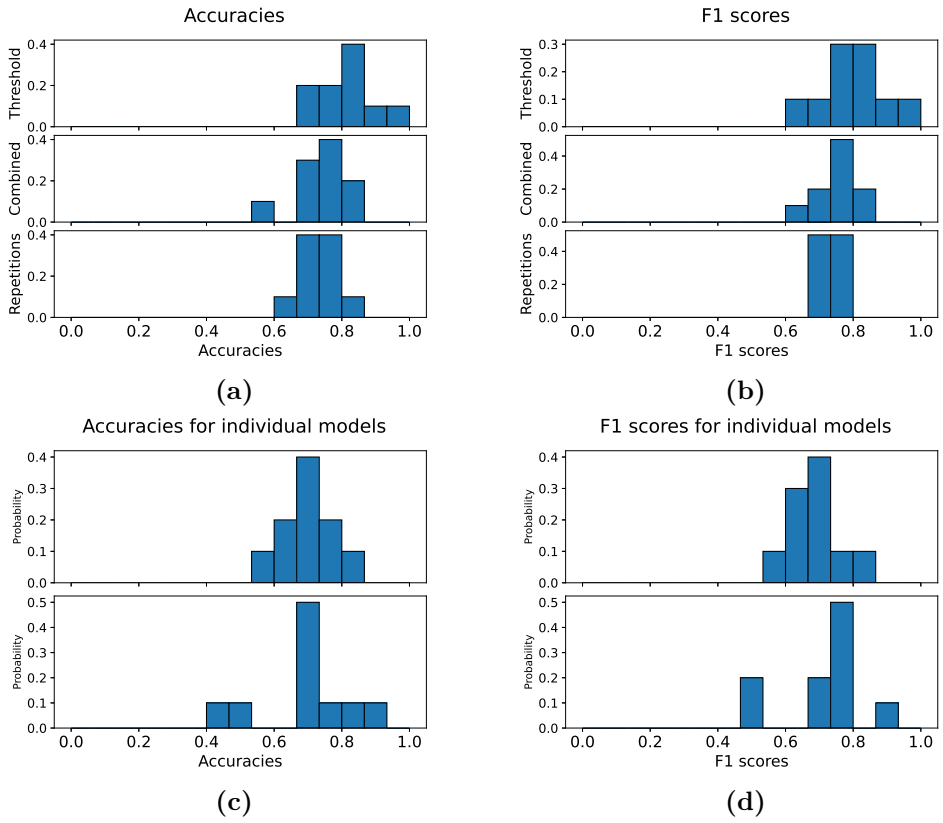


Figure 6.2.2: Histograms of the accuracies and F1 scores summarized in Table 6.2.2 along with the same metrics for the repetition classification for the models making up the ensembles, presented in Table 5.4.1. The high precision models only predicting one class are excluded.

Table 6.2.3: The class distribution in the test data for the trunk POE.

Class	0, Good	1, Fair	2, Poor
Proportion (%)	47.3	38.2	14.5

Table 6.2.4: With what confidence different measures led to improvements, i.e, a higher number means we can be more certain that the performance is increased by performing the corresponding measure. Calculated assuming normal distributions and using pairwise comparisons for the folds. When comparing the ensemble with the individual models the best model is chosen.

	Ensemble - individual models	**Combined - Repetitions**	**Threshold - Combined**
Accuracy	85%	75%	95%
F1 score	75%	85%	95%

sequences where the human expert were certain about the class, seen in Table 6.2.2.

In the results above, a threshold to ignore uncertain classifications and thereby increase the performance was used. What can be seen as an extension to such a threshold, and something important for clinical use, would be to provide a confidence in the classification. Figure 6.2.3 shows the probabilities for the predicted class depending on which the correct class actually is. This is the output the threshold acts on, ignoring any sample with a probability lower than 0.4. As suggested in the figure, this measure could be a suitable metric to use as prediction confidence, with the model rarely predicting incorrectly when it outputs a high class probability. However, although the ensemble weights has been adjusted as discussed in Section 5.4.2, the probabilities for class 1 is generally low.

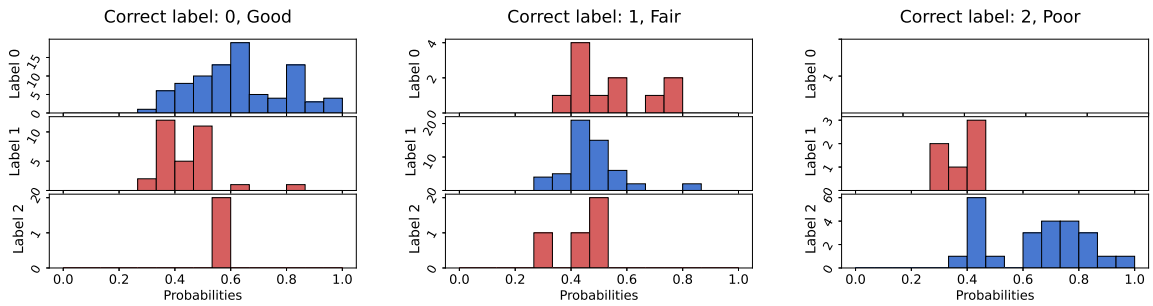


Figure 6.2.3: Figures showing the probabilities for the predicted class, i.e., the argmax of the model output, without threshold, for correct class 0: (a), 1: (b), and 2: (c). Incorrect predictions are shown in red.

6.2.3 Pelvis

As in the previous section, the pelvis results are presented in Table 6.2.5, showing the accuracies, F1 scores, recall, and precisions, and in the confusion matrices in Figure 6.2.4. Along with this, histograms showing the effect of the ensemble, combined score, and threshold can be seen in Figure 6.2.5 and these effects are also summarized in Table 6.2.7. The class distribution in the test set is presented in Table 6.2.6.

Table 6.2.5: Results of the ensemble for the pelvis POE. Rep., Comb., and Thresh. represents the results for the repetitions, combinations, and combinations with thresholds, respectively. The Certainties columns show the results making up the Comb. column, but for the certainty levels of the expert labeling the data. These ranges from certain (0) to uncertain (2), the variable n shows how many datapoints each category contains. All results are the mean from the 10 folds ± the corresponding standard deviations. F1, recall, and precision are macro averaged.

	Rep.	**Comb.**	**Thresh.**	**Certainties**		
				0(n=14)	1(n=7)	2(n=1)
Accuracy (%)	63.6±10.7	69.1±10.1	**73.3±18.9**	67.9±15.0	70.0±19.6	**80.0±40.0**
F1 score (%)	55.7±13.4	66.5±15.0	**73.6±22.3**	58.0±20.1	**63.7±23.8**	31.7±17.4
Recall (%)	57.3±12.0	75.3±14.0	**77.9±14.9**	56.8±20.7	**64.6±22.8**	31.7±17.4
Precision (%)	58.3±14.9	67.0±14.4	**74.9±23.8**	62.5±19.7	**68.4±24.0**	31.7±17.4

Overall it is clear that the performance is worse for the pelvis as compared to the other POEs. Although the different scores are not that much lower compared to the trunk POE they vary much more making it difficult to draw any conclusions from it. It also looks less like a normal distribution, making Table 6.2.7 unreliable. One explanation for this behavior is that the human experts considers this POE to be the most difficult one to assess of the four considered in work. The reason for this is that the hip rotates in several planes making it difficult to assess in 2D. This probably affects the models, both directly as a more difficult pattern to identify, which might rely on information not available in this 2D data. It might also affect the models indirectly as this suggests that the training labels can be more unreliable. The higher uncertainty can also be seen in Figure 6.2.4d as more sequences are ignored due to the threshold, 7.6 on average in this case. The high number of ignored samples is one explanation for the higher variance in the different metrics after applying the threshold.

Table 6.2.6: The class distribution in the test data for the pelvis POE.

Class	0, Good	1, Fair	2, Poor
Proportion (%)	39.1	53.6	7.3

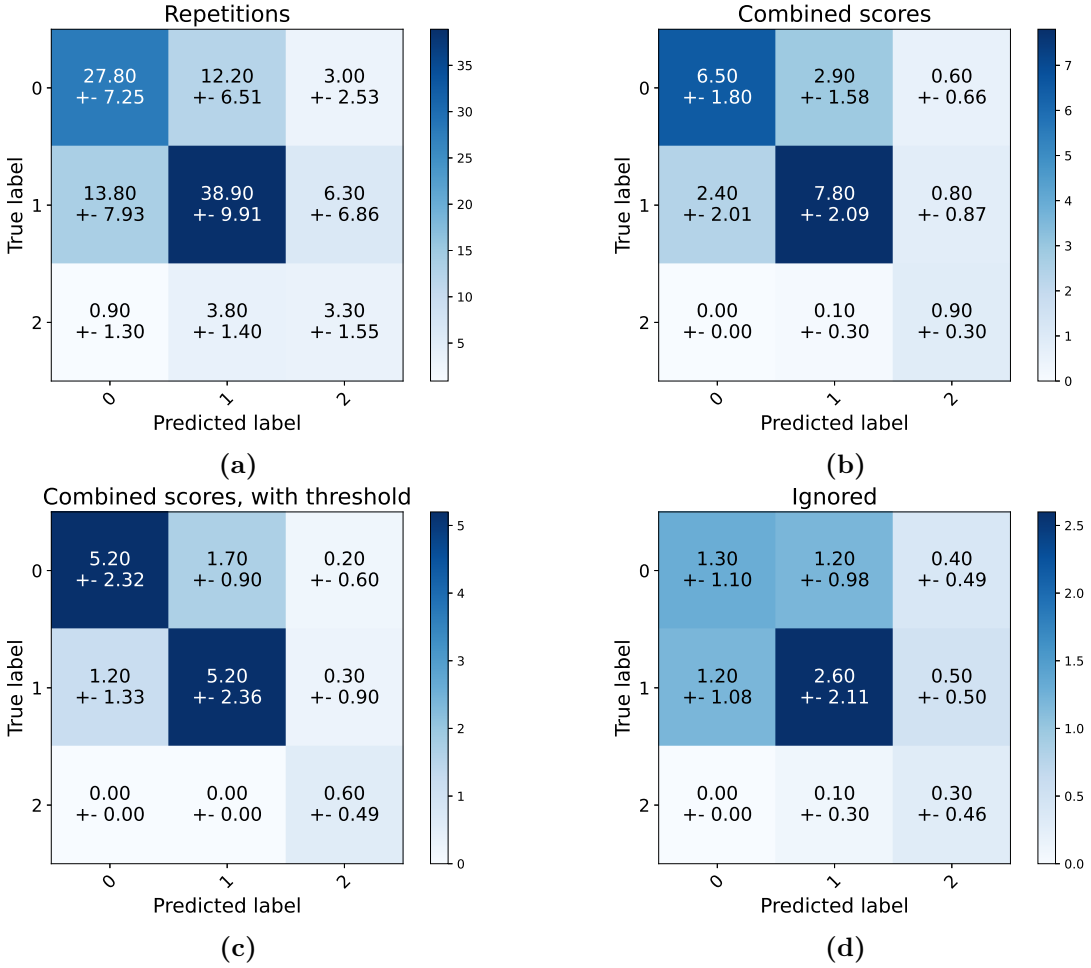


Figure 6.2.4: Confusion matrices for the pelvis classification on the test set. Classification of the individual repetitions is shown in (a), the combined score for the sequences of 5 repetitions is shown in (b). (c) shows the combined score with the threshold suggested in Section 5.4.5, i.e. all scores with a predicted probability higher than 0.4. The scores ignored due to this threshold are shown in (d). The entries in the matrices show the mean and standard deviation of the 10 ensembles trained in the cross validation.

