Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Computer Science

Joachim Eide Eivindsen
Brede Yabo Kristensen

Human Pose Estimation Assisted Fitness Technique Evaluation System

Master's thesis in Master of Science in Informatics
Supervisor: Theoharis Theoharis
June 2020

NTNU
Norwegian University of
Science and Technology

Joachim Eide Eivindsen
Brede Yabo Kristensen

Human Pose Estimation Assisted Fitness Technique Evaluation System

Master's thesis in Master of Science in Informatics
Supervisor: Theoharis Theoharis
June 2020

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Computer Science

NTNU
Norwegian University of
Science and Technology

Abstract

Weight lifting is an effective and popular way to gain the benefits from strength training, but comes at a high risk of injuries for newcomers. Every lifter has their own set of challenges when improving their technique and their feedback needs to reflect this.

With the recent developments in human pose estimation this thesis aims at examine how this technology can be used as a tool to give valuable feedback on weight lifting technique. This task involves the detection of specific technique related issues with high association with risk of injury for common exercises.

This thesis propose an analytical approach through developing a feedback system where the exercise and filming perspective are automatically detected, before the associated technique aspects are tested for. Dynamic time warping is used for the action recognition process, while vector calculations are performed on the human pose estimation data to test for issues related to weight lifting technique. Also, an overall overview of selected existing human pose estimation systems is presented and evaluated.

We demonstrate that this method is effective in detecting technique related issues for multiple users, exercises and technique issues. The result showed considerable scores for subjects facing the camera, while subject with their side to the camera was challenging to analyze. The results indicate that human pose estimation is maturing and produces viable results when analyzing weight lifting technique, although a bigger dataset may be needed to confirm these findings. Granting that this is an interesting application that undoubtedly would gain from further research.

Sammendrag

Vektløfting er en populær og effektiv form for styrketrening, men denne metoden kommer også med høy risiko for skade blant nye løftere. Hver person har ulike utfordringer når de skal forbedre løfte-teknikken sin og det er viktig at de får tilbakemeldinger som reflekterer dette.

Med stor utvikling innenfor human pose estimation de siste årene, har denne avhandlingen som mål å undersøke hvordan denne teknologien kan bli brukt for å gi verdifull tilbakemelding til brukeren om deres vektløfting teknikk. Dette er en oppgave som innebærer å detektere spesefikke feil ved teknikken som har en tett sammenheng med risiko for skade.

Denne avhandlingen presenterer en ny analytisk tilnerming via et utviklet tilbakemeldingssystem der øvelse og vinkel for filming blir automatisk detektert, før de tilhørende teknikkaspektene blir testet for. Dynamic time warping blir brukt for å gjenkjenne øvelsen, mens vektorkalkulasjoner blir utført på human pose estimation dataen for å teste for de ulike problemene relatert til løfteteknikk.

Vi demonstrerer at denne metoden er effektiv for å detektere teknikk-realterte feil for flere brukere, øvelser og teknikkaspekter. Resultatet viste gode tall for personer plassert rett fremfor kameraet, mens brukere med siden sin mot kameraet viste seg å være vanskligere å analysere. Resultatet indikerer at human pose estimation har blitt nøyaktig nok til å produsere gode resultater for å detektere feil ved løfteteknikk, selv om et større datasett muligens er nødvendig for å bekrefte disse funnene. Det er uansett klart at dette er et interessant bruksområdet for teknologien som vil gagne mye av videre undersøkelse.

Acknowledgements

With high motivation and work ethic, we still needed guidance and feedback to make the best out of this thesis, which is why we owe some acknowledgements to those who have contributed a great deal.

We would like to thank our supervisor, Theoharis, for his professional feedback and guiding us through this thesis. The weekly meeting sessions were both inspiring and exciting and we learned a lot from them.

We would also like to thank Marius Steiro Fimland and Svein Ove Tjøsvoll for the rich discussions and help when researching injury risk in weight lifting. As well as Espen Ihlen for his knowledge and efforts when performing the qualitative evaluation of the exercise videos.

Last, but not least we would like to thank our two super athletic weight lifters Helga Sangolt and Kamilla Jacobsen for their contribution to enrich our exercise dataset.

Contents

Appendices

List of Figures

List of Tables

Glossary

AP	Average Precision.
COCO	Common Objects in Context.
DTW	Dynamic Time Warping.
JSON	JavaScript Object Notation.
LOESS	Locally Weighted Scatterplot Smoothing.
RFID	Radio Frequency Identification.
RGB-D	RGB-Depth.
SSB	Statistisk Sentralbyrå(Statistics Norway).

Chapter 1

Introduction

Human pose estimation is a highly focused research field in the computer vision community and have a variety of use cases ranging from robotics to human action recognition. With the technology of 2D pose estimation coming a long way and camera phones in every adults pocket, the table is set for a more widespread application of the technology. Likewise, technology has long been embraced as an important part of analyses in the context of sports and weight lifting [1]. This thesis aims at using human pose estimation to analyse technical aspects of weight lifting and give feedback to increase performance and decrease the risk of injuries.

1.1 Motivation

Training and physical activity have long been seen as an important factor to minimize the risk of chronic diseases and premature death as well as providing mental health benefits [2, 3]. However, sports and weightlifting activities also comes with a great risk of injuries, as a result from improper execution and poor technique [4, 5]. This could be due to muscle fatigue, using too much weight or lack of proper technique training and understanding. Compound exercises such as squat and deadlift are among the most injury-prone, due to the heavy weight involved and high load on muscles and joints, and will therefore be the focus point of this thesis.

Fitness centers are now placed in every major city and smaller towns and is widely available for the general population. According to SSB, the percentage of Norwegians over 16 years old performing strength training at least once the last twelve months increased from 30% to 46% between 2007 and 2019 [6]. Meaning that almost half of the adult population have been doing some kind of strength training over the last year. This is beneficial to society considering the benefits strength training provides. However, it creates a demand for newcomers needing to learn proper weightlifting technique. Watching a video on how to perform a squat or read an article about deadlifting is helpful in understanding the movements, though to truly rule out errors in technique, feedback and practice is necessary. This includes either having a network of people with experience in weightlifting or paying for a personal trainer. Even then, there is no guarantee that the person giving you feedback truly understand how to lift properly and what pitfalls to avoid to stay healthy.

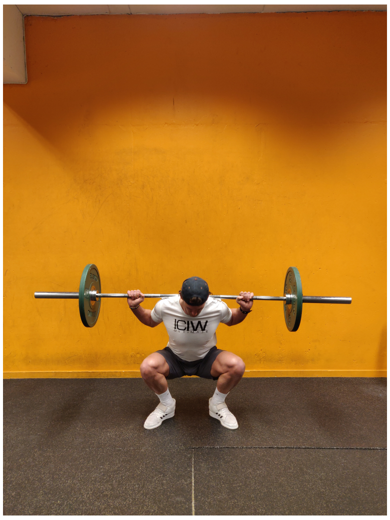


Figure 1.1: Example Squat

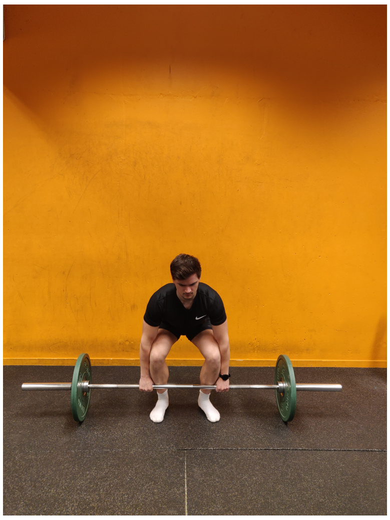


Figure 1.2: Example Deadlift

Weightlifting is also considered an important measure against overweight, being the second most popular physical activity among Norwegians [6]. Internationally findings from the World Health Organization shows that 39% of the global population is overweight (bmi $\geq$ 25) which is almost twice as mush as in 1975 [7]. A tool that can assist the users in correctly lifting technique can hopefully boost confidence of the execution and not only prevent injuries but can also be used as a motivator to start lifting. Building confidence in the movement the user is performing, will also hopefully make it easier to work out in public gyms and crowded places.

