Improving feature extraction for camera-based motion analysis using trajectory similarity

Verbesserung der Feature Extrahierung für kamerabasierte Bewegungsanalyse mithilfe von Ähnlichkeitsanalyse der Trajektorien

Master thesis by Malte Christian Mai

Date of submission: 22.04.2024

1. Review: Prof. Dr.-Ing. Christoph Hoog Antink
2. Review: Sebastian Dill, M.Sc.

Darmstadt

TECHNISCHE UNIVERSITÄT DARMSTADT

Electrical Engineering and Information Technology Department

KIS*MED - Künstlich Intelligente Systeme der Medizin

Studiengang Elektro- und Informationstechnik

Erklärung zur Abschlussarbeit gemäß § 22 Abs. 7 APB TU Darmstadt

Hiermit erkläre ich, Malte Christian Mai, dass ich die vorliegende Arbeit gemäß § 22 Abs. 7 APB der TU Darmstadt selbstständig, ohne Hilfe Dritter und nur mit den angegebenen Quellen und Hilfsmitteln angefertigt habe. Ich habe mit Ausnahme der zitierten Literatur und anderer in der Arbeit genannter Quellen keine fremden Hilfsmittel benutzt. Die von mir bei der Anfertigung dieser wissenschaftlichen Arbeit wörtlich oder inhaltlich benutzte Literatur und alle anderen Quellen habe ich im Text deutlich gekennzeichnet und gesondert aufgeführt. Dies gilt auch für Quellen oder Hilfsmittel aus dem Internet.

Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Mir ist bekannt, dass im Falle eines Plagiats (§ 38 Abs. 2 APB) ein Täuschungsversuch vorliegt, der dazu führt, dass die Arbeit mit 5,0 bewertet und damit ein Prüfungsversuch verbraucht wird. Abschlussarbeiten dürfen nur einmal wiederholt werden.

Bei einer Thesis des Fachbereichs Architektur entspricht die eingereichte elektronische Fassung dem vorgestellten Modell und den vorgelegten Plänen.

Darmstadt, 22.04.2024

M. Mai

Abstract

The assessment of human movements in the context of medical treatments has become increasingly popular in recent years due to progress in the estimation of human poses. This work demonstrates the development of the Motion Quality Assessment (MoQuA) algorithm, which assesses recorded sports exercises from two camera perspectives. The objective is to provide feedback on the quality of exercise performance. Using MediaPipe, human pose estimation (HPE) is conducted from both camera perspectives, and the resultant data is subsequently filtered, fused, and normalized to ensure consistency and accuracy. The precision of the position estimation is validated by comparison with a Motion Capture suit. As a result of the fusion of the two data sources, an improvement in the estimate of 18.49% is achieved. Based on the coordinates, further relative features, such as angles, are determined. The quality of movement is assessed by comparing the test movement with a ”Golden Standard” through multidimensional Dynamic Time Warping (mDTW). The determination of whether the exercise has been performed correctly is made through a classification process for which a wide range of models were trained. A decision tree, identified as the best model, achieves an accuracy of 91.7%. The traceability and importance of the features are evaluated by using Explainable Artificial Intelligence (XAI), including Local Interpretable Model-agnostic Explanations (LIME) and SHapley Additive exPlanations (SHAP). Finally, feature importance is used to identify the limb that causes the errors. The MoQuA algorithm demonstrates the capability to reliably classify a wide range of exercises with a small dataset.

Contents

List of Figures

1. Introduction

1.1. Motivation and background

In modern medicine, physiotherapy plays a key role in the treatment and rehabilitation of a wide range of diseases and physical complaints. Its practice extends from sports injuries to chronic diseases such as osteoporosis, Parkinson's disease [1] and arthritis [2]. Physiotherapeutic treatment mainly involves direct interaction between the patient and a physiotherapist, where the therapist guides exercises or performs massages and manual therapies. The direct interaction ensures that the treatment is supervised by medically trained humans, which is essential for guaranteeing correct exercise performance and, consequently, the success of the therapy [3].

The healthcare system faces the dilemma that the financial costs for regular sessions with professional physiotherapists often face a significant limitation. Statistics are showing this by the fact that in the period from 2010 to 2022, statutory health insurance (SHI) expenditure in Germany increased by around 41% when adjusted for inflation. In addition, the share of physiotherapy costs in SHI expenditure rose from 2.12% to 2.85%, which highlights the increasing relevance of cost expenditure [4, 5]. Patients also carry out their therapeutic exercises independently at home as an integral part of the therapy. While this self-management of treatment may represent a cost-efficient part of physiotherapy, it carries considerable challenges and risks, especially for individuals lacking prior athletic experience or knowledge of body mechanics. The autonomous performance of therapeutic exercises without expert supervision and guidance can lead to difficulties in ensuring the correct technique. The absence of external correction mechanisms, such as mirrors or direct feedback from a therapist, makes it challenging for patients to practice their posture and movement exercises correctly. This can be problematic since individuals without experience in physical exercises often lack the capability to identify critical mistakes in their

technique independently. The risk of incorrect performance of exercises at home is substantial and can lead to a deterioration of therapeutic outcomes [6]. Incorrect movements can not only slow down or stop the healing process but also increase the risk of injuries, which in turn may extend the rehabilitation period and potentially lead to an increased need for medical intervention [7]. The previously described challenges highlight the importance of devising strategies that enhance the accuracy of home-based therapeutic exercises.

A promising approach to overcome the challenges associated with the independent performance of physiotherapeutic exercises lies in developing and implementing automatic assistance systems. These innovative technologies have the potential to fundamentally change the dynamics of home therapy by providing real-time monitoring and precise feedback on the quality of exercise practice. Such systems not only enhance the effectiveness of home therapy but also significantly reduce the risk of injuries that could arise from incorrect exercise practice. To use this technology as a real option for assistance, it is essential that the support system is designed to be cost-efficient and easy to use. This includes broad availability and adoption, creating an affordable extension to on-site therapy sessions accessible to a wide range of patients. Moreover, these systems must be highly user-friendly. The automatic assistance systems should be designed in a way that they can be easily used by individuals with limited technical knowledge or by those with little experience with performing physiotherapeutic exercises. Intuitive operation and accessibility are key elements that ensure all patient groups can benefit from the advantages of this technology. Another important aspect for the development of such assistance systems is the ability to operate with a minimal set of basic data. Considering the potential complexity and significant effort associated with large-scale data collection and processing, these systems should be capable of effectively learning and adapting based on a limited amount of initial data. By utilizing advanced machine learning (ML) and artificial intelligence (AI) algorithms, these systems could continuously learn from user interactions, thereby improving their accuracy and effectiveness over time without the need for extensive pre-programming or manual data entry. In summary, the development of automatic assistance systems for home physiotherapy exercises holds the potential to revolutionize the quality and safety of exercise performance. By combining cost-efficiency, user-friendliness and the ability to operate on minimal basic data, such systems can achieve broad acceptance and make a significant contribution to improving healthcare provision and the quality of life for patients.

1.2. Objective and organization of research

The primary objective of this work is to conceptualize and develop a support system for motion analysis, considering the main aspects presented in the previous section. In the beginning, a scenario is defined for the training of the Algorithm wherein the patient is recorded with two cameras from two perspectives. From both perspectives, an estimate of the human pose is made. The test patient is also equipped with a Motion Capture suit (MoCap suit) that determines posture data based on inertial sensors. This data has a low error rate and is therefore used to provide control data to assess the accuracy of human pose estimation (HPE) [8]. Given the progress in computer science in recent years, robust algorithms for position determination are now available. MediaPipe was chosen because of its high balance between calculation time and estimation quality [9]. Furthermore, the estimates of posture will synchronize and normalize and improve their accuracy through filtering techniques and sensor fusion. Subsequently, a procedure for extracting significant features will be used, utilizing the Dynamic Time Warping (DTW) algorithm to work with minimal data. A multitude of relative features will be computed to identify which features are particularly informative. The determined features are supposed to be used to train a classification algorithm. This classifier is expected to autonomously identify informative features and analyze the classifier through Explainable Artificial Intelligence (XAI) methodologies. The importance of features will be utilized to understand the individual decisions of the classifier. Based on the evaluation of the importance of the features, the limb responsible for the error is identified. For user accessibility and clinical application, simple cameras, like webcams, are intended to maximize user-friendliness and facilitate access for a broad patient group. Furthermore, it is expected that no MoCap suit will be required after the development is completed. This means video recordings alone are sufficient for the motion assessment. The overall aim is to develop technologies that are advanced in their capabilities and integrative in their user-friendliness in order to make state-of-the-art healthcare solutions accessible to a diverse group of patients.

Based on the objectives mentioned, this work is structured as described in the following. Chapter 2 begins with a compilation of the current state of the art. It discusses the achievements and methods relevant to this work and analyzes the milestones achieved so far. It tries to identify gaps in the current research and describes how this work attempts to offer a new perspective compared to existing research. Subsequently, the focus shifts to HPE as a basis for converting video material to positional data about the human body. The functionality of pose estimation is explained, and various existing models are compared.

Chapter 3 shows the development of the Motion Quality Assessment (MoQuA) algorithm. The chapter begins with a detailed description of data collection, specifying the dimension of the dataset used, the experimental setup and the performance of data acquisition. Followed by a description of data collection processes and a comprehensive overview of the structure and core components of the MoQuA algorithm. This overview incorporates key aspects of HPE, coordinate processing, feature extraction using DTW, classification and the approach of XAI. These components constitute the framework of the algorithm and are explored in terms of their interdependence and their contribution to the overall performance of the algorithm. Subsequent sections of the chapter are dedicated to a detailed mathematical explanation of each step within the MoQuA algorithm.

The developed MoQuA algorithm is thoroughly assessed and debated in Chapter 4. The first Section 4.1 is about the accuracy of the HPE, which is an important factor for the precision of the entire system. The quality of the HPE is validated by comparing the determined positional values with those obtained through a high-precision MoCap suit. By using relative features such as hip width as a basis for comparison, a detailed assessment of the algorithm's performance in terms of the accuracy of posture estimation is performed. A significant section of the evaluation deals with improving feature extraction, with a particular focus on the application of various fusion techniques. These techniques aim to combine data from different camera perspectives to enable a more comprehensive and precise capture of movement features. Another key aspect of the evaluation, described in Section 4.2, is optimizing classification performance. For this purpose, features generated by DTW are used to train and evaluate various machine-learning models. The methods considered include Linear Regression, Decision Trees, Random Forest, Support Vector Machines (SVM), and Artificial Neural Networks (ANN). These different approaches are tested for their effectiveness in order to choose the model with the highest accuracy. In the next part of the evaluation (Section 4.3), the explainability of the models is examined using techniques such as LIME (Local Interpretable Model-agnostic Explanations) and SHAP (SHapley Additive exPlanations). These methods contribute to the transparency of the decision-making processes by indicating which features have the greatest impact on model predictions. The resulting feature importance is used to precisely detect limbs which cause errors in the execution of the exercise. The determined feature importance contributes to understanding the model's functionality and increases confidence in the accuracy of limb detection and classification. Finally, Section 4.4 summarises the evaluation.

The final Chapter 5 focuses on a comprehensive summary and conclusive evaluation of the entire outcomes. It offers a reflection on the implications and significance of the findings applying them in a broader field. It concludes with an outlook for future research.

2. Theoretical background

This chapter reviews the progress and methodologies in human movement analysis. It recaps the evolution of motion quality assessment technologies, including machine learning and motion capture, to highlight the challenges and advancements in making these analyses more accurate and accessible. The chapter examines related works and sets the groundwork for developing innovative solutions in physiotherapy motion analysis.

2.1. Related work

The analysis of human movements has become increasingly important in recent years due to new technical possibilities and ever-increasing demand in the field of fitness, rehabilitation and preventive actions [10]. The published research pursues different perspectives and can be divided into two main categories: the recognition of human movements and the evaluation of the quality of human movements. These two research directions are closely linked, as accurately recognising movements is a fundamental prerequisite for subsequent qualitative assessment [11]. This work focuses mainly on the area of motion quality assessment. This implies a trajectory-based human movement analysis to identify and evaluate deviations from a defined standard or norm. Trajectories define the path curves representing the route of a moving object over time. In biomechanics, they specifically describe the paths that a human or a part of their body follows during movement, mapped as a function of time. The analysis of motion quality is of essential importance for the development of targeted rehabilitation measures or the improvement of physical performance [12]. Human Motion Quality Assessment (HMQA) has been applied in various fields, such as sports training programmes [10], competitive sports assessment systems [13], motor rehabilitation [12], medical diagnostic procedures [14], educational performance assessment [15] and ergonomic risk analysis [16]. Even though the subject areas are different, they are suitable for transfer to physiotherapy exercises

due to the similarly abstract task of estimating and comparing movement trajectories.

Exercise is characterized as a deliberate and systematic movement aimed at improving or sustaining physical fitness. It encompasses activities like body conditioning and sports and is intentional, structured, and often repetitive, designed to achieve a particular fitness goal [17, 18]. HMQA describes such procedures for the assessment of these exercises. Healthcare professionals usually perform these for medical purposes, and sports trainers ensure correct movement execution. The professionals can support the development of automated assessment methods and increase the data quality through their expertise [19].

Historically, HMQA has utilised rule or template-based methods. Rule-based methods assess motion through set conditions, such as joint angles [20], while template-based methods, DTW [21], match patterns against established motion recordings. In some cases, a combination of the two methods is also used. These methods are easy to implement and can be used in real-time because they follow clearly defined calculation rules, and the operators used can be calculated quickly. An improvement could generally be achieved by additionally using ML methods [22]. Figure 2.1 shows the abstract schematic representation that most models implement. There may be deviations, for example, the omission of individual steps [11]. The following is a brief description of the steps and their individual uses.

The initial step in the standard process is typically the collection of data, which is defined by the dynamics of movement. There are primarily two categories of data collection: those based on inertial sensors and those reliant on optical capture. The most widely utilized method involves MoCap suits, wherein subjects wear suits equipped with sensors that measure acceleration. Another commonly used approach is based on optical systems, primarily cameras, as utilized in this thesis. The preference for MoCap suits stems from their significantly lower data noise levels compared to data obtained from camera-based systems. The disadvantage, by contrast, is that such a suit is expensive and rarely available and therefore not user-friendly. Regardless of the chosen method, three-dimensional data points (x, y, and z) for various keypoints of the body are output as a time series. This point, therefore, represents a coordinate. The measurements invariably contain inaccuracy, necessitating filtering methods to enhance data quality [23].

Following the collection of motion capture data, the next step typically involves data normalization, which ensures the comparability of the data. The reasons for this are differences in physique as well as errors in the measurement. There are two types of

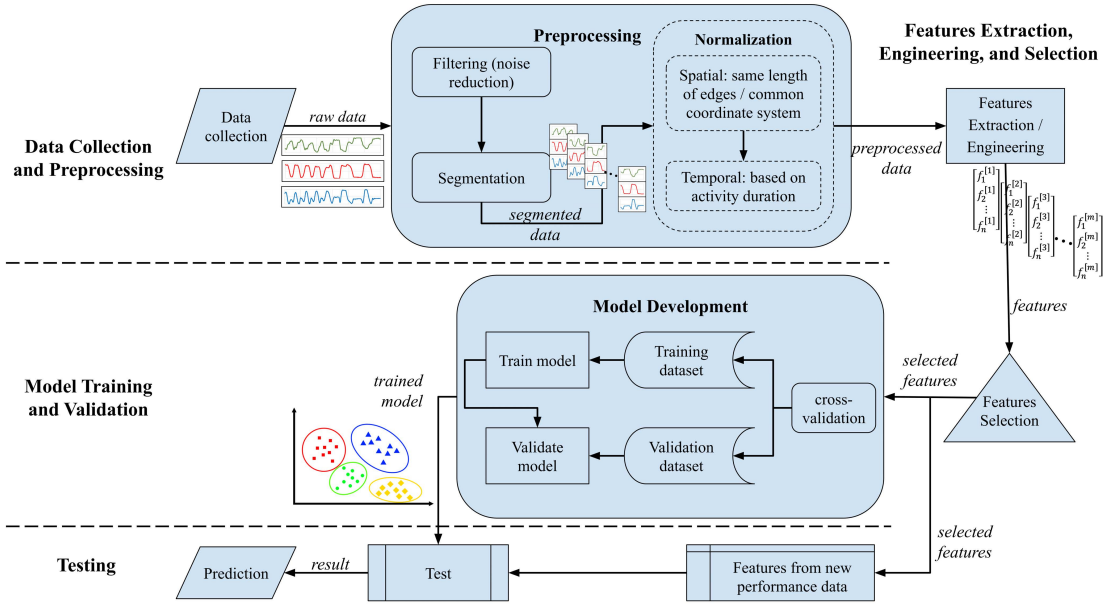


Figure 2.1.: The abstract representation of a human movement analysis. It shows the complete process from data collection to prediction by [11]. This process involves multiple steps: data gathering, preprocessing (which includes filtering, segmenting, and normalizing data), extracting and engineering features, selecting those features, building the machine learning model, and finally, validating and evaluating the model.

normalization: spatial, which aligns data to a consistent coordinate system and plane, and temporal, which adjusts the duration of data segments to a uniform length. Spatial normalization is essential for accurate representation of the human skeleton in assessments. Temporal normalization is important for compatibility with certain machine-learning algorithms like CNNs but may alter the original time-series data. Despite being time-consuming, preprocessing is necessary as it enhances the accuracy of the assessment by standardizing the input data, making it more amenable to analysis. Each application should carefully consider the necessity and extent of these preprocessing steps [19].

The next step is the feature extraction. Different approaches are used in related work, and the most commonly used ones are described in the following. A fundamental feature extraction approach involves utilising each joint's raw data from the recorded signals as a feature vector (e.g., the values of the three axes (X, Y, Z) of a triaxial accelerometer can

serve as features over time). These feature vectors encapsulate all time series data from an exercise execution and serve as inputs for ML algorithms. Another method involves applying various statistical/aggregating functions (e.g., minimum, mean, and standard deviation) that provide descriptive statistics for the time series and are utilized as features. These features are often used with shallow ML algorithms [24]. In addition to the previous methods, feature construction transforms features into different and more efficient representations or dimensions while retaining the expressive power of the original features. Various techniques are employed for such transformations, leveraging different properties of the data. One approach is the application of dimensionality reduction algorithms (e.g., principal component analysis), which capture large variations in the data and omit invariant aspects. Another strategy is transfer learning, which converts features into a format that has been used successfully in other areas. For example, a set of feature vectors representing time series values can be converted into an RGB image, with each column representing a skeletal joint, the rows representing time points, and the RGB values representing the position of (x, y, z) of a joint that features at that time. This generates a heatmap. The trained model may be applied backwards to predict the location [25]. Another frequent transformation for skeletal data is to convert it into a graph that may represent the connection of the joints in the human body. This allows features to be extracted using a graph convolutional network (GCN). This increases the performance of the classifiers [26].

High-dimensional features increase the complexity and accuracy of the ML model. Therefore, to reduce complexity, the extracted features should be selected instead of using all to develop the model. This may be particularly necessary for more complex systems that use multiple input devices and modalities and may contain features irrelevant to the performed movements. In simple cases, this process can be manually performed based on empirical knowledge. However, various algorithms for feature selection have been proposed to choose optimal feature sets. Feature selection algorithms are divided into unsupervised and supervised algorithms. Unsupervised algorithms may use correlation analyses or clustering techniques to identify redundant features. Supervised algorithms are typically categorized into wrapper techniques (e.g., sequential algorithms and genetic algorithms), filter techniques (e.g., correlation criteria and mutual information), and embedded techniques that select features during the training of the model (e.g., Random Forest (RF)) [27].

The extracted features or data series are classified according to their application in the final part. A wide range of classification methods is applied across different studies, from

simple decision trees to complex neural networks and ensemble methods. Many studies address the challenge of limited datasets Khan *et al.* [28], which increases the likelihood of overfitting. Therefore, particular attention is paid to ensuring a clean split between training and test datasets. This is only a brief overview. Chapter 3 provides a more detailed overview of current methods and their applications in this work.

An exemplary instance of a comprehensive system utilizing a MoCap suit is showcased in [29]. Here, position data is sourced from a Tesla Suit, which concurrently facilitates the provision of haptic feedback. A probabilistic model for each exercise is initially developed and then compared with the real-time execution variant. This paper also undertakes an exercise classification followed by segmentation into individual exercises. It was possible to provide immediate feedback during execution, signifying not only the identification of erroneous executions but also the detection of three distinct error types. A Support Vector Machine (SVM) was employed for the classification purpose. The limitation of this study arises from a small dataset. The system's efficacy was rated at 86% for classification and up to 99% for segmentation. A challenge identified is the difficulty in recognizing variations in the execution speed of exercises, indicating that exercises must be performed at a similar speed for the model to recognize them effectively. This problem has not been solved. This limitation highlights that, although haptic feedback is readily provided, its contribution to performance improvement could not be evaluated. Additionally, applying a MoCap suit poses a significant barrier to entry for users. Overall, this paper demonstrates the feasibility of evaluating exercises through position data and error detection.

With advances in computer vision technology, studies have started to utilise optical sensors in the form of cameras in addition to inertial sensors. Although error assessment is still in its infancy, recognising specific exercises and counting repetitions is already being carried out successfully. It is shown by these developments that an era is being approached where the assessment and improvement of human movement using camera-based technology can rival that of inertial sensors [30]. Various methods are used, with the use of a sequence-to-sequence autoencoder being emphasised in Jain and Harit [31]. This is based on videos recorded by experts in order to simulate the possible movement dynamics. The discrepancy between the sequenced movement and the actual video is then measured. However, this method was only evaluated on the basis of a sun salutation exercise, where it was applied relatively successfully. Nevertheless, it was impossible to identify where exactly the error in the execution of the movement was, which meant that no concrete feedback was provided. This is due to the fact that the autoencoder provides an overall output, and individual body points were not considered separately.

In Chariar *et al.* [32], the MediaPipe framework is used to estimate the positions and then a Long Short-Term Memory (LSTM) network is used to determine the type of squat. This recognises that each person performs squats differently and suggests the most appropriate variation based on optimal posture. A distinction is made between seven different types of execution. This suggests a form of abstract error detection and suggestions for improvement, although these are generalised as they relate exclusively to different types of execution. A high accuracy of 94.6% was achieved for this specific use case, although this model is limited solely to squats. In Lei *et al.* [33], the estimation of joint features from data is performed, but only two-dimensional data is utilized. This introduces a challenge in segmenting longer sequences, as the analysis focuses on performances within sports rather than on specific sports exercises. In related applications, movement is fragmented into individual segments and normalised over time, which also enables comparability [34]. Recent research has begun to employ motion assessment methods based on metrics derived from posture estimation, such as angles [35] and the DTW technique for performance assessment. Examples of such research differentiate merely between ”good” and ”bad” performances without providing detailed error descriptions [36–38]. In one of the most recent studies, a similar method of posture estimation and feature extraction of derived quantities and the use of DTW for error correction was used. This method was then tested to see to what extent it actually improves execution quality. It was observed that the execution quality of older people was significantly improved by the system, which indicates that the development of such a system is reasonable [39].

These studies present some of the most successful methodologies, acknowledging that numerous other methods have been partially explored. Two comprehensive literature reviews provides an overview of all related works and methodologies [11, 12]. It was discovered that comparing these methods presents significant challenges. This difficulty arises partly because some systems were tested using only specific exercises and due to varying definitions of error detection and correction. Additionally, the comparability is further limited because the datasets and models used are often not publicly available, and a unique dataset has been created. While there are some public databases, they were generally not designed to correct sports exercises, making them only marginally useful. The diversity in error definitions and using different performance metrics also significantly restrict comparability. This work aims to bridge the gaps identified in previous studies and to view them within a unified context. This entails disregarding the speed of executions and focusing on the applicability to a wide range of exercises, as well as a more precise error localization on a segmental body group level, utilizing simple sensory technology, namely cameras.

2.2. Human pose estimation

One of the fundamental components of HMQA is obtaining the appropriate data foundation. This section deals with the methods available for deriving data points from individual images, which form a temporal trajectory of data points when sequenced from a video containing multiple frames. Initially, the section addresses these methods' general principles and variations, subsequently describing some of the most widely adopted and specifically trained models. It evaluates which model is chosen for this work, notably the BlazePose model, utilised within Google's MediaPipe library. Other methodologies are briefly outlined, while BlazePose receives a more detailed examination. The conclusion thoroughly justifies selecting the BlazePose model, drawing on comparisons, strengths, and weaknesses identified in existing literature.

The selection of the BlazePose model over others is predicated on its superior accuracy, speed, and robustness in various conditions, as established by comparative literature. This evaluation considers not only the technical capabilities of the models but also their applicability to real-world scenarios, emphasizing the importance of precision in human posture estimation. The choice is further supported by BlazePose's integration within the MediaPipe framework, which offers a comprehensive toolset for real-time, cross-platform application development. This decision carefully considers the model's advantages, including its ability to generate high-fidelity posture estimations from video data, thereby enhancing the analysis and understanding of human motion.

2.2.1. Overview

Estimating human pose, which aims to predict the positions of joints in a human body based on an image or video of the individual, has become a widely pursued aim in computer vision in recent years. These endeavours, known as Human Pose Estimation (HPE), provide critical geometric and kinematic information about the human figure and facilitate a multitude of applications ranging from human-computer interaction and motion analysis to augmented reality (AR), virtual reality (VR) and healthcare [40]. Described in the following and shown in Figure 2.2 are the three most commonly used body model types in HPE: skeleton-based, contour-based and volume-based models.

Skeleton-based models represent the human skeleton with joints and associated limbs represented as simple graphs. They are known for their simplicity and flexibility but do not contain any texture or contour information about the body. Contour-based models

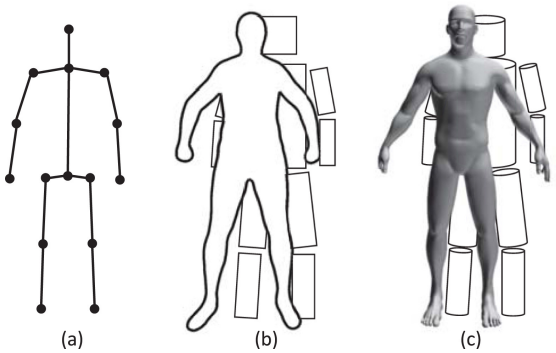


Figure 2.2.: The three most common body model types: skeleton-based model (a); contour-based model (b); volume-based model (c) by [41]

provide a rough representation of the body shape and outline by mapping body parts through geometric shapes such as rectangles or through the silhouette of a person. They have been used in previous HPE methods and include models such as cardboard models and Active Shape Models (ASMs). Volume-based models represent 3D body shapes and postures using geometric shapes or mesh structures. Volumetric models in HPE enable a more precise and comprehensive capture of the three-dimensional structure of the human body, leading to improved accuracy in analysis and application. However, the main drawback is the high computational power and data processing requirements, which increase the cost and complexity of the technology. These are often obtained from 3D scans and include modern models such as SCAPE, SMPL and uniform deformation models [41]. As the selected approach in this thesis compares trajectories, skeleton-based models are chosen because they provide a compact and efficient representation of human motion by reducing the complexity of keypoints and their connections, which facilitates the analysis of motion trajectories. They are robust to external influences such as illumination changes and background variations as they directly capture the structure of the human body. Thanks to advanced deep learning techniques and the recent accumulation of extensive datasets, significant progress has been made in HPE, resulting in many libraries available to address this challenge. The most notable are Openpose [42], DeepCut [43], AlphaPose [44], ShuffleNet [45], High-Resolution Network [46] and BlazePose [47].

Despite rapid development, HPE faces several challenges that affect estimation quality. Such challenges include occlusion and the ambiguity of depth, which remain significant obstacles to overcome. 2D HPE from images and videos labelled with 2D positions is easily

achievable and has seen high performance in estimating the human pose of a single person using deep learning techniques. Due to the results in 2D estimation, the focus has recently shifted towards the HPE of multiple people in complex scenes with significant occlusion. Conversely, 3D HPE is more challenging than its 2D counterpart in obtaining accurate 3D position information. Although motion capture systems can acquire 3D positional data in controlled laboratory settings, their applicability in natural environments is limited. This is primarily due to the challenge of generating 3D information from a 2D image. For instance, a 2D skeleton can correspond to multiple 3D poses, as illustrated. The issue of depth ambiguity could be significantly mitigated by incorporating temporal information, images from various viewpoints, etc. Visual cues, such as shadows and objects of known size, can be utilised to resolve ambiguities in images. However, capturing such information directly from images proves to be exceptionally challenging because the 3D world transforms a 2D projection plane in an image. Estimating the 3D pose of multiple people poses is a more significant challenge than that of a single person because the different people have to be separated from each other and the number of people is unknown. The additional challenge in estimating multiple people from a single-view image lies in occlusion by nearby individuals. When estimating the 3D pose of numerous people from various views, the most significant challenges include a more extensive state space, occlusions, and ambiguities between views [48].

Most existing methods rely on two-stage processes for each frame, which need more efficiency because each step requires more computing power. Another problem is the significant differences across various datasets, as each dataset employs different logic and MoCap suits, making it difficult to force a model to generalise. Summarizing the problem is focusing on single-person estimation, as one always expects precisely one person in the camera, and on 3D estimation from a single image or second estimation for registration through two images, with further details to follow in later Chapter 3. The following presents some of the most important deep learning models used for HPE.

2.2.2. OpenPose

OpenPose [42] is the first real-time technology that can recognise numerous human body keypoints (up to 135) on a single picture proposed by Cao *et al.* The first step is passing an image through a Convolutional Neural Network (CNN) to extract the feature maps of the input. In particular, the first ten layers of the VGG-19 network, which are integrated into the model, are used. A multi-stage CNN generates two output types from these feature

maps: Part Confidence Maps (PCM) and Part Affinity Fields (PAF). These maps represent the probability of the position of body parts or the spatial relationship and orientation of neighbouring body parts respectively.

In the subsequent stages of the CNN, the predictions of each of these outputs are gradually refined. Bipartite graphs based on the PCM are created between pairs of body parts (e.g. the hips, shoulders,...) and help to place the body's individual parts in relation to each other. The PAF is first used to refine the connections between the body parts from the feature maps of the base network. The results of the previous layers are then used to improve the recognition in the confidence maps further. Finally, a greedy algorithm is applied to analyse the final PCM and PAF. This step allows OpenPose to efficiently and accurately identify the positions and connections of the body parts.