Table 6.2.7: With what confidence different measures led to improvements, i.e, a higher number means we can be more certain that the performance is increased by performing the corresponding measure. Calculated assuming normal distributions and using pairwise comparisons for the folds. When comparing the ensemble with the individual models the best model is chosen.

	Ensemble - individual models	**Combined - Repetitions**	**Threshold - Combined**
Accuracy	-	95%	85%
F1 score	-	95%	95%

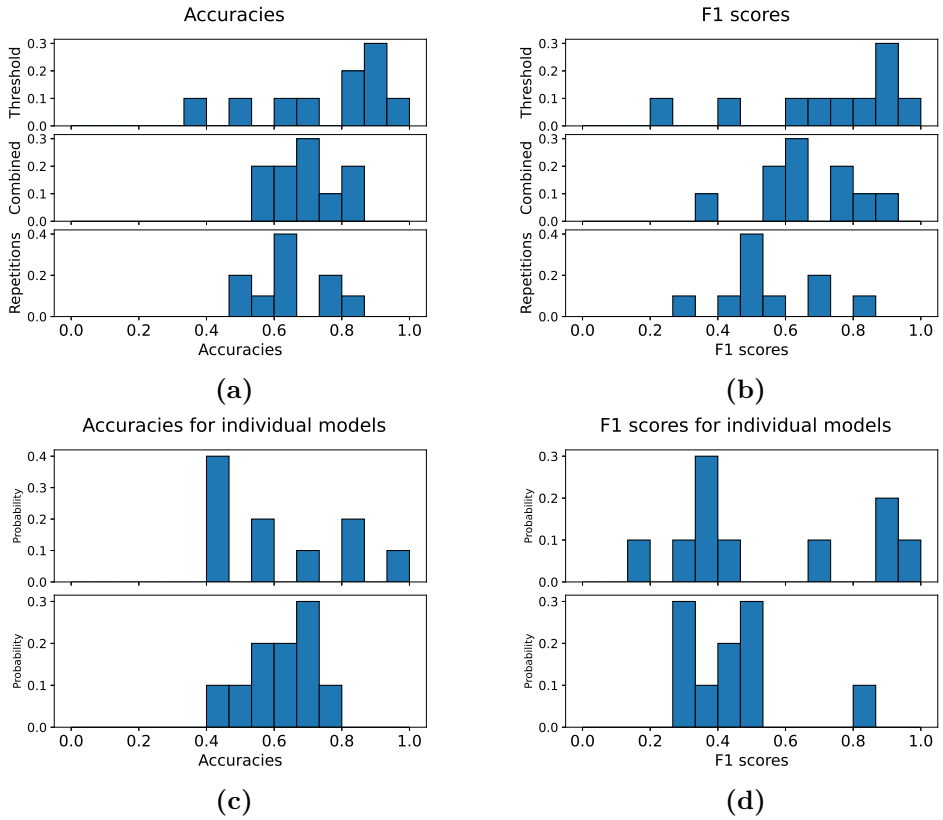


Figure 6.2.5: Histograms of the accuracies and F1 scores summarized in Table 6.2.5 along with the same metrics for the repetition classification for the models making up the ensembles, presented in Table 5.4.1. The high precision models only predicting one class are excluded.

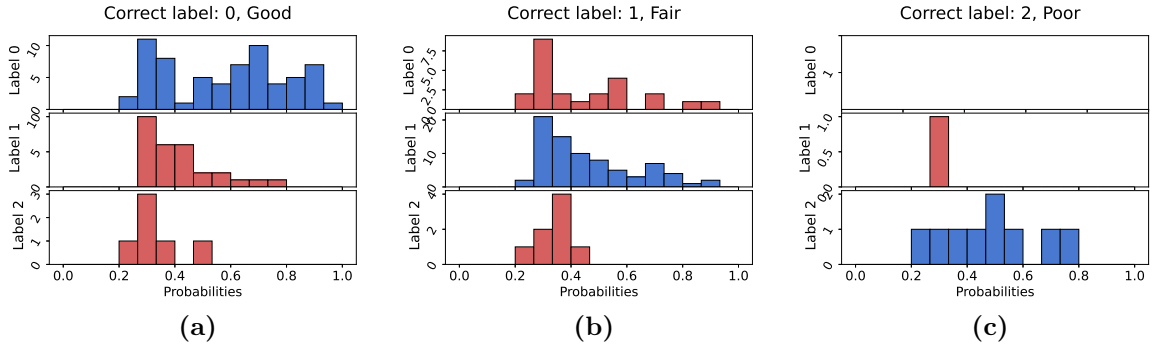


Figure 6.2.6: Figures showing the probabilities for the predicted class, i.e., the argmax of the model output, without threshold, for correct class 0: (a), 1: (b), and 2: (c). Incorrect predictions are shown in red.

6.2.4 Femoral Valgus

The results for the femoral valgus POE can be found in Figures 6.2.7, 6.2.8, and 6.2.9 as well as in Tables 6.2.8 and 6.2.10. This model performs well, but it seems to be slightly biased, at least for this data, towards class 1. To some extent, this can be explained by the class distribution, found in Table 6.2.9. On average 2.2 samples are ignored because of the threshold and 1.1 of these were correct. Notable is that it classifies a class 2 as a 0 three times and none of these are ignored due to the threshold. As can be seen in Figure 6.2.9, two of these have a probability right above the threshold of 0.4.

Table 6.2.8: Results of the ensemble for the femoral valgus POE. Rep., Comb., and Thresh. represents the results for the repetitions, combinations, and combinations with thresholds, respectively. The Certainties columns show the results making up the Comb. column, but for the certainty levels of the expert labeling the data. These ranges from certain (0) to uncertain (2), the variable n shows how many datapoints each category contains. All results are the mean from the 10 folds ± the corresponding standard deviations. F1, recall, and precision are macro averaged.

	Rep.	**Comb.**	**Thresh.**	**Certainties**		
				0(n=15)	1(n=7)	2(n=0)
Accuracy (%)	69.6±6.8	79.1±9.3	**82.3±6.0**	**82.7±9.5**	71.4±11.0	-
F1 score (%)	69.0±6.6	77.6±8.6	**81.0±5.2**	**80.8±8.6**	70.1±9.4	-
Recall (%)	68.4±6.3	76.3±8.1	**79.5±5.5**	**80.4±7.4**	71.1±9.5	-
Precision (%)	74.0±7.4	83.8±7.4	**86.5±5.6**	**85.2±7.6**	81.8±6.6	-

Clearly, the performance is improved by the calculating the combined score and applying the threshold. For this POE, the variance is also decreased by introducing the threshold, which was not the case for trunk, pelvis, nor KMFP. As can be seen in Table 6.2.10, the ensemble is not improving the performance, but it gives a more robust model neglecting the poor performance which can be seen in Figures 6.2.8c and 6.2.8d for the individual models for one of the folds.

Table 6.2.9: The class distribution in the test data for the femoral valgus POE.

Class	0, Good	1, Fair	2, Poor
Proportion (%)	32.7	40.0	27.3

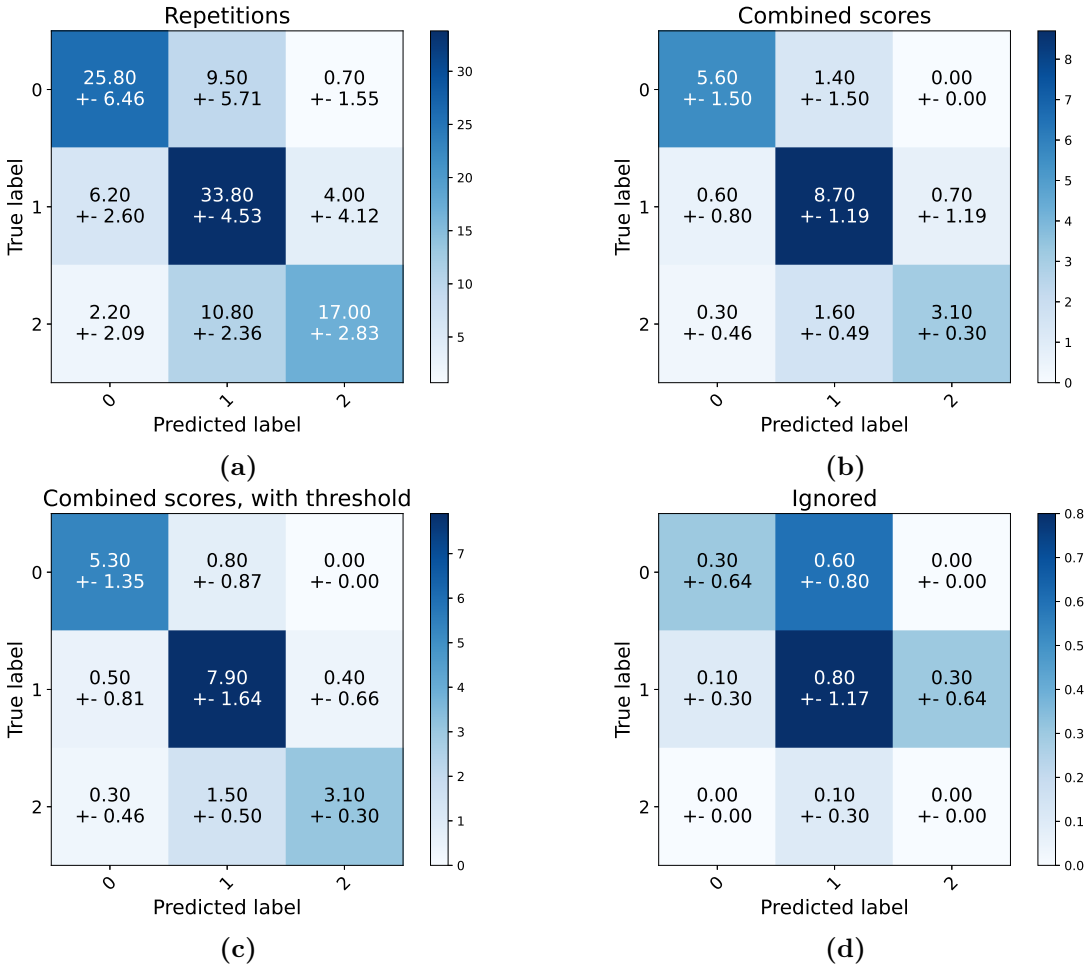


Figure 6.2.7: Confusion matrices for the femoral valgus classification on the test set. Classification of the individual repetitions is shown in (a), the combined score for the sequences of 5 repetitions is shown in (b). (c) shows the combined score with the threshold suggested in Section 5.4.5, i.e. all scores with a predicted probability higher than 0.4. The scores ignored due to this threshold are shown in (d). The entries in the matrices show the mean and standard deviation of the 10 ensembles trained in the cross validation.