With the improvements in the field of deep learning, pose estimation results have improved substantially over the past ten years [8]. State-of-the-art technologies are now able to accurately detect a persons joints, even in complicated situations such as sporting activities and weightlifting.

The use of smartphones is increasing every year, and most of these smartphones include a RGB camera. Further, alternatives like RGB-D cameras or multiple camera angels are either to expensive or hard to configure correctly for most users. This makes it practical to use 2D human pose estimation models which use RGB images as input. With a few instructions anyone can film themselves weightlifting [9], thus improving their technique without the need to interact with anything other than their phone. This can be a great tool for introverts and people with social anxiety as well as being a much cheaper solution than paying for a personal trainer.

1.2 Thesis Goal

The content validity index protocol developed by Sjöberg et al. [10] was developed to cover aspects of technique considered to be associated with risk in weightlifting, both acute and by overuse. If these aspects can be detected and communicated automatically with the help of human pose estimation, action recognition, vector calculations and a standard RGB camera, people could get an accessible and easy-to-use tool to guide them in an injury-free training experience.

To achieve this goal we will use three existing state-of-the-art 2D human pose estimation technologies. OpenPose [11], AlphaPose [12] and WrnchAI [13].

It is important to note that this thesis aims at detecting technique aspects in weightlifting associated with risk and not necessarily perfecting the execution on a given weightlifting exercise. The absence of a detected risk does not mean the technique are not flawed, but that the risk for injuries is minimized in regard to aspects chosen from the protocols.

Research Question: To which extent can 2D human pose estimation be used as a tool to give valuable feedback on weight training technique to minimize risk of injuries?

In order to try and answer this question, the following main tasks were set:

- Gain insight in different state-of-the-art approaches for human pose estimation and pick several candidates to evaluate on.
- Explore aspects of technique in weightlifting considered to have a high risk of injury and pick the best features with respect to technique variations and body composition to evaluate on.
- Produce exercise videos where the chosen technique aspects are present as well as videos where none technique aspects are present. Then generate datasets to be used for testing and evaluation by running the videos on the human pose estimation systems.
- Develop a system to detect which exercise is being performed by the subject and from which angle the video is filmed, so that technique aspects from that particular exercise and view can be automatically tested for.
- Develop universal formulas with a high likelihood of detecting technique aspects associated with risk for the common user.
- Analyze the findings and compare the different pose estimators against each other. Evaluate the systems ability to recognize filming angle and exercise, as well as its ability to detect individual technique errors.

The reminder of this thesis will go through related work in Chapter 2, the system and its implementations in Chapter 3, evaluation methods and results in Chapter 4, discussion of findings and limitations in Chapter 5 and conclusion and future work in Chapter 6.

Chapter 2

Related Work

Finding state-of-the-art candidates that are able to produce a high accuracy pose estimation output is important to give us correct data points to perform calculations on, and thus lays the foundation for our work. Equally important is it to do research that covers weightlifting patterns and associated risk, so that the calculations can give feedback that is meaningful to the user. For this reason, a deep dive into both fields is necessary to answer the research question precisely.

This chapter will first go through the realm of human pose estimation and look at central concepts, its development and innovations, state-of-the art systems and important dataset for training and evaluation. It will then take a look at action and gesture recognition and related technologies within both areas. Then it will look at weight training, its benefits, injury epidemiology and technique aspects related to risk of injury. Lastly it will present applications of technology in sports and weight training. In particular, visual computing and human pose estimation applications.

2.1 Human Pose Estimation

Human pose estimation is a computer vision task that detects and track the location of human joints and pose of a person from an image or a video. This is most commonly done by locating the subject, then detecting keypoints and finally connecting them to the corrosponding subjects limbs, though the order of these steps vary. Keypoints represent major joints in the human body that are useful when describing an action or pose. These keypoints play an important role in understanding the activity being done by the individual. They also play an important role in mapping human body expression over to applications such as robotics or animation.

A human pose estimation system may produce either (x,y) coordinates for a two-dimensional representation or an additional coordinate, z to make the representation three-dimensional. Though 3D keypoints are more favorable, they are also more computationally and infrastructure demanding in the form of multiple or more advanced cameras. 3D pose estimation is also difficult due to the fact that datasets used for training and testing is built using motion capture systems, which are suitable only in a controlled indoor environment, therefore not suitable for wild applications. Nevertheless, one can estimate 3D coordinates using a single RGB camera [14, 15, 16, 17], though the keypoint accuracy will suffer greatly compared to 2D solutions, with a mean euclidean distance error ranging from 40mm to over 100mm. This paper will focus only on (x,y) coordinates as it is the most reliable way to generate accurate keypoint coordinates using only a single RGB camera.

Human bodies are innately hard to locate and perform pose estimation on for several reasons. One is that the human body is dynamic, and we may perform a variety of poses that a model will have to take into account. Humans also express themselves very differently by what type of clothes they wear. This may have a considerable effect on the performance of human pose estimation models by making it harder to pinpoint keypoints to their corresponding non-visible joints. Also, lighting, which generally affects computer vision problems, may lessen the accuracy of the keypoints produced. Out in the "wild", unconstrained background contexts produce more unpredictable results than in an indoor controlled environment.

Human pose estimation models may also be categorized into single or multi pose estimation models. Multi pose estimation is considerably more difficult as it demands the system to detect and differentiate every human. Overlapping of each individual may also lead to false keypoints.

The human body is dynamic and may express itself in many different ways, this makes human pose estimation very useful as input data for applications such as animation [18], gaming, robotics [19] and augmented reality [20].

Body movements can be hard to understand and human pose estimation may be used as tool to understand how and why we move the way we do. In sports it is crucial to analyze how we move in order to improve our techniques and avoid injury. Multi human pose estimation is a useful tool to generate data that can be analyzed in team sports [21, 22, 23].

2.1.1 Human Pose Estimation Methods

Methods for estimating keypoint location on a 2D image has changed drastically over the last decades. Conventional methods have long been proven useful, but have since the past decade been outperformed by methods incorporating deep learning methods. A brief overview of the most important conventional and modern methods are laid out in the following subsections.

Conventional Methods

Human pose estimation is an important research area in computer vision showing great potential in several applications. Pictorial structures, developed by Fishclet and Elschlager [24] led to the first major breakthroughs in articulating human pose estimation using RBG images. This framework is based on a statistical model of objects, that enable recognition of the objects and their connected counterparts. This applies very nicely to human pose estimation, with joints being the objects in mind. Though this method showed promising results, the main flaw was that images were not used as part of the pose model itself.

Deep Learning

One of the major hurdles in computer vision tasks is the variety of angles and scenes that an image consists of. The results produced by classical methods varies greatly from image to image, making them poor at adapting to new environments. Deep learning has shown to outperform all previous state-of-the-art methods in computer vision. Humans have a talent for recognizing and extracting information from images, by using supervised machine learning and deep neural networks one can mimic human brain to do the same.