The advantage of OpenPose is that it offers considerable precision, as it was developed specifically for operation on GPUs. This enables high accuracy in keypoint detection without compromising the quality of the implementation. OpenPose is also free of charge for non-commercial use. In contrast, the disadvantages of OpenPose are that the image has low-resolution the results provide a limited level of detail in the keypoint predictions. In addition, the programme does not provide any information about the depth of the detected objects. OpenPose is based on deep neural networks (DNN), so a powerful machine is required for efficient operation. This may have a slight impact on speed or you get lower precision, when a low computing power is available [49].

2.2.3. DeepCut

DeepCut [43] stands out as an innovative model in multiple HPE, employing a unique bottom-up approach. Developed by Pishchulin *et al.* in 2016, the model is designed to tackle the dual challenges of detecting body parts and estimating poses. This approach delineates DeepCut's operation into three critical stages: detecting body part candidates, classifying and labelling each detected part into various human body subsections like arms, legs, and torso, and finally, grouping these parts according to the individuals they belong to. This last step is notably complex due to the potential presence of multiple individuals in a single image. A key component to DeepCut's methodology is the use of Integral Linear Programming (ILP), which cleverly organises detected keypoints to render a skeletal depiction of human figures in the output image. The model aims to seamlessly blend all operational phases, from the initial recognition of body parts to the ultimate

pose presentation, within a cohesive framework.

The advantages of DeepCut are manifold. It offers a holistic solution that simultaneously solves detection and pose estimation, providing a comprehensive tool for complex scenarios without the necessity for markers or special equipment typically required by marker-based systems. However, the model is not without limitations. The complexity of solving an ILP problem makes DeepCut highly computationally intensive, mainly when dealing with images featuring multiple subjects. Moreover, the precision of the pose estimation and the resolution of the results are inherently limited by the quality of the input images.

DeepCut's utility spans a broad spectrum of applications, from enhancing video surveillance capabilities to advancing sports analysis and facilitating more nuanced human-computer interaction. Its value is especially pronounced in scenarios that necessitate the accurate capture and analysis of multiple individuals' poses simultaneously, such as in motion research or in developing interactive systems designed to interpret human movements [50]. Despite the challenges associated with its computational intensity and precision limits, DeepCut represents a significant leap forward in the automated estimation of human poses, offering a novel and effective solution to a complex problem.

2.2.4. AlphaPose

AlphaPose [44], introduced in 2017 by Fang *et al.*, is an advanced posture estimation system distinguished by its top-down methodology for detecting the poses of multiple individuals. It represents the first open-source system of its kind, aiming to address the challenges of posture estimation "in the wild" which means in natural, uncontrolled environments. The system comprises three components: A Symmetric Spatial Transformer Network (SSTN), Parametric Pose Non-Maximum Suppression (NMS) and a Pose-guided Proposals Generator (PGPG). The process initiates with bounding box proposals provided by a VGG-based SSD512 detector for human detection. These proposals are then forwarded to the Symmetric STN + SPPE module, which generates the pose proposals. During the training phase, a Parallel SPPE module is employed to avoid local minima, enhancing the training process's reliability [51]. A well-acknowledged issue in multi-person posture estimation is the redundancy in detected poses, which means that the same pose may be recognised several times for one person. AlphaPose addresses this challenge with a parametric Pose NMS, which eliminates redundant poses. Additionally,

a Pose Guided Proposals Generator is utilised during training to augment the training data.

AlphaPose is applied in fields requiring accurate full-body multi-person posture estimation and tracking, such as behavioural analysis, where capturing subtle human actions is required. It is a system capable of accurate full-body posture estimation and simultaneous monitoring in real time. With techniques like Symmetric Integral Keypoint Regression, Parametric Pose NMS, and Pose Aware Identity Embedding, AlphaPose precisely locates whole-body keypoints. It tracks individuals concurrently, even with inaccurate bounding boxes and redundant detections. Despite the challenges posed by the top-down approach, such as localisation errors and prediction inaccuracies, AlphaPose represents a significant advancement in posture estimation and has proven to be highly effective across various datasets [52].

The most essential advantage of AlphaPose is the ability to circumvent errors prevalent in traditional systems, such as incorrect identification or localisation. Optimising the network's hyperparameters significantly enhances its performance. Compared to conventional single-stage process frameworks, AlphaPose's two-stage framework delivers more accurate results. The two-stage structure also presents disadvantages, notably impacting speed or runtime efficiency. Moreover, there are scenarios where AlphaPose may not perform as well as compared to other mentioned posture estimation methods. For example, when recognizing poses in very crowded scenes [53].

2.2.5. ShuffleNet

ShuffleNet [45] is an innovative architecture for convolutional neural networks (CNNs) designed specifically for mobile devices with limited computing power. This architecture is designed to perform complex tasks such as object recognition efficiently and pose estimation efficiently [54]. At the heart of ShuffleNet are the ShuffleNet units, which are based on pointwise group convolution and a channel shuffle. These techniques make it possible to drastically reduce the number of computational operations without compromising accuracy. With pointwise group convolution, the channels are divided into groups, and each group is processed separately. This saves computing power compared to traditional convolutional operations. The channel shuffle ensures that the features are swapped between the groups to promote the flow of information between the channels and improve network performance. Another advantage of ShuffleNet is the use of depthwise separable convolutions, which became popular in MobilenetV1. This type of convolution reduces

the complexity of operations by treating the channels individually and then applying a 1x1 convolution to combine the features. ShuffleNet further extends this by using group convolution for the 1x1 convolutions, which further increases efficiency [45].

In summary, the advantages of ShuffleNet are apparent: it is highly efficient in terms of computing power, making it ideal for mobile use. It offers high-speed execution on ARM-based devices while maintaining accuracy comparable to more powerful networks like AlexNet. This efficiency makes ShuffleNet an excellent choice for real-time applications such as 2D posture estimation of multiple people in resource-constrained scenarios. Nevertheless, there are drawbacks. Due to its focus on efficiency, ShuffleNet may not consistently achieve the same accuracy as some larger and more computationally intensive networks. In addition, it may be a challenge to further optimise the architecture to improve maximum performance (mAP) without sacrificing efficiency [55].

Overall, ShuffleNet is an excellent solution for developers and researchers who want to bring powerful CNN capabilities to hardware with limited resources. The architecture has mobile image and video analytic applications, especially where real-time processing and energy efficiency are critical, such as mobile object and gesture recognition and augmented reality applications.

2.2.6. High-Resolution Network

HRNet [46], which stands for High-Resolution Network, represents an advanced approach in HPE characterised by preserving high-resolution representations throughout the entire processing sequence [46]. Unlike most existing methods that reconstruct high-resolution representations from low-resolution ones, HRNet initiates with a high-resolution subnetwork and incrementally integrates subnetworks from high to low resolution. This multi-scale architecture facilitates parallel connections between subnetworks of varying resolutions, leading to repeated multi-scale fusions. Each representation, from high to low, benefits from the information of the other parallel representations, yielding a rich and precise high-resolution depiction. The success of HRNet is attributed to two main aspects: Firstly, the continuous preservation of high resolution circumvents the need to restore it, enhancing the accuracy and spatial precision of the keypoint heatmaps. Secondly, the repeated fusion of multi-resolution representations enables the creation of reliable high-resolution depictions. This methodology results in exceptionally accurate outcomes in pose estimation across various benchmark datasets, including Common Objects in

Context (COCO) and MPII (Max Planck Institute for Informatics Human Pose) for keypoint detection and PoseTrack for pose tracking.

Advantages of HRNet include high accuracy and spatial precision: On the one hand, the HRNet generates more accurate and spatially precise keypoint heatmaps than many other network architectures. By preserving high resolution throughout the process and through repeated representation fusions, an efficient use of the available information is ensured. HRNet has yielded promising in HPE and shows potential for applications in other dense prediction tasks such as facial alignment, object detection, and semantic segmentation. On the other hand, the disadvantages of HRNet are its computational intensity and the need for optimisation regarding computational speed. Although HRNet is praised for its accuracy, its complex architecture can lead to higher computational demands, potentially limiting its deployment on devices with limited resources. There is a continuous need to optimise the network's structure and training methods further to enhance its mAP [56].

In conclusion, HRNet is exceptionally suited for high accuracy in HPE by using images and videos and finds use in sports analytics, surveillance, interactive systems, and the healthcare industry. Its capability to estimate precise poses in complex scenes with multiple individuals and under various conditions renders HRNet a valuable tool for researchers and developers in computer vision and related fields [57].

2.2.7. BlazePose

BlazePose, developed by Google Research, is an advanced technology designed to estimate a single person's pose, particularly tailored for fitness applications [47]. Its two-stage approach distinguishes from another methods. Initially, a detector identifies the Region of Interest (ROI) within an image where the pose is located. Subsequently, a tracker estimates the keypoints within this area. A critical feature of BlazePose is the requirement for an initial pose alignment, necessitating the clear annotation of either the entire person or at least the hip and shoulder points. During the inference phase, shown in Figure 2.3, BlazePose demonstrating remarkable real-time capabilities in various tasks, such as detecting hand and facial features. This system integrates a compact body pose detector and a pose tracker used for predicting keypoint positions, confirming the presence of a person in the image, and determining the refined ROI for the image. If the tracker fails to detect a person, the detection network is prompted to analyze the following image. Most modern object detection solutions rely on NMS in their final post-processing step, which is

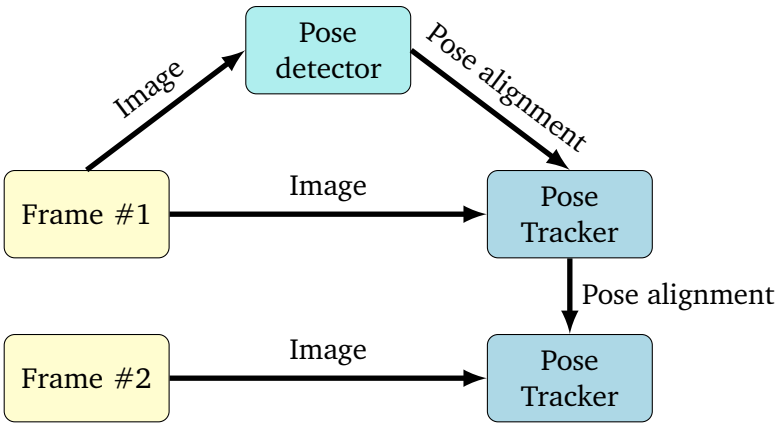


Figure 2.3.: Human pose estimation pipeline overview for the BlazePose model by [47]

effective for rigid objects with limited degrees of freedom. This approach faces challenges when dealing with complex movements, such as gestures of greeting or hugging. These difficulties occur because several overlapping bounding boxes may satisfy the criteria for Intersection over Union (IoU), which the Non-Maximum Suppression (NMS) algorithm uses. To address this issue, the strategy shifts towards identifying the bounding box of

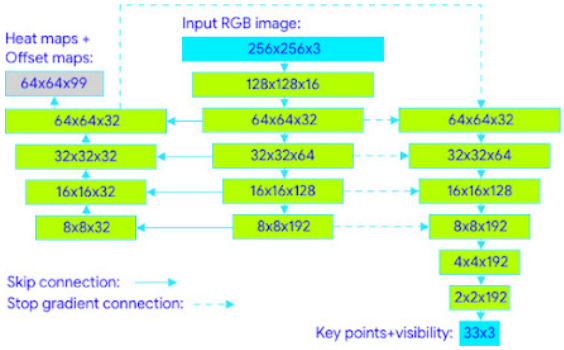


Figure 2.4.: Tracking network architecture: regression with heatmap supervision by [47]

more stable body parts, such as the face or torso. Observations suggest that the face, with its distinct, high-contrast characteristics and consistent appearance, typically provides a clear indication of the torso's location to the neural network. In pursuit of a swift and efficient person detection tool suitable for Augmented Reality (AR) applications, it is assumed that the person's head is always visible in scenarios involving only one individual.

As a result, a rapid face detection system implemented directly on the device serves as the basis for person detection. This system not only recognizes faces but also estimates key person-specific alignment metrics, including the midpoint of the hips, the encompassing circle's diameter, and the tilt, defined as the angle formed by the line connecting the midpoint of the shoulders and hips.

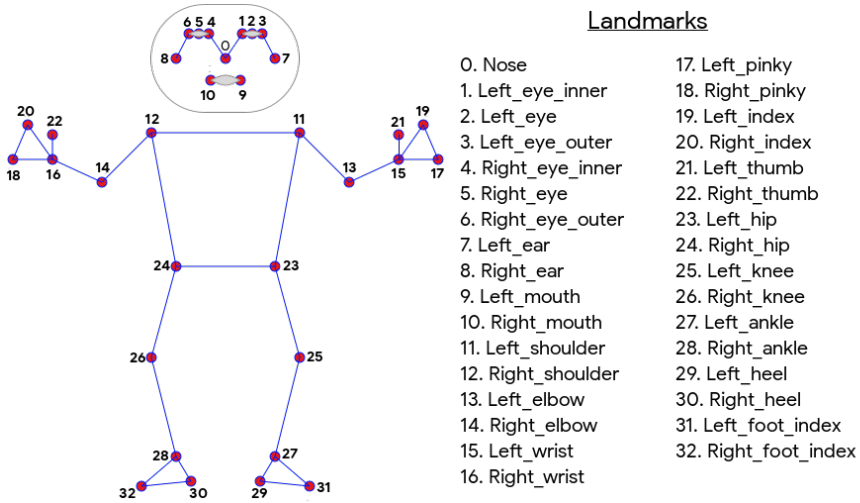


Figure 2.5.: Shows the body model of Blazepose, both the position and the naming of the 33 keypoints by [47]

The pose estimation component of the system calculates the locations of 33 human keypoints, as indicated in Figure 2.5, using the alignment suggestions from the initial stage of the pipeline. The methodology combines heatmaps, offsets, and regression techniques, as depicted in Figure 2.4. In pose detection, heatmaps serve to visualize the potential locations of body parts through color gradients, with each essential body part represented by its own heatmap where more intense colors indicate higher likelihoods. Offsets are used to refine these predictions by providing vectors that adjust the keypoint positions beyond the general areas indicated by the heatmaps. BlazePose utilizes a regression method to directly learn the coordinates of keypoints, enhancing the accuracy and efficiency of pose estimations. During training, both heatmap and offset losses are utilized, but these layers are omitted at the inference stage to streamline the process. This setup allows for the use of heatmaps in tracking the light embedding, which aids the regression encoder network. The architecture incorporates skip connections throughout to integrate high and low-level data effectively. Moreover, it is noted that preventing gradient transfer from

the regression encoder back to the heatmap-trained features enhances heatmap performance and significantly boosts the precision of the coordinate regression. Incorporating

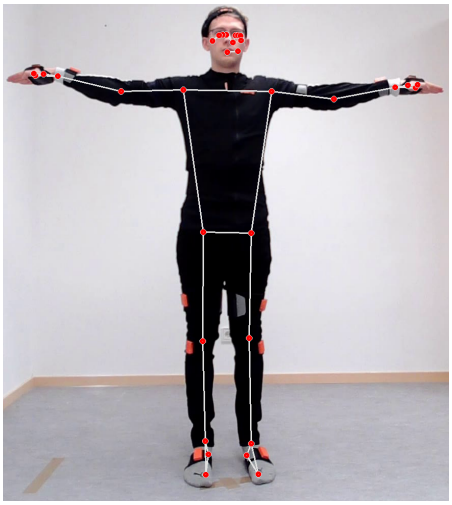


Figure 2.6.: The image shows a position estimate using MediapPipe Pose. The red dots correspond to the keypoints shown in Figure 2.5.

a relevant pose prior is critical to the proposed method. During training, the ranges for angles, scaling, and shifting are intentionally restricted to optimize data augmentation and preparation. This constraint helps to reduce the network's demand on computational and energy resources, enhancing the speed and efficiency of the system on the device. Alignment of the person in the neural network's square image input is based on either the detection from the previous phase or keypoints from the prior image, centering the midpoint between the hips. The system calculates rotation by defining a line L between the centers of the hip and shoulder and adjusts the image to make line L parallel with the y-axis. Scaling is adjusted to ensure all body points are enclosed within a square bounding box. To accommodate movement and alignment discrepancies between images, a 10% scaling and shifting augmentation is also applied, improving the tracker's ability to manage variations in body position.

To aid in the prediction of unseen points, random rectangles filled with different colours (occlusions) are generated during training, introducing a visibility classifier per point that indicates if a certain point is hidden and whether the location prediction is judged erroneous. This enables continuous tracking of a person, even in the presence of severe

occlusions, such as when just the upper body is visible or the rest of the person's body is outside the scene. Figure 2.6 shows a HPE using MediaPipe pose. The red dots represent the keypoints. Each point in this example has a visibility of over 50%. The Blazepose model thus offers a computationally efficient HPE optimised for sports exercises [58].

2.3. Summary

In this chapter, the basics of HMQA and the fundamental methods for high-accuracy HPE were explained, along with an introduction to some of the most significant trained models. Ultimately, BlazePose was chosen for this work. The decision was based on the objective requirements, prioritising a balance between real-time processing capabilities and high estimation accuracy. Furthermore, the focus on fitness exercises plays a significant role [49]. Although AlphaPose and OpenPose deliver superior quality results, their longer computation times render them less suitable for real-time applications [59]. Additionally, BlazePose was specifically developed for fitness exercises, making it the primary choice for tracking in this context. While ShuffleNet also offers high computational speed, it lacks the estimation accuracy found in BlazePose [54]. The drawback of the HRNet is its significantly higher computational intensity compared to the other models, a limitation similarly applicable to DeepCut [51]. Besides this, a single person estimation is sufficient for the case of application of this process. Another essential aspect in choosing MediaPipe, which supports BlazePose, was the need to implement a system that was powerful and efficient in real-time processing and offered cross-platform support. MediaPipe's modular and extensible architecture allows easy customization to meet specific requirements, which is especially valuable for fitness exercise analysis. In addition, it was a goal to develop a system compatible with low-cost hardware and provide a user-friendly experience to ensure broad accessibility and ease of use. This combination of technical performance, adaptability to fitness applications, cross-platform support, cost and ease of use made MediaPipe an ideal choice for developing a HPE system that works in real-time and is easily accessible to the end user. In view of these considerations, the decision is made in favour of BlazePose.

3. Development of the Motion Quality Assessment Algorithm

This chapter introduces the mathematical foundations at the core of the algorithm, providing a comprehensive overview of the entire algorithm, building upon the estimation of human poses discussed in the preceding chapter. Subsequently, it conducts a detailed analysis of each step the algorithm undergoes, as well as the various modes of operation in which it functions. To introduce this chapter, it is necessary first to present the theoretical foundations and methods essential for understanding the structure and functionality of the algorithm. An explanation of the data collection, the setup and a basic description of the dataset are provided at the beginning of the chapter. This methodological approach facilitates a profound understanding of the algorithm's functionality - from the initial estimation of the pose to the complex details of its foundations. Based on the foundations described below, the MoQuA algorithm was implemented, laying the groundwork for the evaluations in the following chapter.

3.1. Data collection and setup

In order to develop a system to improve physiotherapy practice, collecting data on sports exercises is essential. The following outlines the data collection process necessary for the system's development. The primary objective is to record sports exercises using a dual-camera setup. The participant is equipped with a MoCap suit to validate the accuracy of the system's pose estimation. This process will incorporate a variety of perspectives by having the participant assume different orientations towards the cameras and occupy various positions within the room. In the experimental setup, the cameras are strategically positioned in a vertical plane parallel to the ground, oriented at a 90-degree angle relative to each other. The participant is positioned centrally between the two cameras, each covering an approximate area of 5 x 5 meters and 3 x 3 meters. The participant's orientation

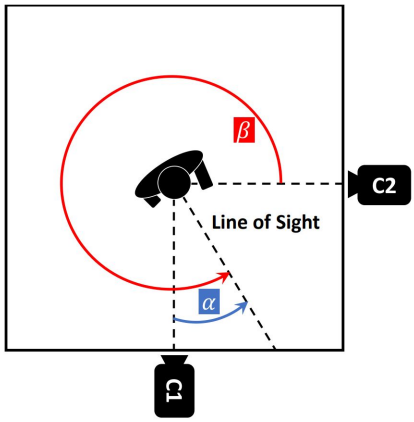


Figure 3.1.: Top-down perspective of the experimental configuration by [60]. Cameras were positioned to ensure that their image planes were perpendicular to the floor surface. The line of sight for the participant is represented by the angle α for Camera 1 (C1) and the angle β for Camera 2 (C2).

towards the cameras is quantified with the line of sight using two angular parameters: α for Camera 1 and β for Camera 2. The experimental setup is shown in Figure 3.1. This alignment ensures comprehensive coverage and optimal data acquisition from multiple viewpoints. This arrangement is designed to maximise the efficacy of the data collection process, ensuring a robust foundation for developing the physiotherapy support system.

3.1.1. Equipment

The material utilised in this study includes two EMEET Full HD Webcam - C960 cameras[1] or the Reolink RLC-510A[2], which is an low-cost security type camera and an MTw Awinda Motion Capture system by Movella[3]. The Webcam cameras, chosen for their cost-effectiveness, boast a resolution of 1080p and a frame rate of 29.5 frames per second. The Reolink has a resolution of 2560×1920 and a frame rate of 30 frames per second. The MoCap suit, on the other hand, records data at a rate of 60 Hz. Therefore, it necessitates down-sampling the captured data to 29.5 (or 30) Hz to synchronise with each camera's frame rate. Due to the length of the recording sequence, it is necessary to adjust the

[1] https://emeet.com/en-eu/products/webcam-c960
[2] https://reolink.com/de/product/rlc-510a/
[3] https://www.movella.com/products/wearables/xsens-mtw-awinda

frame rate precisely, otherwise the synchronisation will not be successful. The MoCap suit is designed to provide the x-y-z coordinates of 23 different body segments and 66 joint angles, such as the flexion and extension of the right knee. The MoCap suit has proven to provide reliable and accurate data for human movement [61] and has been successfully used in exercise-related motion analysis [62]. The previous mentioned arguments leads to the conclusion that the MoCap suit is suitable for validating the data.

3.1.2. Data and setup

The data collected encompass a total of 372 execution units, which were recorded over three separate sessions. The first data collection session utilised Reolink cameras, while a different camera setup was employed for the subsequent two sessions. Overall, recordings were made of seven different individuals. Throughout the course of the study, 20 rounds of various exercises were conducted. During the initial session, five different exercises were performed, followed by seven different exercises in the subsequent session. The first session was distinguished by the professional accompaniment of a physiotherapist. For the first session, an extended field of 5 x 5 meters was used, whereas, in the following two exercise sessions, a more compact field of 3 x 3 meters was utilised. Each exercise was correctly performed with two repetitions, and two specific errors were documented for each, allowing for a comprehensive analysis of the executions, including the errors that occurred. An overview of all rounds with corresponding angles can be found in the appendix A.1.

The push-up, a fundamental exercise, commences from a prone position with arms extended, facilitating the elevation and lowering of the upper body. A prevalent error in this exercise is the misalignment caused by excessively high hips, which disrupts the ideal body line that should extend from the shoulders to the feet, primarily affecting the torso. Additionally, the inclination to elevate the gaze upwards rather than maintaining it downwards leads to a misalignment of the spine and neck, further detracting from the exercise's effectiveness.

Squats, integral for leg and hip strength, involve lowering the body by bending the knees and hips. However, inadequacies such as failing to descend sufficiently, resulting in a knee angle exceeding 90 degrees, compromise leg muscle engagement. Moreover, positioning the feet wider than shoulder-width can detrimentally impact the legs and hips, deviating from the exercise's intended benefits.

Kick-backs performed on all fours require one leg to be extended backward and upward, presenting errors such as looking upwards, which strains the neck. Additionally, positioning the knees too close to the hands can disrupt the intended alignment, affecting both the upper body and legs.

The simulation of swimming movements (breaststroke), often performed dry, underscores the importance of proper head positioning. Directing the head upwards instead of towards the ground strains the neck, while bringing the shoulder too close to the head can adversely affect the shoulders, undermining the exercise's mimicry of swimming dynamics.

A second variant of the push-up, where the knees are on the ground, introduces variations in execution yet shares common errors with the first variant, such as the upward gaze and a curved downward back, which lead to misalignment and strain on the neck, spine, and torso.

Lunges, involving a step forward followed by lowering the body until both knees are bent, can be compromised by taking too small a step or turning the front knee inward, affecting the legs, hips, and knee, and detracting from the exercise's effectiveness in strengthening these areas.

Finally, sit-ups, aimed at engaging the abdominal muscles by lifting the upper body towards the knees, present challenges when the knees are not bent sufficiently or the feet lift off the ground, affecting the abdominal muscles and legs, and reducing the exercise's efficacy. A detailed description of the exercise can be found in the appendix B.

The variety of exercises recorded, as well as their performance by various individuals, allows for the examination of both the effects of the exercises and the effects caused by individual variations between participants. The dataset, while not overly large, is sufficiently comprehensive to train models effectively. This setup should enable the development of a highly accurate model, demonstrably showing that, for instance, even a single correct recording can be used to achieve significantly reasonable results.

3.2. Motion Quality Assessment (MoQuA) Algorithm

Secondly, the MoQuA algorithm is described, illustrating the pipeline from the input of video material through preprocessing, HPE, similarity comparison, to classification. This section briefly outlines the individual steps. The precise mathematical functionality and background are detailed in the subsequent Subsections 3.3 to 3.7. A schematical view

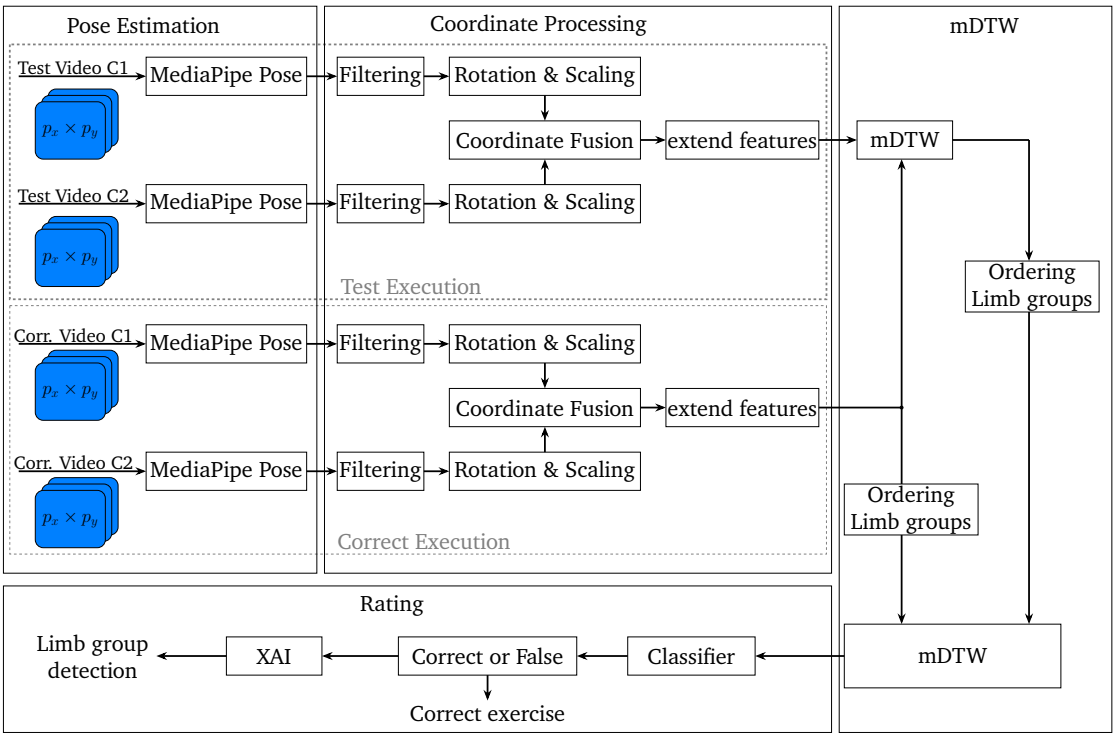


Figure 3.2.: This figure illustrates the proposed end-to-end algorithm utilised for the MoQuA Algorithm of sports exercises. Using Mediapipe, videos are converted into position trajectories, followed by filtering, fusing and normalisation. The mDTW is then used to determine the similarity to the Golden Standard, and the features are evaluated using a classifier. Finally, the limb group is determined by XAI.

of the algorithm is shown in Figure 3.2. The collected data consists of a person who is filmed by two cameras during the execution of sports activities, as described in Section 3.1. The process begins with footage from Camera 1 and Camera 2, each recording a video comprising a series of images, resulting in

$$\mathbf{V}_{cam,type} \in \mathbb{R}^{p_x \times p_y \times f_l}, \tag{3.1}$$

where p_x and p_y denote the number of pixels in the x and y direction respectively, f_l denotes the length of the video, $cam \in \{1, 2\}$ represents the individual camera and $type \in \{\text{Test}, \text{Correct}\}$, indicates whether it is a test video or the Golden Standard. In the given context, the term ”test video” implies that there is a recording for which it needs to be determined whether its execution was correct or incorrect. Furthermore, the method aims

to identify which group of limbs is responsible for any possible incorrect performance. The term ”Golden Standard” refers to the method where, from the exercises that were correctly performed, one is randomly selected and used as a basis for comparison.
The videos are then processed using the MediaPipe posture estimation as previously described in Chapter 2.2, producing features of

$$\mathbf{Z}_{cam,type} \in \mathbb{R}^{33\times 3\times f_l}, \tag{3.2}$$

due to the 33 keypoints from MediaPipe, with each keypoint comprising 3 coordinates for every frame. Subsequently, the features are filtered, rotated, and scaled to normalise them, to improve the estimation and to create a better comparability among each other. Following this step, the coordinates are merged by combining the camera footage. This means $\mathbf{Z}_{1,type}$ and $\mathbf{Z}_{2,type}$ merged to $\widetilde{\mathbf{Z}}_{\text{type}}$. In addition to the joint coordinates, relative features are calculated. This creates additional features over and above the initial 33. These features are not 3D points but scalar values and are expressed by

$$\widehat{\mathbf{Z}}_{\text{type}} \in \mathbb{R}^{\kappa\times f_l}, \tag{3.3}$$

where κ describes the number of additionally calculated features, which are described in Section 3.4.5. These steps are done for all videos regardless of their type. The trajectories from the test video and the Golden Standard have different lengths, f_{l1} and f_{l2}. In order to match the sequence length the multidimensional Dynamic Time Warping (mDTW) [63] is applied, as described in Section 3.5.1. This changes the length of the signal to $\widetilde{f}_{l1}$ where applies $max(f_{l1}, f_{l2}) \leq \widetilde{f}_{l1} \leq (f_{l1} + f_{l2}–1)$ and $\widetilde{f}_{l1} \in \mathbb{N}$. The data is then categorised into different limb groups as a noise reduction measure. The categorization into various limb groups is a parameter that can be adjusted, changing the data dimensions

$$\begin{aligned}\widetilde{\mathbf{Z}}^{33\times 3\times f_l}_{\text{type}} &\rightarrow \widetilde{\mathbf{G}}_{\text{type}} \in \mathbb{R}^{g_{l1}\times 3\times f_l},\\ \widehat{\mathbf{Z}}^{\kappa\times f_l}_{\text{type}} &\rightarrow \widehat{\mathbf{G}}_{\text{type}} \in \mathbb{R}^{g_{l2}\times f_l}.\end{aligned}$$

In this context, g_{li} denotes various categorizations into limb groups (number of limb groups) based on a foundational order that commences with the head, torso, left arm, right arm, left leg, and right leg. This sequential arrangement facilitates a structured approach to analyzing and processing data pertaining to human movement, by segmenting the body into its principal limb groups. Such, the approach not only simplifies the assessment of posture and movement through noise reduction, but is also intended to increase the specificity and accuracy of the analysis. The noise reduction is achieved by averaging the individual trajectories. The synchronised test video and correct video signals are then further processed in the mDTW calculation to determine the distance measure

($\mathbf{d} \in \mathbb{R}^{g_{l1}+g_{l2}}$). This distance measure, calculated for each limb group, is then fed into a classifier that decides whether the execution is correct or incorrect. In case of an incorrect execution, the most probable error is identified within the group using XAI, namely LIME and SHAP, providing feedback to the subject. The following sections will elaborate on these steps in detail. This structured approach ensures a thorough analysis and correction of posture through video analysis, leveraging the mathematical framework of DTW for synchronization and MediaPipe posture estimation for initial data capture. Incorporating a classifier for final assessment enhances the system's utility by providing actionable feedback, which is essential for applications in physiotherapy, sports science, and personal fitness.