Table 6.2.10: With what confidence different measures led to improvements, i.e, a higher number means we can be more certain that the performance is increased by performing the corresponding measure. Calculated assuming normal distributions and using pairwise comparisons for the folds. When comparing the ensemble with the individual models the best model is chosen.

	Ensemble - individual models	**Combined - Repetitions**	**Threshold - Combined**
Accuracy	-	95%	95%
F1 score	-	95%	95%

As for the trunk model, the probability for class one is low. This is clearly not a problem for the classification, as mentioned above this model seems to be slightly biased towards class 1. It is however something to consider for the confidence score desirable in a clinical setting.

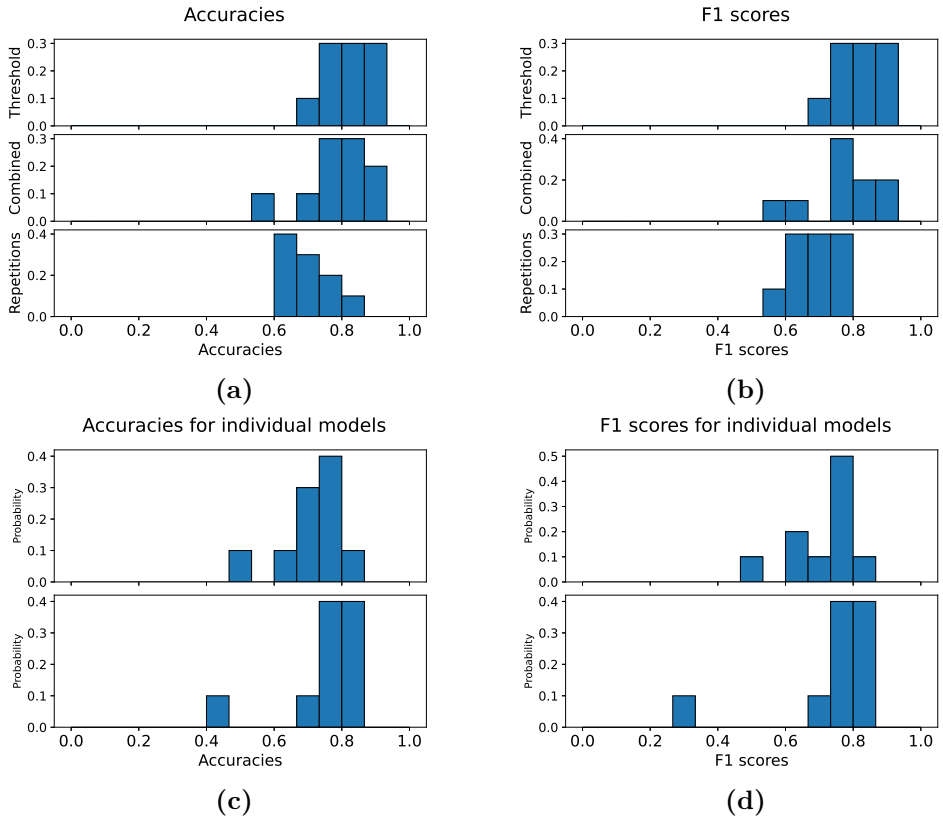


Figure 6.2.8: Histograms of the accuracies and F1 scores summarized in Table 6.2.8 along with the same metrics for the repetition classification for the models making up the ensembles, presented in Table 5.4.1. The high precision models only predicting one class are excluded.

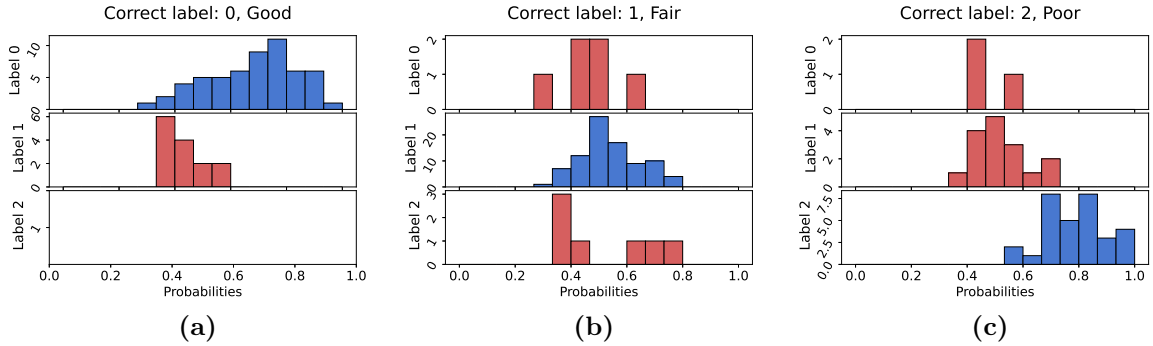


Figure 6.2.9: Figures showing the probabilities for the predicted class, i.e., the argmax of the model output, without threshold, for correct class 0: (a), 1: (b), and 2: (c). Incorrect predictions are shown in red.

With the ensemble consisting of different models, some of the explainability from the X-InceptionTime architecture is lost. It is still possible to get importance values for each time step, but the regular models will not differentiate between the inputs. Figure 6.2.10 shows the activation maps for some different inputs to a single X-InceptionTime model, both for correctly and incorrectly classified samples. It can be seen how the same regions seems to be deemed important for the same predictions.

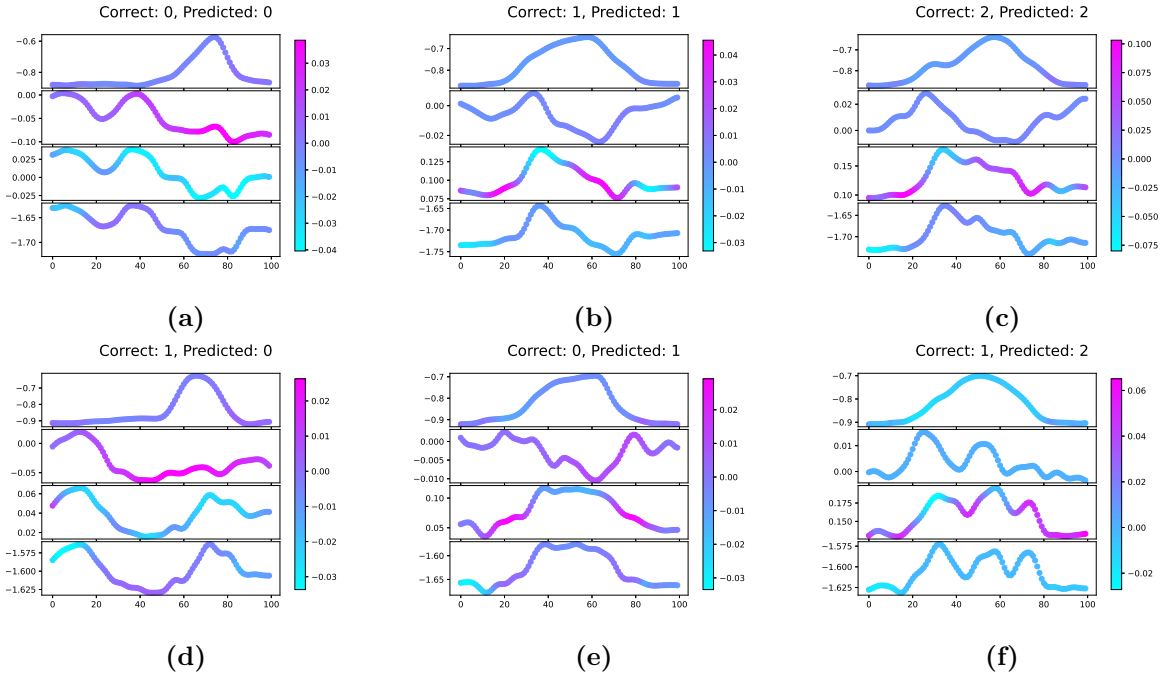


Figure 6.2.10: Activation maps applied to different inputs showing what was important for decision. The subplots show from the top: y coordinate of the right shoulder, x coordinate of the right hip, x coordinate of the right knee, and angle between right knee and right ankle.

6.2.5 Knee Medial-to-Foot Position

As can be seen in Table 6.2.11, the KMFP data is heavily unbalanced making it somewhat difficult to analyze. As long as this training set is a representable class distribution this model might perform satisfactory. However, it has seen very few examples of both class 1 and 2, so the risk of out-of-distribution samples if deploying this model is probably rather high. Regarding the performance metrics this model achieves a good accuracy which is natural since it will do so by just classifying most data as class 0. Based on the other metrics and the confusion matrices it becomes clear that it does not simply do this. To avoid such behavior some of the models used in the ensemble were slightly biased towards class 1 and 2. This shows in the result mainly as a low precision for class 1 compared to the other POEs. The low standard deviation for the accuracy shows that the number of misclassification is similar for all examples. Due to the macro averaging of the other metrics together with the class imbalance, variations in a single classification of class 1 or 2 results in the large standard deviations seen for F1, recall, and precision.