Top-down vs Bottom-up

Two popular approaches to multi pose estimation using deep neural networks are top-down [25, 26, 27, 28, 29] and bottom-up [11, 30, 31, 32, 33]. The top-down approach essentially bottles down to performing object detection to find a bounding box containing a person in an image, followed by estimating the pose in each of these boxes. While being a viable solution, it suffers from poor performance due to the need of running pose estimation for every person found in the image. The performance is directly correlated with the number of people in the scene, thus lowering the systems performance. Also, if the object detection step fails, it will run bad results through the pipeline. Since top-down approaches makes use of an object detector, one can choose a huge variety of existing object detection models such as YOLOv3 [34] and SSD [35] or create a new one such as HRNet [27]. This makes the top-down approach flexible by letting developers tune the speed and accuracy of their pose estimation models to their needs. The other approach, bottom-up, consists of identifying and localizing all the key points in an image and then connecting them into the individual. Starting with the smallest cases and combining them into a general representation of the human pose.

Encoder-Decoder Architecture

Most deep learning architecture for 2D human pose estimation start with an encoder that uses RGB images as input and extracts features using multiple convolutions. Some neural network models, such as mask-RCNN [25], use an encode-decoder architecture, where the output from the encoder is directly fed into a decoder. Which then produces a heatmap that represent the probability of where the keypoints may be located. The exact keypoints may then be located by selecting the keypoints from the heatmap with the highest likelihood of being the correct one. The downside to this approach is that it may result in a low-resolution output, which in turn is used to create the high-resolution representational keypoints [27]. Using higher resolution images as input may alleviate the problem, but will hurt the performance.

2.1.2 Datasets & Keypoints

There exists several datasets that are used in training and evaluation of human pose estimation systems. Some of them vary in terms of number of keypoints and what they correspond to in the human body. This section will briefly mention a few important datasets.

COCO [36], short for Common objects in context, is a large dataset of labeled objects, first presented in a paper in 2014 and used to aid computer recognition systems in training and testing. The dataset has been an important driver for evaluating computer vision systems and a motivating base for competition among professionals and hobbyists. The researchers proposed that in order to build systems that solve computer vision tasks and be effective out in the wild, the training images needed to represent a diverse background context.

One major flaw with the COCO human keypoint dataset, is that it lacks sufficient keypoints for the feet. Without foot coordinates, it is hard to say how the subjects interact with the floor. An estimation has to be made in the case of collision detection with the floor or other applications has to be applied, which are often prone to errors. With the release of OpenPose, they included annotated foot keypoints, which were a subset of the COCO dataset, consisting of 14K images from the training set and 545 images from the validation set. This lead to a total of 25 keypoints produced by OpenPose and has showed to improve the overall performance of the system.

Instead of detecting keypoints that correspond to human limbs, a research team presented a study in 2018 [37], along with their dense pose estimation system, a dataset of annotated pixels that correspond to the 3D surface of that individual. This dataset consists of 50K COCO images that are manually annotated to describe the image-to-surface data. The dataset enables a more accurate mapping of RGB pixels to a semantic 3D object representation.

Until 2017, most pose estimation dataset did not include tracking of multiple people over video. This made it hard to evaluate the tracking capabilities of human pose estimation systems. In a paper published in 2017 a dataset named PoseTrack [38] was proposed that contained over 150,000 annotated poses including tracking. By using the VATIC Tool [39] the research team was able to effectively annotate a total of 15 keypoints for each visible individual in each image.

2.1.3 Human Pose Estimation Systems

The realm of human pose estimation is constantly in motion and new technologies are introduced every year. Ranging from realtime bottom-up models to commercial closed source and dense pose estimation systems. Thus making it one of the most interesting computer vision fields, with a jungle of technologies to explore. Here we will present a few handpicked state-of-the-art options for human pose estimation.

DeepPose

In 2014, Alexander Toshev and Christian Szegedy, published a paper on DeepPose, a solution to pose estimation using deep neural networks [8]. This model achieved top results on several datasets, such as FLIC, Buffy and LSP. The paper was in many ways a turning point for solving the pose estimation problem and outperformed many more classical approaches.

OpenPose

OpenPose, developed in 2017 by Zhe Cao et al. [11] was the first open source realtime bottom-up multi pose estimation that uses both body and foot detector. They later in 2019 made improvements to the model which lead to an increase in accuracy at a shorter runtime. However, their license restricts the use of their code in sports activities and any commercially use without paying a royalty fee.

HRNet

Several existing human pose estimation models try to apply the output to a high resolution representation, despite the fact that output being produced by a high to low resolution network. A study done in 2019 showed that preserving the high resolution representation throughout the model will result in much more accurate keypoints [27]. The pose estimation method presented in the paper is a top-down model that achieved the highest score on the COCO test-dev dataset at the time of publication.

WrnchAI

WrnchAI [40] is a commercialized human pose estimation tool that achieves some higher precision than OpenPose for small images, but at triple speed [13]. It's closed source thereby isolating future improvements to accuracy, speed and availability to WrnchAI employees themselves. WrnchAI also tries to predict key point for occluded parts of the body which can be to great help when one part of the body covers the other.

AlphaPose

AlphaPose [29] is an open source top-down based multi-person pose estimation system. When it was introduced in a paper from 2016 the researchers proposed that top-down methods suffer from imperfect human boundary box detection leading to redundant boxes and inaccurate bounding box coordinates. This in turn leads to keypoint locations that are inaccurate. To address this innate problem with top-down approaches the research team developed a regional multi-person pose estimation (RMPE) framework to increase the accuracy of keypoints by limiting the bounding box error.

DensePose

DensePose, developed by Güler, Rıza Alp and Neverova, tries to map "all human pixels of an RGB image to the 3D surface of the human body." [37]. Along with it, they introduced the DensePose-COCO with 50k COCO images manually mapped to the surface of the 3D model. They present a DensePose-RCNN, a variant of Mask-RCNN where each pixel is first mapped to a specific body part before deciding on what part of the 2D plane of that body part the pixel corresponds to. This technology provides promising results for tasks in future applications like graphics and augmented reality.

2.2 Action & Gesture Recognition

Human activities can be divided into the four following sub categories based on complexity and keypoints active in the movement: gesture, action, interaction and group activities [41]. The first two only take into account movements done by a single person and is the basis for two important and interesting fields in computer vision, human action recognition and human gesture recognition. The tasks involves recognising actions or movements based on a series of observations and has been successfully applied to applications such as surveillance, animation, gaming and sign language translation. The distinction between the two is usually in how much of the body they track. Gesture recognition is only concerned with some specific parts of the body like the face or hand whereas action recognition usually tracks the whole body at all times. However, the two applications have many similarities and can be researched together or achieved using similar approaches [42, 43].

2.2.1 Action Recognition Approaches

Human action recognition or activity recognition are used interchangeably in the community and refer to the same task. The task of detecting an activity based on data from one or more sensors. These sensors can be cameras, wearable sensors or sensors in the environment itself. The traditional classification is to distinguish between sensor-based activity recognition and vision-based activity recognition. Where the latter only uses a camera to capture the information about movements, the first can use other forms of sensors to capture the action as well. Models can be built using two methods and are therefore often divided into data-driven and knowledge-driven activity recognition as well [44].

Sensor Based

Hussain, Sheng and Zhang divide the sensor-based approach into three distinct sub-fields wearable, object-tagged and dense sensing in their survey of sensor based approaches [45]. Here object-tagged refer to a device bound sensor and dense sensing to a device free or environment sensor. The latter being a popular research area in recent years for its device free approach, where RFID is often seen as a popular choice of technology.

Vision Based

Vision based approaches using only a camera as sensor provides exciting opportunities in areas such as surveillance, human–computer interaction and security. But this approach also comes with its own set of real life challenges caused by uncontrolled environments. Low quality data, inter-class similarity and intra-class variability, low quality videos, camera motions and insufficient data are some challenges vision based action recognition faces [46].