3.3. Camera geometry

The first part of the MoQuA algorithm deals with the systematic processing and analysis of video material, which is the primary data source for the explained algorithm. The challenge is to extract structured and actionable data from the available video content to capture the dynamics and complexity of the depicted scenarios adequately. A key aspect in the processing of video material is the application of techniques for transforming the three-dimensional reality of the world into a two-dimensional image representation. This transformation enables us to describe phenomena and processes captured in the videos on a 2D plane, making them accessible for further analysis. Another emphasis is on the fusion of data from various sources to achieve a more comprehensive and detailed representation. Combining information captured from different perspectives and with various technical aims to enrich the analytical base and improve the significance of the findings. To maximise the quality and expressiveness of the derived image information, it is necessary to identify and correct distortions caused by suboptimal camera positioning and settings. This involves using models and techniques developed explicitly for correcting such distortions, such as the pinhole camera model and lens distortion correction techniques. These approaches play a vital role in improving image quality and forming the foundation for precise 3D reconstructions of the recorded scenes. By carefully considering and applying these methods ensuring the high quality and reliability of the data extracted from the video material, which is important for the subsequent analysis steps.

3.3.1. Pinhole camera model

The pinhole camera model is a simple model that illustrates the fundamentals of photography and projection. The concept behind this model originates from the observation that light, passing through a small opening (a ”pinhole”) in a dark room, can create an image on the opposite side. This model is frequently used as a foundation for understanding more complex camera and optical systems. The camera geometry described below refers to Hartley and Zisserman [64] and Escalera *et al.* [65].

The central concept in the pinhole camera model is shown in Figure 3.3:

- Camera Center (**c**): The origin of the coordinate system through which central projection occurs.
- Principal Axis (Z): The line perpendicular from the camera centre to the image plane.
- Principal Point (**p**): The point on the image plane where the principal axis intersects it.
- Image Plane (Z = f): The plane where the image is formed, located at a distance f (focal length) from the camera center.

The basic equation that describes the projection of a 3D point in space$[x, y, z]$ onto a 2D point $[x_{im}, y_{im}]$ on the image plane is derived using similar triangles:

$$x_{im} = f\frac{x}{z}, \qquad y_{im} = f\frac{y}{z}. \tag{3.4}$$

The assumptions of the pinhole camera model are idealised, and in real cameras, lenses and other optical elements are used to further focus and direct light. Despite its simplicity, the pinhole model provides a fundamental explanation of how images are created through projection and illustrates the geometric relationship between objects in space and their images on a flat image plane.
Equation 3.4 implies that the coordinate origin in the image plane is located at the principal point. In practice, this is often not the case, meaning that a mapping generally exists where $[c_x, c_y]^T$ are the coordinates of the principal point, forming an offset for the

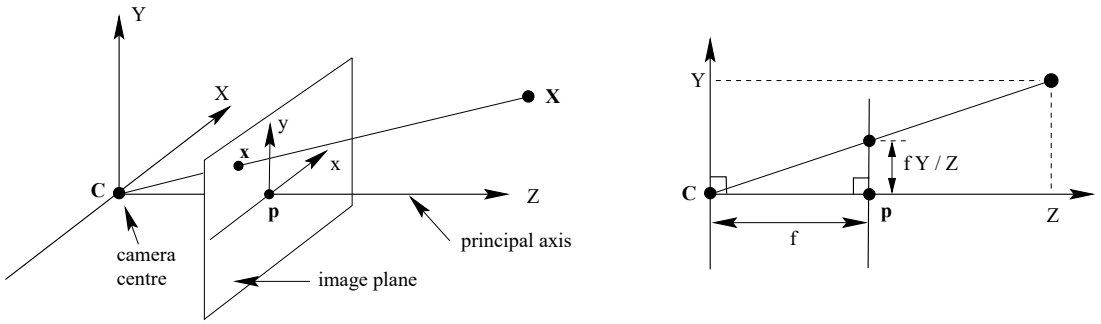


Figure 3.3.: A systematic overview of the geometry of a pinhole camera [64]

computation of the image's origin. The mapping is expressed by

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} \rightarrow \begin{bmatrix} f\frac{x}{z} + c_x \\ f\frac{y}{z} + c_y \end{bmatrix}. \tag{3.5}$$

Expanding this mapping to homogeneous coordinates results in

$$\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \rightarrow \begin{bmatrix} fx + zc_x \\ fy + zc_y \\ z \end{bmatrix} = \begin{bmatrix} f & 0 & c_x & 0 \\ 0 & f & c_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}. \tag{3.6}$$

Homogeneous coordinates are a coordinate system used in projective geometry to represent points at infinity and to perform transformations such as translations, rotations and scaling. Homogeneous coordinates are obtained by adding an additional dimension to traditional Cartesian coordinates, simplifying mathematical operations and perspective representation. Now, the matrix is written as

$$\mathbf{K} = \begin{bmatrix} f & 0 & c_x \\ 0 & f & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{3.7}$$

following in

$$\mathbf{x} = \mathbf{K}[I|0]\mathbf{x}_{\text{cam}}. \tag{3.8}$$

The matrix **K** is referred to as the camera calibration matrix. In Formula 3.8, the vector $[x, y, z, 1]^T$ is denoted as $\mathbf{x}_{\text{cam}}$. The point $\mathbf{x}_{\text{cam}}$ is represented in its own coordinate system, which can be called the camera coordinate system.

To elaborate further, the camera coordinate system is a special reference frame in which the camera serves as the origin. All measurements and positions of objects are made relative to this point. The Z-axis of this system runs along the optical axis of the camera, pointing in the direction the camera is facing. The camera's principal point, the point on the image sensor that lies directly opposite the lens, is located on the Z-axis in this system.

The camera calibration matrix **K** includes the intrinsic parameters of the camera, which are independent of the scene being viewed. These parameters include the camera's focal length, represented by the diagonal element f in the matrix, the position of the principal point on the sensor, given by the elements c_x and c_y, with the neglect of the tilt of the x- and y-axis of the sensor (skew factor) due to the insignificance.

Through camera rotation and translation, it is possible to consider the relationship between the world coordinate system and the camera coordinate system. These systems are linked through rotation and translation to describe the position and orientation of points in three-dimensional space from the camera's perspective. Given a point $\mathbf{x}_{\text{world}}$ with coordinates $[x, y, z]$ in the world coordinate system, to transform this point into the camera coordinate system, the camera's position and orientation is accounted for in the world coordinate system. The camera is located at a specific point **c** in the world coordinate system, given by the vector $\mathbf{c} = [c_x, c_y, c_z]$. To determine the position of a point relative to the camera position, the camera centre is subtracted vector **c** from the world coordinate vector $\mathbf{x}_{world}$, resulting in $\mathbf{x}_{world} - \mathbf{c}$. The camera can be oriented in various directions. The orientation of the camera in space is described by a rotation matrix **R**. This 3x3 matrix transforms points from the world coordinate system into the camera coordinate system, taking into account the orientation of the camera. The transformation is given by $\mathbf{R}[\mathbf{x}_{\text{world}} - \mathbf{c}]$. To represent these transformations in homogeneous coordinates, the vectors are extended to 4-dimensional vectors by adding an additional dimension with the value 1. This allows us to represent translations as matrix multiplications. Finally, the camera's intrinsic parameters are considered through the camera calibration matrix **K**. In summary, the transformation of a point $\mathbf{x}_{\text{world}}$ in the world coordinate system into a point $\mathbf{x}_{cam}$ in the camera coordinate system and finally onto the image plane is:

$$\mathbf{x}_{\text{cam}} = \mathbf{R}[\mathbf{x}_{\text{world}} - \mathbf{c}]. \tag{3.9}$$

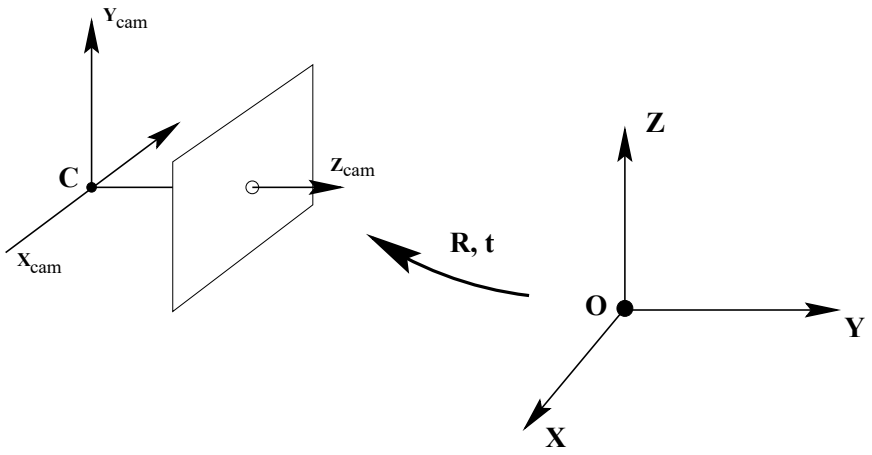


Figure 3.4.: The Euclidean transformation between the world and camera coordinates [64]

for homogeneous coordinates, the following ˜ is used. In homogeneous coordinates:

$$\tilde{\mathbf{x}}_{\text{cam}} = \begin{bmatrix} \mathbf{R} & -\mathbf{R}\tilde{\mathbf{c}} \\ \mathbf{0} & 1 \end{bmatrix} \tilde{\mathbf{x}}_{\text{world}}. \tag{3.10}$$

And then using the camera calibration matrix:

$$\mathbf{x}_{\text{im}} = \mathbf{K}\mathbf{x}_{\text{cam}}. \tag{3.11}$$

By combining these steps, the projection matrix **P** is obtained, which incorporates both the external calibration (position and orientation of the camera) and the internal calibration (intrinsic camera parameters):

$$\mathbf{x}_{im} = \mathbf{P}\mathbf{x}_{world} = \mathbf{K}\begin{bmatrix} \mathbf{R} & -\mathbf{R}\mathbf{c} \end{bmatrix} \mathbf{x}_{world}. \tag{3.12}$$

Therefore:

$$\mathbf{P} = \mathbf{K}\begin{bmatrix} \mathbf{R} & -\mathbf{R}\mathbf{c} \end{bmatrix}. \tag{3.13}$$

The projection matrix **P** has a total of 9 degrees of freedom: 3 from the calibration matrix **K** (focal length and coordinates of the principal point), 3 from the rotation matrix **R**, and 3 from the translation vector $\mathbf{t} = -\mathbf{Rc}$ [64]. Equation 3.13 can thus be rewritten as

$$\mathbf{P} = \mathbf{K}\left[\mathbf{R}|\mathbf{t}\right]. \tag{3.14}$$

A chessboard pattern can be employed to calculate the camera matrix, rotation, and translation, which is a standard method of camera calibration based on the precise geometric properties of the chessboard. The process initiates with capturing multiple photographs of the chessboard pattern from various angles and distances, ensuring that the pattern is clearly visible in each image. Subsequently, specialised algorithms are utilised to accurately identify the corners of the chessboard pattern in these images. Each detected corner is considered as a point, the image position of which is known. The chessboard pattern facilitates the precise determination of corresponding points, enabling an optimal estimation. However, a disadvantage is that additional recordings must be made, which requires additional recordings. In MediaPipe, the output of 2D image coordinates is also available. These coordinates can likewise be used as corresponding points to determine the transformation between the two systems. To determine **R** and **t**, the corresponding points $\mathbf{x}_1, \mathbf{x}_2$ of both cameras are used to solve the following equations:

$$\mathbf{x}_1^\top \mathbf{E}\mathbf{x}_2 = 0,$$

where the following relationship is given

$$\mathbf{E}_{21} = \mathbf{t}_{21} \times \mathbf{R}_{21}.$$

Here, **E** is referred to as the essential matrix. The described camera geometry enables the derivation of both intrinsic and extrinsic camera parameters, facilitating a comprehensive calibration of the camera system. This model describes all necessary intrinsic and extrinsic parameters [65].

3.3.2. Lens distortion

The emergence of lens distortion in the context of the pinhole camera model is related to the limitations and physical properties of real lens systems. In reality, lenses are used instead of pinholes to capture more light and produce a brighter image. The curvature and material of the lens cause light rays not to be perfectly focused on a point, leading to distortions in the image. There are two main types of lens distortions: barrel distortion

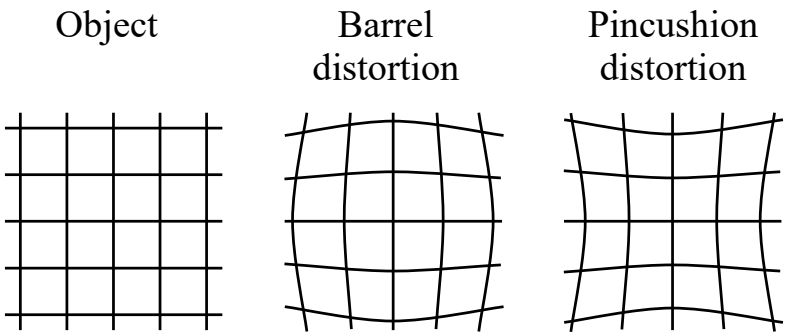


Figure 3.5.: On the left is an undistorted object; on the right is pincushion distortion, where the lines are curved towards the centre and on the middle is barrel distortion, where the lines curve outwards. These images illustrate the typical effects that optical aberrations in lens systems can cause [66].

and pincushion distortion, as show in Figure 3.5. Barrel distortion occurs when rays at the edges are bent more than those closer to the centre, causing objects near the edges of the image to appear curved and further from the centre. The result is an image that appears curved, akin to the surface of a sphere. In pincushion distortion, the edge rays are bent less than the central rays, causing objects near the edges of the image to appear bent inwards and closer to the centre, creating a cushion-shaped appearance. Barrel distortion and pincushion distortion are specific types of radial distortion. While radial distortion pertains to the shape-altering effects in an image, tangential distortion arises from the misalignment of the image sensor and the lens, causing the image to appear skewed. The radial distortion can be described by the following equations [67]:

$$x_{\text{corr}} = x_{\text{dist}}(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \tag{3.15}$$

$$y_{\text{corr}} = y_{\text{dist}}(1 + k_1 r^2 + k_2 r^4 + k_3 r^6). \tag{3.16}$$

The tangential distortion is represented by:

$$x_{\text{distorted}} = x + [2p_1 xy + p_2(r^2 + 2x^2)] \tag{3.17}$$

$$y_{\text{distorted}} = y + [p_1(r^2 + 2y^2) + 2p_2 xy]. \tag{3.18}$$

By combining and simplifying the equations, one obtains the distortion coefficients:

$$\text{Distortion coefficients} = (k_1, k_2, p_1, p_2, k_3). \tag{3.19}$$

With the help of the camera matrix **K**, the image can be corrected after determining the distortion coefficients. To correct an image impaired by lens distortions, the process begins with calculating the undistorted coordinates for each pixel in the distorted image. Thereby, the calculation uses previously determined distortion coefficients and refers to the corresponding mathematical equations that model both radial and tangential distortions. This step determines a new, corrected location in space for each pixel in the distorted output image, reflecting its position in the idealised, undistorted image. However, these newly calculated coordinates do not always result in integer pixel positions, preventing a direct assignment of pixel values in the digital image grid because pixels always have whole numbers as coordinates. To overcome this challenge, an interpolation technique is applied. Techniques such as bilinear or bicubic interpolation allow for the estimation of the intensity values of the undistorted pixels by considering the values of neighbouring pixels, thereby creating a smooth and coherent image surface. Finally, the reconstruction of the undistorted image is completed by using the interpolated intensity values to create a new image free from the original distortions. The final step presents the visual result of the preceding mathematical and processing efforts, delivering a corrected image that more accurately reflects the real scene features than the originally distorted image. Through these carefully coordinated steps, the image quality is significantly improved, enhancing visual precision for various applications, from photography to precise image analysis [67]. Due to the often enhanced distortion in use, correction is essential to improve the accuracy of estimating body positions.

3.3.3. 3D reconstruction

3D reconstruction is an advanced computational process aimed at capturing the three-dimensional structure of the physical world from two-dimensional data, such as images or videos. Thus, a important bridge between digital perception and tangible reality has been forged. As an alternative to 3D world coordinates, 2D image coordinates can also be estimated, allowing the transformation of data from two cameras into 3D coordinates. This process assumes a successful camera calibration, which was described in Section 3.3.2. By calculating the projection matrices $\mathbf{P}_1$ and $\mathbf{P}_2$ for Camera 1 and Camera 2, as described in Section 3.3.1, and assuming that the two 2D image coordinates of the keypoints for each camera have been computed, the 3D points can be reconstructed using Direct Linear Transform (DLT) and Singular Value Decomposition (SVD) [68]. DLT is a mathematical method frequently employed in computer vision and photogrammetry to determine the parameters of a projection matrix or transformation between two coordinate systems. This method is particularly useful for solving problems such as camera calibration, 3D

point reconstruction from image pairs, and image registration. DLT involves solving a linear equation of the form

$$\mathbf{Ax} = \mathbf{0}, \tag{3.20}$$

where $\mathbf{A}$ is a matrix constructed from the coordinates of corresponding points in the two coordinate systems, and $\mathbf{x}$ is the vector of unknown transformation parameters that should be determine. The solution to this equation is typically obtained through SVD, a technique that allows finding the best-fitting solution in terms of minimizing the square of the errors, even when the system is overdetermined or impaired by measurement errors [69]. This means that it is assumed

$$\mathbf{u_1} = \begin{bmatrix} u_1 \\ v_1 \\ 1 \end{bmatrix}, \quad \mathbf{u_2} = \begin{bmatrix} u_2 \\ v_2 \\ 1 \end{bmatrix}$$

as homogenised 2D-pixel coordinates. The homogenised 3D point

$$\mathbf{x} = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

is related to the 2D points through

$$\mathbf{u_i} = \mathbf{P}_i\mathbf{x}, \tag{3.21}$$

as described in Equation 3.12. The objective is to determine the unknown elements within $\mathbf{x}$. Given that $\mathbf{u_1}$ and $\mathbf{p_i}$(part of $\mathbf{P}_i$), $\mathbf{x}$ are vectors that run parallel to each other, taking their cross-product should result in zero. This leads to the following conclusion [70, 71]:

$$\begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} \times \begin{bmatrix} \mathbf{p_1x} \\ \mathbf{p_2x} \\ \mathbf{p_3x} \end{bmatrix} = \begin{bmatrix} v_i\mathbf{p_3x} - \mathbf{p_2x} \\ \mathbf{p_1x} - u_i\mathbf{p_3x} \\ u_i\mathbf{p_2x} - v_i\mathbf{p_1x} \end{bmatrix} = \begin{bmatrix} v_i\mathbf{p_3} - \mathbf{p_2} \\ \mathbf{p_1} - u_i\mathbf{p_3} \\ u_i\mathbf{p_2} - v_i\mathbf{p_1} \end{bmatrix} \mathbf{x} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}. \tag{3.22}$$

The row vectors of $\mathbf{p_1}$ are represented as $\mathbf{p_i}$, which are four-dimensional vectors. The formulation 3.22 leads to an equation of the type $\mathbf{Ax} = \mathbf{0}$. The third row is a linear combination of the first two rows, resulting in only two equations that are insufficient for solving the three unknowns in $\mathbf{x}$. This outcome is anticipated, as determining a 3D coordinate from a single camera perspective is not feasible. Assuming that two cameras

are utilised, it's possible to augment the matrix with additional rows. Indeed, extra rows corresponding to any number of views can be appended, leading to the following equation:

$$\mathbf{Ax} = \begin{bmatrix} v_1\mathbf{p_3} - \mathbf{p_2} \\ \mathbf{p_1} - u_1\mathbf{p_3} \\ v_2\mathbf{p_3} - \mathbf{p_2} \\ \mathbf{p_1} - u_2\mathbf{p_3} \\ \vdots \end{bmatrix} \mathbf{x} = \mathbf{0}. \tag{3.23}$$

The goal is to find the non-trivial solution for an equation structured as $\mathbf{Ax} = \mathbf{0}$. Considering the presence of noise in practical scenarios, one could reformulate the equation to $\mathbf{Ax} = \mathbf{w}$, aiming to solve for $\mathbf{x}$ in a manner that minimises $\mathbf{w}$. The initial step involves calculating the (SVD) of $\mathbf{A}$ [68, 71].

$$\mathbf{Ax} = \mathbf{USV}^T\mathbf{x}. \tag{3.24}$$

To reduce $\mathbf{w}$ for a given $\mathbf{x}$ can be achieved through the computation of the dot product:

$$\mathbf{w}^T\mathbf{w} = \left(\mathbf{x}^T\mathbf{VSU}^T\right) * \left(\mathbf{USV}^T\mathbf{x}\right) = \mathbf{x}^T\mathbf{VS}^2\mathbf{V}^T\mathbf{x}. \tag{3.25}$$

$\mathbf{U}$ and $\mathbf{V}$ are orthonormal matrices, with $\mathbf{S}$ being a diagonal matrix. Furthermore, the diagonal values of $\mathbf{S}$ decrease in magnitude, making the final diagonal entry the smallest. The resulting characteristic is assured by the SVD decomposition process. Using the orthonormal form of $\mathbf{V}$, selecting $\mathbf{x}$ as one of the column vectors from $\mathbf{V}^T$:

$$\mathbf{v_i}^T\mathbf{VS}^2\mathbf{V}^T\mathbf{v_i} = s_i^2. \tag{3.26}$$

Based on the previous explanations, it can be concluded that the i'th diagonal element of $\mathbf{S}$ is denoted as s_i. Given the objective to minimise $\mathbf{w}_T\mathbf{w}$, it becomes evident that this equates to selecting the smallest value within $\mathbf{S}^2$ by choosing the corresponding $\mathbf{v}_i$ column vector from $\mathbf{V}^T$ as $\mathbf{x}$. The minimum value is achieved when opting for the final column vector of $\mathbf{V}^T$ as $\mathbf{x}$. Consequently, the $\mathbf{Ax} = \mathbf{w}$ equation has been addressed amidst the noise. This SVD approach is effective even in the absence of noise [69]. In the presence of noise, a filtering effect occurs, which attenuates the noise. This characteristic makes SVD a powerful tool not only for achieving a precise decomposition of data but also for enhancing data quality by reducing noise impact. Through the selective retention of singular values that represent significant data features while diminishing those associated with noise, SVD inherently incorporates a denoising process into its application. The dual functionality of SVD, combining data decomposition with noise reduction, significantly benefits various data analysis and signal processing applications, ensuring clearer and more reliable outcomes even in noisy environments [70].

3.4. Coordinates preprocessing

This chapter gives an overview of the complex and multifaceted processes required to transform raw data from various sensors into useful, accurate environmental information. Initially, there is a description of the synchronization necessary for the temporal alignment of data captured by different sensors. Since sensors often collect data at different times and with varying frequencies, a robust system must be developed to synchronise these data streams, allowing them to be meaningfully combined and analyzed. Rotations and scaling are required for full coordinate processing. Rotations are necessary to understand the orientation of objects in space, while scaling is used to bring measurement data to a typical magnitude. Both concepts are essential for the correct interpretation of spatial relationships.

A significant part of data processing is filtering. Here, the methods of moving average and low-pass filter are employed to reduce noise and increase the accuracy of sensor data. Noise reduction allows for an accurate assessment of a system's state, even with uncertainties and measurement errors. A necessary next step is turning to sensor fusion, a process that combines data from different cameras to obtain a more comprehensive and accurate view of the environment. Angle calculation is essential as a relative size that describes human posture. Finally, relative values are determined, a method that makes it possible to describe features independently of the coordinate system, For example, the MoCap suit has a different coordinate origin than the MediaPipe out. This enables the evaluation of the HPE compared to the MoCap suit.

3.4.1. Camera synchronization

Synchronization is indispensable to integrate data from diverse sources effectively. Data from a specific position share identical timestamps, essential when fusing data from two cameras. Synchronization uses the MoCap suit as a reference, aligning y-plane movement from MediaPipe with the z-coordinate from the MoCap suit, necessitating matching sampling rates. Signals with higher frequencies undergo downsampling, adjusting the MoCap suit's 60 Hz to the camera's 30 or 29.5 Hz. These signals, indicating frame-wise differences on the specified axis, serve as robust features for assessing sequence dynamics. The method applies outlier removal to diminish noise and artifacts impact, employing a 15 Interquartile Range (IQR) threshold from the 25th and 75th percentiles of motion amplitude, enhancing correlation analysis reliability.

The synchronization process utilises cross-correlation between the motion signals to identify the temporal offset (lag) that maximises the alignment of these signals across data sources. Followed by segmenting the relevant segments, computing the full cross-correlation, and determining the lag corresponding to the highest correlation coefficient. The resulting lag indicates the necessary adjustment: a positive value implies that the first signal starts earlier, while a negative value indicates the opposite. The mathematical representation of the correlation z between two motion signals follows. If $\mathbf{x}$ and $\mathbf{y}$ denote arrays and $\mathbf{z} = \text{correlate}(\mathbf{x}, \mathbf{y})$, then the relationship can be expressed as:

$$\mathbf{z}[k] = (\mathbf{x} * \mathbf{y})(k - N + 1) = \sum_{l=0}^{|\mathbf{x}|-1} \mathbf{x}_l \mathbf{y}_{k-l-N+1}$$

where k ranges from 0 to $|\mathbf{x}| + |\mathbf{y}| - 2$. Here, $|\mathbf{x}|$ represents the length of $\mathbf{x}$, N is the larger of the lengths of $\mathbf{x}$ and $\mathbf{y}$, and $\mathbf{y}_m$ is considered to be 0 for any m not within the actual bounds of $\mathbf{y}$ [72].

Thereby, the resulting determination provides a reliable indication of the time difference, enabling the alignment of the signals to a common starting point. The synchronization approach, as described, offers a robust framework for aligning data from diverse sources based on their movement characteristics.

3.4.2. Filtering

Two methods are introduced to reduce the noise in the measurements: the moving average and the low-pass filter. Applying a moving average filter is a fundamental technique in data processing, beneficial for smoothing out short-term fluctuations and highlighting longer-term trends or cycles within a dataset. The rationale behind filtering data, especially using a moving average filter, stems from the need to mitigate the impact of noise and outliers that can obscure meaningful insights in raw data. Noise can originate from various sources, including sensor inaccuracies, environmental variability, or random disturbances and can significantly distort the actual signal. A moving average filter calculates a series of averages from various subsets of the entire dataset. The main idea is to calculate the average of the data points within a specific window that slides across the data. This process smooths out short-term fluctuations and reveals the underlying trend by averaging adjacent values over a specified period. The basic formula for a simple moving average of a dataset is given by:

$$\text{MA}(t) = \frac{1}{N} \sum_{n=0}^{N-1} \mathbf{x}(t-n), \tag{3.27}$$

where $\text{MA}(t)$ is the moving average at time t, N is the number of time periods in the moving average window, and $\mathbf{x}(t-n)$ represents the data point at time $t-n$ [73].

The choice of N, the window size, is important as it determines the filter's smoothing extent. A larger window will provide a smoother signal but can cause significant changes in the data, whereas a smaller window will follow the data more closely but with less smoothing effect. This filtering method is favoured for its simplicity, effectiveness, and the fact that it requires no assumptions about the underlying data distribution, making it a versatile tool in various data analysis applications [73].

Another essential tool in data filtering is the low-pass filter, which aims to attenuate high frequencies while allowing low frequencies to pass through. This type of filter is particularly useful for reducing noise manifested as high-frequency fluctuations and for highlighting the fundamental, often low-frequency signals in a dataset. Low-pass filters are commonly used in signal processing to smooth out unwanted disturbances or rapid changes in the data, which can be considered noise [74].

In digital signal processing, a low-pass filter can be implemented through discrete convolution of the input signal with a filter kernel or impulse response function. The formula for a simple digital low-pass filter can be expressed as follows:

$$y[n] = \sum_{k=0}^{M} b_k \cdot x[n-k], \tag{3.28}$$

where $\mathbf{y}[n]$ is the output signal, $\mathbf{x}[n]$ the input signal, $\mathbf{b}_k$ the filter coefficients defining the impulse response of the filter, and M the order of the filter. The choice of coefficients b_k and the order M of the filter are adjustable parameters that determine the filter's frequency characteristics, thus controlling how effectively the filter suppresses high frequencies. Low-pass filters are similar in function to moving averages, but use weighted averages to refine the data analysis. It is required that the sum of the weights is equal to one in order to preserve the original data magnitude [74].

The application of a low-pass filter removes high-frequency noise while preserving the essential, slow changes in the signal. This renders it an indispensable tool in various fields, from electronics to data analysis, where maintaining signal integrity while reducing noise is required. Both filtering techniques, the moving average and the low-pass filter, are utilised to improve the signal quality. According to the evaluation, the moving average provides superior outcomes and is consequently selected as the default method.