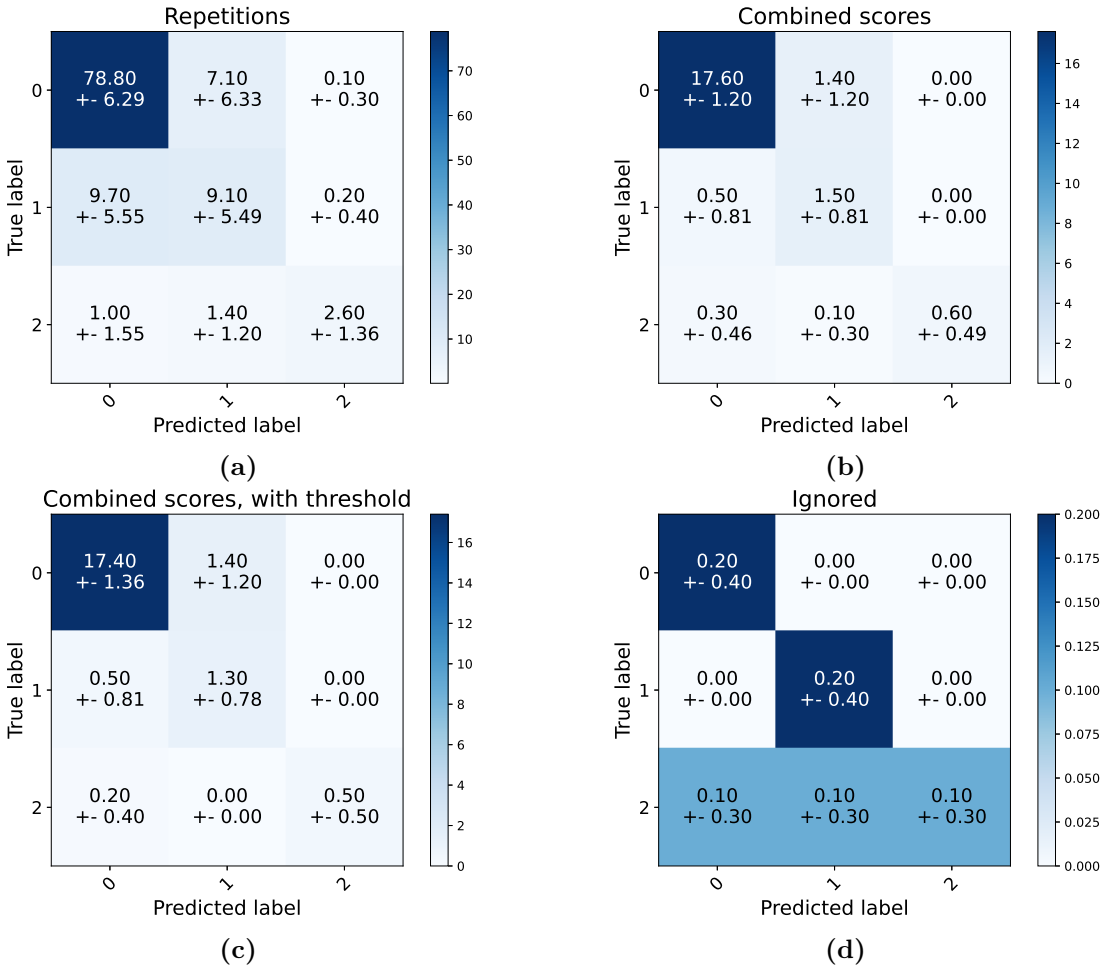


Figure 6.2.11: Confusion matrices for the KMFP classification on the test set. Classification of the individual repetitions is shown in (a), the combined score for the sequences of 5 repetitions is shown in (b). (c) shows the combined score with the threshold suggested in Section 5.4.5, i.e. all scores with a predicted probability higher than 0.4. The scores ignored due to this threshold are shown in (d). The entries in the matrices show the mean and standard deviation of the 10 ensembles trained in the cross validation.

Table 6.2.11: The class distribution in the test data for the KMFP POE.

Class	0, Good	1, Fair	2, Poor
Proportion (%)	78.2	17.3	4.5

Table 6.2.12: Results of the ensemble for the KMFP POE. Rep., Comb., and Thresh. represents the results for the repetitions, combinations, and combinations with thresholds, respectively. The Certainties columns show the results making up the Comb. column, but for the certainty levels of the expert labeling the data. These ranges from certain (0) to uncertain (2), the variable n shows how many datapoints each category contains. All results are the mean from the 10 folds ± the corresponding standard deviations. F1, recall, and precision are macro averaged.

	Rep.	**Comb.**	**Thresh.**	**Certainties**		
				0(n=19)	1(n=2)	2(n=1)
Accuracy (%)	82.3±3.1	89.5±4.5	**90.3±4.3**	**92.6±5.3**	90.0±20.0	30.0±45.8
F1 score (%)	65.1±13.7	69.0±24.6	**74.0±22.3**	**69.3±28.6**	57.8±17.8	10.0±15.3
Recall (%)	63.8±14.4	75.9±26.7	**81.1±24.7**	**75.3±29.4**	60.0±13.3	10.0±15.3
Precision (%)	**78.5±8.2**	66.1±24.0	71.4±22.6	**67.2±28.9**	56.7±20.0	10.0±15.2

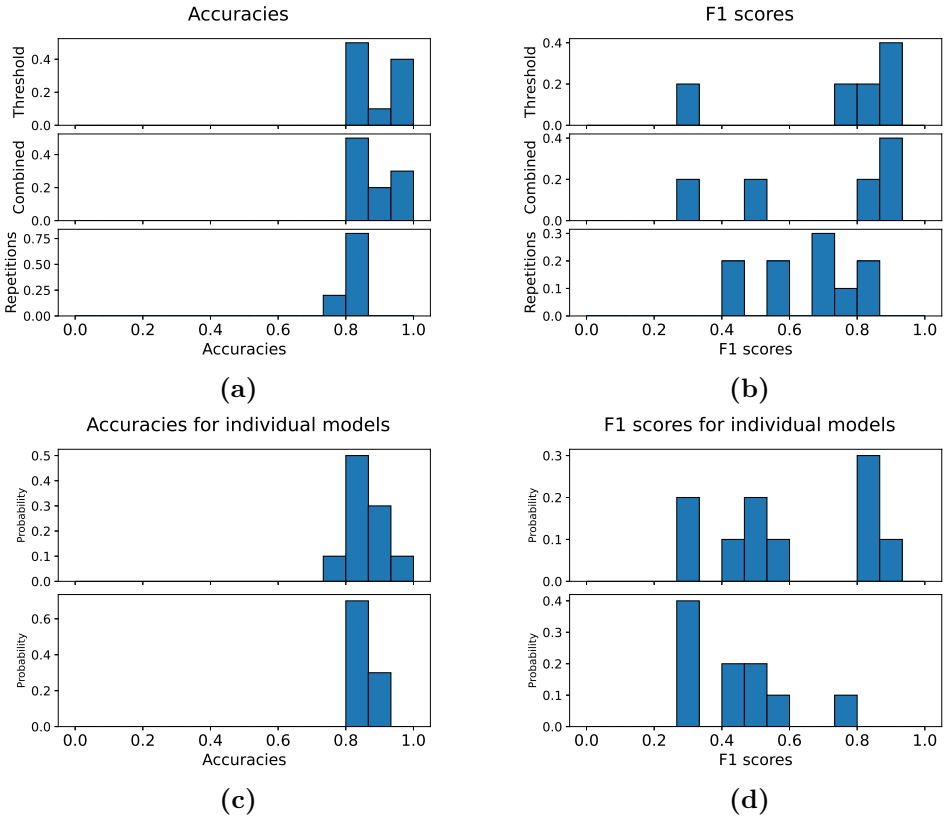
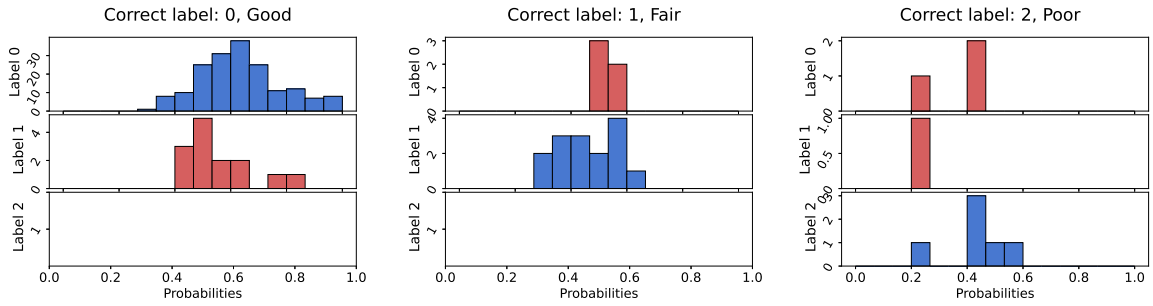


Figure 6.2.12: Histograms of the accuracies and F1 scores summarized in Table 6.2.12 along with the same metrics for the repetition classification for the models making up the ensembles, presented in Table 5.4.1. The high precision models only predicting one class are excluded.

Table 6.2.13: With what confidence different measures led to improvements, i.e, a higher number means we can be more certain that the performance is increased by performing the corresponding measure. Calculated assuming normal distributions and using pairwise comparisons for the folds. When comparing the ensemble with the individual models the best model is chosen.

	Ensemble - individual models	**Combined - Repetitions**	**Threshold - Combined**
Accuracy	-	95%	95%
F1 score	-	95%	95%



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

Figure 6.2.13: Figures showing the probabilities for the predicted class, i.e., the argmax of the model output, without threshold, for correct class 0: (a), 1: (b), and 2: (c). Incorrect predictions are shown in red.

6.2.6 Summary

The results with their 95% confidence intervals are summarized in Table 6.2.14. These confidence intervals assumes Gaussian distributions, which is not unrealistic for trunk, femoral valgus, and to some extent KMFP. For the pelvis however, this assumption does not seem to be very reasonable. Based on this, it can be said that the model for femoral valgus seems to perform the best followed by trunk. KMFP achieves a high accuracy, but it is difficult to draw any conclusions about the performance for the Fair (1) or Poor (2) classes due to the very few examples. The pelvis model does not perform very well which might be due to that it is based on information not available in the 2D joint position data. For all POEs the developed models clearly outperforms the baseline methods.

The only difference between the different fold models is a slight variation in the training data. Still, the result, generated on the same testing data, clearly varies. This suggests that more data would be very welcome for all models. Both to improve the performance through bigger training sets and to give more reliable evaluations thanks to more testing data.