The task of classifying between actions can be achieved by using multiple methods. Template-based approaches where extracted data gets compared to existing templates is a common procedure to measure the similarity. Template matching and dynamic time warping (DTW) are two popular examples of this. Generative models such as hidden markov models (HMM) and dynamic bayesian network, or discriminative models such as supported vector machines (SVMs) and conditional random fields (CRFs) are common alternative choice of implementation. Lastly, deep learning architectures have emerged as a popular choice where especially convolutional neural network (CNNs) have showed promising results [47].

2.2.2 Gesture Recognition Approaches

Gesture Recognition is the task of recognizing expressions of motion from distinct body parts, usually the arm, hand, face or head. The application of this technology has been popular in areas such as sign language translation, robotics, virtual reality and surveillance [48]. However, the task are challenging for a number of reasons including different environmental surroundings such as lightning diversity and complex background and diverse training data resulting in small or insufficient data sets [49, 50]. The area is mainly divided into two gesture recognition system, device-based and vision-based. The vision-based approach has emerged as the most popular choice, with big developments in visual computing and deep learning technologies the last decades.

Vision Based

This method, as with vision-based action recognition, concern itself with recognising movement patterns based only on data from a camera as sensor, either as a single image or an image sequence. Multiple approaches have been used for achieving vision-based gesture recognition. Model based approaches like kinematic models, view based, low level feature based and template based approaches such as dynamic time warping are some of the most popular approaches [50, 51].

2.2.3 Dynamic Time Warping

A popular method used in both action and gesture recognition are the template-based method dynamic time warping. This is a distance function for time series with possibly different progress rates. The goal is to find the optimal alignment of two time series and the method can be used for measuring similarity or doing classifications on datasets. This method is popular in speech recognition, but has also been applied to applications such as robotics and data mining [52].

The method has a quadratic time complexity, limiting its performance on smaller datasets. However, FastDTW [53] is an approximation of the dynamic time warping method which present an approach with linear time and space complexity. The method avoids the brute force dynamic programming approach and finds a near optimal warping path between two time series.

As mentioned, dynamic time warping has been a popular tool in action and gesture recognition. Recognising simple actions using pose estimation [54], applying dynamic time warping with skeleton data for gesture recognition [55] and a differential evolution approach to optimize weights in dynamic time warping [56] have all yielded good results. Schneider et al. presented a method that uses dynamic time warping on RGB image sequences. The processing pipeline included normalization, smoothing and dimension selection, along with dynamic time warping and pose estimation to classify gestures [57]. The method showed promising result when used i collaboration with a k-nearest neighbour classifier.

2.3 Weight Training and Injuries

To effectively detect weightlifting errors it is important to understand what a proper technique consists of. Human bodies in regards to body composition and functionality may vary greatly from person to person. There is also a disagreement in what defines a proper posture when lifting, due to the difficulty in measuring the biomechanics. This makes it an interesting topic to investigate, but also harder to gather valuable information when there is much disagreement among experts in the field.

One aspect however, has a major agreement in the training community. Which are the many health benefits that training and physical activity provides. In a study by Darren E.R. Warburton et al. they evaluated current literature and found a clear correlation between physical activity and reduced risk of chronic diseases and premature death [2]. Another literature review by Frank Penedo and Jason Dahn showed much of the same physiological results, but also found that training provides higher quality of life and better mood states [3]. One especially interesting finding from Darren E.R. Warburton et al. is that the groups that have the most to gain from physical activity are the ones that are the least fit. Hence, also has the least experience with training and are in need of learning proper technique and form when they start lifiting.

Strength training in particular has showed to have a clear effect on muscle size regardless of gender or age [58]. And thus indicates that strength training is beneficial for the general population including all ages and genders. A review performed by Rebecca Seguin and Miriam E. Nelson looked at previous work done on strength training for older adults. The results showed major strength gains, fewer injury related falls, better endurance and even higher bone density [59]. Work has even been done on strength training for children and adolescents and demonstrates that also young athletes gain advantages from performing strength training without any higher risk than older athletes [60, 61]. This may contributes to more new people, both young and old, with a desire to begin their weight training journey.

But the hard truth is that strength training, although its many benefits, does not come without any risk. The injury rates may be low compared to other sports similar to American football or boxing, but a review of the epidemiology of injuries in weight training shows that injuries also occur regularly in different weight lifting activities [5]. Mark E Lavallee and Tucker Balam take this further and shows all injuries, both acute and by overuse, related to different weight lifting approaches [62]. Improper movements of joints, loss of form with heavy weight and wrongful repeated placed stress on tissue are all seen as a recurrently causes for injuries. Strains, tendinitis, and sprains were found to be the most common types of injuries.

The risk of injuries was found to be highest when free weights were involved and used aggressively, even though injuries also occurred when using weight machines [63]. This makes it interesting to look at common free weight exercises, such as squat and deadlift, where heavy weights also are involved. The community agrees that good coaching on correct technique is the most important factor to minimize the risk of injuries.

In 2018 an article on evaluating lifting technique in the powerlifting squat and deadlift using content validity index and reliability was published [10]. The paper consists of powerlifting experts doing a review of literature and reaching a consensus of lifting risks in regards to deadlifting and squatting. The aspects where then rated related to risk of injury and given given a content validity index score. The final result where 17 aspects of the squat technique and 10 aspects of the deadlift technique with a high association with risk of injuries. They state the following on the protocols created:

"The protocols, formed in this study, will provide evidence-based recommendations on safe lifting technique for coaches and strength practitioners' to use to make relevant assessments and instructions."

This provides a great basis for selecting aspects related to risk of injuries to evaluate on. By using features that are heavily agreed on, the solution will have support for its findings and all recommendations on technique changes will likely have a positive effect on the risk of injuries for the athlete.

2.4 Pose Estimation in Exercise Activities

Technological assistance is becoming a popular tool in sport activities and strength training to analyze athletes performance, technique and movements in different situations. Human pose estimation with its ability to track human joints and limbs have a lot of potential to gather useful information about athletes and to provide feedback on their performance. With the prediction accuracy of pose estimation in continuously development, the possible applications of the technology have become many.

Applying deep learning to improve performance in the fitness industry is nothing new. Artificial intelligence has already been applied to give analytic feedback on performance in sport like basketball [64]. Human pose estimation has also been used to identify correct movements of a given exercise using OpenPose, machine learning and vector geometry [65]. This proved that promising results are possible when using common pose estimation models with few or none tweaks.

The option of using motion capture suits is another way to yield accurate results, making it easier to evaluate weight lifting technique. Unfortunately these suits cost at least 2495$ [66], making them inaccessible to use as an evaluation tool for the general public.

2.4.1 RGBD Camera Applications

The first camera application used to analyze training activities was the use of depth cameras to track and analyze body movements. The Microsoft Kinect consisting of a RGB camera and a depth sensor was a popular choice because of its consumer friendly technology and price tag. A study done by Š. Obdržálek et al. measured the accuracy of the Kinect pose estimation in coaching of elderly [67]. They compared the technology to more expensive motion capture systems and presented the Microsoft Kinect as a low cost alternative. The Kinect was found to be useful in given scenarios, but the variability of the implementation was high, thus making it more helpful in assessing general movement trends than precisely estimate body positions.

Other research done by Joe Sarsfield et al. showed similar results [68]. Their goal was to assess if the Microsoft Kinect could be used as a supervision technology in rehabilitation applications. They found the technology to be mostly inadequate for this application, due to variable performance. Problems with jitter and inaccurate tracking made it hard to assess correctly. Even a silhouette-based approach has been tested [69], but also here was the error rate too high to actually give valuable feedback to the users.

A system using topological skeleton generation to assist self-training [70], later develop further as a yoga-training system [71] showed promising result using the Kinect camera. The latter research was able to use posture analyzing to provide posture rectification instructions to the users for twelve different yoga poses. Showing that this might be a way to implement feedback in self-training systems.