3.4.3. Rotation and scaling

In the process of pose estimation, two significant effects can occur. Firstly, the estimation may deviate in scaling because the image provides only a 2D projection surface. Consequently, the BlazePose model, described in the previous Chapter (2.2), might result in a variation in the overall scale, such that the person's body height is not accurately represented. Secondly, individuals generally differ in body stature, both in terms of height and width. While the ratios between width and height are not always the same, they tend to follow a specific scale. To counteract these effects, the estimations are normalised. This normalization is achieved by scaling the body to the hip width, meaning that the hip width is calculated using the following formula:

$$\kappa = \frac{1}{d(K_{23}, K_{24})}. \tag{3.29}$$

where K_{23} is the keypoint left hip and K_{24} the keypoint right hip. $d(K_{23}, K_{24})$ is calculated with

$$d(K_{23}, K_{24}) = \sqrt{(\mathrm{x2} - \mathrm{x1})^2 + (\mathrm{y2} - \mathrm{y1})^2 + (\mathrm{z2} - \mathrm{z1})^2}. \tag{3.30}$$

This scaling factor κ is then applied in subsequent operations as follows [64]:

$$\mathbf{x}_{scaled} = \begin{bmatrix} \kappa & 0 & 0 \\ 0 & \kappa & 0 \\ 0 & 0 & \kappa \end{bmatrix} \mathbf{x}.$$

As a result, the hip width is normalised to 1 at every instance, ensuring higher comparability between different subjects and across various videos. This approach effectively addresses the challenges of scale variation and body stature differences in pose estimation,

facilitating a more accurate representation and comparison of human poses.

As a second normalization step, a rotational adjustment is made in addition to scaling. This adjustment is necessary because the different cameras have varying rotations in the coordinate system. Additionally, deviations in the estimation itself, including both model errors and measurement errors, are addressed. To counteract these issues, a rigid transformation is employed, limited to rotation and translation, without altering the distances between points. This fully preserves the geometric integrity of the object. This method effectively minimises distortions that may occur in many estimations. Moreover, it is flexible in the coordinate axis, which enhances its effectiveness compared to a rotation constrained to a fixed axis.

Mathematically, a rigid transformation can be expressed using a rotation matrix $\mathbf{R}$ and a translation vector $\mathbf{t}$, which together act to transform an original point $\mathbf{p}$ into a new point $\mathbf{p}'$, following the formula $\mathbf{p}' = \mathbf{R}\mathbf{p} + \mathbf{t}$. The rotation matrix rotates the object around the three axes of the space while the translation vector shifts its position.

A practical example of estimating a rigid transformation is aligning two sets of points in different coordinate systems. First, it typically involves calculating the centroid of both sets of points. This is achieved by computing the mean of the start and target points along their axes, resulting in $\mathbf{A}_{\text{mean}}$ and $\mathbf{B}_{\text{mean}}$. The resulting centroids serve as reference points from which the relative movement of the point sets can be assessed.

To align the points correctly, the mean is subtracted from each point, leading to points centred around the origin. This operation aims to minimise the difference between the two sets of points. Subsequently, a matrix $\mathbf{H}$ is formed by the product of the subtracted points $\mathbf{A}_{\text{mean}}$ and the transpose of $\mathbf{B}_{\text{mean}}$. The SVD of $\mathbf{H}$ yields the matrices $\mathbf{U}$, $\mathbf{S}$, and $\mathbf{V}^T$, from which the rotation matrix $\mathbf{R}$ is derived as $\mathbf{R} = \mathbf{V}^T \cdot \mathbf{U}^T$.

A particular case occurs when the determinant of $\mathbf{R}$ is less than zero, indicating a reflection. In this case, a correction is made by reversing the sign of the last row of $\mathbf{V}^T$ and recalculating $\mathbf{R}$. Finally, the translation vector $\mathbf{t}$ is determined by rotating centroid $\mathbf{A}$ with $\mathbf{R}$ and then subtracting it from $\mathbf{B}$ to obtain the displacement [75].

The benefits of this method are manifold, consisting of allowing an efficient and robust estimation of the rigid transformation between two sets of points while completely preserving the geometric integrity of the transformed object. This technique is beneficial in applications where precision and the preservation of the physical properties of objects are

critical, such as in the alignment of robotic components or the registration of medical images. Preserving the distances and angles between points enables accurate manipulation of objects in 3D space without altering their fundamental structure.

3.4.4. Coordinate fusion

The presence of two cameras has a number of advantages. To gain an advantage from this setup, the data from both sources must be fused, a process known as data fusion, for which several options exist. Fusion can occur on three levels:

- Image level, where 2D data from each camera are determined and reconstructed into 3D data.
- Coordinate level, meaning that 3D coordinates are combined.
- Relative level, where relative sizes are directly combined.

At the first option (Image level), the 3D reconstruction, as described in the previous Section 3.3, is applied. All in all, 2D points are extracted from each camera and then fused using camera geometry, with the Singular Value Decomposition (SVD) method minimizing noise in the process. This modification alters the algorithm by first implementing fusion, followed by rotation and scaling. This modification is shown in Figure 3.6.

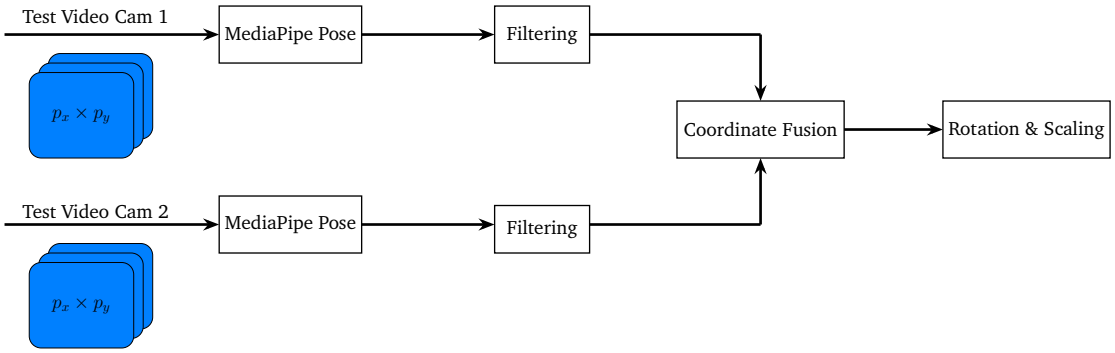


Figure 3.6.: This illustration provides a clear description of the process depicted, emphasising the reconstruction of 3D data from 2D inputs and the adjustment made to the algorithm's sequence to accommodate this process.

On the Coordinate level, three approaches are pursued: The first approach involves issuing a visibility score (vis)for each estimated point, allowing the fusion to be based on the

visibility of estimates, shown in the formula

$$\mathbf{x}_{\text{fusion}} = \mathbf{x}_{\text{cam1}} \cdot \left(\frac{\text{vis}_{\text{cam1}}}{\text{vis}_{\text{cam1}} + \text{vis}_{\text{cam2}}} \right) + \mathbf{x}_{\text{cam2}} \cdot \left(\frac{\text{vis}_{\text{cam2}}}{\text{vis}_{\text{cam1}} + \text{vis}_{\text{cam2}}} \right). \tag{3.31}$$

The second approach incorporates prior knowledge, meaning that the position of the subject relative to the camera is known through rotation. This means the Formula 3.31 is used again, except that the visibility is fixed. This makes it possible to determine which parts are likely to be clearly visible. To summarise, this means that when the camera is positioned directly in front of a person, both the left and right halves are estimated as equally significant. Nonetheless, when the person is sideways to the camera, the half of the body that is visible in the camera is valued at 80%, while the more obscured half is valued at 20%. The stated percentage values are based on a grid search using different weightings. The analysis was carried out on the interval $[60, 90]$ in 5 percentage point increments. Besides, these primary factors are adjusted depending on the angle at which the person faces the camera. Another approach involves combining the predetermined fixed fusion with the fusion based on visibility, where half the factor is applied in each case.

There is also the approach of selecting the coordinates associated to the highest visibility score, but this is not recommended. Potential deviations could lead to distortions in the skeletal model if points are selected from individual estimates rather than combined. Therefore, this method has been omitted. The level of calculating relative sizes is excluded from consideration because it no longer utilises the potential of coordinate data.

In summary, this means that 4 different fusion types are calculated and compared with each other in Section 4.1. Fusion based on:

- ...the visibility from MediaPipe (fusion vis).
- ...the orientation of the subject to the camera (fusion a prior).
- ...the mixing of the orientation of the subject to the camera and the visibility of MediaPipe (fusion mixed).
- ...the 3D reconstruction of two 2D HPEs (fusion rc)

In conclusion, integrating data from two cameras and applying these fusion techniques at different levels can achieve a more accurate and robust representation of human motion, enhancing the analysis within the used MoQuA algorithm.

3.4.5. Relative values

In the current context, the analysis of human movements is discussed based on coordinate information. Relative sizes are required to compare these with other coordinate systems, such as a MoCap suit. Furthermore, these relative sizes can be used to obtain additional features related to the coordinates themselves. These allow for a detailed examination of the relative sizes within human anatomy. A key component of this analysis is the angles within the body, defined by the relationship between two vectors. Three points in three-dimensional space are required to determine the angles between these vectors. Each point is either the start, middle or end point. This definition is necessary to get a consistent angle. The following formula represents the calculation of an angle based on such points [68]:

$$\theta = \cos^{-1}\left(\frac{(\mathbf{b}-\mathbf{a})\cdot(\mathbf{c}-\mathbf{a})}{\|\mathbf{b}-\mathbf{a}\|\|\mathbf{c}-\mathbf{a}\|}\right).$$

Here **a**, **b**, and **c** represent the positions of the three points in space, with θ being the calculated angle between them. This method allows to compute specific body angles, such as:

- Elbow angle: utilizing the keypoints of the elbow, hand, and shoulder.
- Knee angle: defined by the points hip, knee, and ankle.
- Hip angle: based on the points knee, hip, and shoulder.
- Side angle: determined through the elbow, arm, and hip.
- Head angle: established by the points shoulder, midpoint of the shoulders, and nose.

Above angles, hip and shoulder width are classic relative sizes that can be derived from coordinate information. These measurements are generally constant, with hip width typically exhibiting less variance than shoulder width due to the shoulders having a greater degree of freedom.

Additional relative sizes can be ascertained by calculating the distance between key and hip points, for example, distance between hip and wrist. This is achieved by applying the distance formula 3.30. Here, x_1, y_1, z_1 and x_2, y_2, z_2 represent the coordinates of the points under consideration. Through these analytical methods, a profound understanding of human motion is gained, which is essential for various applications from sports science to rehabilitation.

3.4.6. Limb and feature groups

In this section, the sorting of features into distinct limb groups as well as the different groups of used features described in the preceding section, are delineated. The 33 features are each allocated to a specific limb group. This allocation is undertaken both to enhance understanding and to minimize noise. The Table 3.1 displays the assignment of individual features to their respective limb groups. There are three variants defined for the features,

Landmark	Limb Group	Dimension
0-10	head	11
11, 12, 23, 24	torso	4
13, 15, 17, 19, 21	left arm	5
14, 16, 18, 20, 22	right arm	5
25, 27, 29, 31	left leg	4
26, 28, 30, 32	right leg	4

Table 3.1.: Assignment of the features from Figure 2.5 to the respective limb group [63].

each distinguished by a varying number of features. The first variant (f1) serves as the baseline. This set comprises 27 features sourced from six limb groups, with each axis contributing one feature. Additionally, it includes six primary angles that capture the spatial orientation of the limbs. Except for the head angle, there are corresponding features for both the left and right sides. The second variant (f2) builds upon the features included in f1 by adding relative widths and distances between the limbs, such as the hip-to-wrist and hip-to-ankle distances. This expansion increases the total feature count to 55. This group is designed to evaluate the utility of these relative measurements. The final variant (f3) is the most comprehensive feature set, combining the attributes of f2 and each of the 33 keypoints without sorting them into limb groups, resulting in a total of 163 features. This approach aims to encapsulate all possible information, making classification more challenging yet potentially leading to improved outcomes due to the increased data available. Furthermore, each feature can be individually compared against the others, facilitating a detailed analysis.

3.4.7. Summary

A comparison with the data from a MoCap suit is essential to evaluate the quality of all the preprocessing steps and the overall estimation. A direct comparison is challenging

because the MoCap suit uses a different skeletal model. This discrepancy means that the individual keypoints do not align, preventing a direct comparison.

Specific relative measures, such as the elbow and knee angles, are compared to circumvent this issue and still perform an evaluation. These measurements are chosen because they do not depend on the coordinate system, making them viable for comparison despite the differences in skeletal models. Additionally, the hip and shoulder width variance is examined and normalised to the hip width for cases where scaling has not been applied.

These evaluation methods are designed to assess the effectiveness of the estimation process. By comparing relative sizes independent of the coordinate system and analyzing the variance in key measurements, it is possible to gauge the accuracy and reliability of the motion analysis. These evaluation methods allow for an indirect but insightful evaluation of how well the preprocessing and estimation steps perform in capturing human motion, even when direct comparisons are not feasible due to model discrepancies.

3.5. Similarity measures for trajectories

The investigation and selection of suitable similarity measures for trajectories generated during the execution of sports exercises demand a careful analysis of each method's characteristics to identify the one that can best handle variable speeds and potential execution errors. Various similarity measures, including Euclidean distance, Hausdorff distance, Fréchet distance, DTW, Time Warp Edit Distance (TWED), and Longest Common Subsequence Distance (LCSS), offer different approaches to evaluating the similarity between trajectories. In the following, these similarity measures are briefly introduced and compared with each other. For simplicity, the explanations and definitions of the individual similarity measures (excluding DTW) are newly introduced and should be considered as only applicable within this section. Followed by a detailed description of DTW as a selected similarity measure.

The Euclidean distance is a fundamental metric measure that calculates the direct geometric distance between two points in space. Assuming there are two points $\mathbf{r}^n$ and $\mathbf{s}^n$ in n-dimensional space, then the Euclidean distance results in

$$ED(\mathbf{r}^n, \mathbf{s}^n) = \sqrt{\sum_{i=1}^{n} (\mathbf{r}_i - \mathbf{s}_i)^2}. \tag{3.32}$$

It directly compares points to each other and requires the trajectories to be the same length. This measure is unsuitable for comparing trajectories that are temporally shifted or have different speeds, as it does not account for local time shifts and is not robust to noise [76].

The Hausdorff distance broadens comparison capabilities by measuring the maximum distance between two sets of points (**A** and **B**), allowing for evaluating trajectories of differing lengths.

$$H(\mathbf{A}, \mathbf{B}) = \max \left\{ \sup_{\mathbf{a} \in \mathbf{A}} \inf_{\mathbf{b} \in \mathbf{B}} d(\mathbf{a}, \mathbf{b}), \sup_{\mathbf{b} \in \mathbf{B}} \inf_{\mathbf{a} \in \mathbf{A}} d(\mathbf{b}, \mathbf{a}) \right\}, \tag{3.33}$$

where **a** and**b**are points in **A** and **B** and $d(\mathbf{a}, \mathbf{b})$ is a selectable metric between two points. However, this approach also does not consider the order of points, making it less suitable for trajectories with variable speeds [77].

The Fréchet distance accounts for the arrangement of points along the trajectories by utilizing the metaphor of a person and a dog positioned at either end of the trajectories, having to move so that the ”leash” between them remains as short as possible. The Fréchet distance between two curves, **A** and **B**, is defined mathematically as follows:

$$FD(\mathbf{A}, \mathbf{B}) = \inf_{\alpha, \beta} \max_{t \in [0,1]} \|\mathbf{A}(\alpha(t)) - \mathbf{B}(\beta(t))\|, \tag{3.34}$$

where $t \in [0, 1]$ and α and β are continuous and strictly increasing functions mapping from the interval $[0, 1]$ to parameters describing the curves **A** and **B**. This approach is more intuitive for comparing trajectories that may change in speed and direction but are also not robust to noise [77].

DTW offers an advanced method for comparing trajectories by allowing the trajectories to stretch or compress over time to find an optimal match. This makes DTW particularly suitable for evaluating sports exercises, where the execution speed can vary without affecting the correctness of the execution. DTW is flexible enough to compare trajectories of different lengths and can adjust to account for local time shifts [76].

TWED and LCSS provide alternative approaches that entitle temporal distortions in different ways. The TWED between two time series **A** and **B** is determined by the formula $\min\{D(m, n)\}$, where **A** and **B** are the time series being compared. The function $D(m, n)$ calculates the minimal edit distance considering deletions and insertions, with penalty parameters λ for deletions or insertions and ν for mismatches. The special feature of TWED is that it can assign penalty points for deleting and inserting points to improve

the match [78]. The LCSS between two sequences **A** and **B** is expressed by the function $L(m, n)$, which determines the length of the longest common subsequence between the two sequences. Here, **A** and **B** are the sequences being compared, and m and n are their respective lengths. The special feature of LCSS is that it allows skipping elements, offering some robustness against outliers [79].

Considering the specific requirements in evaluating sports exercises, particularly the tolerance for variable speeds in execution, DTW emerges as the superior similarity measure for different reasons. DTW's ability to adjust trajectories by stretching or compressing them over time to achieve an optimal match makes it especially suitable for analyzing movements where speed tells little about the correctness of execution. This flexibility, combined with the capability to handle different lengths and account for local time shifts, provides a decisive advantage over other measures that either require a direct point-to-point match or do not adequately consider the order of points. Hence, DTW is the preferred choice for assessing the correct execution of sports exercises, considering the inherent variability in execution speed [80]. Considering these attributes, the DTW method proves to be a suitable measure for extracting meaningful features from trajectory data. These features are necessary to train classifiers, especially in a context where only a small dataset is available. Therefore, the following explains the precise functionality of DTW.

3.5.1. Dynamic Time Warping

The following description of DTW is based on [81]. Suppose there are the time series $\mathbf{Q} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_M]$ and $\mathbf{C} = [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_N]$, where, in the multidimensional case, for example, the points consist of $\mathbf{x}_i = [x_{i,1}, x_{i,2}, \ldots, x_{i,K}]$ and $\mathbf{y}_i = [y_{i,1}, y_{i,2}, \ldots, y_{i,K}]$. Here, **K** describes the dimension, which is the same for both time series. M describes the length of the first time series and N the length of the second time series. The goal is to find the matching point from the other time series for each point so that the distance is minimised. This process is known as warping, clearly indicating that the algorithm is non-linear. To create this warping, a path is chosen for which the following constraints apply:

1. Monotonicity: The mapping must be monotonically increasing to preserve the temporal order of points.
2. Continuity: Each point in a sequence must be assigned to at least one point in the other sequence.

3. Boundary conditions: The first point of a sequence must be assigned to the first point of the other sequence and similarly for the last point.

This means a path is formed that links the two signals. The path is defined by $\pi = \{(\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_p, \mathbf{y}_q), \ldots, (\mathbf{x}_M, \mathbf{y}_N)\}$ with f_{l1} where applies $max(M, N) \leq f_{l1} \leq (M + N–1)$ and $f_{l1} \in \mathbb{N}$. The length of the path is determined by the minimised cumulative total distance. Thus, DTW can be described by:

$$DTW(Q, C) = \min_{\pi \in A(x,y)} \sqrt{\sum_{(i,j)\in\pi} d_{ij}^2}. \tag{3.35}$$

To create an $m \times n$ matrix to determine the path, considering the entire space, all possible paths **A** need to be calculated. The size of **A**, i.e., the number of possible paths from one corner of the matrix to the opposite corner, is defined by the Delannoy number. The sum of the individual distances gives the total distance. In the minimization process, it is determined which of the following steps are chosen: insertion, deletion, match (these steps need further elaboration).

The determination of the path can be found through dynamic programming:

$$D(i, j) = D(i, j) + \min\{\underbrace{D(i-1, j)}_{Insertion}, \underbrace{D(i, j-1)}_{Deletion}, \underbrace{D(i-1, j-1)}_{Match}\}.$$

The termination condition of the recursive formula for calculating the edit distance occurs when one of the sequence indices i or j becomes zero, indicating that one of the sequences has been completely traversed. In this case, the distance directly corresponds to the number of remaining characters in the other sequence, which need to be either inserted or deleted to achieve equality. These three operations enable the adaptation of two time series to each other by determining how the elements of one sequence are matched to the elements of another sequence. In the following, a detailed description of the steps is presented.

1. Insertion: This operation adds an extra point into the first time series to achieve a better match with a point in the second time series. Insertion allows bridging differences in the time series caused by temporary stretches or expansions in one of the sequences. In distance calculation, an insertion contributes to a point in one sequence matching multiple points in the other sequence.

2. Deletion: Conversely, deletion allows for the removal of a point from the first time series, thereby achieving a better fit with the second time series. This is useful for minimizing the effects of contractions or brief events in the time series. Similar to insertion, deletion helps to align the sequences despite local differences in duration or the number of events.

3. Match: A match occurs when a point in the first time series is directly matched to a point in the second time series without requiring an insertion or deletion. This means that both points are considered equivalent based on a predefined similarity or distance measure. Matches are ideal as they directly contribute to the alignment of the two sequences without increasing complexity through additional operations.

The distance is adapted from Equation 3.32:

$$d(i,j) = \sum_{k=1}^{K} (\mathbf{x}_{i,k} - \mathbf{y}_{j,k})^2.$$

The recursive calculation using these three operations enables determining the cost (or distance) of matching for each pair of elements in the two time series. The total cost of a path through the matrix of distance values is minimised by choosing the optimal combination of insertions, deletions, and matches for each step. This leads to determining the optimal path, which exhibits the least cumulative distance (or the highest similarity) between the two time series. Dynamic programming is employed to enhance the efficiency of the calculation by storing and reusing already computed partial solutions. Calculating the Delannoy numbers helps to understand the complexity of pathfinding in an $m \times n$ matrix and provides a mathematical basis for analyzing algorithms that find such paths, such as DTW. By using this number in the context of pathfinding, the efficiency of algorithms for finding paths in temporal sequences or spatial grids can be better assessed and optimised.

Figure 3.7 illustrates the application of the DTW algorithm. Initially, DTW enables a non-linear alignment between time series. This means it can identify patterns that have been stretched or compressed in their temporal sequence, which is particularly valuable when dealing with datasets that undergo the same phases but not within the same time frame.

Another significant advantage, which is important for the MoQuA algorithm discussed in this thesis, is the robustness of DTW against changes in speed. As the example in Figure 3.7 demonstrates, DTW can recognise similar patterns in datasets, even if they are

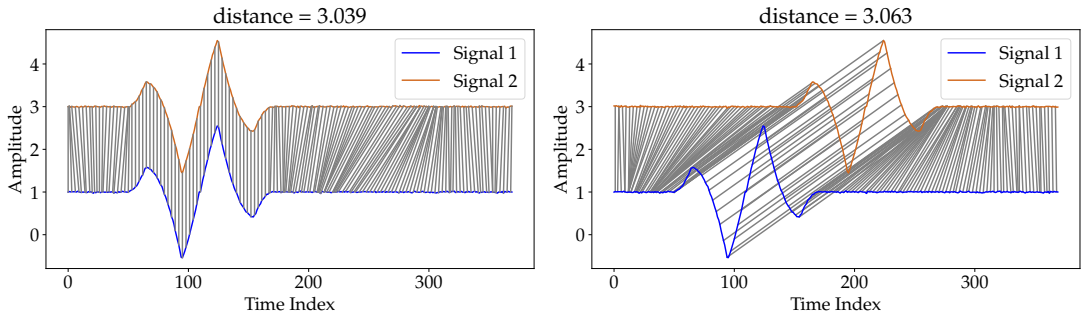


Figure 3.7.: An illustration of the advantage of DTW similarity: On the left is a synchronized signal. The amplitude is shifted for display purposes. The gray lines show the connections of the individual data points. The dummy movement is shifted on the right. The distance of the two variants differs only slightly, which shows the elimination of the temporal shift.

traversed at different speeds. This is crucial in speech recognition, for instance, where the speaking rate can vary greatly among different people.

DTW is known for its enhanced performance in pattern recognition compared to other techniques that rely on simple point-to-point distance measurement. The algorithm finds the optimal match between sequences, leading to more accurate recognition. Furthermore, DTW is highly versatile. It is not limited to acoustic signals but can be applied to a wide range of sequential data, such as financial time series, biomedical signals, or motion data. Lastly, DTW can be customised to the specific needs of a dataset by adjusting the cost function and other parameters, further improving pattern recognition. The minor differences in the distance indicated in the diagrams highlight how DTW handles subtle differences in the temporal extension of sequences to allow for precise matching between data points, thereby quantifying the similarity between sequences.

3.5.2. Summary

DTW is an effective method for measuring the similarity between two time-sequential datasets that may operate at different speeds. Therefore, it is chosen to calculate the features between the different time series. In the classification of exercise routines, DTW enables the comparison of time series data from movement patterns, even when the execution speeds vary, because it can adjust the temporal sequence to find an optimal match.

This flexibility makes DTW particularly suitable for the analysis of sports movements, where the consistency of the movement is more important than the speed at which it is performed.

3.6. Classifiers

The last Chapter (3.5) describes how individual time series were transformed into distance features by applying the DTW method. To make statements based on these features, various classifiers are introduced, considering both simple classifiers and those with higher model complexity. The aim is to let the different classifiers compete against each other on the features.

3.6.1. Linear classification

Linear classification is a basic ML technique that offers a straightforward yet practical approach for distinguishing between two or more classes. At its core, linear classification seeks to separate classes using a decision boundary that is linear in nature. This simplicity facilitates a clear understanding of the model's decision-making process, enhancing interpretability.

The essence of linear classification can be encapsulated in the formulation of a linear equation or hyperplane, defined as $\mathbf{w}^T\mathbf{x} + \mathbf{b} = 0$, where $\mathbf{w}$ represents the weight vector, $\mathbf{x}$ is the feature vector, and $\mathbf{b}$ is the bias. The goal is to learn $\mathbf{w}$ and $\mathbf{b}$ such that the hyperplane optimally separates the classes in the feature space. For binary classification tasks, this method assigns labels based on the side of the hyperplane on which the data points reside, making it a model of high utility for problems with clear linear separability [82].

Linear classifiers, including Logistic Regression and Linear Support Vector Machines, are renowned for their computational efficiency and ease of implementation. These characteristics make linear classification models particularly appealing for tasks with large datasets and in scenarios where a rapid model deployment is crucial. Moreover, the linear approach serves as a foundation upon which more complex nonlinear models can be constructed, often through kernel methods or polynomial feature expansion, thereby extending their applicability to more intricate datasets.

However, the simplicity of linear classifiers comes with its constraints. The primary limitation lies in their inherent assumption of linear separability among classes, which is only sometimes present in real-world data. Complex datasets with highly intertwined classes may elude the grasp of linear classification, necessitating alternative approaches that can capture the nonlinear relationships within the data.

Despite these limitations, the value of linear classification in the ML toolkit is undeniable, and is therefore a good choice for meaningful characteristics, which this paper attempts to achieve. Its straightforward interpretability and computational efficiency render it an essential first line of analysis for many classification problems. Whether used on its own for linearly separable datasets or as a stepping stone towards more complex models, linear classification continues to play a key role in developing ML solutions across diverse domains, from text categorization and image recognition to medical diagnosis and beyond [82].

3.6.2. Support Vector Machine

The Support Vector Machine (SVM) technique is a classification method used to differentiate between two or more classes. Unlike what the name might suggest, Support Vector Machines are not physical machines but purely mathematical procedures. The basic functioning and potential applications of SVMs are outlined in [83], explaining that SVMs aim to establish a multidimensional, typically non-linear, decision boundary between classes. This separating boundary is known as a hyperplane. The data points closest to this hyperplane are termed Support Vectors, representing a subset of the distinctive input data. The margin MM is defined as the distance between the hyperplane and the nearest point. Ideally, this distance should be maximised to ensure the most distinct separation between classes. Once a hyperplane is identified, it results in a simple decision function, often represented by a step function. The dimensionality of the space is increased to accommodate the number of features. Furthermore, dimensionality is increased until a linear hyperplane can separate the generally non-linear data. This process may result in a significant increase in dimensionality, subsequently increasing computational demands. The ”Kernel Trick” offers a solution by providing a function that behaves similarly to a dot product, eliminating the need to know the high-dimensional space to compute the required dot product. The hyperplane and the Support Vectors are determined through optimization techniques based on the training data. Unlike neural networks, SVMs are robust against overfitting, though, like neural networks, they only predict the most likely class, not the probability of its occurrence. However, this limitation can be mitigated

by employing a Bayesian filter to estimate the probability, thus compensating for this disadvantage [84].

3.6.3. Decision Tree

Decision trees [85] are a fundamental tool in ML theory, serving as the foundation for numerous classification and regression methods. They model decision-making processes as a series of branches, each representing a decision based on an attribute. These tree structures are intuitively understandable, functioning similarly to a series of ”If-Then” rules.

In evaluating human movements, especially concerning exercises and training, decision trees can be employed to classify the quality of movement execution. Here, each node in the tree serves as a decision point for specific movement characteristics.

The basics of a decision tree can be described as follows:

- Root Node: This is the starting point of the tree, representing the entire population or dataset being analyzed. From this point, subgroups are created based on the most distinguishing attributes.
- Branches: Each branch corresponds to a decision made based on an attribute, such as the height of a jump or the consistency of stride frequency in a running exercise.
- Internal Nodes: These nodes represent points where an attribute check occurs, further dividing the datasets.
- Leaves: The endpoints of the tree, making a final decision, in this context, whether a movement was executed correctly or incorrectly.

In analyzing exercise movements, which is the basis of this work , a decision tree might first distinguish the type of exercise at an internal node. Subsequent nodes could then assess the technique, speed, and consistency of the execution. Ultimately, the tree leaves would classify the movement as correct, needing improvement, or incorrect.

The selection of attributes and decisions at each node is based on statistical methods to find the best division. Techniques such as maximizing the Information Gain Ratio, Gini impurity, or the mean squared deviation in regression problems are employed. A significant advantage of decision trees lies in their interpretability. They can be easily visualised, facilitating the comprehensibility of decisions. This is particularly useful in medical diagnostics or sports science, where understanding the decision-making process

is almost as important as the decision itself. Methods like tree pruning can be applied to prevent overfitting, where unnecessary branches are removed to simplify the model. Moreover, performance can be enhanced through ensemble methods like Random Forests, where the predictions of many decision trees are combined. In practice, applying decision trees for exercise evaluation would begin with collecting data, such as video footage or sensor recordings of individual training. After processing and extracting relevant features, these data would be used to train the decision tree, enabling it to make evaluations and provide feedback independently [85].