Table 6.2.14: Approximative 95% confidence intervals for the performance of the combined scores with thresholds for different POEs, assuming Gaussian distributions.

	Trunk	**Pelvis**	**Femoral Valgus**	**KMFP**
Accuracy (%)	80.0±4.9	73.3±11.9	82.3±3.8	**90.3±2.7**
F1 score (%)	79.9±5.6	73.6±14.1	**81.0±3.3**	74.0±14.1
Recall (%)	**81.6±4.6**	77.9±9.4	79.5±3.5	81.1±15.6
Precision (%)	83.0±4.3	74.9±15.1	**86.5±3.5**	71.4±14.3

Regarding the model selection, it is notable that padded and normalized data yields similar results. This suggests that, at least with this amount of data, the frequency content or the speed of the movements is not an indicator of the POE score. Instead, the overall shape of the movements of the different body parts are important, which matches the descriptions in Appendix A. Normalizing the data length probably unifies these shapes slightly over all data, which might make it easier to find patterns.

Another thing to note is that deep models does not seem to be effective as models with depths of one and two performed the best. Similarly, the InceptionTime model where interactions between the inputs are taken into account does not perform significantly better than X-InceptionTime. This suggests, at least with this amount of data, that the discriminating features are not that complicated and does not depend on direct interactions between the inputs, apart from difference or angles in some cases. With more data this might be different as new patterns could be found.

Chapter 7

Conclusions and Future work

As discussed in the introduction, the goal with this work was to evaluate whether the POE assessments could be automated using these kinds of approaches, and to get an idea of what is needed to make that a reality. Based on the results in this thesis, this seems promising, clearly it is possible to assess the POEs from the videos. To further assess the model, more data is needed. Assessments made by several experts, providing a more reliable ground truth, would be beneficial from both a training and validation perspective. The current assessments are made by the physiotherapist who designed the method. Beyond these, it would be interesting to have assessments made by others to provide a more valuable baseline comparison.

One commonly discussed risk when using machine learning methods on data including humans is that unwanted biases from the training data should affect the decisions. Such biases usually includes race or sex. In this case an example could be that women, as they suffer more from ACL injuries, also might have more issues with poor postural orientation. An undesirable behavior would then be for the model to use features such as long hair and certain clothes, typically identifying women, to affect the decision. As no training is performed in the pixel space, the risk of such biases are probably reduced. How to avoid bias in the labeled data is discussed below.

7.1 Future work and improvements

As discussed in Chapter 6, the performance for the pelvis POE assessment seems to be worse than the other POEs and this might be due to rotations difficult to see in the 2D joint positions. Hence, it would be interesting to investigate how well this could be done using 3D information. At the initial stages of this work the 3D reconstruction method by Pavllo et al. [52] was briefly evaluated, but a decision was taken to focus on the modeling of the assessments. As about half of the available videos were recorded in a motion capture lab there are 3D information available for these allowing some 3D estimation model to be trained or fine tuned for our movements.

The data used to train and evaluate our method was labeled by one physiotherapist. To make sure bias from this person does not find its way into the data and model assessments from other physiotherapists should be used as well. The current data has been labeled on a per video basis which might introduce bias towards the surrounding repetitions.

To split the videos or record single repetition videos and shuffle these before assessment would probably be desirable for the future. When doing this a repetition certainty score could also be gathered which could be used during training. This was evaluated with the certainty data available at the moment as well. However, it did not yield any model improvements which was rather reasonable as this data was gathered for the combined scores and not the repetitions. The ability to give a measure of the model's confidence is important for use in a clinical setting. This can be used to, for instance, ask the patient to repeat the task or, if still uncertain, ask a physiotherapist to assess certain videos and thereby hopefully avoid misclassifications. As of now the output probability of the models seems to be a somewhat good measure for this, but it could potentially be improved by providing the certainty as guidance during training.

Regarding the way the videos are recorded, this method should be rather invariant to factors such as frame rate, orientation, and resolution. However, what is very important for the data collection is to make sure that important body parts are not occluded in the image. Likewise, it is very important to have a consistent recording angle for all the videos. As a human unconsciously deduces the positions of occluded objects or rotates planes this might not be considered as important when collecting the data. Hence, this is something to stress when videos are recorded.

With the current approach the lowest hanging fruit for improvements might be the combinations of scores, both combining repetition scores to the final score and the calculation scores. Potentially this could be improved by using some non-linear scoring combination or by learning functions for this as well. Furthermore, combining the proposed method with some of the more traditional features could improve the results. Although the baseline methods did not perform nearly as good, it is probably reasonable to assume that some features contain valuable information which, due to the limited amount of data, might be difficult to find in other ways.

Bibliography

[1] Hussein A. Abbass, Jason Scholz, and Darryn J. Reid. *Foundations of Trusted Autonomy*. Springer, 2018. DOI: 10.1007/978-3-319-64816-3.

[2] Eva Ageberg. "Consequences of a ligament injury on neuromuscular function and relevance to rehabilitation — using the anterior cruciate ligament-injured knee as model". In: *Journal of Electromyography and Kinesiology* 12.3 (2002). Sensory Function of Ligaments, pp. 205–212. ISSN: 1050-6411. DOI: 10.1016/S1050-6411(02)00022-6.

[3] Alan Agresti. *An Introduction to Categorical Data Analysis (2nd ed.)* Hoboken, New Jersey: Wiley, 2007.

[4] Jenny Älmqvist Nae. "Is seeing just believing? Measurement properties of visual assessment of Postural Orientation Errors (POEs) in people with anterior cruciate ligament injury". English. PhD thesis. Department of Health Sciences, June 2020. ISBN: 978-91-7619-940-4.

[5] Julia Amann, Alessandro Blasimme, Effy Vayena, Dietmar Frey, and Vince I. Madai. "Explainability for artificial intelligence in healthcare: a multidisciplinary perspective". In: *BMC Medical Informatics and Decision Making* 20.1 (Dec. 2020), p. 310. ISSN: 14726947. DOI: 10.1186/s12911-020-01332-6.

[6] A. Bagnall, J. Lines, J. Hills, and A. Bostrom. "Time-Series Classification with COTE: The Collective of Transformation-Based Ensembles". In: *IEEE Transactions on Knowledge and Data Engineering* 27.9 (2015), pp. 2522–2535. DOI: 10.1109/TKDE.2015.2416723.

[7] Anthony Bagnall, Jason Lines, Aaron Bostrom, James Large, and Eamonn Keogh. "The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances". In: *Data Mining and Knowledge Discovery* 31.3 (May 2017), pp. 606–660. ISSN: 1573756X. DOI: 10.1007/s10618-016-0483-9.

[8] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[9] Wenzhi Cao, Vahid Mirjalili, and Sebastian Raschka. "Rank consistent ordinal regression for neural networks with application to age estimation". In: *Pattern Recognition Letters* 140 (Jan. 2019), pp. 325–331. DOI: 10.1016/j.patrec.2020.11.008. arXiv: 1901.07884.

[10] Yucheng Chen, Yingli Tian, and Mingyi He. "Monocular Human Pose Estimation: A Survey of Deep Learning-based Methods". In: *Computer Vision and Image Understanding* 192 (June 2020). DOI: 10.1016/j.cviu.2019.102897. arXiv: 2006.01423.

[11] Bowen Cheng, Bin Xiao, Jingdong Wang, Honghui Shi, Thomas S. Huang, and Lei Zhang. "HigherHRNet: Scale-Aware Representation Learning for Bottom-Up Human Pose Estimation". In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (Aug. 2019), pp. 5385–5394. arXiv: 1908.10357.

[12] François Chollet. *Deep Learning with Python*. Manning, 2018.

[13] Mayo Clinic. *ACL injury*. 2019. URL: https://www.mayoclinic.org/diseases-conditions/acl-injury/symptoms-causes/syc-20350738 (visited on 03/05/2021).

[14] R. Crichlow, P. Andres, S. Morrison, S. Haley, and M. Vrahas. "Depression in orthopaedic trauma patients. Prevalence and severity." In: *The Journal of bone and joint surgery. American volume* 88 9 (2006), pp. 1927–33.

[15] Hoang Anh Dau, Anthony Bagnall, Kaveh Kamgar, Chin-Chia Michael Yeh, Yan Zhu, Shaghayegh Gharghabi, Chotirat Ann Ratanamahatana, and Eamonn Keogh. "The UCR Time Series Archive". In: *IEEE/CAA Journal of Automatica Sinica* 6.6 (Oct. 2018), pp. 1293–1305. arXiv: 1810.07758.

[16] Mengnan Du, Ninghao Liu, and Xia Hu. "Techniques for Interpretable Machine Learning". In: *arXiv* (July 2018). arXiv: 1808.00033.

[17] V. B. Duthon, C. Barea, S. Abrassart, J. H. Fasel, D. Fritschy, and J. Menetrey. *Anatomy of the anterior cruciate ligament*. Mar. 2006. DOI: 10.1007/s00167-005-0679-9.

[18] Kevin Fauvel, Tao Lin, Véronique Masson, Élisa Fromont, and Alexandre Termier. "XCM: An Explainable Convolutional Neural Network for Multivariate Time Series Classification". In: (Sept. 2020). arXiv: 2009.04796.

[19] Martin A. Fischler and Robert A. Elschlager. "The Representation and Matching of Pictorial Structures Representation". In: *IEEE Transactions on Computers* C-22.1 (1973), pp. 67–92. ISSN: 00189340. DOI: 10.1109/T-C.1973.223602.

[20] Kunihiko Fukushima. "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position". In: *Biological Cybernetics* 36.4 (Apr. 1980), pp. 193–202. ISSN: 03401200. DOI: 10.1007/BF00344251.

[21] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. http://www.deeplearningbook.org. MIT Press, 2016.

[22] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep residual learning for image recognition". In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Vol. 2016-December. IEEE Computer Society, Dec. 2016, pp. 770–778. ISBN: 9781467388504. DOI: 10.1109/CVPR.2016.90. arXiv: 1512.03385.

[23] Timothy E. Hewett, Gregory D. Myer, Kevin R. Ford, Robert S. Heidt Jr., Angelo J. Colosimo, Scott G. McLean, Antonie J. van den Bogert, Mark V. Paterno, and Paul Succop. "Biomechanical Measures of Neuromuscular Control and Valgus Loading of the Knee Predict Anterior Cruciate Ligament Injury Risk in Female Athletes: A Prospective Study". In: *The American Journal of Sports Medicine* 33.4 (2005), pp. 492–501. DOI: 10.1177/0363546504269591.