2.4.2 RGB Camera Applications

Recent research has seen some promising application of human pose estimation to assess in training and provide relevant feedback to the user. In 2019 H. Xie, A. Watatani and K. Miyata used a normal web camera to give visual feedback on core training [72]. OpenPose was used in combination with human mesh recovery methods to create a 3D model of the user. The given model was compared to a SMPL target pose model and feedback was then given to the user based on the comparison. The solution was found to be helpful for the users to effectively perform correct core training.

Another study from 2019 by Jiaqi Zou et al. aimed at creating a full fitness trainer system that also give feedback to the user based on human pose estimation technologies [73]. The system recognize the movement the user is doing and compares it with a standardized action to give correction feedback to the user. The solution was found to have good influence on accuracy of the movement, thus making the users exercise movements better.

In a study from 2018 a team of researchers presented GymCam [74], a software that uses images from a training studio to recognize which exercise the subjects in the image are performing and how many repetitions. The software proved to be promising by detecting up to 17 exercise types with an accuracy of about 80.6%.

Other approaches involving human pose estimation in combination with vector geometry has been proposed. Pose Trainer [65] by Steven Chen and Richard Yang suggested a solution where movement of skeleton points either indicated wrongful movements or correct performance of four movements: biceps curl, front raise, shoulder shrug and shoulder press. The solution showed good precision at detecting error for most exercises and present a promising angle to investigate further.

2.5 Opportunities in the Research Field

The substantial improvements made to Human Pose Estimation systems over the last decade creates interesting opportunities in new and beneficial applications. It is important to find the human pose estimation systems that best solve a specific problem, in this thesis, detecting weight lifting aspects. Comparing human pose estimation systems will help other researchers and developers make more informed choices when building their applications.

The fitness industry has barely scratched the surface with regards to what might be possible using data output from computer vision. Software such as GymCam and Pose Trainer demonstrate that computer vision and human pose estimation systems may be valuable in giving users feedback. By creating a system that reaps the benefits of pose estimation systems, one is able to give valuable feedback to the user. Researching weight lifting feedback systems that uses video as input can therefore help answer if this technology is mature enough to be used in the fitness domain.

Further, the main research topic in this area has been on Human Pose Estimation accompanied with depth cameras or multiple sensors to get information on the three-dimensional plane. Other human pose estimation research have even used different techniques to transform the two-dimensional pose information into a 3D model. Thus leaving much room to investigate how 2D human pose estimation alone, can be used to assess and analyze movement patterns in fitness and weight training.

In addition, much of the research only distinguishes between correct and incorrect executions of an exercise or movement. Not taking into account exactly what the subject is doing wrong or which technique aspects that makes the technique suffer. Knowing what the subject is doing wrong is an important aspect to be able to give valuable feedback to the user. As well as giving them information that they can actually use to improve their technique and minimize risk of injuries.

Chapter 3

Methodology

This chapter will first go through the system architecture and present all the unique components that make up the system thereby familiarizing the reader with the system as a whole.

Then, a discussion of the research done early on to pick favorable human pose estimation systems, strength exercises and technique aspects is presented. These sections will first discuss the decisions made when selecting human pose estimation systems, why the chosen candidates were picked and what makes these candidates interesting. Then a necessary prerequisite about weight training in general and the reasoning behind exercise and technique selections will be proposed.

After introducing an overview of the system and related prerequisites for human pose estimation and strength exercises, the thesis continues by presenting the solution itself. First the video generation method and its resulting dataset is described. Then finally, each of the subsystems are described in great detail in their own section. This includes, pose estimation extraction tasks, action recognition and technique analysis. All related methods and implementations for each of the subsystem will be presented and discussed extensively.

3.1 System Architecture

The system as a whole takes an exercise video from the user as input and outputs a table of detected technique issues for the given video. However, before the final result is presented to the user, the data has to be processed by multiple components within the system. The overall process will be presented shortly before each component will be described further in their own separate subsection.

The input video is first passed on to the **Pose Extraction System(3.1.1)** where it is processed by either OpenPose, AlphaPose or WrnchAI. For this thesis each video is run through all of the systems to compare their individually ability to detect technique issues in weight training. The data from the unique human pose estimation systems are then processed and passed on to the document database.

Then the **Action Recognition System(3.1.2)** extract the keypoints from the database and run them through the classification algorithm to detect the performed exercise and filming angle. The exercise is classified by using dynamic time warping along with a k-nearest neighbour algorithm. The result of the classification is then stored back to the document database with a reference to the related keypoint dataset. Lastly the **Technique Evaluation System(3.1.3)** retrieve all related data from the database and run specific vector calculations based on the predicted exercise and detection angle. The resulting technique analysis is then stored in the database and is ready to be presented to the user along with the detected exercise and filming angle. An overview of the full system architecture is shown in Figure 3.1. The data flow between different sub-systems is presented in subsection 3.1.4.

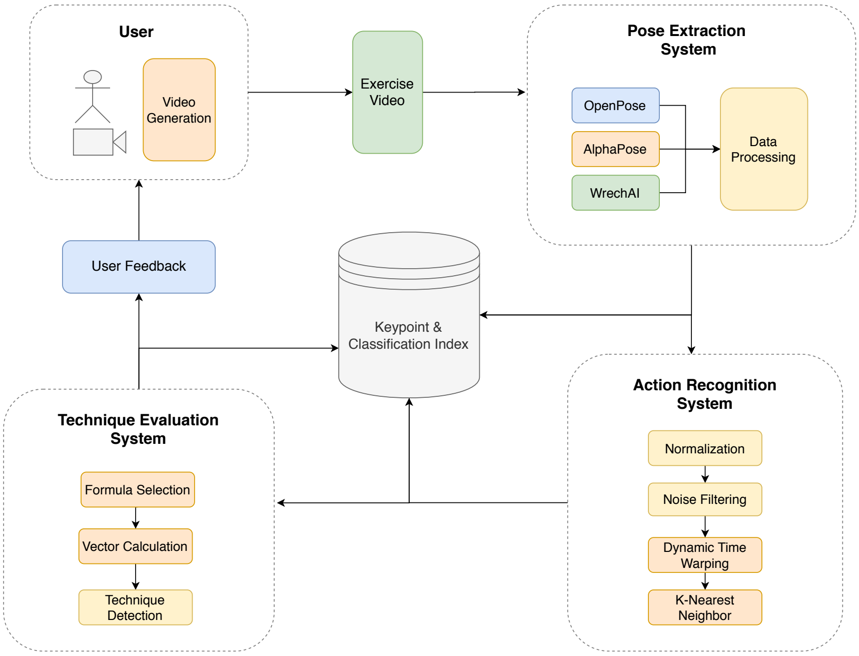


Figure 3.1: System Architecture

3.1.1 Pose Extraction System

The input to this part of the system is an raw unprocessed and unfiltered exercise video. The preconditions for the input video and the video format is described in great detail in Section 3.4. The exercise video is first processed by the three implemented human pose estimation systems OpenPose, AlphaPose and WrnchAI. The resulting dataset is then stripped of unnecessary data, transformed to an universal format and filtered for inaccurate estimations. The resulting keypoints are then indexed and stored in the database. The pipeline for the **Pose Extraction System** is shown in Figure 3.2. The subsystem implementation and all its details are described thoroughly in Section 3.5.