3.6.4. Random Forest

Random Forest is a robust ensemble learning algorithm that leverages the strengths of multiple decision trees to create a robust model for classification and regression tasks. By constructing several trees based on random samples of the training data (bootstrap aggregation or bagging) and random subsets of features at each split, Random Forest achieves high accuracy and control over overfitting. Each tree in the forest makes an independent prediction, and the final decision is made through voting (in classification) or averaging (in regression) across all trees [86].

In addition to the advantages already mentioned, Random Forest possesses several notable strengths. For instance, it can determine the importance of features for prediction, which is valuable for the model's interpretability and understanding of the underlying data structures. This capability renders Random Forest a predictive tool and an exploratory analysis instrument.

Another advantage is its flexibility: Random Forest can handle missing data by using the internal structures of the forest to impute missing values or minimise their impact. This reduces the need for extensive data preprocessing.

However, like all models, Random Forest has its limitations. Beyond the potentially time-consuming training process, the model's complexity can complicate interpretation. While decision trees are straightforward to understand, amalgamating many trees in a Random Forest model can make it challenging to discern the specific reasons behind a prediction.

Moreover, the model size, especially with a large number of trees and features, can lead to significant storage requirements. This may limit the use of Random Forest in resource-constrained environments or applications requiring fast prediction.

Despite these challenges, Random Forest remains a popular choice for many ML problems, thanks to its high accuracy, robustness against overfitting, and ability to handle complex data structures. Its versatility makes it a valuable tool for various applications, from disease prediction in healthcare to customer classification in marketing [87].

3.6.5. Artificial Neural Network

Artificial Neural Networks (ANNs) are computational models inspired by the human brain's structure and function, designed to recognize patterns and make decisions with a level of complexity that traditional algorithms struggle to match. They consist of interconnected processing units, or neurons, which work in parallel to solve specific problems through learning. ANNs are capable of learning from data, making them highly effective for tasks such as image recognition, natural language processing, and predictive modeling. One of the main advantages of ANNs is their ability to handle and make predictions from large and complex datasets, adapting their structure as they learn from more data. However, a significant disadvantage is their "black box" nature, which makes it difficult to understand how decisions are made, complicating the process of debugging and verification. Additionally, training ANNs can be computationally expensive and time-consuming, requiring substantial hardware resources and energy consumption. Despite these drawbacks, the versatility and efficiency of ANNs in processing and analyzing vast amounts of data make them indispensable in advancing fields like artificial intelligence and machine learning [88].

3.6.6. Gradient Boosting

Gradient Boosting is a machine learning technique that builds models in a stage-wise fashion, where new models are created to correct the errors made by existing models [89]. It works by combining multiple weak learning models, typically decision trees, to create a strong predictive model. Each tree is trained on the residual errors of the previous ones, gradually improving the model's accuracy. A key advantage of Gradient Boosting is its effectiveness in handling various types of data, including binary, categorical, and continuous outcomes, making it highly versatile for a range of predictive tasks. Additionally, it often provides high predictive accuracy that can outperform other models. However, a major drawback is its tendency to overfit the data, especially with noisy datasets or when too many trees are used without proper regularization techniques. Gradient Boosting also requires careful tuning of its parameters, such as the number of trees and learning

rate, which can be time-consuming and requires a significant amount of trial and error to achieve optimal performance [90].

3.7. Explainable Artificial Intelligence

Upon the classification of each instance of the sports exercise, the initial task is to ascertain which body part is accountable for the erroneous execution. Additionally, the determination of which features are significantly informative will be made. Consequently, an assessment of the informativeness of a feature and its value in the classification process will be conducted. Subsequently, the principles of XAI and two prevalent methodologies, LIME and SHAP, will be discussed.

XAI aims to make the outcomes of AI models understandable by humans. This field is gaining importance as AI systems become more complex and widespread, necessitating decision-making transparency, especially in critical healthcare, finance, and law applications. Within the scope of XAI, various ML models, including linear classification, Support Vector Machines (SVM), decision trees, and Random Forests, offer varying degrees of interpretability [91].

Linear Classification stands out for its simplicity and interpretability. It works by establishing a linear decision boundary that separates classes. The model's decisions are based on the relationship between input features and the target variable, which is directly interpretable. For example, in logistic regression, a type of linear classifier, the coefficients of input features indicate the importance and direction of the relationship with the target variable, offering clear insights into the model's decision-making process. Support Vector Machines (SVM) offer a slightly more complex scenario. SVMs work by finding the hyperplane that best separates the classes in the feature space. While maximizing the margin between classes is straightforward, using kernel functions to enable nonlinear classification can obscure interpretability. However, the support vectors data points that lie closest to the decision boundary provide direct insight into the decision process, as they are the critical elements defining the boundary. Decision Trees are inherently interpretable by design, as they mimic human decision-making processes through a series of binary choices—essentially, a flowchart of decisions. This model's structure allows for easy visualization and understanding of how input features affect the outcome, with each node offering a clear, rule-based decision point. The path from the root to a leaf can be directly translated into a set of conditions leading to a particular decision, making decision trees an excellent tool for XAI. Random Forests, while building on the interpretability of

individual decision trees, introduce complexity by aggregating the outcomes of numerous trees. This ensemble method improves prediction accuracy and robustness but at the cost of direct interpretability. However, Random Forests contribute to XAI through feature importance measures, which quantify the contribution of each feature to the prediction accuracy across all trees in the forest. This offers a macro-level insight into the model's decision-making process, although it needs the decision path clarity found in a single decision tree [92].

In conclusion, the quest for explainability in AI necessitates a balance between model complexity and interpretability. The straightforward, rule-based insights provided by linear classifiers and decision trees are valued for their high interpretability, although they may fall short when dealing with complex, non-linear relationships. On the further hand, models like SVMs and Random Forests, which are capable of capturing more complex patterns, pose challenges to interpretability. It has been recognised that tools and techniques within the XAI framework, such as LIME and SHAP, must be employed to shed light on the decision-making processes of these intricate models. The continuous evolution of AI underscores the importance of research in XAI methods that aim to enhance the transparency and understandability of models, ensuring that the field progresses with a commitment to clarity and accountability .

3.7.1. Local Interpretable Model-agnostic Explanations

LIME [93] is an innovative technique aimed at breaking through the black-box nature of complex ML models by providing locally interpretable explanations for their predictions. This method is particularly valuable as it enhances the explainability and transparency of ML models, which is crucial for the acceptance and trust in AI. LIME generates explanations by creating a new, human-understandable representation of the data, which might include the presence or absence of certain words in a text or specific areas in an image. It then constructs a local approximation of the complex model around the prediction using a simple model, often a linear model. This simple model is trained to closely approximate the predictions of the original, complex model near the instance being explained. LIME selects features that are relevant to the prediction and weights them to show how they influence the prediction. The refined explanation for a data point x involves the model g that minimises the locality-aware loss $L(f, g, \Pi_x)$. This loss quantifies the inaccuracy of g in approximating the model to be explained, f, within its vicinity, Π_x, while simultaneously maintaining a low model complexity. LIME seeks to identify an interpretable model g that closely replicates the behavior of the complex model f near x. The foundational

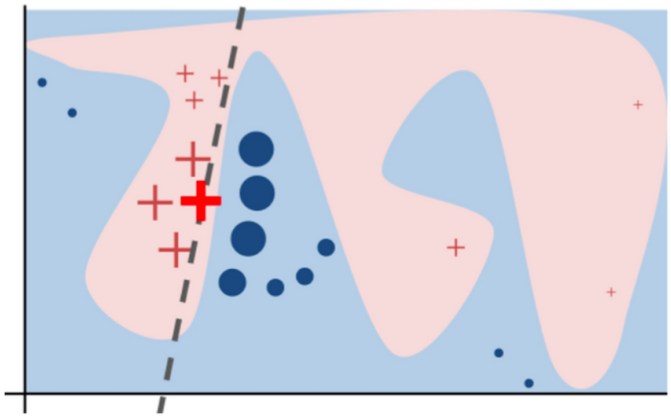


Figure 3.8.: Visualizing LIME: This illustration simplifies the complex decision function of a black-box model into a local, linear approximation, centered on the significant instance denoted by the bold red cross from [93].

mathematical expression for LIME is presented as:

$$\min_{g \in G} L(f, g, \Pi_x) + \lambda \cdot \text{Complexity}(g)$$

where f represents the complex model to be explained (for example, a Deep Learning model), g is the interpretable model providing the explanation (often a simpler model such as linear regression), $L(f, g, \Pi_x)$ denotes the locality-aware loss, measuring the accuracy of g in approximating f in the vicinity of x, Π_x is the weighting function that defines the ”proximity” or relevance of points around x, λ is a regularization parameter that controls the balance between the fidelity of approximation and the complexity of model g, Complexity(g) measures the complexity of the explanatory model g, ensuring interpretability. This formulation highlights LIME's core objective: to create a locally valid and interpretable model that elucidates the predictions of a complex model in the vicinity of a specific data point, balancing the trade-off between approximation accuracy and model simplicity for effective explanation. This basic idea is presented in Figure 3.8 The benefits of LIME are manifold. First, it allows for better traceability of predictions from complex models, which can be crucial for professionals in various fields, from medicine to finance. Second, it fosters trust in ML models by demonstrating the features on which predictions are based. Third, LIME can help uncover biases and injustices in model predictions by showing which features are overvalued or undervalued. Fourth, it is model-agnostic, meaning it can work with any classifier without understanding its internal workings.

However, LIME also has drawbacks. One of the main criticisms is the locality of its explanations. Since LIME provides local explanations around the prediction of a single instance, these explanations may only be generalizable to some of the models or other instances. This can lead to inconsistencies in explanations when analyzing different instances. Another drawback is the complexity associated with interpreting explanations generated by LIME, especially if the identified relevant features could be more intuitive to the end-user. Moreover, the effectiveness of LIME can depend on the choice of human-understandable data representation and the quality of the simple, local model, posing additional challenges in implementation.

In summary, LIME offers a valuable approach to improving the transparency and comprehensibility of complex AI models by providing locally interpretable explanations. Despite its limitations, such as limited generalizability and potential interpretation challenges, it represents a significant step towards responsible AI by laying the groundwork for understanding and verifying AI decisions [93].

3.7.2. SHapley Additive exPlanations (SHAP)

SHAP [94] is an innovative interpretation method for ML models based on cooperative game theory principles. It utilises the Shapley value, a method from game theory, to fairly and accurately assess the contribution of each individual feature to a model's prediction. SHAP aims to quantitatively capture the impact of features on the predictions of a model, thus providing highly detailed insight into the "why" behind a model's decisions. This method is model-agnostic, meaning it can be applied across a broad spectrum of ML models, from simple linear models to complex deep learning structures. Mathematically, the contribution of a feature i to the prediction of a model f for a specific instance x is expressed by the Shapley value $\phi_i(f, x)$ of that feature is calculated as follows:

$$\phi_i(f,x) = \sum_{\mathbf{S}\subseteq\mathbf{N}\setminus\{i\}} \frac{|\mathbf{S}|!(|\mathbf{N}| - |\mathbf{S}| - 1)!}{|\mathbf{N}|!} \left(f_x(\mathbf{S}\cup\{\mathbf{N}_i\}) - f_x(\mathbf{S})\right),$$

where $\mathbf{N}$ is the set of all features, $\mathbf{S}$ is a subset of $\mathbf{N}$ excluding the feature i, $f_x(\mathbf{S})$ is the prediction value of the model f, considering only the features in $\mathbf{S}$ (in addition to a base value), $f_x(\mathbf{S}\cup\{\mathbf{N}_i\})$ is the prediction value of the model f, considering the features in $\mathbf{S}$ along with the feature i, $\phi_i(f, x)$ is the SHAP value for the feature i, indicating the average marginal contribution of feature i to the difference between the model's prediction and the base value. In the context of sets, the absolute value line, or the so-called "cardinality",

denotes the number of elements in a set. The calculation of SHAP values is based on the principle that the fair distribution of the ”gain” among all features should occur, so that the contribution of each feature is measured based on its effects in all possible combinations of features. This method ensures a consistent and fair allocation of contributions based on the logic of marginal increment.

A key advantage of SHAP is enhancing the transparency and interpretability of complex models, which is significant in many fields, especially in critical ones such as medicine and finance. By quantifying the influence of individual features on model predictions, SHAP helps to strengthen trust in ML decisions. Moreover, the fair allocation of contributions by considering all possible combinations of features ensures a balanced assessment of feature importance. This leads to a consistent and accurate representation of the influences of features on predictions across different models. Nevertheless, the application of SHAP also comes with challenges. The most significant drawback is the high computational cost, especially for models with many features and data points, which may limit its application in real-time systems or on very large datasets. Additionally, the amount and complexity of information generated by SHAP can be overwhelming for end users without specific technical background, complicating interpretation. The accuracy and usefulness of SHAP explanations can also depend on how data was prepared for the model, potentially leading to misinterpretations.

Despite these challenges, SHAP represents a significant advancement towards responsible and interpretable Artificial Intelligence. By providing detailed insights into the decision-making processes of models, SHAP promotes understanding and trust in ML systems. The ability to quantify and explain the contribution of individual features to the predictions of a model is an invaluable asset in the world of Artificial Intelligence, laying the foundation for more transparent and fair decisions [94].

3.8. Summary

In summary, the individual steps of the MoQuA algorithm have been detailed, and the entire algorithm pipeline has been explained. The resulting attributes show that the MoQuA algorithm fulfils the expectations for the implementation of assessments in the field of physiotherapy. A particular focus is on the use of cost-effective cameras and the use of a small dataset. Based on the described development, the evaluation and analysis of the algorithm now follow.

4. Evaluation and Discussion

This chapter contains an evaluation of the developed MoQuA algorithm and its individual components, looking closely at various aspects such as exercises, subjects, orientations, and differences in camera perspective. The term „subject“is used in the following to refer to the patient more specifically the person performing the sports exercises. The chapter is focused on three main parts: The evaluation of the pose estimation and the improvements achieved through preprocessing is the first part. The second part focuses on the evaluation of the classification processes. The final part is about the identification of relevant limb groups. Here, the assessment of the significance of individual features is of particular interest to determine their relative importance and contribution to the overall performance of the algorithm. In addition, the limb that is likely to cause the error in the performance of the exercise is determined. Through this strategy, both the strengths and weaknesses of the comprehensive MoQuA algorithm are illustrated in detail. The following analysis provides a keen insight into the effectiveness and efficiency of the analysed MoQuA algorithm and should contribute significantly to the understanding of its applicability and performance in different contexts.

4.1. Human pose estimation

The following section analyzes and evaluates the accuracy of pose estimation. Data from a MoCap suit is used as a benchmark. The two estimation approaches, BlazePose Model from MediaPipe and MoCap suit cannot be directly compared because they differ in the skeletal model. Therefore, the comparison is made through relative sizes to eliminate model and coordinate system discrepancies.

The comparison between MediaPipe and the MoCap suit is based on six metrics, described in Section 3.4.5: the widths of the hips and shoulders, which serve as comparison parameters due to their anatomical constancy, and four angles (left and right for both knees and

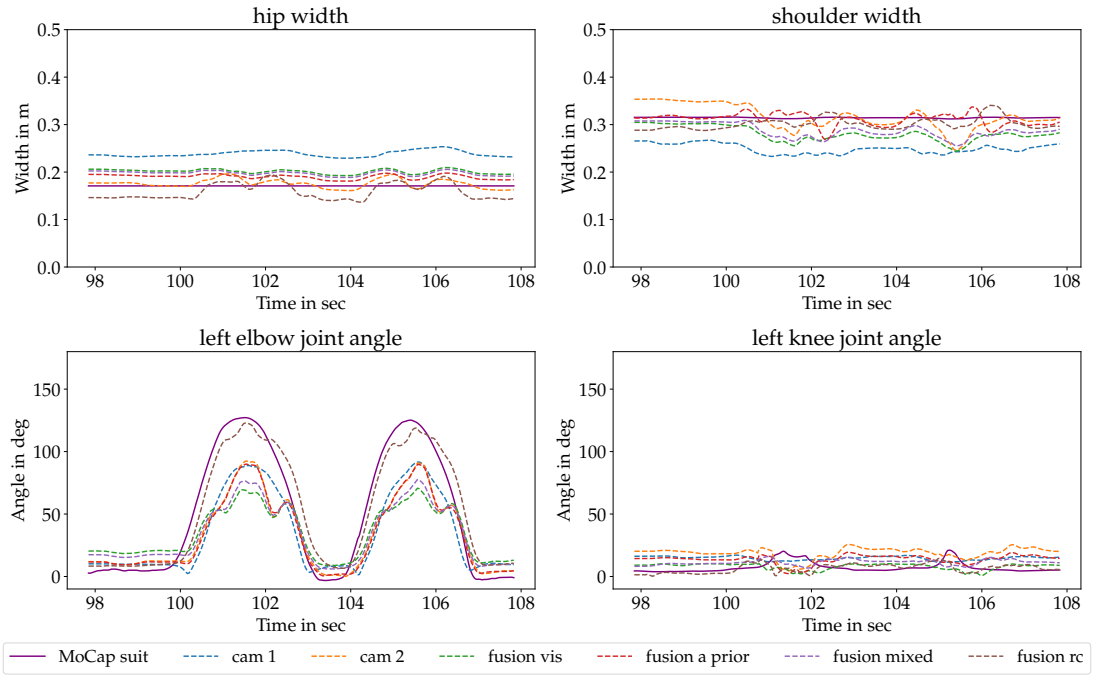


Figure 4.1.: A plot demonstrating the execution of two consecutive push-ups. Displayed are the hip and shoulder width, as well as the angles of the left elbows and knees. These values are presented for the various types of HPE.

elbows angle) that capture the relative positions of the limbs to each other. The constancy of hip width is assumed due to anatomical stability even during movements and the lower constancy of shoulder width, despite possible minimal movement variations.

Figure 4.1 exemplifies a push-up exercise, where the double execution is displayed by two pronounced wave peaks at the elbow while the legs remain extended and the knee angle is almost constant. The analysis of the shown data indicates that the MoCap suit shows little variance for hip and shoulder width, pointing to high accuracy. In general, the lower the variance, the higher the accuracy of the HPE. In contrast, the camera-based estimations show significant deviations, although the relative size fluctuation is similar, and synchronization makes it possible to overlay the movements. This means that 3D reconstructions can provide a good measure of relative sizes.

In the realm of angular movements, it is shown that exercises are never performed perfectly,

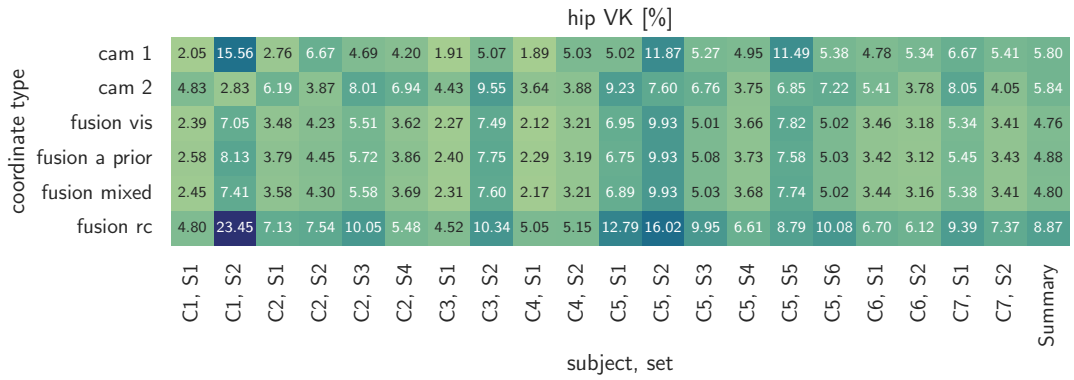


Figure 4.2.: A table presenting the coefficient of variation of the hip width in percentages for the combinations of subject, set (CX, SX) and various HPE methods. The value of 0.017% of the MoCap suit is a basis for comparison.

as indicated by slight peaks, even in MoCap recordings. This implies that no exercise is done flawlessly. The evaluation of these exercises includes the preprocessing steps carried out, but it excludes normalization and scaling based on hip width from the analysis. These specific preprocessing activities are not taken into account in this phase of evaluation, because then the hip width is the same for all subjects at all times.

The application of a low-pass filter and a moving average was intended to smooth the signal without significant signal delay. A moving average with a window size of $n = 10$ proved to be superior to the low-pass filter, which fell below the Nyquist limit, half of the recording frequency of $30\,\text{Hz}$. Therefore, the smoothing of the estimate was successful. Figure 4.1 shows just an example of a single exercise, but it is necessary to make a more comprehensive analysis across all exercises in order to get a generally valid statement. Therefore, comparability is established by differentiating according to subject, set and data type. For hip and shoulder width, the coefficient of variation is given in percent, formalized by the equation [95]:

$$VK = \frac{\sigma}{\mu} * 100\%.$$

The coefficient of variation has the advantage of normalization to the mean, thereby eliminating scaling factors and thus enabling comparability of values. The coefficient of variation of the MoCap suit serves as a reference value, achieving 0.017%, indicating

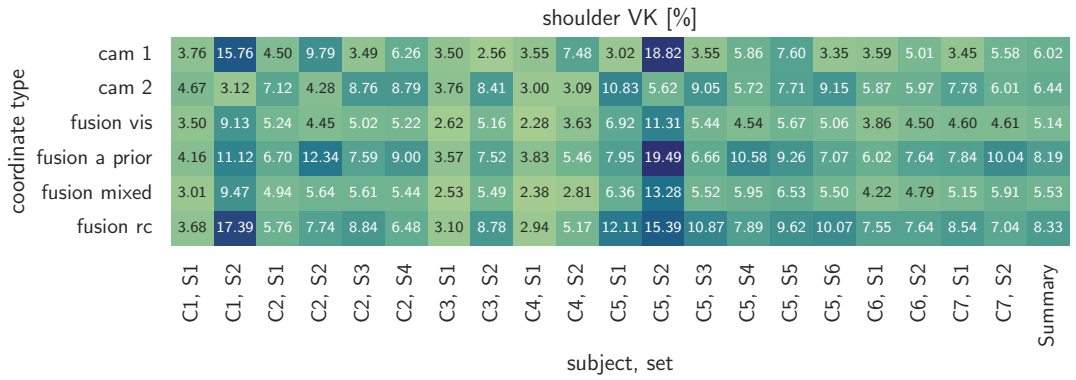


Figure 4.3.: A table presenting the coefficient of variation of the shoulder width in percentages for the combinations of subject, set (CX, SX) and various HPE methods. The value of 3.1% of the MoCap suit is a basis for comparison.

theoretically high accuracy of estimation. For shoulder width, a coefficient of variation of approximately 3.1% is reported, which was already expected due to the anatomical mobility of the shoulder.

Figures 4.2 and 4.3 display the coefficients of variation of the pose estimation for various scenarios, described in 3.4.4. It is shown that hip and shoulder widths yield similar results, indicating a consistent performance of the algorithm. Generally, values for shoulder width are higher, because of their anatomical condition.

The analysis of all subjects and sets shows that Camera 1 and Camera 2 deliver similar results with a coefficient of variation of about 5.8%. This can be explained by the balancing of good and bad estimates caused by the different orientations of the subjects. It was found that fusion based on visibility achieves an improvement in estimation by $18,49\%$, highlighting the effectiveness of this method. Fusion based on a priori knowledge or a combination of a priori knowledge and mix also shows improvements of around 18%, although they are not quite as effective as pure visibility fusion.

The 3D construction through both cameras shows a more inferior result with 8.87%, attributable to its dependence on both the pose estimation and the 2D data estimation. The estimation for rotation and translation (R and T) is often only valid for a specific image area and is difficult to determine accurately, as described in Section 3.3.3.

Differences between the individual sets, i.e., combinations of subject and set, are clearly

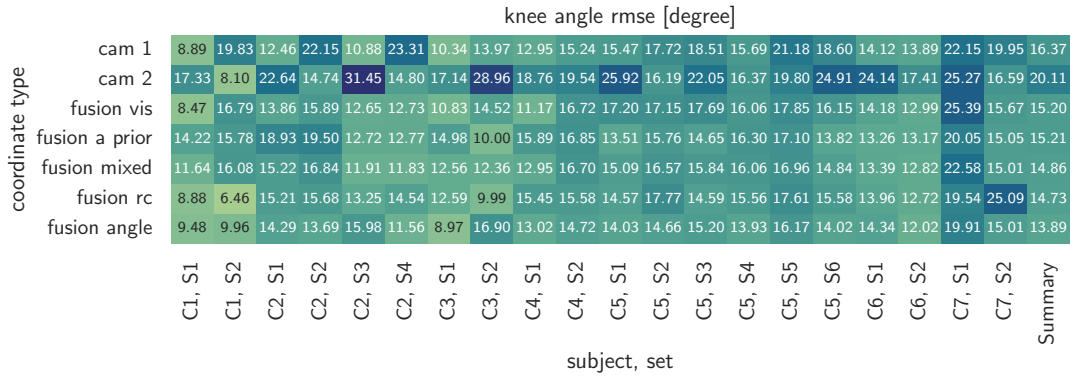


Figure 4.4.: A table presenting the RMSE of the knee angle for the combinations of subject, set (CX, SX) and various HPE methods.

observable. Unsatisfactory results, due to their performance values, are particularly apparent in C1, S2 and C5, S2. In contrast, C4, S1 and C1, S1 stand as examples of particularly good results. Differences between subjects, such as C1 and C5, could be traceable to factors like clothing and general body structure affecting the estimation accuracy of MediaPipe.

Following the analysis of the coordinates, the analysis of relative positions is now described by examining the angle ratios. The Root Mean Square Error (RMSE) serves as the benchmark for evaluation, with its results presented in Figures 4.4 and 4.4. An initial finding indicates that the knee angle consistently achieves better results than the elbow angle, suggesting that the position of the knee angle can generally be estimated with greater accuracy than that of the elbow. This might partially be attributed to the increased difficulty in localizing the position of the hand.

Camera 2 delivers slightly worse results than Camera 1, likely due to the limited visibility of the angle from Camera 2's perspective. Furthermore, it is shown that fusion based on visibility allows for an improvement of about 25% over using just one camera for estimating the elbow angle, although only a minor improvement is achieved for this angle.

Compared to relative coordinates, both the a priori and the mix fusion show improved results, suggesting that the relative positions are captured more accurately by these methods. Notably, the fusion based on reconstruction delivers even better values, highlighting the potential benefit of reconstruction since the relative positions are depicted more accurately,

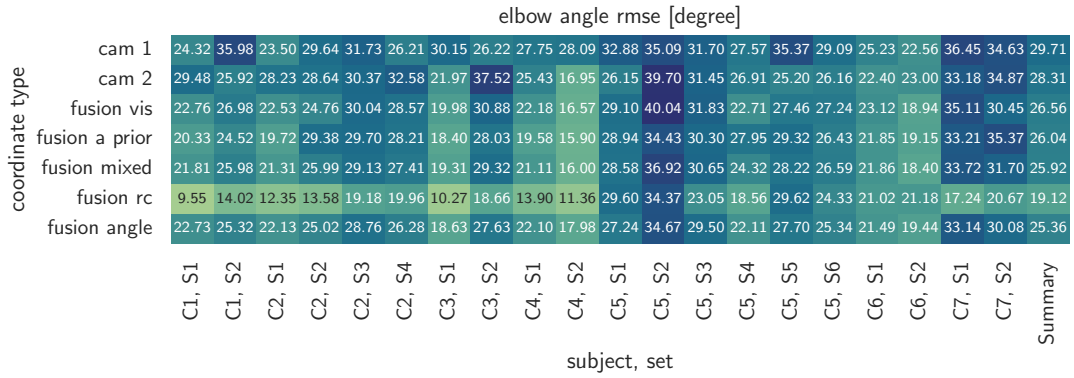


Figure 4.5.: A table presenting the RMSE of the elbow angle for the combinations of subject, set (CX, SX) and various HPE methods.

even if this leads to a higher variance in absolute values.

Additionally, fusion based on angles (relative level) further improves the results, though the advantage here is less usable since no fusion occurs at the coordinate system level. Generally, it can be noted that the differences within the subject and set variations, with some exceptions, are smaller. This leads to the conclusion that estimating relative positions is simpler and more accurate than estimating absolute positions.

In the analysis of the effects of camera selection on the accuracy of pose estimation, the difference between the two types of cameras, Reolink and webcam, is distinguished. By analyzing all subjects and sets and associating them with the respective cameras, it was found that the Reolink, on average, delivers 23.09% better results than the webcam. Specifically for hip width, the webcam scored an average of 5.24%, while the Reolink achieved 4.03%. There are two reasons, which can mainly explain this difference. First, the Reolink cameras have a significantly higher resolution than webcams, providing a more detailed pixel basis for estimations. Second, the Reolink recordings were conducted in a larger space of 5×5 meters, compared to 3×3 meters for the webcam. This resulted in placing the subjects more centrally in the image, where camera efficiency is higher. However, the significance of the comparison between camera types is limited by several factors. The number of subjects and the number of performances per orientation were not identical between the camera types, therefore a comparison is hard to make.

These inaccuracies cannot be eliminated from the data, as no error modelling has per-

formed in this case. While they do not reduce the validity of the observed differences between the cameras, they complicate the quantification of the actual extent of these differences. Therefore, it is important to emphasize that the conclusions drawn about the impact of camera selection on the accuracy of pose estimation are bound by the mentioned limitations.

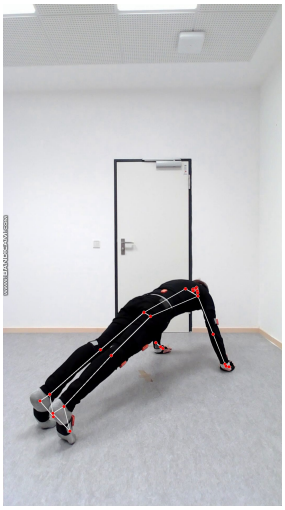


Figure 4.6.: C5, S2

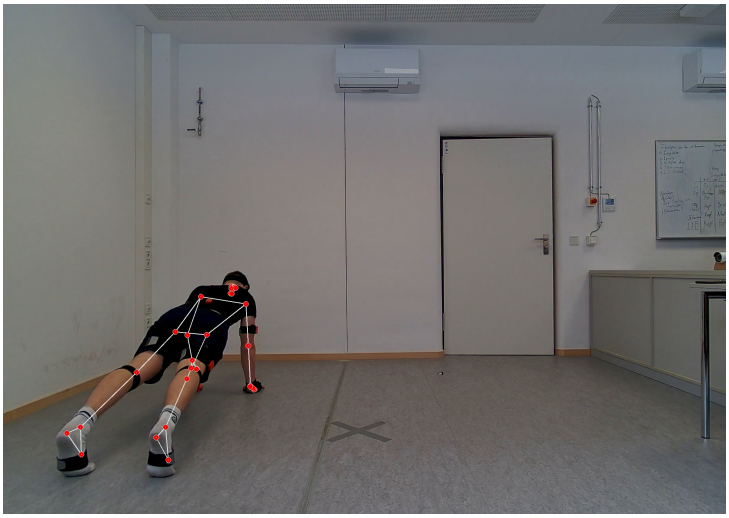


Figure 4.7.: C1, S2

Figure 4.8.: Two images showing the orientation for case (C1, S2 and C5, S2). This illustrates the difficulty of estimation, as the subject is turned away from the camera and is in the depth of space.