[24] Fay B. Horak. "Postural orientation and equilibrium: what do we need to know about neural control of balance to prevent falls?" In: *Age and Ageing* 35.suppl$_2$ (Sept. 2006), pp. ii7–ii11. ISSN: 0002-0729. DOI: 10.1093/ageing/afl077.

[25] H. Ismail Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P. Muller. "Deep Neural Network Ensembles for Time Series Classification". In: *2019 International Joint Conference on Neural Networks (IJCNN)*. 2019, pp. 1–6. DOI: 10.1109/IJCNN.2019.8852316.

[26] Hassan Ismail Fawaz, Germain Forestier, Jonathan Weber, Lhassane Idoumghar, and Pierre Alain Muller. "Deep learning for time series classification: a review". In: *Data Mining and Knowledge Discovery* 33.4 (July 2019), pp. 917–963. ISSN: 1573756X. DOI: 10.1007/s10618-019-00619-1. arXiv: 1809.04356.

[27] Hassan Ismail Fawaz, Benjamin Lucas, Germain Forestier, Charlotte Pelletier, Daniel F. Schmidt, Jonathan Weber, Geoffrey I. Webb, Lhassane Idoumghar, Pierre Alain Muller, and François Petitjean. "InceptionTime: Finding AlexNet for time series classification". In: *Data Mining and Knowledge Discovery* 34.6 (Nov. 2020), pp. 1936–1962. ISSN: 1573756X. DOI: 10.1007/s10618-020-00710-y. arXiv: 1909.04939.

[28] A. G. Ivakhnenko and V. G. Lapa. *Cybernetic Predicting Devices*. CCM Information Corpo-ration, 1965.

[29] Sheng Jin, Lumin Xu, Jin Xu, Can Wang, Wentao Liu, Chen Qian, Wanli Ouyang, and Ping Luo. "Whole-Body Human Pose Estimation in the Wild". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 12354 LNCS (July 2020), pp. 196–214. arXiv: 2007.11858.

[30] R. Kianifar, A. Lee, S. Raina, and D. Kulić. "Classification of squat quality with inertial measurement units in the single leg squat mobility test". In: *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. 2016, pp. 6273–6276. DOI: 10.1109/EMBC.2016.7592162.

[31] Diederik P. Kingma and Jimmy Lei Ba. "Adam: A method for stochastic optimization". In: *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*. International Conference on Learning Representations, ICLR, Dec. 2015. arXiv: 1412.6980.

[32] Matthias Krause, Fabian Freudenthaler, Karl-Heinz Frosch, Andrea Achtnich, Wolf Petersen, and Ralph Akoto. "Operative Versus Conservative Treatment of Anterior Cruciate Ligament Rupture". In: *Dtsch Arztebl International* 115.51-52 (2018), pp. 855–862. DOI: 10.3238/arztebl.2018.0855.

[33] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In: ed. by Peter L. Bartlett, Fernando C. N. Pereira, Christopher J. C. Burges, Léon Bottou, and Kilian Q. Weinberger. 2012, pp. 1106–1114.

[34] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. "Backpropagation Applied to Handwritten Zip Code Recognition". In: *Neural Computation* 1.4 (Dec. 1989), pp. 541–551. ISSN: 0899-7667. DOI: 10.1162/neco.1989.1.4.541.

[35] Ling Li and Hsuan-tien Lin. "Ordinal Regression by Extended Binary Classification". In: *Advances in Neural Information Processing Systems*. Ed. by B. Schölkopf, J. Platt, and T. Hoffman. Vol. 19. MIT Press, 2007, pp. 865–872.

[36] Y. Liao, A. Vakanski, and M. Xian. "A Deep Learning Framework for Assessing Physical Rehabilitation Exercises". In: *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 28.2 (2020), pp. 468–477. DOI: 10.1109/TNSRE.2020.2966249.

[37] Tsung Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. "Microsoft COCO: Common objects in context". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Vol. 8693 LNCS. PART 5. Springer Verlag, May 2014, pp. 740–755. DOI: 10.1007/978-3-319-10602-1_48. arXiv: 1405.0312.

[38] J. Lines, S. Taylor, and A. Bagnall. "HIVE-COTE: The Hierarchical Vote Collective of Transformation-Based Ensembles for Time Series Classification". In: *2016 IEEE 16th International Conference on Data Mining (ICDM)*. 2016, pp. 1041–1046. DOI: 10.1109/ICDM.2016.0133.

[39] Jason Lines and Anthony Bagnall. "Time series classification with ensembles of elastic distance measures". In: *Data Mining and Knowledge Discovery* 29.3 (Apr. 2015), pp. 565–592. ISSN: 13845810. DOI: 10.1007/s10618-014-0361-2.

[40] L. Stefan Lohmander, P. Martin Englund, Ludvig L. Dahl, and Ewa M. Roos. "The Long-term Consequence of Anterior Cruciate Ligament and Meniscus Injuries: Osteoarthritis". In: *The American Journal of Sports Medicine* 35.10 (2007), pp. 1756–1769. DOI: 10.1177/0363546507307396.

[41] Warren S. McCulloch and Walter Pitts. "A logical calculus of the ideas immanent in nervous activity". In: *The Bulletin of Mathematical Biophysics* 5.4 (Dec. 1943), pp. 115–133. ISSN: 00074985. DOI: 10.1007/BF02478259.

[42] *MMPose - OpenMMLab Pose Estimation Toolbox and Benchmark*. URL: https://github.com/open-mmlab/mmpose.

[43] A. Paul Monk, Loretta J. Davies, Sally Hopewell, Kristina Harris, David J. Beard, and Andrew J. Price. "Surgical versus conservative interventions for treating anterior cruciate ligament injuries". In: *Cochrane Database of Systematic Reviews* 4 (2016). ISSN: 1465-1858. DOI: 10.1002/14651858.CD011166.pub2.

[44] Alicia M Montalvo, Daniel K Schneider, Laura Yut, Kate E Webster, Bruce Beynnon, Mininder S Kocher, and Gregory D Myer. ""What's my risk of sustaining an ACL injury while playing sports?" A systematic review with meta-analysis". In: *British Journal of Sports Medicine* 53.16 (2019). Ed. by Emir Veledar, pp. 1003–1012. ISSN: 0306-3674. DOI: 10.1136/bjsports-2016-096274.

[45] Jenny Nae, Mark W Creaby, and Eva Ageberg. "Extended Version of a Test Battery for Visual Assessment of Postural Orientation Errors: Face Validity, Internal Consistency, and Reliability". In: *Physical Therapy* 100.9 (May 2020), pp. 1542–1556. ISSN: 1538-6724. DOI: 10.1093/ptj/pzaa092.

[46] Jenny Nae, Mark W. Creaby, Gustav Nilsson, Kay M. Crossley, and Eva Ageberg. "Measurement Properties of a Test Battery to Assess Postural Orientation During Functional Tasks in Patients Undergoing ACL Injury Rehabilitation". In: *Journal of Orthopaedic & Sports Physical Therapy* 47.11 (Oct. 2017), pp. 1–42. ISSN: 0190-6011. DOI: 10.2519/jospt.2017.7270.

[47] Christopher Nagelli and Timothy Hewett. "Should Return to Sport be Delayed Until 2 Years After Anterior Cruciate Ligament Reconstruction? Biological and Functional Considerations". In: *Sports Medicine* 47 (Feb. 2017). DOI: 10.1007/s40279-016-0584-z.

[48] Md Nazmus Saadat and Muhammad Shuaib. "Advancements in Deep Learning Theory and Applications: Perspective in 2020 and beyond". In: *Advances and Applications in Deep Learning*. IntechOpen, Dec. 2020. DOI: 10.5772/intechopen.92271.

[49] Alejandro Newell, Kaiyu Yang, and Jia Deng. "Stacked hourglass networks for human pose estimation". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Vol. 9912 LNCS. Springer Verlag, 2016, pp. 483–499. ISBN: 9783319464831. DOI: 10.1007/978-3-319-46484-8_29. arXiv: 1603.06937.

[50] Zhenxing Niu, Mo Zhou, Le Wang, Xinbo Gao, and Gang Hua. "Ordinal Regression with Multiple Output CNN for Age Estimation". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 4920–4928. DOI: 10.1109/CVPR.2016.532.

[51] Mark Paterno, Mitchell Rauh, Laura Schmitt, Kevin Ford, and Timothy Hewett. "Incidence of Contralateral and Ipsilateral Anterior Cruciate Ligament (ACL) Injury After Primary ACL Reconstruction and Return to Sport". In: *Clinical journal of sport medicine : official journal of the Canadian Academy of Sport Medicine* 22 (Mar. 2012), pp. 116–21. DOI: 10.1097/JSM.0b013e318246ef9e.

[52] Dario Pavllo, Christoph Feichtenhofer, David Grangier, and Michael Auli. "3D human pose estimation in video with temporal convolutions and semi-supervised training". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019.

[53] Rajat Raina, Anand Madhavan, and Andrew Y. Ng. "Large-Scale Deep Unsupervised Learning Using Graphics Processors". In: *Proceedings of the 26th Annual International Conference on Machine Learning*. ICML '09. Montreal, Quebec, Canada: Association for Computing Machinery, 2009, pp. 873–880. ISBN: 9781605585161. DOI: 10.1145/1553374.1553486.

[54] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39.6 (June 2017), pp. 1137–1149. ISSN: 01628828. DOI: 10.1109/TPAMI.2016.2577031. arXiv: 1506.01497.

[55] D. E. Rumelhart and J. L. McClelland. "Learning Internal Representations by Error Propagation". In: *Parallel Distributed Processing: Explorations in the Microstructure of Cognition: Foundations*. 1987, pp. 318–362.

[56] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. "Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization". In: *International Journal of Computer Vision* 128.2 (Oct. 2016), pp. 336–359. DOI: 10.1007/s11263-019-01228-7. arXiv: 1610.02391.

[57] Karen Simonyan and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition". In: *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*. International Conference on Learning Representations, ICLR, Sept. 2015. arXiv: 1409.1556.

[58] Socialstyrelsen. *Digitala vårdtjänster och artificiell intelligens i hälso- och sjukvården*. 2019-10-6431. Office of the United Nations High Commissioner for Human Rights, Oct. 31, 2019. URL: https://www.socialstyrelsen.se/globalassets/sharepoint-dokument/artikelkatalog/ovrigt/2019-10-6431.pdf.