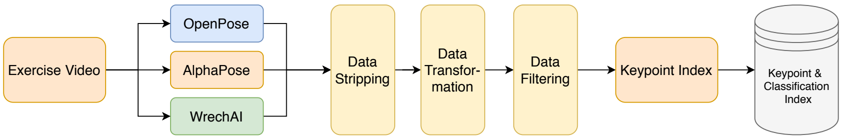


Figure 3.2: Pose Extraction System: Pipeline

3.1.2 Action Recognition System

This system start by extracting all processed keypoints from the document database. Further each keypoint is processed as a time series to detect both the filming angle and the performed exercise. Data normalization and noise filtering is applied, before the filming angle and exercise detection are performed separately. A *dynamic time warping* algorithm is used to compare similarity between time series and classify each sequence using a *k-nearest neighbors* algorithm. The pipeline for the **Action Recognition System** is shown in Figure 3.3. Detailed information about the implementation and technologies used are presented later on in Section 3.6.

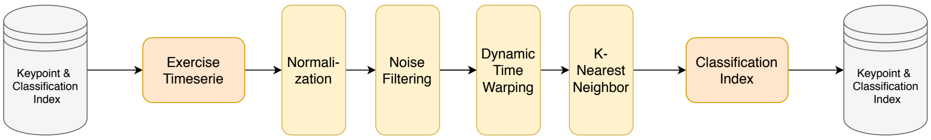


Figure 3.3: Action Recognition System: Pipeline

3.1.3 Technique Evaluation System

This system initially extracts both the processed keypoints and the related prediction of exercise and detection angle generated from the previously presented systems. Then it selects a subset of relevant vector formulas based on the predicted exercise and filming angle. All selected vector formulas is then calculated to detect if any technique issues is present in the given pose estimation dataset. The output is a list containing all detected technique issues for the dataset. If none are detected, an empty list is returned. The pipeline for the **Technique Evaluation System** is shown in Figure 3.4. A complete explanation of the systems and vector formulas are presented in Section 3.7.

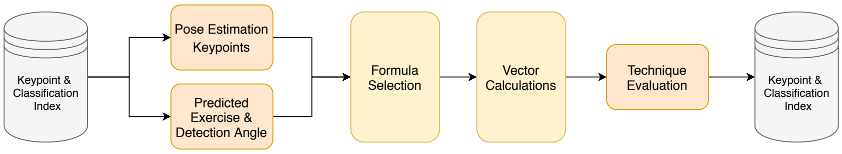

Keypoint & Classification Index
Pose Estimation Keypoints
Predicted Exercise & Detection Angle
Formula Selection
Vector Calculations
Technique Evaluation
Keypoint & Classification Index

Figure 3.4: Technique Evaluation System: Pipeline

3.1.4 System Data Flow

The document database is the link between the different components within the system and is responsible for information flow across distinct sub-systems. Extracted keypoints are initially rendered from the **Pose Extraction System** and stored in the document database. The **Action Recognition System** then extract the keypoints and use them to predict the angle the keypoints are filmed from and the exercise they represent. This information is then stored to the document database with a reference to the related dataset already stored. Consecutively the **Technique Evaluation System** extracts both the keypoint information from the **Pose Extraction System** and the predictions from the **Action Recognition System** to evaluate for different technique aspects on dataset. The overview of data flow between different sub-systems and the document database is shown in Figure 3.5.

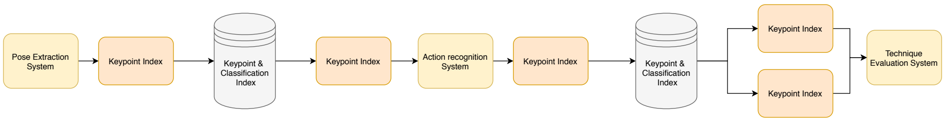

Pose Extraction System
Keypoint Index
Keypoint & Classification Index
Keypoint Index
Action recognition System
Keypoint Index
Keypoint & Classification Index
Keypoint Index
Keypoint Index
Technique Evaluation System

Figure 3.5: System Data Flow

3.1.5 Implemented Technologies & Libraries

In creation of these systems, multiple libraries were used and different technologies implemented. The most important ones are described briefly below. All technologies related to human pose estimation will be presented in Section 3.2.

MongoDB

MongoDB is a document-based database that is suitable for storing JSON data [75]. It provides an expressive query language that enables fast and efficient queries. Due the large amount of data generated by each pose estimation system MongoDB was a fitting choice.

Docker

Docker is a virtualization software on the OS-level that makes it easier to create, deploy and run applications [76]. This makes it more convenient for other developers to contribute to the project both now and in the future, regardless of their operating system or system configuration.

Matplotlib

Matplotlib is a library for creating visualization of data mainly through graphs [77]. This enables users to understand their data from an overview. In this project the graphs were valuable in understanding and detecting patterns between different lifting aspects.

Statsmodels

Statsmodels is a python library with many classes and functions for a diverse number of statistical models [78]. This was a valuable tool when working with time series data in the action recognition process.

NumPy

NumPy is a popular library for handling and processing arrays in Python [79]. The library is fast, efficient and has good support for different dimensional arrays. Thus, being a essential tool when working with big arrays and matrices like human pose estimation data.

3.2 Pose Estimation System Selection

To be able to answer the research question precisely, the first major decision to be taken was the selection of human pose estimation candidates to be used for the solution. With new candidates presented every year, the options are many and the features to consider even more. Speed, accuracy, body models and availability are all important aspects to consider when picking pose estimation systems and will be discussed further in this section.

Research into the realm of human pose estimation revealed some clear state-of-the-art candidates. Some being well tested and applied in multiple research and others with less exploration. But all stating to be among the best pose estimation technologies available. Here we will shortly introduce the main prospects and their preeminent benefits.

- **DensePose:** Published by Facebook in 2019 and aims at mapping all human pixels from a RBG Image to a 3D model. Unique of its kind and provides opportunities never examined before. Open source.

- **OpenPose:** Released as an open-source project in 2017. Since then, it has become the most popular human pose estimation library available. Big community, great documentation and well tested.
- **HRNet:** A recent project released in 2019 that maintains a high resolution representation and has so far outperformed all existing models on keypoint detection earlier tested on the COCO dataset.
- **WrnchAI:** Is the only closed source software on the list. However, third party testing against OpenPose revealed more than 2x faster processing speed, significantly smaller model sizes and lower GPU RAM requirement.
- **AlphaPose:** open-source software released in 2018 and receiving further developing in 2020. Scores remarkably better than OpenPose for several tests on the COCO and MPII datasets.

3.2.1 Accuracy

Accuracy is the highest priority when it comes to choosing a model that fulfills the research question of this study. Without data that realistically and correctly captures the core movement of the exercise, the evaluation will be ineffective. This solution will only use a single RGB camera which limits us to 2D models.

Out of all the pose estimation systems presented here, HRNet maintains the highest AP with a score of about 77.0% [27] on the COCO test-dev dataset [36]. In comparison, OpenPose scores 61.8% on the same dataset. Considering that OpenPose won the 2016 COCO keypoint challenge, this signifies a substantial improvement in accuracy for pose estimation systems. AlphaPose has also demonstrated a high score in accuracy with a 73.3 mean average precision [80] on the COCO dataset. The real accuracy score of WrnchAI is unknown as it has never been published, but they claim to achieve the same accuracy as OpenPose [13].

3.2.2 Speed

At the time of writing, the extraction of keypoints is done offline, and not in realtime. The purpose of this is to reap the benefits of models that yield the highest precision. It is also unnecessary to give realtime feedback on the exercise as it is safer to assess the technique and form errors when not performing the exercise. Due to this, speed will not be considered as an important factor, unless the speed is unreasonably low to the user. The speed of pose estimation systems do not contribute to answering our research question.

3.2.3 Keypoint Information

To fully understand each exercise movement, it is necessary to have as many well placed keypoints on the body as possible. Feet for example play an important role in understanding how well the squat and deadlift are performed, considering that the weight is always pulling you towards the ground. Each human pose estimation system may vary on which keypoints they have and how many they are trained to detect. This plays an important role in choosing the most optimal system as it determines which flaws the solution can detect.