The previous analysis in this section has already shown differences between individual sets, with the ones depicted in Figures 4.7 and 4.6 being particularly noteworthy. These illustrations show that the referenced represent scenarios in which the subjects are turning away from or looking sideways at the camera. This is a significant challenge for estimation accuracy, as the MediaPipe model is primarily designed to recognize persons who are directly faced towards the camera or in a lateral position.

An additional complicating factor is the positioning of subjects deep within the room, which means a greater distance to the cameras. This results in fewer pixels per square meter being available, reducing the pixel density and thus the available resolution for estimations. The combination of these two factors leads to a significant deterioration in estimation quality, which stands out in the respective tables with deviations of over 482.97% compared to the

average. As a result, it can be inferred that for effective classification it is necessary that the patient are in a position in which they are facing the camera. Alternatively, the model used must be powerful enough to provide reliable estimates even under suboptimal conditions - such as reduced pixel density and unfavourable orientations of the subjects, which is note the case yet. Figure 4.9 presents another subject and exemplifies several, previously

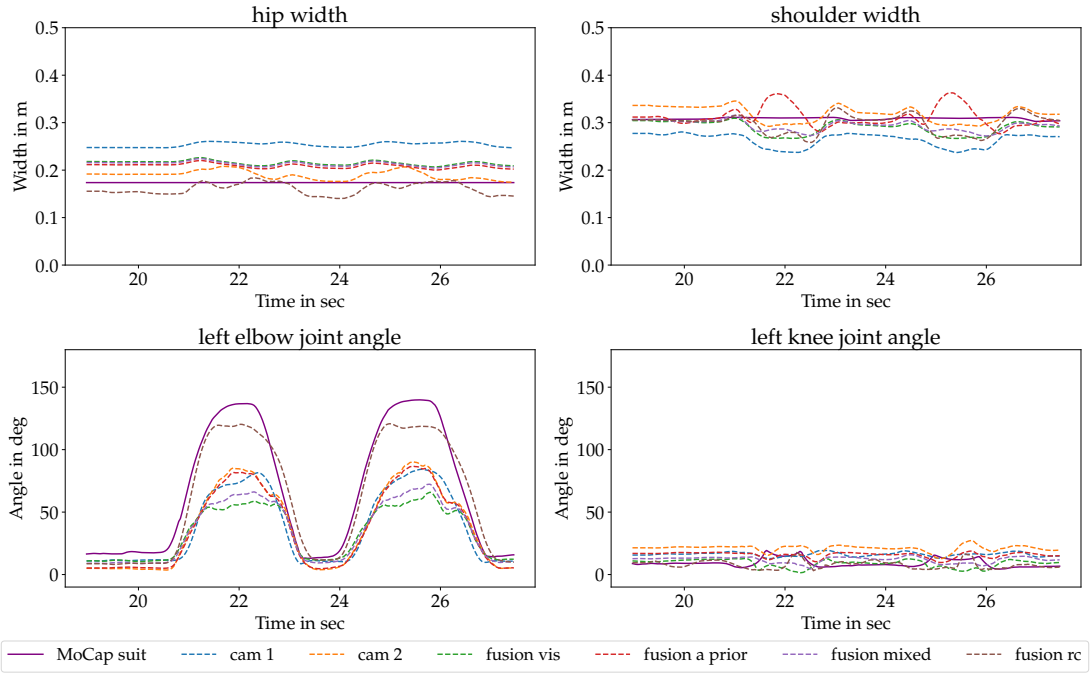


Figure 4.9.: A plot demonstrating the execution of two consecutive push-ups. Displayed are the hip and shoulder width, as well as the angles of the left elbows and knees. These values are presented for the various HPE estimates.

described findings. First, it is noticeable that significant differences in estimation accuracy can occur between Camera perspectives 1 and 2, depending on the subject's orientation, particularly visible at the left elbow. Second, the consistency of the 2D reconstruction is remarkable, remaining steady regardless of whether the left or right side is viewed. When examining the left elbow, it is observed that the estimations consistently show an offset compared to the MoCap suit. Moreover, it is evident that the fusion techniques, both based on visibility and a priori knowledge, form a good average between the estimates from Cameras 1 and 2. Specifically, fusion techniques based on visibility prove to be

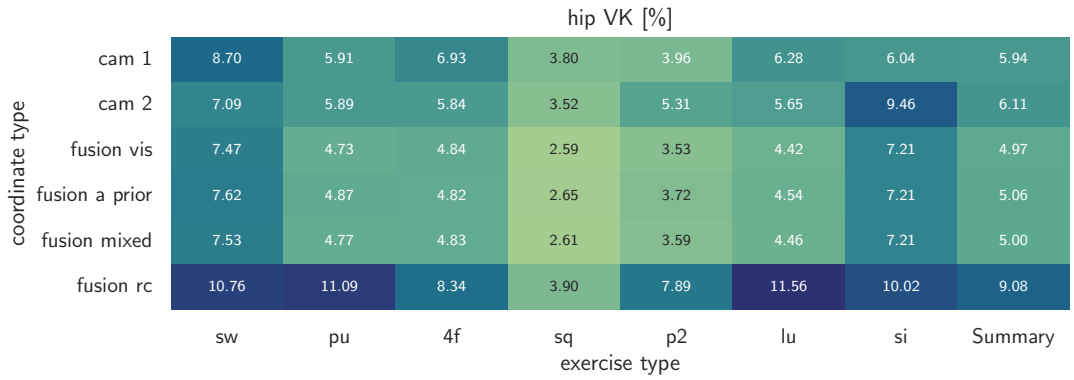


Figure 4.10.: A table presenting the coefficient of variation of the hip width in percentages for the combinations of exercise types and various HPE methods. The evaluated exercises are swimming (sw), push-ups (pu), kick-backs on all fours (4f), squats (sq), push-ups variation 2 (p2), lunges (lu) and sit-ups (si).

significantly advantageous, because they are closer to more accurate estimates.

The following part of the evaluation is focused on the differences in recognition performance across the seven recorded exercises. It begins by considering the absolute sizes, represented by the hip and shoulder width, listed in Tables 4.10 and 4.11. Based on these data, a consistent ranking regarding the effectiveness of the various fusion techniques can be created, similar to the examination of individual subjects and sets. These data match the previously described results.

In Tables 4.12 and 4.13 the RMSE of the knee and elbow angle are shown. Significant differences in recognizability are evident in the different exercises. The squat exercise shows the lowest error value, making it the most accurately captured. In contrast, the exercises swimming and fit apps are noticeable for significantly worse recognition rates. This is because the MediaPipe model was primarily trained on recognizing standing or moving persons and less on lying down or curled-up positions. This discrepancy makes correct estimation by MediaPipe difficult and leads to increased error rates for these specific exercises. Therefore it is more difficult for the classifiers. The exercises push-up, squat, and lunge, as well as an alternative variant of push-up, can be found in the middle of the ranking. These results imply that certain physical positions and movements can be captured by MediaPipe with greater accuracy than others, attributable to the specific training data basis of the model. In the further analysis of relative sizes and

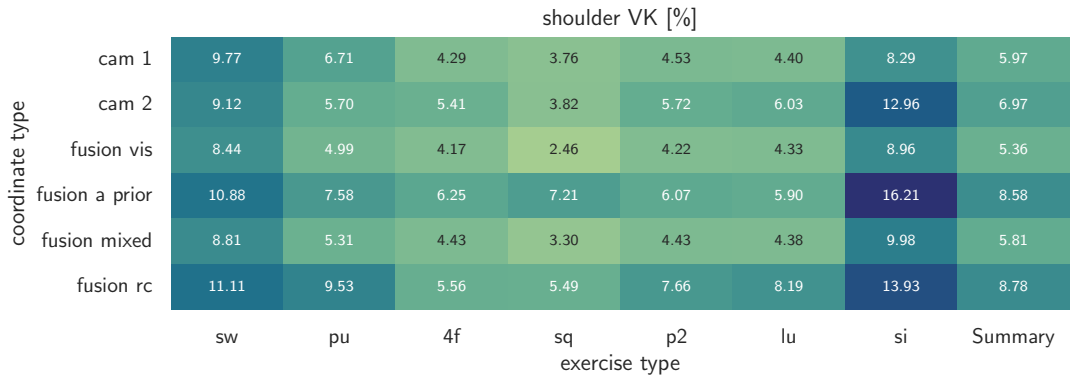


Figure 4.11.: A table presenting the coefficient of variation of the shoulder width in percentages for the combinations of exercise types and various HPE methods. The evaluated exercises are swimming (sw), push-ups (pu), kick-backs on all fours (4f), squats (sq), push-ups variation 2 (p2), lunges (lu) and sit-ups (si).

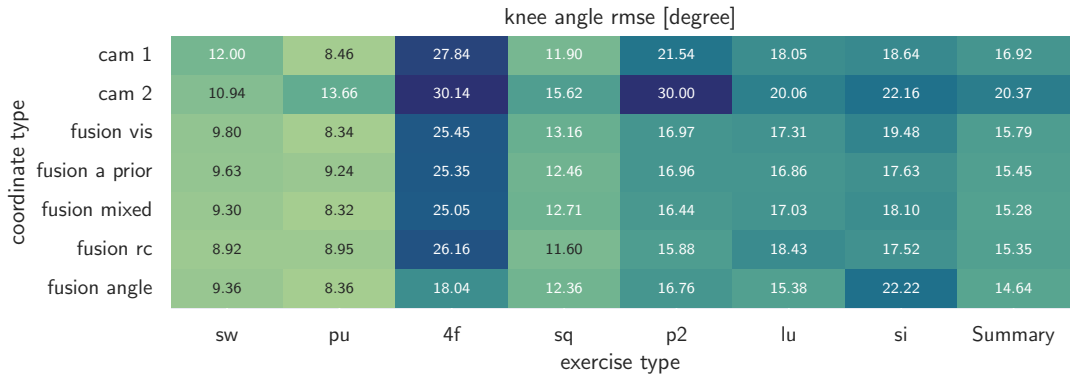


Figure 4.12.: A table presenting the RMSE of the knee angle for the combinations of exercise types and various HPE methods. The evaluated exercises are swimming (sw), push-ups (pu), kick-backs on all fours (4f), squats (sq), push-ups variation 2 (p2), lunges (lu) and sit-ups (si).

their correlation to the subjects and their positions in space, the previously discussed outcome is verified too. It appears that some exercises yield better estimation results when considering relative sizes than absolute sizes. A striking example of this is the swimming exercise, where the knee angle can be estimated relatively precisely, whereas the elbow

angle exhibits significantly poorer results. This difference can primarily be attributed to the fact that, in recordings with the webcam, the hands are often positioned at the edge of the frame, negatively impacting the estimation accuracy, as previously discussed. The exercises performed on all fours and the second variant of the push-up for the knee angle also stand out, by possessing a lower performance. It can be assumed that in these cases, an unusual, rarer position leads to an increased error degree. This emphasizes that the difficulty of correct recognition and estimation heavily depends on the specific exercise. Particularly challenging seems to be the exercises sit-ups and swimming, as well as the all-fours position, which turn out to be particularly difficult to recognize. Verified with the presented VK values. In contrast, the other exercises can be estimated with significantly higher accuracy.

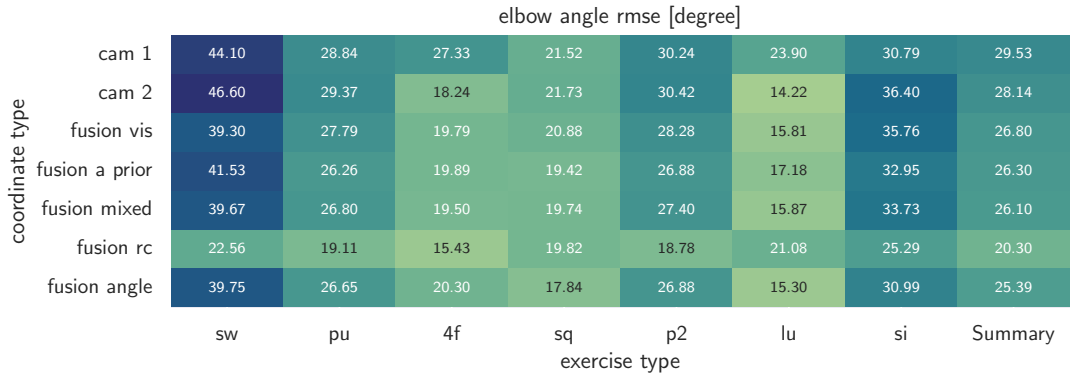


Figure 4.13.: A table presenting the RMSE of the elbow angle for the combinations of exercise types and various HPE methods. The evaluated exercises are swimming (sw), push-ups (pu), kick-backs on all fours (4f), squats (sq), push-ups variation 2 (p2), lunges (lu) and sit-ups (si).

All in All, it can be stated that estimations using camera technology generally show lower accuracy compared to inertial sensors, as used in MoCap suits. However, the error margins are similar, making the results comparable. In this section it has been presented that fusing data from two cameras improves the accuracy of estimations, thereby potentially facilitating subsequent classification.

The evaluation of the HPE with MediaPipe has also shown that the type of recording, the orientation of the persons towards the camera, and the type of exercises performed significantly influence estimation accuracy. This means that the system has to deal with

varying quality in the datasets. Another finding discusses the differences between relative and absolute sizes: in many cases, relative sizes prove to be superior to absolute sizes. This is due to the greater error tolerance in relative sizes, especially when all points exhibit the same error and therefore the relative positioning to each other does not change. This points out that features based on relative sizes are likely to have a positive impact on the accuracy of the estimation.

Overall, the high-quality estimations should enable effective classification. The previous described aspects highlight the importance and potential of camera technology for pose estimation, even if it currently does not quite reach the level of inertial sensors.

4.2. Exercise classification

Based on the evaluation of HPE, this section evaluates exercise classification. The following section emphasizes the utilization of features computed through the DTW method, evaluating their efficacy in conjunction with various classification algorithms. The aim is to develop a robust model capable of classifying the performing of various exercises with high precision by distinguishing between correctly and incorrectly performed movements.

This part of the evaluation begins with an exploration of different models, which are trained on the fusion vis and on the fusion rc (3D reconstruction). The aim is to determine which is more effective: better focus on visibility fusion or 3D reconstruction. These fusion types are chosen because, in the previous part of the evaluation, the performance of visibility data in terms of overall accuracy and the precise relative positioning achieved through 3D reconstruction is superior. The evaluation encompasses a spectrum of classic ML algorithms, including Decision Trees (DT), Random Forests (RF), SVM (SVC), Gradient Boosting (GB), Logistic Regression (LR), and Artificial Neural Networks (ANNs), detailed in Chapter 3.6. A total of 365 exercise recordings are utilized for training and testing, partitioned in an 80:20 ratio, selected due to established allocation [88]. Each ML model is evaluated considering a range of hyperparameters. For Decision Trees, aspects such as the maximum number of features, maximum depth, selection criterion (Gini Coefficient or Entropy) and the minimum number of samples per split or leaf are considered. Similar adjustments are made for Random Forests, Gradient Boosting and SVM, with the latter focusing on the kernel type and the regularization parameter C. In the case of ANNs, various architectures are explored, from simpler models with two layers (e.g., 64 and 32 neurons) to more complex configurations with up to five layers and an increasing number of neurons, aiming to maximize the network's capacity for feature extraction. Activation

functions include ReLU (Rectified Linear Unit) and Hyperbolic Tangent, culminating in an output layer with a sigmoid function.

Optimal combinations of hyperparameters are determined through Hyperparameter Tuning via Random Search, supported by a 5-fold Cross-Validation. Cross-validation is a statistical technique for evaluating and enhancing the accuracy of prediction models by partitioning the data into several parts and systematically testing the model with different parts such as training and test data. This process shall achieve that the ML model's robustness against diverse datasets and mitigate overfitting. This traninigs method enables efficient exploration of the hyperparameter space, ensuring a robust estimate of model performance across different data segmentations. The hyperparameter exploration is conducted through Random Search, with the combination of hyperparameters capped at a maximum of one thousand due to the computing time.

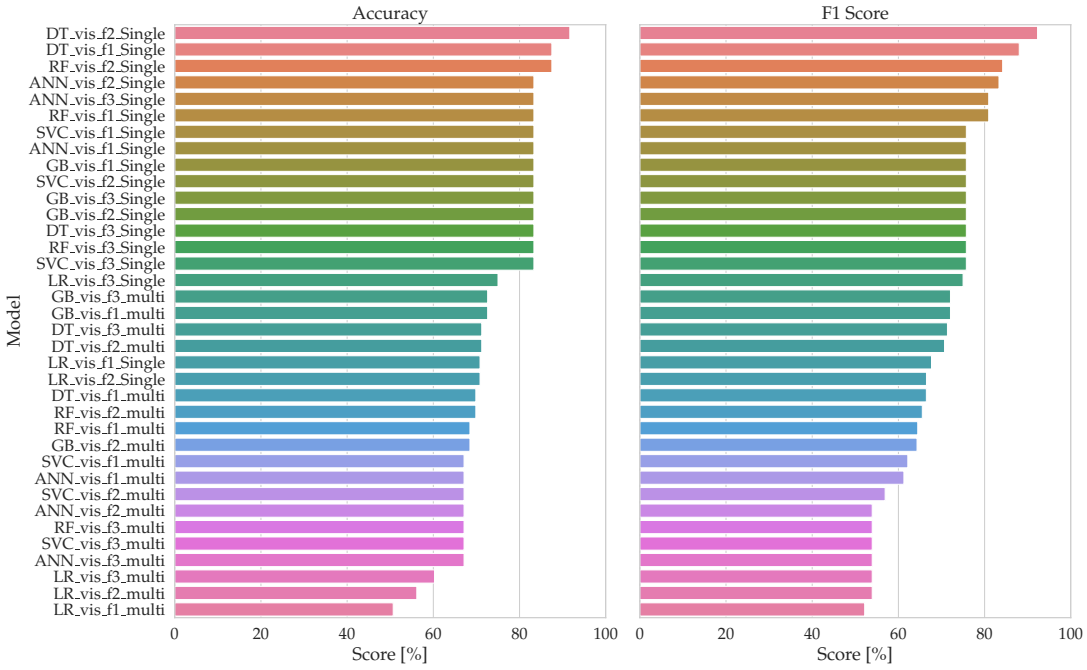


Figure 4.14.: Results of the model trained on fusion vis with feature variants f1,f2,f3, Single-Subject (single) or Multi-Subjects (multi) and the ML-Models Decision Trees (DT), Random Forests (RF), SVM (SVC), Gradient Boosting (GB), Logistic Regression (LR), and Artificial Neural Networks (ANNs). The best model with 91.7% acc is a decision tree f2 single.

The ML models are trained on different variants of the dataset's pre-processing. First, models based on the dataset from fusion vis were evaluated. These models were configured with three distinct feature sets to examine the impact of feature selection on model performance (as described in Section 3.4.6).

- Variant 1 (f1): This feature set comprises 27 features originating from seven limb groups, with each axis forming a feature. Additionally, it includes six primary angles that capture the spatial orientation of the limbs.
- Variant 2 (f2): Alongside the features contained in f1, this set further includes relative widths and distances between limbs, raising the total number of features to 55.
- Variant 3 (f3): This most comprehensive feature set combines the features of f1 and f2 with individual keypoints and their groupings, resulting in a total of 163 features.

Furthermore, two data variants were considered in the following evaluation: Multi-Variant (Multi-Subjects) and Single-Variant (Single-Subject). The Multi-Variant utilized the entire dataset to provide a broad view across multiple subjects. In contrast, the Single-Variant was limited to data from a single subject, thereby reducing the dataset to 119 records. The performance of the models was evaluated using various metrics, including Accuracy and the F1-Score. The F1-Score is defined as

$$F1 = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$

and represent the harmonic mean of precision and recall. Precision measures the proportion of true positive predictions among all positive predictions made by the model, while recall assesses the proportion of actual positives that were correctly identified [88]. As depicted in Figure 4.14, the results showcase a broad spectrum of model accuracies. The metrics are based on the test data set.

As expected, the Single-Variant demonstrated significantly better performance. This leads to the assumption that models trained on data from a single subject are more reliable. The most effective Single-Variant model achieved an accuracy of 91.7% and an F1-Score of 92.3%. In contrast, the best Multi-Variant model achieved an Accuracy of only 72.6%, emphasizing the significance of feature selection. Notably, the best-performing model was trained with the feature set from Variant 2.

Upon examining the performance of individual classifiers, no unequivocally superior methods were identified. Decision Trees are particularly effective, exhibiting nearly

similar good performances in detecting both positive and negative cases. This indicates a balanced model capability, without a tendency for misinterpretation in any specific direction. This speaks in favour of a well-balanced classification.

The observed similarity in model performance can be attributed to the limited size of the dataset, which restricts the variance in performance metrics.

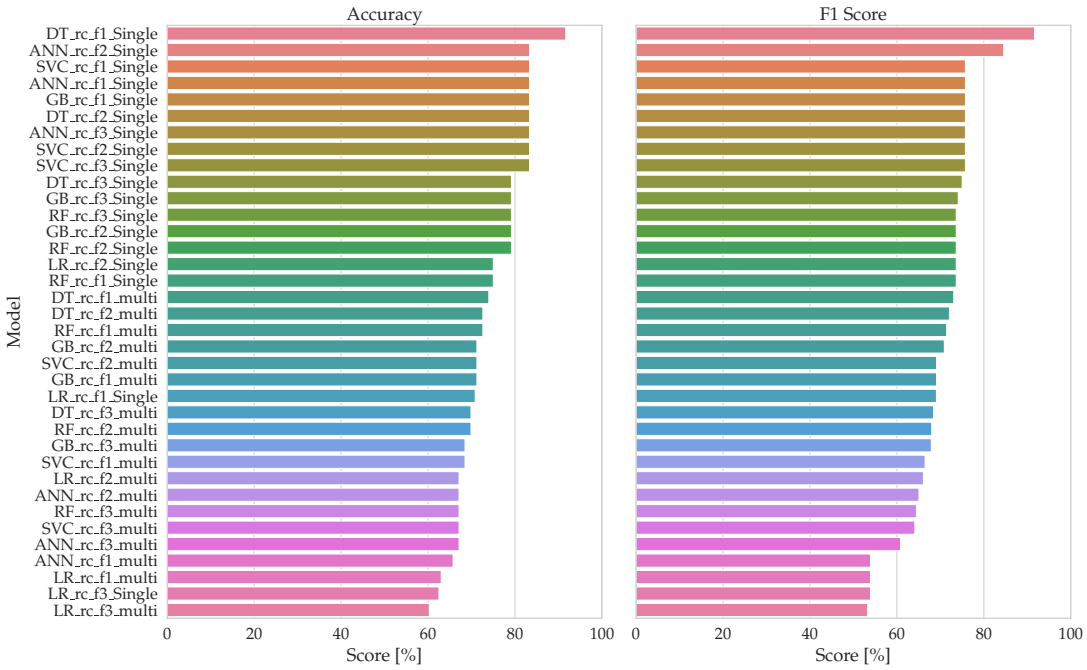
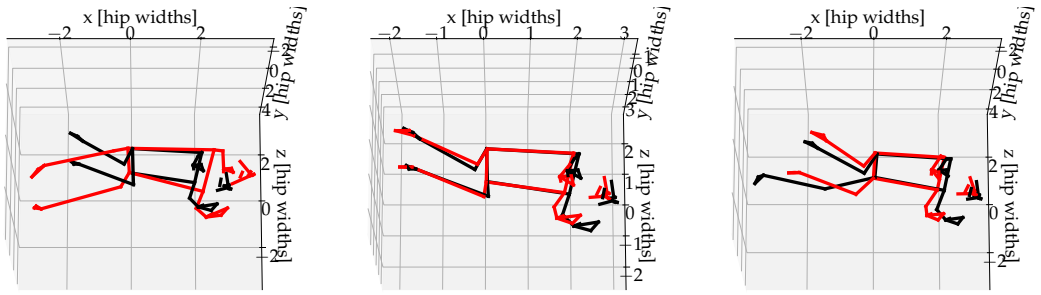


Figure 4.15.: Results of the model trained on fusion rc with feature variants f1,f2,f3, Single-Subject (single) or Multi-Subjects (multi) and the ML-Models Decision Trees (DT), Random Forests (RF), SVM (SVC), Gradient Boosting (GB), Logistic Regression (LR), and Artificial Neural Networks (ANNs). The best model with 91.7% acc is a decision tree f1 single.

Figure 4.15 illustrates the training results of models developed based on 3D construction features. Notably, the model utilizing fusion vis displays comparable performance to the best models in this category. Subsequent models, based on fusion rc, exhibit a slight reduction in performance, approximately 3 to 4 percentage points on average. Interestingly, the performance of the most inferior models in this category exceeds that of models based on fusion vis. A consistent observation is that ML models trained exclusively on

data from a Single-variant (Single-Subject) yield significantly better results compared to those using data from multi-variant (multi-subjects). This pattern highlights the balanced classification capability of the again, as no evidence was found suggesting disproportionate representation of either class. The analysis of individual classifiers suggests that Decision Trees continue to lead in this context as well. A direct comparison between models based on 2D features and those utilizing 3D construction features indicates that both approaches deliver satisfactory results. While 3D construction tends to offer a higher average accuracy, models based on fusion vis achieve a higher maximum accuracy. This leads to the conclusion that both approaches have their respective strengths. Therefore it is difficult to say which construction is the best because both have their advantages.



(a) Differences in body proportions

(b) Differences in positioning of the participants

(c) Differences in the execution of movements

Figure 4.16.: A Comparative Analysis of Movement Variability: Highlighting Differences in Body Proportions (a), Participant Positioning (b), and Movement Execution (c). Black and red each represent different subjects performing the same exercise, the movements are synchronised. These distinctions, along with errors, are captured as variances in DTW, complicating the classification process.

The superior performance of Single-Subject models compared to Multi-Subject models can be explained by three primary, non-error-related differences, as depicted in Figure 4.16. These differences contribute to the widening distance between the test exercise and the Golden Standard. As a result, these differences perceived as errors

- Differences in body proportions: Despite applying a scaling factor based on hip width, significant variabilities in proportions persist, such as the ratio of leg length to hip width, arm length to width, and shoulder width to hip width. A simple scaling factor based on hip width cannot fully neutralize these differences.
- Differences in the positioning of the participants: The initial body posture varies

among subjects, leading to different starting points for the performance of movements.

- Differences in the exercise of movements: Even for standardized exercises, such as squats, arm movements and other aspects of performance can vary between individuals. These variations are not errors in the traditional sense but lead to deviations which should be taken into account.

In Single-Subject models, these variabilities are minimized due to the consistency of body proportions and exercise performance within an individual. This is not the case with Multi-Subjects approaches, resulting in increased challenges and correspondingly lower performance. So-called error correction models have been developed to minimize the effects of these three differences. These models randomly adjust small rotations and stretches as well as randomly chosen bias. The adjustment is retained if such adjustments reduce the distance between model prediction and actual data. This method aims to improve performance, particularly in Multi-Subjects settings, by eliminating recognizable but unwanted described bias.

Figure 4.17 shows the results of ML models that have been enhanced using Modeling Error Correction techniques, from which two primary observations emerge. ML models predicated on Multi-Subjects data exhibit an average performance enhancement of 3 to 4 percentage points. This suggests that the implemented error correction techniques are particularly efficacious in the processing of heterogeneous data sources. In contrast, models based on Single-Subject data experience an average performance decrement of approximately 5 percentage points. This phenomenon may be attributed to the specific nature of error correction, which may not offer the same benefits in scenarios characterized by more consistent data sources and could potentially impair model performance through overfitting or by worsening errors less pronounced in homogeneous data. Moreover, it is significant that complex models like Artificial Neural Networks (ANN) or Gradient Boosting exhibit improved performance after error modelling is applied. This enhancement is notable when compared to their performance before the error modelling. This indicates that the characteristics rectified through error modelling become more complex, thereby enabling more sophisticated algorithms to more adeptly capture and generalize the underlying patterns.