[59] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. "Deep high-resolution representation learning for human pose estimation". In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Vol. 2019-June. IEEE Computer Society, June 2019, pp. 5686–5696. ISBN: 9781728132938. DOI: 10.1109/CVPR.2019.00584. arXiv: 1902.09212.

[60] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. "Going deeper with convolutions". In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Vol. 07-12-June-2015. IEEE Computer Society, Oct. 2015, pp. 1–9. ISBN: 9781467369640. DOI: 10.1109/CVPR.2015.7298594. arXiv: 1409.4842.

[61] K. Thorborg, M. S. Rathleff, P. Petersen, S. Branci, and P. Hölmich. "Prevalence and severity of hip and groin pain in sub-elite male football: a cross-sectional cohort study of 695 players". In: *Scandinavian Journal of Medicine & Science in Sports* 27.1 (2017), pp. 107–114. DOI: 10.1111/sms.12623.

[62] Eric J. Topol. "High-performance medicine: the convergence of human and artificial intelligence." In: *Nature Medicine* 25.1 (2019-01-07 00:00:00.0), pp. 44–56. ISSN: 1078-8956. DOI: 10.1038/s41591-018-0300-7.

[63] A. Toshev and C. Szegedy. "DeepPose: Human Pose Estimation via Deep Neural Networks". In: *2014 IEEE Conference on Computer Vision and Pattern Recognition*. 2014, pp. 1653–1660. DOI: 10.1109/CVPR.2014.214.

[64] *tsfresh*. URL: https://tsfresh.readthedocs.io.

[65] J. Wang, K. Sun, T. Cheng, B. Jiang, C. Deng, Y. Zhao, D. Liu, Y. Mu, M. Tan, X. Wang, W. Liu, and B. Xiao. "Deep High-Resolution Representation Learning for Visual Recognition". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2020), pp. 1–1. DOI: 10.1109/TPAMI.2020.2983686.

[66] Shui-Hua Wang, Chaosheng Tang, Junding Sun, Jingyuan Yang, Chenxi Huang, Preetha Phillips, and Yu-Dong Zhang. "Multiple Sclerosis Identification by 14-Layer Convolutional Neural Network With Batch Normalization, Dropout, and Stochastic Pooling". In: *Frontiers in Neuroscience* 12 (2018), p. 818. ISSN: 1662-453X. DOI: 10.3389/fnins.2018.00818.

[67] Zhiguang Wang, Weizhong Yan, and Tim Oates. "Time series classification from scratch with deep neural networks: A strong baseline". In: *Proceedings of the International Joint Conference on Neural Networks*. Vol. 2017-May. Institute of Electrical and Electronics Engineers Inc., June 2017, pp. 1578–1585. ISBN: 9781509061815. DOI: 10.1109/IJCNN.2017.7966039. arXiv: 1611.06455.

[68] Nathan Wetters, Alexander E. Weber, Thomas H. Wuerz, David L. Schub, and Bert R. Mandelbaum. "Mechanism of Injury and Risk Factors for Anterior Cruciate Ligament Injury". In: *Operative Techniques in Sports Medicine* 24.1 (2016). Anterior Cruciate Ligament Injury and Reconstruction: From Perioperative Management to Rehabilitation and Return-to-Play, pp. 2–6. ISSN: 1060-1872. DOI: 10.1053/j.otsm.2015.09.001.

[69] Feng Zhang, Xiatian Zhu, Hanbin Dai, Mao Ye, and Ce Zhu. "Distribution-Aware Coordinate Representation for Human Pose Estimation". In: Institute of Electrical and Electronics Engineers (IEEE), Aug. 2020, pp. 7091–7100. DOI: 10.1109/cvpr42600.2020.00712. arXiv: 1910.06278.

[70] Yi Zheng, Qi Liu, Enhong Chen, Yong Ge, and J. Leon Zhao. "Exploiting multi-channels deep convolutional neural networks for multivariate time series classification". In: *Frontiers of Computer Science* 10.1 (Feb. 2016), pp. 96–112. ISSN: 20952236. DOI: 10.1007/s11704-015-4478-2.

Appendix A: POE-task combinations [45]

Segment-Specific POEs	Functional Tasks in Which Each POE Is Assessed*	Scoring of "0" Good (no POE)†	Scoring of "1" Fair (minor POE)†	Scoring of "2" Poor (major POE)†
Foot pronation	SLS	The absence of pronation of the medial arch of the foot, navicular bone and the medial malleolus indicates no POE	A slight position of pronation of the medial arch of the foot, navicular bone and the medial malleolus indicates a minor POE	A clear position of pronation of the medial arch of the foot, navicular bone and the medial malleolus indicates a major POE
KMFP	SLS SD FL SLHD SH	Mid-point of patella is in line with or lateral to the second toe	Mid-point of patella is placed medial to the second toe	Mid-point of patella is clearly placed medial to the big toe
Deviation of pelvis in any plane	SLS SD FL SLHD SH	The absence of pelvis into lateral deviation, pelvic tilt and/or rotation of pelvis respectively indicates no POE	A slight position of the pelvis into lateral deviation, pelvic tilt and/or rotation of pelvis respectively indicates minor POE	A clear position of the pelvis into lateral deviation, pelvic tilt and/or rotation of pelvis respectively indicates major POE
Deviation of trunk in any plane	SLS SD SLHD SH	The absence of a trunk position into forward lean, lateral lean and/or rotation indicates no POE	A slight position of the trunk into forward lean, lateral lean and/or rotation indicates minor POE	A clear position of the trunk into forward lean, lateral lean and/or rotation indicates major POE
Femur medial to shank	**SLS** **SD** **FL** **SLHD** **SH**	Mid-point of medial and lateral femoral condyles is lateral to tibial tuberosity  	Mid-point of medial and lateral femoral condyles is in-line with tibial tuberosity  	Mid-point of medial and lateral femoral condyles is medial to tibial tuberosity  
Femoral valgus (the angle created by the intersection of a longitudinal line and a line from mid-point patella toward ASIS)	**SLS** **SD** **FL** **SLHD** **SH**	The absence of femoral valgus indicates no POE 	A slight position of femoral valgus indicates minor POE 	A clear position of femoral valgus indicates major POE 

Appendix B: Models

B.1 Trunk

	T1	T2	T3	T4	T5
Training weights	[1 5]	[1 5]	$\begin{bmatrix} 0.6 & 0.04 & 0.04 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 \\ 0.3 & 0.45 & 0.3 \\ 0 & 0 & 1 \end{bmatrix}$	[1 10]
Ensemble weights	[1/3 1.15/3 1/3]	[1/3 1.15/3 1/3]	[1/3 0 0]	[0 1.15/3 0]	[0 0 1/3]
Trainable parameters	197890	197890	47747	119879	47651

B.2 Pelvis

	P1	P2	P3	P4	P5
Training weights	[2 1]	[2 1]	$\begin{bmatrix} 0.6 & 0.04 & 0.04 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 \\ 0.3 & 0.45 & 0.3 \\ 0 & 0 & 1 \end{bmatrix}$	[1 5]
Ensemble weights	[1/3 1.05/3 1/3]	[1/3 1.05/3 1/3]	[1/3 0 0]	[0 1.05/3 0]	[0 0 1/3]
Trainable parameters	183044	183044	112577	112577	41543

B.3 Femoral Valgus

	F1	F2	F3	F4	F5
Training weights	[1 1]	[1 1]	$\begin{bmatrix} 0.6 & 0.04 & 0.04 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 \\ 0.3 & 0.45 & 0.3 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0.1 & 0.1 & 0.6 \end{bmatrix}$
Ensemble weights	[1/3 1.25/3 1/3]	[1/3 1.25/3 1/3]	[1/3 0 0]	[0 1.25/3 0]	[0 0 1/3]
Trainable parameters	90648	23720	23727	23727	23727