DensePose outputs 2D keypoints that can be mapped to a 3D model. This surface data might be useful for detecting bad technique such as rounding of lower or upper back by calculating the curvature of the back. Applying this data to other technique errors, there are several keypoints for each limb to choose from, making it hard to define when the error starts. This makes the DensePose system suboptimal for detecting flaws in weightlifting.

The COCO keypoints dataset consists of 17 keypoints in total for each human, though this does not include keypoint labels on feet. HRNet and AlphaPose have primarily been tested on this dataset and it's not known how they work on the 25 keypoint dataset from OpenPose. Based on the data generated from WrnchAI it does include one extra keypoint on each tip of the feet. The OpenPose solution by default provide the most number of keypoints by including an additional 8 key points from the feet adding up to 25 keypoints in total. This makes OpenPose the best choice in regards to keypoint information.

It is important to note that while OpenPose detects the most keypoints it is still possible to train the other human pose estimation models to also detect the same. However, in this paper we will not customize or remodel any of the systems.

3.2.4 Availability

To effectively create a reliable and useful application around these human pose estimation systems, it is decisive that these systems are accessible and easy to integrate. Our five pose estimation system candidates can be categorized as open-source or closed-source. WrnchAI is closed source, which makes it inaccessible to retrain this model on other datasets that might include more keypoints. This also hinders any developer from adding or removing any neural layers to the model to tweak the performance. When using WrnchAI the application will be dependent on the developers and WrnchAI company existing.

WrnchAI does introduce simplicity by enabling the processing of RGB images on the cloud through an API. This simplifies the application development substantially by removing the task of integrating code tightly coupled. This permits simple smartphone apps connected to the internet to process their videos in a matter of minutes.

3.2.5 Conclusion

We have chosen three remaining pose estimation system candidates that will produce our 2D keypoints. These are **AlphaPose, WrnchAI** and **OpenPose**. The reason for choosing AlphaPose is that it achieves the second highest AP among the candidates while scoring high on availability and documentation. HRNet scores the highest but lacks sufficient documentation and speed to build a reliable application

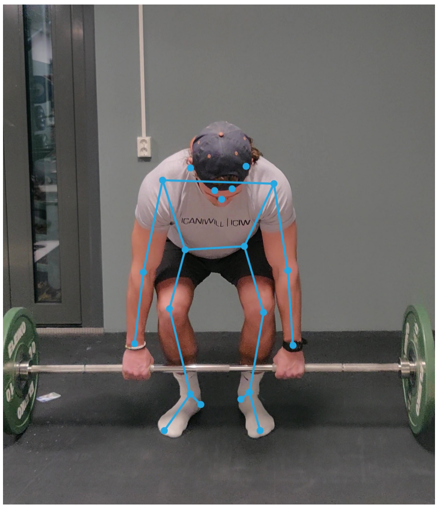


Figure 3.6: WrnchAI: Deadlift Example

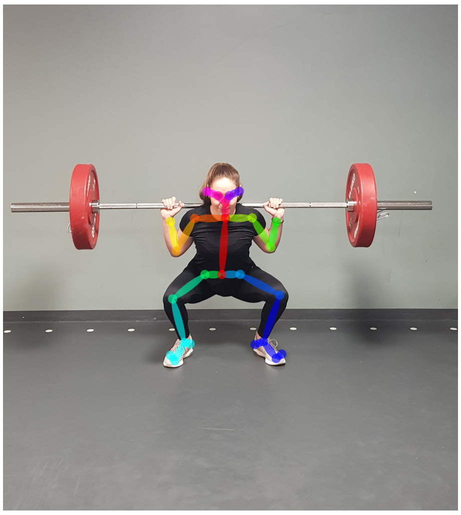


Figure 3.7: OpenPose: Squat Example

on top of it. WrnchAI represents the closed source solution out of our candidates, which makes it interesting to explore. DensePose introduces complexity due to the number of keypoints, therefore it will not be used for this application. OpenPose scores the lowest in terms of accuracy, but this human pose estimation system includes a foot dataset, which might be necessary in order to the technique errors outside the scope of this thesis.

3.3 Exercise and Technique Selection

This section will present which strength exercises that will be used for the evaluation and further, which technique aspects to consider for each selected exercise. The importance of this choice can not be understated. The exercise selection lays the foundation for the systems usability and usefulness by picking the most common movements with the highest probability of risk of injury. In addition, the technique aspect chosen for each given exercise, determines both the risk of causing an serious injury and to what degree the system has an ability to detect it. Thus making the topic and discussion around it just as important as the human pose estimation selection problem.

The given selection problem can be stated precisely as two separate questions:

- Which exercises are most popular in the weight training community and simultaneously has the highest risk of injuries?
- Which technique aspect for each selected exercise has the highest association with risk of injuries and has a high probability to be detected using 2D human pose estimation?

3.3.1 Exercise Selection

There exists hundreds of different strength exercises involving both strength training machines and free weights, all with different complexity, popularity, movement pattern and weight involved. Here we will go through the decision processes in detail and the train of thought that lead to the selected exercises.

As discussed in Chapter 2, the occurrence of injuries is higher when free weights are involved and can help to narrow the exercises scope somehow by removing machine related exercises. It is worth mentioning that some strength training machine exercises like the leg, chest and shoulder press could be interesting to investigate due to the heavy weight involved. However, since the epidemiology clearly shows a higher prevalence of injuries using free weights, machine exercises were discarded as candidates.

Further, the comparison between compound exercises and isolation exercises helps to reduce the pool of relevant exercises even further. Compound exercises are multi-joint movements working several muscle groups at the same time, where isolation exercises only work one muscle at a time, such as the biceps curl and leg extension. This group of exercises usually have a strict movement pattern, reducing the number of possible errors to perform. These exercises also involve lighter weights than compound movements, reducing the risk of injuries. For this reason, compound movements are much more intriguing to investigate. By moving multiple joints concurrently, the complexity of the movement increases and incorrect motions will occur more easily, giving compound exercise a higher probability of injuries.

However, increased complexity also makes the exercises harder to perform for the general population, and thus make the exercise less applied among athletes. An example are powerlifting movements like the clean and jerk or snatch which represent some of the most complex exercises out there. For this reason, they are only performed by a small selection of skilled athletes. These exercises have a high occurrence of injuries, but the complexity makes them both hard to analyze and less popular to the general population. For this reason these exercises, regardless of the injury rate, were also discarded while more popular movements were assessed.

The popularity of an exercise is an important factor to consider to correctly select movements the general public will benefit from. By selecting exercises recommended to all ages and genders and with a wide user base, the solution could benefit as many people performing weight training as possible. Recommendations from athletes, personal trainers and experts usually includes the likes of squat, deadlift, bench press, shoulder press and hip thrust as essential to your training routine. Since these movements are compound exercises as well as being popular, they fulfill both the popularity and injury prone requirement.

After discussing the few remaining candidates, two widely applied and extensively recommended compound exercises, namely **The Squat** and **The Deadlift** were selected.

The Squat

The squat is a compound multi-joint exercise involving all the large lower body muscles in addition to the core. The exercise is performed using a weighted barbell placed on the upper part of the back. The starting position is standing with feet placed at shoulder width slightly pointing outwards and the weight comfortably placed on the back. From here the athlete creates tension by tucking the bar between their arms and the upper back and bracing the core. Then the athlete starts the descent by driving their hip backwards and moving the knees over the toes. The bottom positions is reached when the femur is parallel with the floor. From here you start to move the weight back up by pressing trough the feet using the lower body muscles, keeping the core tight. The weight should be moved as vertically as possible in both directions and be at the center of gravity at all times during the movement. The back should be straight and the feet stable to avoid injury. The starting position and in action position for the squat is shown in Figure 3.8 and 3.9.