The improvement in Multi-Subjects model performance accentuates the critical importance of the targeted application of error correction techniques. Whereas in contrast the performance seen in Single-Subject models due to error modelling shows a decline. These methods need to be carefully customized to the unique features of the dataset and the specific phenomena being modelled in order to improve model performance. In

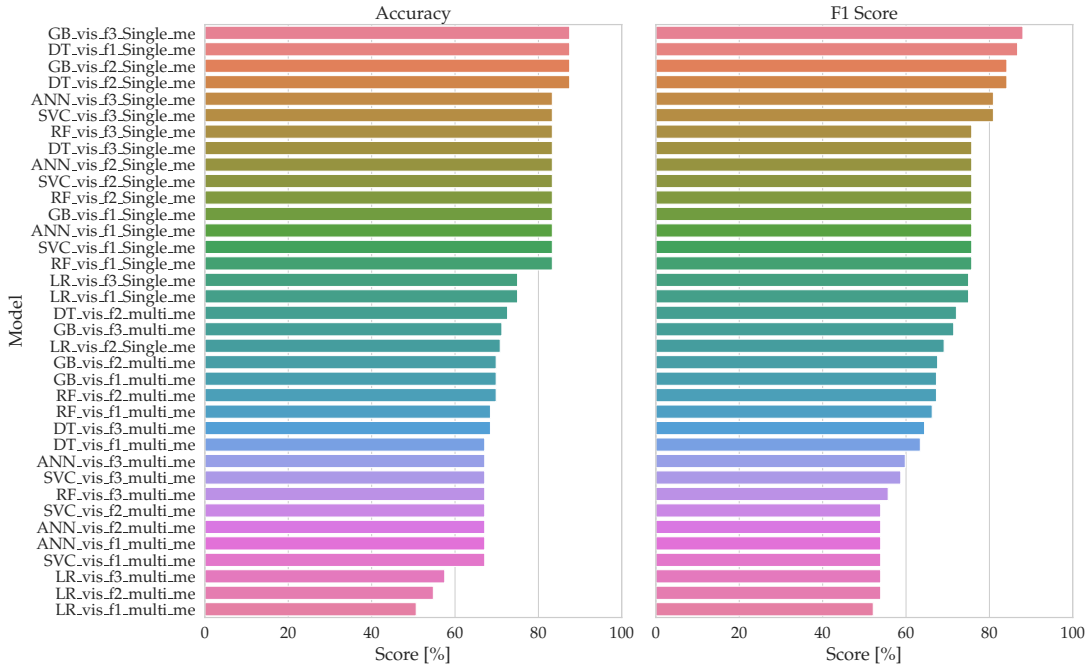


Figure 4.17.: Results of the model trained on fusion vis with feature variants f1,f2,f3, Single-Subject (single) or Multi-Subjects (multi) and with reduced model errors (me) and the ML-Models Decision Trees (DT), Random Forests (RF), SVM (SVC), Gradient Boosting (GB), Logistic Regression (LR), and Artificial Neural Networks (ANNs). The best model with 87.5% acc is a Gradient Boosting f3 single.

addition, emphasize these findings and the necessity to comprehend the complexity of modelling errors and effectively address them through customized modelling approaches. Thus ensuring that the chosen methods are congruent with the data's inherent properties and the analysis's objectives. Moreover, analysis was performed with model error reduction for fusion rc, shown in Figure 4.18. This confirmed the previously observed trend. These findings support the hypothesis that both fusion methods display similar performance ranges and that error correction is generally beneficial, although it can lead to performance losses in specific individual applications. This contributes to a decrease in the overall performance of the MoQuA algorithm. When dissecting the analysis into Single- and Multi-Subjects scenarios, it was found that Model *DT vis f2 Single* was the most effective for Single-Subject scenarios with an accuracy of 91.7%, whereas the best model for Multi-Subjects scenarios was *DT rc f1 multi me*, reaching only 74% accuracy,

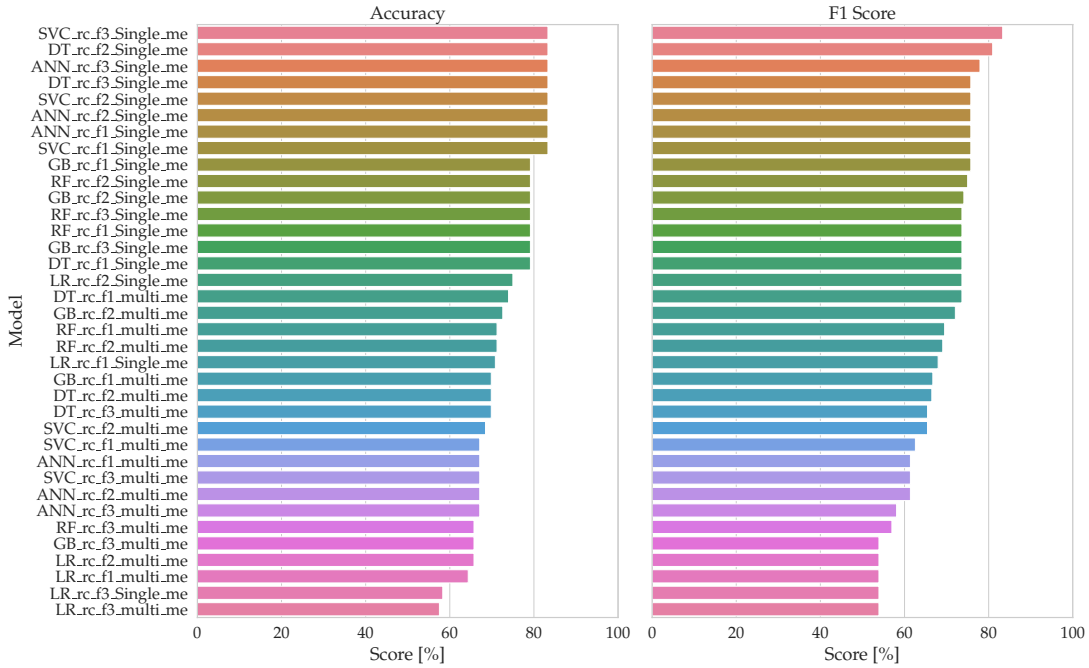


Figure 4.18.: Results of the model trained on fusion rc with feature variants f1,f2,f3, Single-Subject (single) or Multi-Subjects (multi) and with reduced model errors and the ML-Models Decision Trees (DT), Random Forests (RF), SVM (SVC), Gradient Boosting (GB), Logistic Regression (LR), and Artificial Neural Networks (ANNs). The best model with 83.3% acc is a SVC f3 single.

reflecting a substantial discrepancy of 17.7%. This significant variation implies that post-hoc adjustments are essential for managing side effects when using data from multiple subjects. While such modifications may negatively affect the performance when analyzing data from a Single-Subject, highlighting the importance of context-specific optimizations in applying error correction techniques to maximize model efficacy.

Further on a comparison between the two camera types, Reolink and webcam, is analysed. This is possible because the same subject took part in both recording variants. Using the same parameters as the best model previously described for this subject, a difference can be identified. The Reolink variant achieves an accuracy of 71.42, while the webcam variant achieves an accuracy of 79.1. These results should be viewed in the context that around

two-thirds of the data comes from the webcam variant, so it can only be generalised to a limited extent.

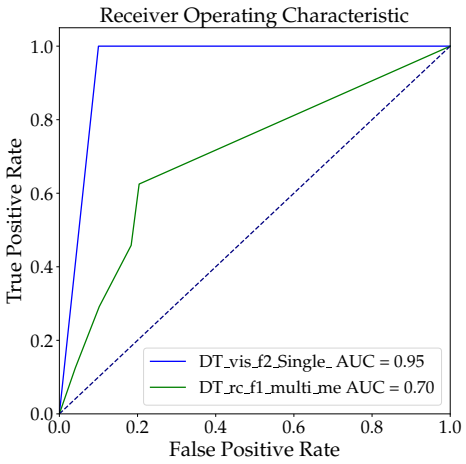


Figure 4.19.: Comparative ROC curves of the two best models for Single and Multi- Subjects: Single-Subject model shows superior classification performance with an AUC of 0.95, while Multi-Subject model demonstrates moderate discrimination capabilities with an AUC of 0.70.

In an additional analysis of the classification performance of the best models for Single-Subject and Mulit-Subject significant differences in their ability to differentiate between the considered classes are revealed. This analysis is represented by the Receiver Operating Characteristic (ROC) curves in Figure 4.19. The models, specified as *DT vis f2 Single* and *DT rc f1 multi me* (error reduction), are subsequently evaluated based on their Area Under the Curve (AUC) values. *DT vis f2 Single* (Blue Curve). This curve achieves an AUC of 0.95, indicating excellent class discrimination ability. An AUC near the ideal value of 1 signals a high model performance, with the curve's proximity to the upper left corner of the graph serving as an indicator of superior model performance. The position of the blue curve, close to this corner, highlights the strong classification capacity of Model *DT vis f2 Single*. *DT rc f1 multi me* (Green Curve). In comparison to both models, Model *DT rc f1 multi me* with an AUC of 0.70, exhibits satisfactory but significantly lower performance. This means that *DT rc f1 multi me* is capable of class differentiation but with less precision compared to Model *DT vis f2 Single*. The dashed line in the diagram represents a random classifier with an AUC of 0.5, which makes no distinction between the classes and is used as the baseline. The performance of both models above this line confirms their ability to classify beyond

random levels. The ROC curve plots the False Positive Rate (FPR) on the X-axis against the True Positive Rate (TPR, also known as sensitivity) on the Y-axis. FPR is derived as 1 minus the specificity. A ML model that achieves a high TPR while maintaining a low FPR effectively identifies the actual positive cases while minimizing the number of false-positive classified cases. The previously presented evaluation demonstrates that, based on the ROC curves and AUC values, Model DT vis f2 Single is the preferred classifier over DT rc f1 multi me due to its superior ability to distinguish between classes precisely. This analysis highlights the importance of AUC as a metric for evaluating model performance in classification tasks, highlighting its utility in identifying models with an optimal ability to distinguish capabilities.

In summary, for this section, the best model for single-subject scenarios, DT vis f2 Single, achieved an accuracy of 91.7%, while the leading multi-subject model, DT rc f1 multi me, reached 74%. This substantial difference of 17.7 percentage points underscores the need for customized error correction methods, especially when analyzing data from multiple subjects, to enhance accuracy. Overall, two satisfactory models were developed.

4.3. Feature importance and limb group detection

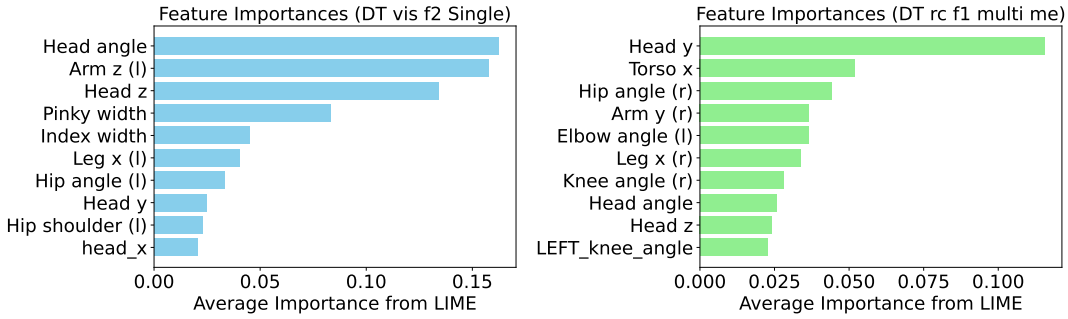


Figure 4.20.: Shows the 10 most influential features for the two best models according to LIME analysis

This section is based on the main goal of evaluating the features extracted from motion data, which means the sequence of HPE. To reach this goal an initial detailed analysis of feature importance is conducted using the LIME technique LIME facilitates the interpretation of individual models' decision-making processes by isolating the influence of individual

features on the prediction, as explained in Chapter 3.7.2. The average importance of the features across all tested exercises is calculated to make general statements. The two previously developed and chosen models (Section 4.2) for Single-Subject and Multi-Subjects will be analysed in the following section, referring to the aspect of feature importance.

The analysis with LIME, as illustrated in Figure 4.20, reveals that both evaluated models place particular focus on features describing the position of the head. Notably, in the *DT vis f2 Single* model, the head angle is considered significantly important. This is due to the fact that a substantial portion of movement errors is directly related to head position, suggesting that our identified error types can be predominantly captured through head features. But this focus also has drawbacks, as it may complicate the identification of other types of errors not directly associated with head position. A differentiated view of individual head characteristics indicates that the models tend to focus on certain aspects, which can lead to redundancies in the characteristics. Such redundancy is not necessarily desirable, as it potentially compromises the efficiency of feature extraction and the generalizability of the models. The following analysis shows a major difference in feature importance, with errors related to head position dominating the dataset, emphasizing the essential role of posture and head alignment in evaluating movement quality. The focus on the head position raises doubts about the model's capacity to identify and weigh other error types, highlighting the necessity for a balanced approach in feature selection and model training to ensure accurate and comprehensive assessments.

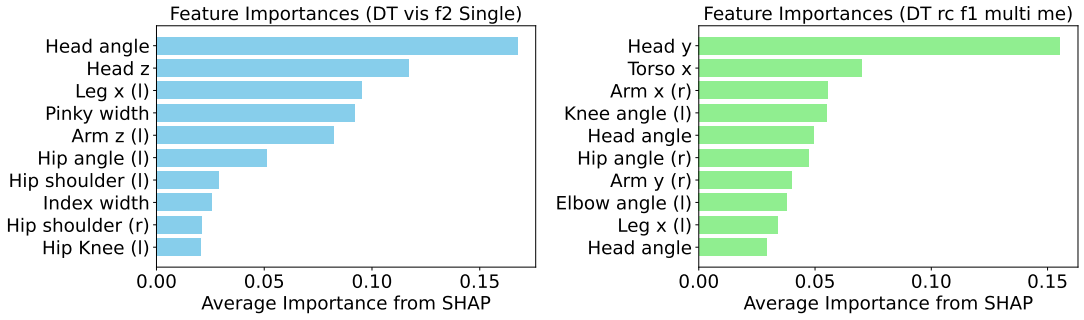


Figure 4.21.: Shows the 10 most influential features for the two best models according to SHAP analysis

To validate the findings of the LIME analysis, an additional approach was conducted

using the SHAP algorithm, with its results presented in Figure 4.21. The SHAP algorithm provides a result similar to LIME, highlighting key feature head y as particularly relevant. A noteworthy aspect of the SHAP algorithm is its more nuanced view of the features, attributable to its ability to represent not just the significance of individual decisions but the overall importance of features in the decision-making process. Although the order of feature importance varies between results determined by LIME and SHAP, with some features swapping positions, the similarity of the evaluations is significant enough to affirm the reliability of the underlying assumptions, as shown in Figure 4.20 and 4.21. This consistency emphasizes the robustness of the identified key features as determinants of model decisions.

The use of both analysis methods (LIME and SHAP) contributes to increasing the transparency and comprehensibility of model decisions by enabling a comprehensive understanding of the feature's importance for the developed algorithm. Supplementing LIME with SHAP highlights the importance of a multidimensional approach in explaining ML models. This dual-method approach of LIME and SHAP not only strengthens the credibility of the explanatory analysis but also provides a more detailed and reliable understanding of how and why models make certain predictions, facilitating the development of more interpretable and effective ML systems.

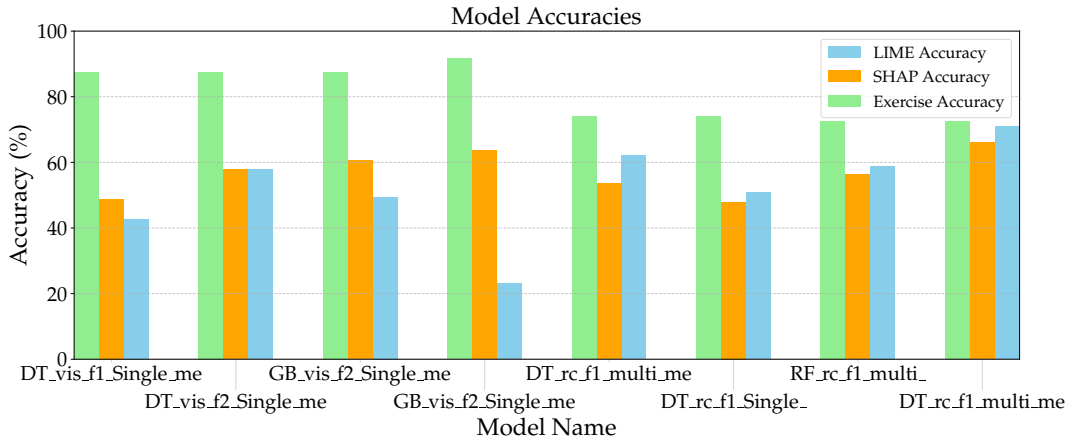


Figure 4.22.: Visualization of the accuracy of the classification of the incorrect limp group using SHAP and LIME.

In this part of the evaluation, the significance of features (Feature Importance) of both classification methods is utilized to assess decisions regarding the correct or incorrect

execution of exercises. More specifically, this described approach is employed to identify a corresponding limb group for each incorrectly classified exercise, which significantly influenced the error. For each decision, the features are sorted and aggregated according to their association with a limb group, with errors also categorized by limb groups. Subsequently, the sum of the weights for each limb group is calculated. The group with the highest cumulative weighting is identified as the primary source of error. Figure 4.22 presents the accuracies of various models taking into account this limb group analysis. The assumption is that the accuracy of limb group-specific classification is lower than that of the accuracy of general classification due to a finer differentiation being made. An important result from the tested data is that the SHAP method is superior in single-subject classification, while the LIME method achieves better results in Multi-Subject classification. This illustrates the different nature of the methods. For more powerful ML models, the accuracy of the limb group analysis is not significantly higher, likely due to the previously noted focus on head features. Another finding is that errors attributable to other limb groups are more challenging to identify.

In summary, these XAI approaches enable the extraction of additional insights into the sources of errors in motion analysis. Although the specific limb group classification may not achieve the same accuracy as the general classification evaluated in Section 4.2, it provides valuable information for improving training and feedback methods in motion analysis applications. This understanding of error sources is important to significantly enhance the development of targeted interventions and corrective measures, leading to more effective and personalized training programs.

4.4. Summary

All in all, the application and evaluation of the MoQuA algorithm's core components indicate promising results. While improvements are definitely needed in Human Pose Estimation (HPE), the overall quality of the outcomes is encouraging. While the presented results do not match the precision of MoCap suit, they stay within an acceptable range. The evaluation of HPE demonstrates that integrating two different data sources, in this case two cameras, could achieve a significant improvement in quality by approximately 17.7%, as shown in Section 4.1. This highlights the importance of diversifying information sources to enhance the accuracy of motion analysis. Potential future optimizations in camera technology and ML may open the opportunity to focus on a single data source.

A fundamental factor for the effectiveness of the MoQuA algorithm is the methodology of data fusion. This research analysed various approaches, such as visibility fusion and three-dimensional (3D) reconstruction. The 3D reconstruction and visibility fusion proved to be effective, even if each has its strengths in different aspects. While visibility fusion provides precise general positional data, 3D reconstruction offers superior results in determining relative positions and angles, which can be required for specific application areas. The evaluation also clarifies that, despite certain limitations of the previously described methods, a comparison between the motion data generated by the algorithm and those captured with a MoCap suit is possible. This reaffirms the initial assumption, made in section 2.2, that 3D data and visual material are sufficient for accurate motion analysis. Considering the rapid development pace in the field of ML libraries, it seems necessary to continuously evaluate the available tools in order to use the best model. Current results indicate that future advances in alternative libraries, such as the MediaPipe method, could potentially lead to further increases in the efficiency and accuracy of motion analysis. Specifically, there is a need for improvements in estimating body positions in less expected postures, such as sit-ups, as these are underrepresented in training datasets. The use of Reolink brand surveillance cameras compared to conventional webcams showed slight improvements in data collection, although the differences were not statistically significant. Nevertheless, the wider capture area in the initial recording sessions is for the used data notably positive. Thus implies the recommendation to aim for as comprehensive a recording area as possible to optimize the quality and completeness of the motion data. These findings imply that the continuous improvement and adaptation of algorithms and data collection methods are essential for the advancement of camera-based motion analysis, especially regarding the extraction of relevant features and the enhancement of trajectory similarity. The results of this research are valuable starting points for future research in this dynamic field.

The multidimensional DTW was used to refine the features based on the previously described data. This technique has turned out to be robust, particularly in its ability to neutralize differences in the execution speeds of the analyzed movements, effectively eliminating a potential source of error. This underlines the outstanding importance of the DTW procedure within the algorithm, which finds its strength in the precise analysis and comparability of movement patterns. However, the application of DTW necessitates a reference standard for comparison. This approach is based on one recorded video due to the limited availability of data. In a scenario where a more extensive dataset is available, selecting various correctly performed exercises could achieve wider variability, which would enable a more refined estimation of exercise performances. This was not feasible in the current work due to the deliberate decision to minimize effort by using a limited

volume of data, as discussed in Chapter 1. Categorizing the captured movements into so-called ”limb groups” turned out to be an effective method for reducing noise. This is evident as models utilizing a median number of features show the best performance, as shown in Figure 4.14. Based on the extracted features, the classification of the motion data was subsequently carried out. The consideration, evaluation and ultimately choice of different methods and the resulting insights significantly contribute to the advancement of precision and efficiency in camera-based motion analysis. These are fundamental for future research, especially regarding improving feature extraction and expanding the data basis to enable an even more detailed and comprehensive analysis of movement patterns. The findings from the application of multidimensional DTW offer promising approaches for optimizing motion analysis algorithms, particularly by eliminating variances in execution speed, which were previously seen as a challenge in precise motion analysis.

The preceding classification of extracted features, conducted using a comprehensive range of models, reveals significant insights regarding the suitability of various algorithms for motion analysis. Among these, the Decision Tree algorithm has proven to be effective in handling the characteristic features of the dataset. This emphasizes the importance of carefully selecting classification models that can appropriately consider the specific properties of the data. The presence of different subjects in the dataset led to the introduction of Single-Subject and Multi-Subjects classifications to allow broader generalizability of the results while simultaneously increasing the comparability of the models. In Single-Subject classification, which focuses on data from only one individual, an outstanding accuracy of over 91,7% was achieved. This result contrasts with the accuracy of 74% reached using the entire dataset (Multi-Subject). The discrepancy between these two approaches is caused by various factors, including the inherent variability among subjects regarding body proportions, starting positions and execution types. The challenge of achieving high generalizability across different subjects throws light on the potential limitations of universal models in motion analysis. The issues identified in the Single-Subject analysis indicated that differences in physical characteristics and performance styles of individual subjects could lead to a deterioration in model performance if not adequately considered. This highlights the need for an adaptive approach, where individually tailored models are developed that consider specific features and needs of users. In practice, this could mean that in physiotherapy, a base model is initially used, which is then individually adjusted by recording and analyzing a ’Golden Standard’ – consisting of several correctly performed exercises by the patient. This method enables combining the advantages of generalization across multiple subjects with the specific strengths of Single-Subject classification to ensure a customized and effective motion analysis.

The evaluation of feature quality and the traceability of how classification models make

their decisions are essential aspects of developing precise motion analysis systems. By using interpretability techniques such as LIME and SHAP it is revealed that features related to the head exert a significant influence on the ML models' decision-making processes. This is mainly due to a significant proportion of detected errors being associated with head position, complicating the identification of other error types. The model's focus on head movements underscores the need for a more diversified data basis to improve classification accuracy across a broader range of movement errors. Although the approach of specifically training models to distinguish between correct and incorrect exercise performances was discarded due to the limited amount of data per exercise and type of error, this emphasizes the importance of an expanded and diversified dataset.

Regardless, the challenge of collecting sufficient data for comprehensive training remains due to the high effort involved in data collection. Furthermore, the analysis of feature importance showed that angle calculations, in particular, represent a valuable metric for the classification of motion sequences. Integrating these specific features is important for achieving more precise classification results. In order to enable the assignment of errors to specific limb groups, LIME and SHAP utilised the explainability of the models. These XAI methodologies enabled targeted identification of which limb group a detected error pertains to. Despite these improvements, a limit to accuracy improvement through more complex models exists, partly traceable to the disproportionate consideration of head movements. These findings necessitate the introduction of a wider differentiation of errors in future research and the corresponding expansion of datasets to avoid overemphasis on certain types of errors. Especially in the context of physiotherapeutic exercises, it is important to train models on a wide range of motion sequences to enable comprehensive analysis and classification. This can ensure a consistently high classification performance across various types of exercises and errors, significantly enhancing the practical utility in physiotherapy.

Overall the implementation and development of the MoQuA Algorithm appears to be a promising approach. A detailed focus on specific exercises as well as individual error types is possible. However, a significant expansion of variability within the datasets is required for comprehensive application to ensure precise and generalizable analysis. Notably, the benefit of simple models, as described in Section 3.6, that allow fast execution is important for real-time applications. Future development could aim to achieve an even more flexible adaptation of the Golden Standard by integrating new methods and algorithms for HPE. Maybe this could be realized by combining several correctly performed exercise variants to create a comprehensive basis for evaluating exercise performance. Furthermore, the expanded classification, especially the direct assessment of specific error types and their assignment to certain body groups (Limb Groups), offers a promising

perspective for refining analysis methods. Such differentiation requires a significant expansion of the existing dataset. A special focus should be the application of these technologies in physiotherapy. The evaluation of HPE through MediaPipe shows that the accuracy of movement estimation varies depending on the type of exercise performed. The goal for a broad application should be to ensure consistently high-quality motion analysis, regardless of the specific exercise. This would not only enhance the effectiveness of therapeutic measures but also improve cost efficiency by enabling patients to perform exercises correctly and independently at home. Using a webcam, which is available in most households, already provides a solid foundation for practical implementation. In summary, the MoQuA algorithm represents a targeted solution with the potential to improve the quality and accessibility of motion analyses significantly. Continuous research and development in this field probably to produce powerful systems that can offer valuable support both in physiotherapy and in independent health care at home.

5. Conclusion

All in all, this work presents the development of an advanced algorithm (the MoQuA Algorithm), which assesses the quality of sports exercise performance by using video material. This process spans from comprehensive preprocessing and the sophisticated refinement of features using DTW to the final classification and ensuring traceability. Notably, the achieved accuracy of 91.7% in the analysis of individuals and up to 74% in the evaluation of group exercises highlights the effectiveness and reliability of the developed algorithm. In brief, this method enables combining the advantages of generalization across multiple subjects with the specific strengths of Single-Subject classification to ensure a customized and effective motion analysis. In practice, this could mean that in physiotherapy, a base model is initially used, which is then individually adjusted by recording and analyzing a 'Golden Standard' consisting of several correctly executed exercises by the patient as described in the previous evaluation.

A significant result of this thesis is that the fusion of data from two camera perspectives significantly improved the quality of the motion data. Despite the inherent challenges of estimating positions by using video cameras, results that are comparable to those of the MoCap suit, which were previously considered the gold standard, were achieved. This validates the practicability and innovative character of the MoQuA approach, which enables precise error detection and feature analysis in the context of the performance of exercises and lays the foundation for understandable and traceable results. A key value of the MoQuA algorithm is its ability to identify specific erroneous motion groups (Limb Groups). This goes beyond a simple assessment of right or wrong, as it is essential not only to detect errors but also to provide concrete indications of how the performance can be improved. Identifying specific weaknesses in movement execution offers a deep understanding for trainers and athletes alike to refine their technique in a targeted manner. Future enhancements could include integrating XAI to maintain or even expand the system's explainability, perhaps through action segmentation to identify errors in specific phases of a movement. Furthermore, the thesis presents various suggestions for improvement, such as expanding the dataset and conducting specific estimates for individual exercises

to further increase the accuracy and effectiveness of the system. It also shows that even a limited dataset is sufficient to develop a powerful model, highlighting the accessibility and applicability of the MoQuA algorithm in practice. A practical implementation of the MoQuA system should be extended by fine-tuning the trained ML model for individual patients. Additionally, introducing an exercise-specific classification could allow a more accurate assessment of different types of exercises through a two-stage process where the exercise type is initially classified before recognizing specific errors. Furthermore, improvements in HPE should be explored, especially for exercises where current posture estimations, such as sit-ups, do not perform as effectively as desired based on existing training data. Moreover, the accessibility of webcams as an already available and easy-to-use tool for data collection emphasizes the practical feasibility of implementing MoQuA in various settings, making advanced motion analysis more accessible to a broader audience. The use of small data sets can be considered sufficient, but the enlargement of the data set can contribute to improvement. So it is possible to start with small data sets and improve the model as new data sets are created

In conclusion, based on the findings and improvements by the MoQuA algorithm, it is not only possible to identify mistake-causing limbs but also provides a well-founded basis for possible improvement. The previously discussed results offer valuable starting points for future research aimed at refining the HMQA process and expanding its application areas. The previous analysis and developed approach contribute to improving feature extraction for camera-based motion analysis using trajectory similarity, laying a foundation for future innovations in this field. In this way, the main goal of supporting physiotherapy with the MoQuA algorithm is achieved.

Bibliography

[1] F. Alonso-Frech, J. J. Sanahuja, and A. M. Rodriguez, "Exercise and physical therapy in early management of parkinson disease", *The Neurologist*, vol. 17, 2011, ISSN: 1074-7931.

[2] H. P. French *et al.*, "Exercise and manual physiotherapy arthritis research trial (empart) for osteoarthritis of the hip: A multicenter randomized controlled trial", *Archives of Physical Medicine and Rehabilitation*, vol. 94, no. 2, pp. 302–314, 2013, ISSN: 0003-9993.

[3] J. L. Helbostad, O. Sletvold, and R. Moe-Nilssen, "Effects of home exercises and group training on functional abilities in home-dwelling older persons with mobility and balance problems. a randomized study", *Aging Clinical and Experimental Research*, vol. 16, no. 2, pp. 113–121, Apr. 2004, ISSN: 1720-8319.

[4] U. Elsner, "Vdek-basisdaten des gesundheitswesens in deutschland", *Verband der Ersatzkassen e. V. (vdek)*, 2023.

[5] GKV-Spitzenverband, "Bundesbericht: Gkv-heilmittel-schnellinformation für deutschland", *Spitzenverband Bund der Krankenkassen (GKV-Spitzenverband)*, 2023.

[6] B. H. Jones, D. N. Cowan, and J. J. Knapik, "Exercise, training and injuries", *Sports Medicine*, vol. 18, no. 3, pp. 202–214, Sep. 1994.

[7] J. B. Lauersen, D. M. Bertelsen, and L. B. Andersen, "The effectiveness of exercise interventions to prevent sports injuries: A systematic review and meta-analysis of randomised controlled trials", *British Journal of Sports Medicine*, vol. 48, no. 11, pp. 871–877, 2014. DOI: 10.1136/bjsports-2013-092538.

[8] S. Dill *et al.*, "Accuracy evaluation of 3d pose estimation with mediapipe pose for physical exercises", in *Current Directions in Biomedical Engineering*, De Gruyter, vol. 9, 2023, pp. 563–566.

[9] D. Setiawan, M. H. Purnomo, and E. M. Yuniarno, "Multi-human pose detection based on eelan-blazepose model", in *2023 International Seminar on Intelligent Technology and Its Applications (ISITIA)*, 2023, pp. 105–109.

[10] A. Rajšp and I. Fister, "A systematic literature review of intelligent data analysis methods for smart sport training", *Applied Sciences*, vol. 10, no. 9, 2020, ISSN: 2076-3417.

[11] F. Frangoudes, M. Matsangidou, E. C. Schiza, K. Neokleous, and C. S. Pattichis, "Assessing human motion during exercise using machine learning: A literature review", *IEEE Access*, vol. 10, pp. 86 874–86 903, 2022.

[12] Y. Liao, A. Vakanski, M. Xian, D. Paul, and R. Baker, "A review of computational approaches for evaluation of rehabilitation exercises", *Computers in Biology and Medicine*, vol. 119, p. 103 687, 2020, ISSN: 0010-4825.

[13] Q. Lei, J.-X. Du, H.-B. Zhang, S. Ye, and D.-S. Chen, "A survey of vision-based human action evaluation methods", *Sensors*, vol. 19, no. 19, 2019, ISSN: 1424-8220.