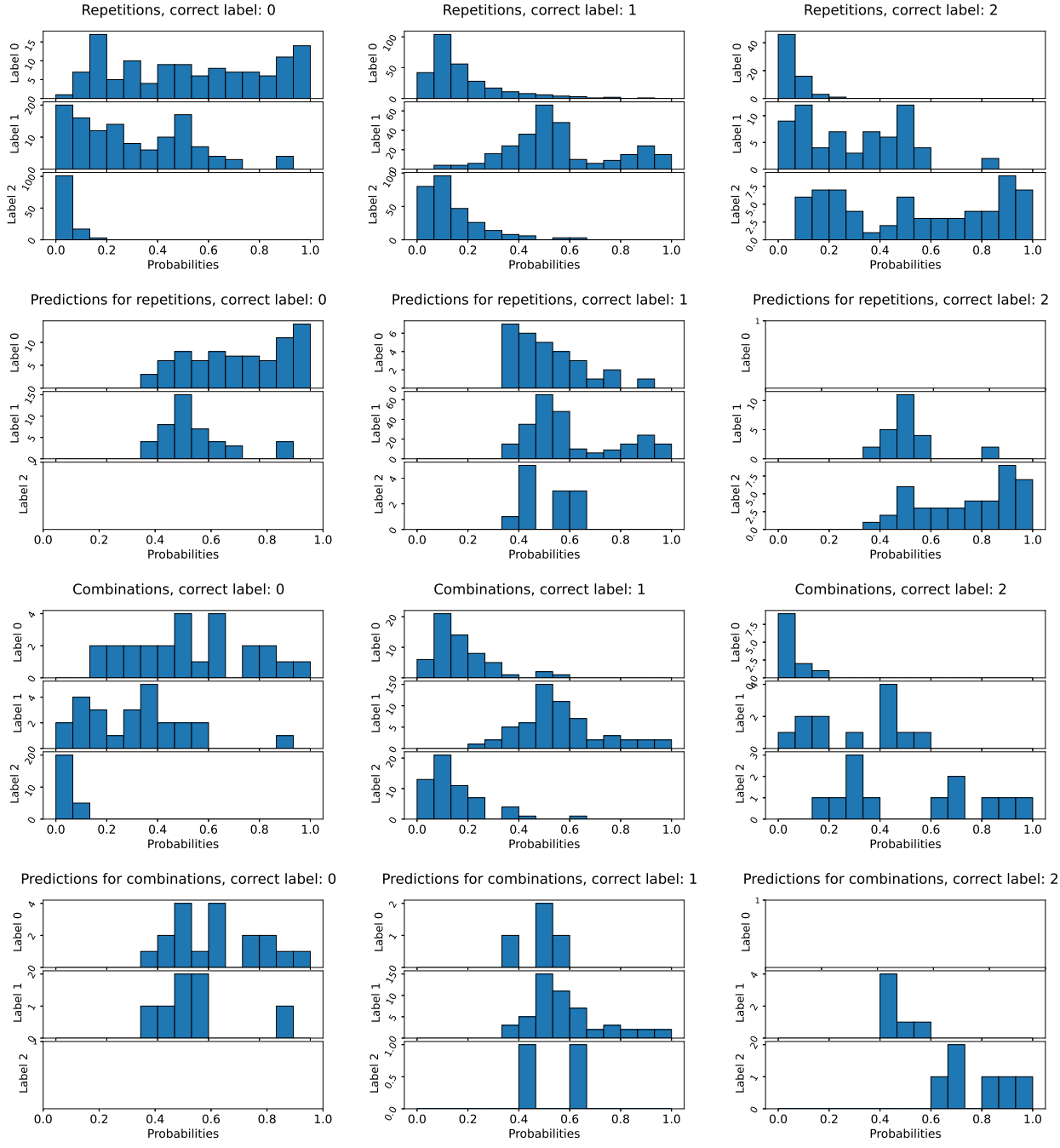
B.4 Knee Medial-to-Foot-Position

	K1	K2	K3	K4	K5
Training weights	[1 1 1]	[1 1 1]	$\begin{bmatrix} 0.6 & 0.04 & 0.04 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 \\ 0.3 & 0.45 & 0.3 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0.05 & 0.05 & 0.7 \end{bmatrix}$
Ensemble weights	[1/3 1.25/3 1/3]	[1/3 1.25/3 1/3]	[1/3 0 0]	[0 1.25/3 0]	[0 0 1/3]
Trainable parameters	168195	35843	35843	105265	35843

Appendix C: Histograms over probabilities on validation sets

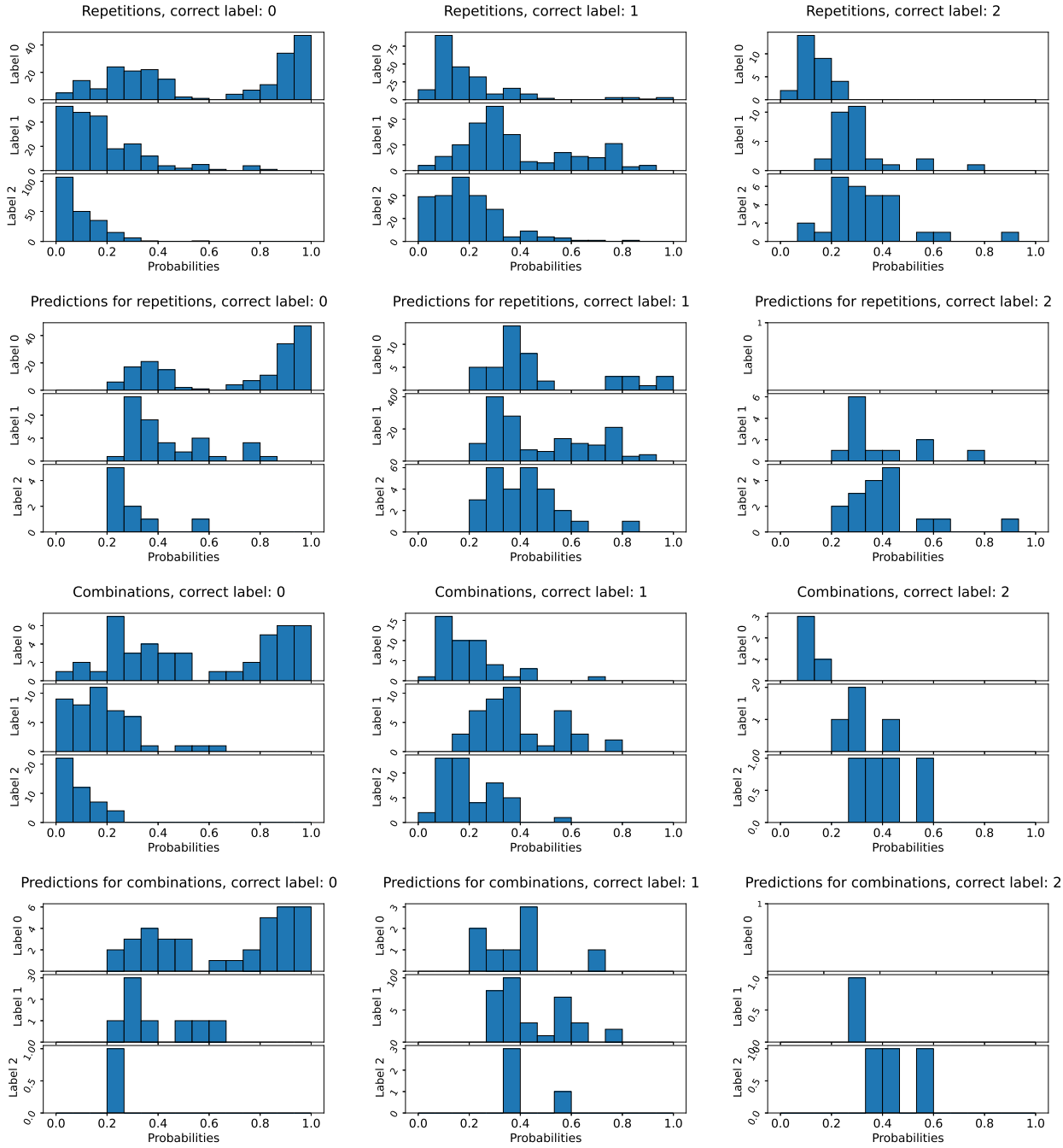
C.1 Trunk

The first row shows all the predicted probabilities for the different classes. The second row shows the probabilities for the predicted class for all repetitions, i.e. the histograms of the highest probabilities for the different classes for all repetitions. The third and fourth rows show the corresponding histograms for the combined scores.



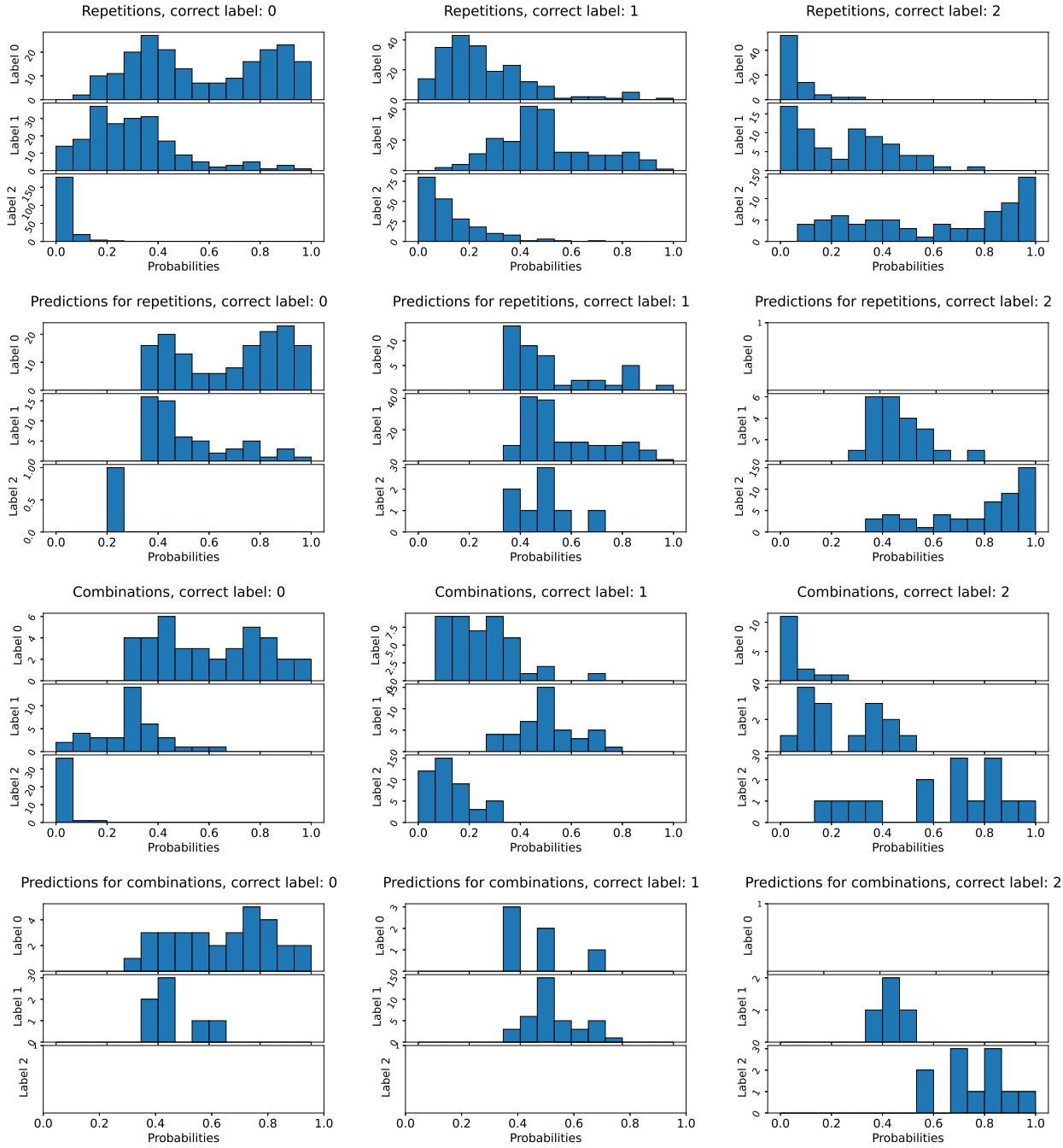
C.2 Pelvis

The first row shows all the predicted probabilities for the different classes. The second row shows the probabilities for the predicted class for all repetitions, i.e. the histograms of the highest probabilities for the different classes for all repetitions. The third and fourth rows show the corresponding histograms for the combined scores.



C.3 Femoral Valgus

The first row shows all the predicted probabilities for the different classes. The second row shows the probabilities for the predicted class for all repetitions, i.e. the histograms of the highest probabilities for the different classes for all repetitions. The third and fourth rows show the corresponding histograms for the combined scores.



C.4 Knee Medial-to-Foot Position

The first row shows all the predicted probabilities for the different classes. The second row shows the probabilities for the predicted class for all repetitions, i.e. the histograms of the highest probabilities for the different classes for all repetitions. The third and fourth rows show the corresponding histograms for the combined scores.