[14] G. J. Luvizutto, G. F. Silva, and M. R. Nascimento, "Use of artificial intelligence as an instrument of evaluation after stroke: A scoping review based on international classification of functioning, disability and health concept", *Topics in Stroke Rehabilitation*, vol. 29, no. 5, pp. 331–346, 2022.

[15] P. Parmar, J. Reddy, and B. Morris, "Piano skills assessment", in *2021 IEEE 23rd International Workshop on Multimedia Signal Processing (MMSP)*, 2021, pp. 1–5.

[16] E. Stefana, F. Marciano, D. Rossi, P. Cocca, and G. Tomasoni, "Wearable devices for ergonomics: A systematic literature review", *Sensors*, vol. 21, no. 3, 2021, ISSN: 1424-8220.

[17] C. CJ, P. KE, and C. GM., "Physical activity, exercise, and physical fitness: Definitions and distinctions for health-related research", *Public Health Rep*, 1985.

[18] V. O.-K. chel and V. Oertel-Knöchel, "Aktiv für die psyche", *Springer*, 2016.

[19] F. Mortazavi and A. Nadian-Ghomsheh, "Continues online exercise monitoring and assessment system with visual guidance feedback for stroke rehabilitation", *Multimedia Tools and Applications*, vol. 78, no. 22, pp. 32 055–32 085, 2019, ISSN: 1573-7721.

[20] V. Fernandez-Cervantes, N. Neubauer, B. Hunter, E. Stroulia, and L. Liu, "Virtualgym: A kinect-based system for seniors exercising at home", *Entertainment Computing*, vol. 27, pp. 60–72, 2018, ISSN: 1875-9521.

[21] R. de Souza Baptista, A. P. L. Bó, and M. Hayashibe, "Automatic human movement assessment with switching linear dynamic system: Motion segmentation and motor performance", *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 25, no. 6, pp. 628–640, 2017.

[22] A. Ebert, M. T. Beck, A. Mattausch, L. Belzner, and C. Linnhoff-Popien, “Qualitative assessment of recurrent human motion”, in *2017 25th European Signal Processing Conference (EUSIPCO)*, 2017, pp. 306–310.

[23] J. Albert, P. Glöckner, B. Pfitzner, and B. Arnrich, “Data augmentation of kinematic time-series from rehabilitation exercises using gans”, in *2021 IEEE International Conference on Omni-Layer Intelligent Systems (COINS)*, 2021, pp. 1–6.

[24] S. G. de Villa, A. M. Parra, A. J. Martín, J. J. G. Domínguez, and D. Casillas-Perez, “Ml algorithms for the assessment of prescribed physical exercises”, in *2021 IEEE International Symposium on Medical Measurements and Applications (MeMeA)*, 2021, pp. 1–6.

[25] J.-K. Kim, K. B. Lee, J.-C. Kim, and S. G. Hong, “Patient identification based on physical rehabilitation movements using skeleton data”, in *2021 International Conference on Information and Communication Technology Convergence (ICTC)*, 2021, pp. 1572–1574.

[26] S. H. Chowdhury, M. A. Amin, A. K. M. M. Rahman, M. A. Amin, and A. A. Ali, “Assessment of rehabilitation exercises from depth sensor data”, in *2021 24th International Conference on Computer and Information Technology (ICCIT)*, 2021, pp. 1–7.

[27] J. Hagelbäck, P. Liapota, A. Lincke, and W. Löwe, “The performance of some machine learning approaches in human movement assessment”, in *Proc. 13th Multi Conf. Comput. Sci. Inf. Syst.(MCCSIS)*, 2019, pp. 35–42.

[28] A. Khan *et al.*, “Generalized and efficient skill assessment from imu data with applications in gymnastics and medical training”, *ACM Trans. Comput. Healthcare*, vol. 2, no. 1, 2021.

[29] P. Caserman, C. Krug, and S. Göbel, “Recognizing full-body exercise execution errors using the teslasuit”, *Sensors*, vol. 21, no. 24, 2021, ISSN: 1424-8220.

[30] T. Hakim, “A comprehensive review of skeleton-based movement assessment methods”, *arXiv preprint arXiv:2007.10737*, 2020.

[31] H. Jain and G. Harit, “An unsupervised sequence-to-sequence autoencoder based human action scoring model”, in *2019 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, 2019, pp. 1–5.

[32] M. Chariar, S. Rao, A. Irani, S. Suresh, and C. S. Asha, “Ai trainer: Autoencoder based approach for squat analysis and correction”, *IEEE Access*, vol. 11, pp. 107 135–107 149, 2023.

[33] Q. Lei, H.-B. Zhang, J.-X. Du, T.-C. Hsiao, and C.-C. Chen, "Learning effective skeletal representations on rgb video for fine-grained human action quality assessment", *Electronics*, vol. 9, no. 4, 2020, ISSN: 2079-9292.

[34] M. Gassen *et al.*, "I³: Interactive iterative improvement for few-shot action segmentation", in *2023 32nd IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, 2023, pp. 378–385.

[35] R. Kianifar, A. Lee, S. Raina, and D. Kulić, "Automated assessment of dynamic knee valgus and risk of knee injury during the single leg squat", *IEEE Journal of Translational Engineering in Health and Medicine*, vol. 5, pp. 1–13, 2017.

[36] J. M. Palomares-Pecho, G. F. M. Silva-Calpa, C. A. Sierra-Franco, and A. Barbosa Raposo, "End-user programming architecture for physical movement assessment: An interactive machine learning approach", in *Digital Human Modeling and Applications in Health, Safety, Ergonomics and Risk Management. Posture, Motion and Health*, V. G. Duffy, Ed., Cham: Springer International Publishing, 2020, pp. 335–348, ISBN: 978-3-030-49904-4.

[37] B. Garg, "Deep learning approach to skeletal performance evaluation of physical therapy exercises", English, Ph.D. dissertation, 2023, p. 33.

[38] A. Kryeem, S. Raz, D. Eluz, D. Itah, H. Hel-Or, and I. Shimshoni, "Personalized monitoring in home healthcare: An assistive system for post hip replacement rehabilitation", in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, 2023, pp. 1868–1877.

[39] Q. Zheng, X. Fu, Y. Li, and S. Cai, "Adaptive real-time rectifying exercise porsture of sport rehabilitation system based on mediapipe", in *2023 2nd International Conference on Health Big Data and Intelligent Healthcare (ICHIH)*, 2023, pp. 176–181.

[40] C. Zheng *et al.*, "Deep learning-based human pose estimation: A survey", *ACM Computing Surveys*, vol. 56, no. 1, pp. 1–37, 2023.

[41] Y. Chen, Y. Tian, and M. He, "Monocular human pose estimation: A survey of deep learning-based methods", *Computer Vision and Image Understanding*, vol. 192, p. 102 897, 2020, ISSN: 1077-3142.

[42] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime multi-person 2d pose estimation using part affinity fields", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[43] L. Pishchulin *et al.*, "Deepcut: Joint subset partition and labeling for multi person pose estimation", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[44] H.-S. Fang, S. Xie, Y.-W. Tai, and C. Lu, "Rmpe: Regional multi-person pose estimation", in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017.

[45] X. Zhang, X. Zhou, M. Lin, and J. Sun, "Shufflenet: An extremely efficient convolutional neural network for mobile devices", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[46] K. Sun, B. Xiao, D. Liu, and J. Wang, "Deep high-resolution representation learning for human pose estimation", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[47] V. Bazarevsky, I. Grishchenko, K. Raveendran, T. Zhu, F. Zhang, and M. Grundmann, "Blazepose: On-device real-time body pose tracking", *arXiv preprint arXiv:2006.10204*, 2020.

[48] J. Wang *et al.*, "Deep 3d human pose estimation: A review", *Computer Vision and Image Understanding*, vol. 210, p. 103 225, 2021, ISSN: 1077-3142.

[49] S. Mroz *et al.*, "Comparing the quality of human pose estimation with blazepose or openpose", in *2021 4th International Conference on Bio-Engineering for Smart Technologies (BioSMART)*, 2021, pp. 1–4.

[50] A. Mathis *et al.*, "Deeplabcut: Markerless pose estimation of user-defined body parts with deep learning", *Nature Neuroscience*, vol. 21, no. 9, pp. 1281–1289, 2018, ISSN: 1546-1726.

[51] S. Dubey and M. Dixit, "A comprehensive survey on human pose estimation approaches", *Multimedia Systems*, vol. 29, no. 1, pp. 167–195, 2023, ISSN: 1432-1882.

[52] X. Zhao, F. Hou, J. Su, and L. Davis, "An alphapose-based pedestrian fall detection algorithm", in *Artificial Intelligence and Security*, Cham: Springer International Publishing, 2022, pp. 650–660.

[53] H.-S. Fang *et al.*, "Alphapose: Whole-body regional multi-person pose estimation and tracking in real-time", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 6, pp. 7157–7173, 2023.

[54] S. Kulkarni, S. Deshmukh, F. Fernandes, A. Patil, and V. Jabade, "Poseanalyser: A survey on human pose estimation", *SN Computer Science*, vol. 4, no. 2, p. 136, 2023.

[55] C.-z. Guan, “Realtime multi-person 2d pose estimation using shufflenet”, in *2019 14th International Conference on Computer Science and Education (ICCSE)*, 2019, pp. 17–21.

[56] M. Wang, J. Tighe, and D. Modolo, “Combining detection and tracking for human pose estimation in videos”, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[57] H.-C. Nguyen, T.-H. Nguyen, J. Nowak, A. Byrski, A. Siwocha, and V.-H. Le, “Combined yolov5 and hrnet for high accuracy 2d keypoint and human pose estimation”, *Journal of Artificial Intelligence and Soft Computing Research*, vol. 12, no. 4, pp. 281–298, 2022.

[58] R. Chauhan, I. Dhyani, and H. Vaidya, “A review on human pose estimation using mediapipe”, in *2023 3rd International Conference on Innovative Sustainable Computational Technologies (CISCT)*, 2023, pp. 1–6.

[59] J.-W. Kim, J.-Y. Choi, E.-J. Ha, and J.-H. Choi, “Human pose estimation using mediapipe pose and optimization method based on a humanoid model”, *Applied Sciences*, vol. 13, no. 4, 2023, ISSN: 2076-3417.

[60] S. Dill *et al.*, “Accuracy evaluation of 3d pose estimation with mediapipe pose for physical exercises”, *Current Directions in Biomedical Engineering*, vol. 9, pp. 563–566, Sep. 2023. DOI: 10.1515/cdbme-2023-1141.

[61] M. Paulich, M. Schepers, N. Rudigkeit, and G. Bellusci, “Xsens mtw awinda: Miniature wireless inertial-magnetic motion tracker for highly accurate 3d kinematic applications”, May 2018.

[62] S. Di Paolo *et al.*, “Rehabilitation and return to sport assessment after anterior cruciate ligament injury: Quantifying joint kinematics during complex high-speed tasks through wearable sensors”, *Sensors*, vol. 21, p. 2331, Mar. 2021.

[63] S. Dill and M. Rohr, “Mnmdtw: An extension to dynamic time warping for camera-based movement error localization”, *arXiv preprint arXiv:2310.09170*, 2023.

[64] R. Hartley and A. Zisserman, “Multiple view geometry in computer vision”, *Cambridge University Press*, 2004.

[65] D. la Escalera, Armingol, and J. María, “Automatic chessboard detection for intrinsic and extrinsic camera parameter calibration”, *Sensors*, vol. 10, no. 3, pp. 2027–2044, 2010, ISSN: 1424-8220.

[66] Y. Liu, Q. Ren, J. Geng, M. Ding, and J. Li, “Efficient patch-wise semantic segmentation for large-scale remote sensing images”, *Sensors (Basel, Switzerland)*, vol. 18, Sep. 2018. DOI: 10.3390/s18103232.

[67] J. Weng, P. Cohen, M. Herniou, *et al.*, "Camera calibration with distortion models and accuracy evaluation", *IEEE Transactions on pattern analysis and machine intelligence*, vol. 14, no. 10, pp. 965–980, 1992.

[68] R. Szeliski, "Computer vision : Algorithms and applications", *Springer Nature*, Texts in computer science, 2022.

[69] H. Andrews and C. Patterson, "Singular value decomposition (svd) image coding", *IEEE Transactions on Communications*, vol. 24, no. 4, pp. 425–432, 1976.

[70] R. I. Hartley and P. Sturm, "Triangulation", *Computer Vision and Image Understanding*, vol. 68, no. 2, pp. 146–157, 1997, ISSN: 1077-3142.

[71] T. Batpurev. "Direct linear transforms (dlt)". (Feb. 6, 2021), [Online]. Available: https://temugeb.github.io/computer_vision/2021/02/06/direct-linear-transorms.html (visited on 04/21/2023).

[72] F. J. Taylor, "Signal processing, digital", in *Encyclopedia of Physical Science and Technology (Third Edition)*, R. A. Meyers, Ed., Third Edition, New York: Academic Press, 2003, pp. 737–760, ISBN: 978-0-12-227410-7.

[73] E. Robles, J. Pou, S. Ceballos, J. Zaragoza, J. L. Martin, and P. Ibañez, "Frequency-adaptive stationary-reference-frame grid voltage sequence detector for distributed generation systems", *IEEE Transactions on Industrial Electronics*, vol. 58, no. 9, pp. 4275–4287, 2011.

[74] L. Zhongshen, "Design and analysis of improved butterworth low pass filter", in *2007 8th International Conference on Electronic Measurement and Instruments*, 2007, pp. 1-729-1–732.

[75] K. S. Arun, T. S. Huang, and S. D. Blostein, "Least-squares fitting of two 3-d point sets", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-9, no. 5, pp. 698–700, 1987.

[76] N. Magdy, M. A. Sakr, T. Mostafa, and K. El-Bahnasy, "Review on trajectory similarity measures", in *2015 IEEE Seventh International Conference on Intelligent Computing and Information Systems (ICICIS)*, 2015, pp. 613–619.

[77] H. Alt, "The computational geometry of comparing shapes", *Springer*, S. Albers, H. Alt, and S. Näher, Eds., pp. 235–248, 2009.

[78] P. Marteau, "Time warp edit distance with stiffness adjustment for time series matching", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 2, pp. 306–318, 2009.

[79] M. Vlachos, G. Kollios, and D. Gunopulos, “Discovering similar multidimensional trajectories”, in *Proceedings 18th International Conference on Data Engineering*, 2002, pp. 673–684.

[80] X. Wang, A. Mueen, H. Ding, G. Trajcevski, P. Scheuermann, and E. Keogh, “Experimental comparison of representation methods and distance measures for time series data”, *Data Mining and Knowledge Discovery*, vol. 26, no. 2, pp. 275–309, 2013.

[81] M. Shokoohi-Yekta, B. Hu, H. Jin, J. Wang, and E. Keogh, “Generalizing dtw to the multi-dimensional case requires an adaptive approach”, *Data Mining and Knowledge Discovery*, vol. 31, no. 1, pp. 1–31, 2017, ISSN: 1573-756X.

[82] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, “Liblinear: A library for large linear classification”, *the Journal of machine Learning research*, vol. 9, pp. 1871–1874, 2008.

[83] I. Steinwart and T. Christmann Andreas, “Support vector machines -”, *Springer Science und Business Media*, 2008.

[84] M. Trommer, “Ein beitrag zur anwendung von support-vektor-maschinen zur robusten nichtlinearen klassifikation komplexer biologischer daten”, Ph.D. dissertation, Technische Universität Ilmenau, Germany, 2017. [Online]. Available: https://www.db-thueringen.de/receive/dbt%5C_mods%5C_00032299.

[85] C. Kingsford and S. L. Salzberg, “What are decision trees?”, *Nature Biotechnology*, vol. 26, no. 9, pp. 1011–1013, 2008, ISSN: 1546-1696.

[86] L. Breiman, “Random Forests”, *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[87] X. Wang, G. Gong, N. Li, and S. Qiu, “Detection Analysis of Epileptic EEG Using a Novel Random Forest Model Combined With Grid Search Optimization”, *Frontiers in Human Neuroscience*, vol. 13, 2019, ISSN: 1662-5161.

[88] J. Zou, Y. Han, and S.-S. So, “Overview of artificial neural networks”, *Artificial neural networks: methods and applications*, pp. 14–22, 2009.

[89] J. H. Friedman, “Greedy function approximation: A gradient boosting machine”, *Annals of statistics*, pp. 1189–1232, 2001.

[90] A. Natekin and A. Knoll, “Gradient boosting machines, a tutorial”, *Frontiers in neurorobotics*, vol. 7, p. 21, 2013.

[91] A. Adadi and M. Berrada, “Peeking inside the black-box: A survey on explainable artificial intelligence (xai)”, *IEEE Access*, vol. 6, pp. 52 138–52 160, 2018.

[92] E. Tjoa and C. Guan, “A survey on explainable artificial intelligence (xai): Toward medical xai”, *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 11, pp. 4793–4813, 2021.

[93] M. T. Ribeiro, S. Singh, and C. Guestrin, “”why should i trust you?”: Explaining the predictions of any classifier”, in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD ’16, San Francisco, California, USA: Association for Computing Machinery, 2016, pp. 1135–1144, ISBN: 9781450342322.

[94] S. M. Lundberg and S.-I. Lee, “A unified approach to interpreting model predictions”, in *Advances in Neural Information Processing Systems*, I. Guyon *et al.*, Eds., vol. 30, Curran Associates, Inc., 2017.

[95] H. Abdi, “Coefficient of variation”, *Encyclopedia of research design*, vol. 1, no. 5, 2010.

A. Overview of Subject/Set combinations

Rounds	Subject	Set	α [deg]	β [deg]
1	C1	S1	90	0
2	C1	S2	180	90
3	C2	S1	90	0
4	C2	S2	30	-60
5	C2	S1	90	0
6	C2	S2	135	45
7	C3	S1	90	0
8	C3	S2	-90	180
9	C4	S1	90	0
10	C4	S2	45	-45
11	C5	S1	90	0
12	C5	S2	-135	135
13	C5	S3	90	0
14	C5	S4	45	45
15	C5	S5	-45	135
16	C5	S6	90	0
17	C6	S1	90	0
18	C6	S2	45	45
19	C7	S1	90	0
20	C7	S2	45	45

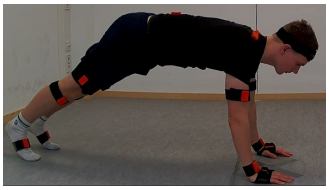
Table A.1.: Table of Subject/Set combinations with corresponding angles. See Figure 3.1 for the setup.

B. Exercise execution

The specific exercises and the associated errors are outlined in detail, providing clear guidance for the participant on executing each variation. It is crucial to make the mistakes clearly so that they are recognisable in the end.

B.1. Push-up variant 1

(a) Correct movement



(b) Mistake 1

(c) Mistake 2

Figure B.1.: Shows the correct movement of a push-up (a); Mistake 1: hips too high (b); Mistake 2: looking upwards (c).

The following describes the correct execution of a push-up.

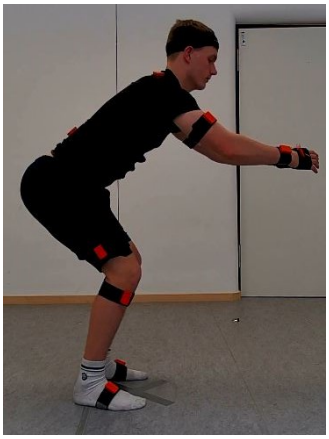
- **Starting Position:**
  - Begin in a plank position with your hands placed slightly wider than shoulder-width apart.
  - Your arms should be extended, with your hands positioned directly under your shoulders.
  - Keep your body in a straight line from your head to your heels.
- **Lowering:**

  - Lower your body by bending your elbows.
  - Ensure that your body forms a straight line during the descent.
  - Go as low as possible without touching the ground.
- **Rising:**
  - Press back up to return to the starting position.
  - Fully extend your arms but avoid overextending your elbows.
- **Breathing:**
  - Inhale as you lower your body and exhale as you rise.
- **Tips:**
  1. Maintain a straight line with your body throughout the entire movement.
  2. Engage your core muscles to stabilize your torso.
  3. Ensure that your elbows are positioned at approximately a 45-degree angle to your body.

The description of the two simulated mistakes follows.

- **Mistake 1:**
  - Positioning the hips too high. This results in a push-up in misalignment with the shoulders and feet and disrupts the ideal straight body line.
- **Mistake 2:**
  - Looking upwards during a push-up instead of maintaining a downward gaze, leading to misalignment of the spine and neck.

(a) Correct movement



(b) Mistake 1

(c) Mistake 2

Figure B.2.: Shows the correct movement of a squat (a); Mistake 1: not going low enough (b); Mistake 2: having feet too wide (c).

B.2. Squat

The following describes the correct execution of a squat.

- **Starting Position:**
  - Stand upright with your feet approximately shoulder-width apart.
  - Your toes may point slightly outward.
  - Keep your back straight and your shoulders pulled back.
- **Lowering:**
  - Simultaneously bend your knees and hips as if you were going to sit down.
  - Lower your body as though sitting back into a chair.
  - Ensure your knees do not extend beyond your toes.
- **Depth:**
  - Descend as deeply as possible while maintaining proper form.
  - Ideally, your thighs should be parallel to the ground or lower.

- **Rising:**
  - Push through your heels to return to the starting position.
  - Be mindful not to overextend your knees as you rise.
- **Breathing:**
  - Inhale as you lower your body and exhale as you rise.

The description of the two simulated mistakes follows.

- **Mistake 1:**
  - The participant does not lower themselves sufficiently, keeping their knee angle above 90 degrees.
- **Mistake 2:**
  - The participant positions their feet more than shoulder-width apart.

B.3. Kick-backs on all fours

(a) Correct movement (b) Mistake 1 (c) Mistake 2

Figure B.3.: Shows the correct movement of kick-backs on all fours (a); Mistake 1: looking upwards (b); Mistake 2: knees too close to the hands (c).

The following describes the correct execution of kick-backs on all fours.

- **Starting Position:**
  - Position yourself on all fours by placing your hands and knees on the ground.
  - Your hands should be directly under your shoulders and your knees under your hips.

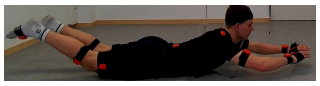
  - Maintain a neutral spine throughout the exercise.
- **Extension:**
  - Lift one leg and extend it backwards.
  - Keep the leg straight and the sole of your foot parallel to the ground.
  - Focus on activating your gluteal muscles as you move your leg back.
- **Hold and Contraction:**
  - Hold the extended leg at its highest position for a moment to contract the muscles.
  - Feel the tension in your gluteal muscles.
- **Return to Starting Position:**
  - Lower the leg back to the starting position in a controlled manner, without touching the ground.
  - Repeat the movement with the other leg.
- **Breathing:**
  - Exhale as you lift your leg and inhale as you lower it.
- **Tips:**
  - Ensure your back remains straight throughout the exercise.
  - Avoid using momentum; perform the movement in a controlled manner.
  - Focus on engaging the gluteal muscles for maximum effectiveness.

The description of the two simulated mistakes follows.

- **Mistake 1:**
  - The participant looks upwards during the execution.
- **Mistake 2:**
  - The participant positions their knees too close to their hands, disrupting the intended alignment of hands under shoulders and knees under hips.

B.4. Swimming

(a) Correct movement



(b) Mistake 1

(c) Mistake 2

Figure B.4.: Shows the correct movement of swimming (a); Mistake 1: looking upwards (b); Mistake 2: shoulder too close to the head (c).

The following describes the correct execution of swimming.

- **Starting Position:**
  - Lie flat on your stomach on a comfortable, flat surface. Ensure there is enough space around you for arm and leg movements.
  - Extend your arms straight in front of you, palms facing each other. Your legs should be straight and together, toes pointed.
- **Head Position:**
  - Keep your head in a neutral position, aligned with your spine. Avoid lifting it too high or pressing it too far down.
  - Gaze should be downwards, slightly in front of you.
- **Movement:**
  1. Raise your right arm and left leg simultaneously, lifting them a few inches off the ground.
  2. Stretch both limbs as if you are reaching forward with your hand and backwards with your foot, elongating your body.
  3. Lower your right arm and left leg, returning them to the starting position.
  4. Immediately repeat the movement with your left arm and right leg.
  5. Continue alternating sides in a smooth, controlled motion, simulating a swimming action.
- **Breathing:**

- Coordinate your breathing with your movements. Inhale as you lift your limbs and exhale as you lower them.

The description of the two simulated mistakes follows.

- **Mistake 1:**
  - The participant incorrectly directs their head upwards towards the ceiling instead of towards the ground.
- **Mistake 2:**
  - The participant pulls the shoulder too close to the head.

B.5. Push-up variant 2

(a) Correct movement

(b) Mistake 1

(c) Mistake 2

Figure B.5.: Shows the correct movement of a push-up (variant 2) (a); Mistake 1: looking upwards (b); Mistake 2: back is not straight (c).

The following describes the correct execution of a push-up (variant 2).

- **Starting Position:**
  - Place your knees on the ground and set your hands slightly wider than shoulder-width apart.
  - Your arms should be extended, with your hands positioned directly under your shoulders.
  - Keep your body in a straight line from your head to your knees.
- **Lowering:**
  - Lower your upper body by bending your elbows.
  - Ensure that your body forms a straight line while descending.

  - Go as low as possible without touching the ground.
- **Rising:**
  - Press up to return to the starting position.
  - Fully extend your arms but avoid overextending your elbows.
- **Breathing:**
  - Inhale as you lower your upper body, and exhale as you raise it.

The description of the two simulated mistakes follows.

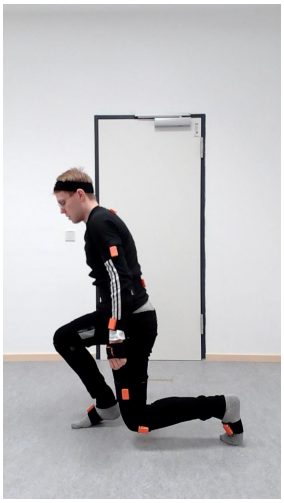
- **Mistake 1:**
  - Looking upwards during a push-up instead of maintaining a downward gaze, leading to misalignment of the spine and neck.
- **Mistake 2:**
  - The participant's back is not straight but curved downwards.

B.6. Lunge

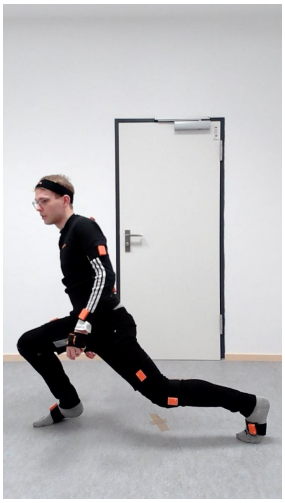
The following describes the correct execution of a Lunge.

- **Starting Position:**
  - Stand upright with your feet about shoulder-width apart.
- **Forward Step:**
  - Take a large step forward with one foot.
  - Lower your body by bending the rear knee until both knees are bent at approximately right angles.
  - Ensure that the front knee is aligned over the ankle and does not extend past it.
- **Hold Position:**
  - Keep your upper body upright, back straight, and core (abdominal muscles) engaged.

(a) Correct movement



(b) Mistake 1



(c) Mistake 2

Figure B.6.: Shows the correct movement of a Lunge (a); Mistake 1: Taking a step that is too small. (b); Mistake 2: The front knee turns towards the inside (c).

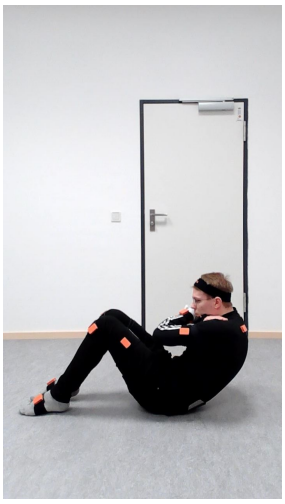
- **Return to Starting Position:**
  - Push back with the front foot to return to the upright position.
  - Repeat the lunge with the other leg.
- **Tip:**
  - Ensure your movements are controlled and stable to prevent injuries.
  - Adjust the length of your stride to suit your individual needs and capabilities.

The description of the two simulated mistakes follows.

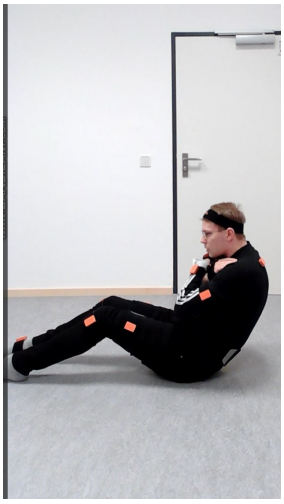
- **Mistake 1:**
  - Taking a step that is too small.
- **Mistake 2:**
  - The front knee turns towards the inside.

Sit-up

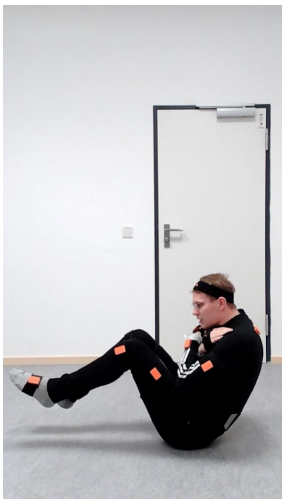
The following describes the correct execution of a sit-up.



(a) Correct movement



(b) Mistake 1



(c) Mistake 2

Figure B.7.: Shows the correct movement of a Sit-up (a); Mistake 1: Knees not bent enough (b); Mistake 2: Feet leave the ground (c).

- **Starting Position:**
  - Lie flat on your back on a suitable surface, such as a sports mat.
  - Bend your knees so that your feet are flat on the ground, with the soles touching the floor.
  - Your arms can either be crossed in front of your chest or placed lightly behind your head.
- **Upward Movement:**
  - Engage your abdominal muscles.
  - Lift your upper body by flexing forward at the abdomen.
  - Aim to reach your hands towards your knees.
- **Downward Movement:**

  - Lower your upper body back to the mat in a controlled manner without momentum.
  - Avoid fully touching the ground to maintain tension in the abdominal muscles.
- **Tip:**
  - Perform the exercise at a controlled pace to prevent injuries.
  - Exhale during the upward movement and inhale during the downward movement.
  - Avoid using momentum and use the abdominal muscles to lift the upper body.

The description of the two simulated mistakes follows.

- **Mistake 1:**
  - Knees not bent enough.
- **Mistake 2:**
  - Feet leave the ground.