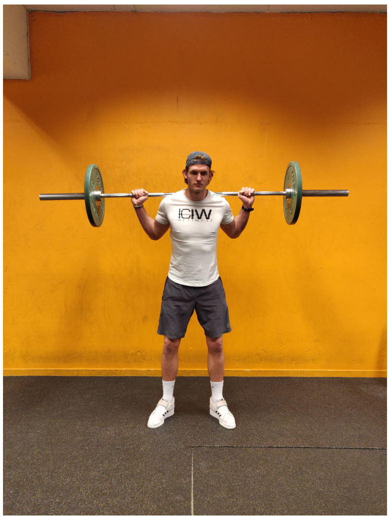


Figure 3.8: Squat: Starting Position

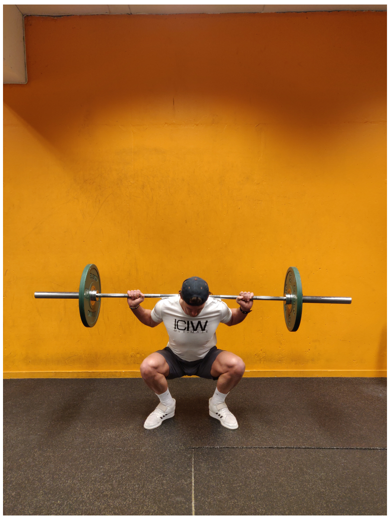


Figure 3.9: Squat: In Action Position

The Deadlift

The deadlift is a multi-joint compound exercise working the shoulder girdle all the way down to the major lower body muscles such as the gluteus maximus, hamstring and quadriceps. However, the main activation for the exercise is in the core, with the abdominals and lower back muscles used for stabilization. The movement is performed using a barbell placed on the floor in front of the athlete. The starting position is with the feet at shoulder width and the bar placed over the center of the foot. The athlete is gripping the bar at shoulder width, with the hip and ankle joints bent so the back and pelvis is kept straight. From here the athlete takes slack out of the bar by activating the lower body muscles, embracing the core and contracting the upper back muscles. Then they push through their feet moving the hip and upper body at the same pace so the bar travels straight up. The top position is reached when the athlete is standing upright with the bar in their hands at hip height. From here the bar is lowered back to the staring position resting on the ground in front of the athlete. The bar should be at the center of gravity at all times during the movement and travel as horizontally and close to the body ass possible. The back and pelvis should be kept straight throughout the exercise to avoid injuries. The starting and in action position for the deadlift is shown in Figure 3.10 and 3.11.

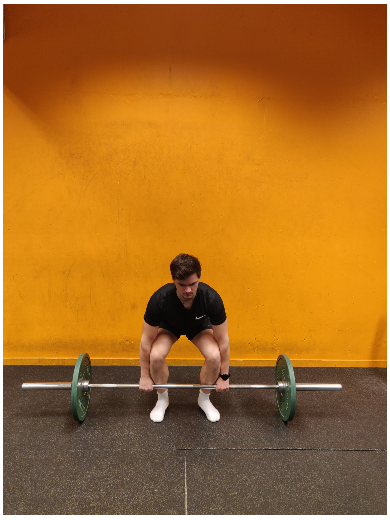


Figure 3.10: Deadlift: Starting Position

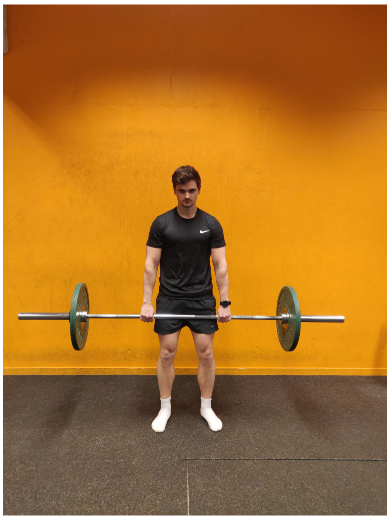


Figure 3.11: Deadlift: In Action Position

3.3.2 Technique Selection

Performing an exercise correctly is a difficult task and often requires guidance from professionals, movement pattern understanding, sufficient mobility and many hours of drilling. Many users neglect one or more of these these issues which causes the technique to suffer and in consequence increases the risk of injury. There are many different technique aspects to consider so we will look at two main factors when assessing witch issues to focus on: the risk of injury the technique issue present and the probability of detecting this risk using human pose estimation and vector calculations.

As mentioned in Related Work, Sjöberg et al. developed two protocols [10] one for the squat and one for the deadlift where aspects of lifting technique and their associated risk were presented. All 27 aspects presented in the article were hand-picked by experts and has an high agreement among powerlifters on their risk of injury and will therefore be the basis for discussion in this section. This gives us a pool of injury prone technique aspects to pick from when considering which issues it is possible to detect.

Detecting risk of injury using only selected keypoints of joints requires accurate pose estimations and deviations large enough to detect. Luckily, aspects with the highest injury risk often are the aspect with highest deviation from regular movement patterns. For this reason, many of the selected issues presented in the previously mentioned protocols are more likely to be detected. At least in the extreme to high deviation cases.

However, some cases where the deviation from the optimal pattern is small, it is hard to detect errors due to inaccuracy of the pose estimation and variation between trailing frames. Also variation resulting from holding the camera makes some issues hard to detect. Example could be one knee travelling slightly more forward than the other or the foot loosing contact with the floor at some point during the lift. Other cases that could be hard to detect is due to the simplicity of the human pose models. Cases when rounding of back and shoulders, twisting of hips and pelvis movement occurs are hard to detect without multiple keypoints located at the back and not only at the shoulders and hips.

By assessing the different technique aspects presented by the protocols and filtering away technique issues a two-dimensional representation will have major problems detecting, 9 errors are left. These are evaluated to be detectable only using human pose estimation and vector calculations. The technique aspects are presented in Table 3.1 together with the associated exercise and the view that the issue will be filmed from.

Table 3.1: Selected Technique Aspects

Number	Exercise	Technique Aspect	Detection View
1.1	Squat	The knee travel inside of the foot seen from the front.	Front
1.2	Squat	The feet are pointed inward-toward on another.	Front
1.3	Squat	Overextension of the knee in the lock-out phase.	Side
1.4	Squat	Asymmetrical rotation of the hips.	Front
1.5	Squat	Moving center of pressure to the sides as seen from the front.	Front
2.1	Deadlift	The knee travel inside of the foot seen from the front.	Front
2.2	Deadlift	Asymmetrical rotation of the hips.	Front

Table 3.1: Selected Technique Aspects

Number	Exercise	Technique Aspect	Detection View
2.3	Deadlift	Excessive arching of the lower back in the lockout of the lift instead of/in addition to hip straightening.	Side
2.4	Deadlift	Lifting with flexion in the elbow.	Front

3.4 Video Generation

This section will present the data foundation for the system. It will go through how the exercise videos were generated and different technique aspects provoked in order to get a video foundation that covers all technique issues, as well as correct execution of the exercise. Different subjects were used to cover for body variations and multiple filming angles were adopted to cover different technique aspects.

3.4.1 Prerequisites for Filming

Before starting the filming process, it was necessary to create a few rules to set the foundation for the video generation process. The results were the following requirements described more thoroughly later on:

- Four subjects should be used to cover for body variations. Two male and two female athletes.
- Each subject should perform all of the technical aspects listed, for both exercises and filming angle.
- Each subject should do each technique issue at two severity degrees. Moderate and high.
- Each subject should do one correctly performed video of each exercise from each of the specified angles.
- No other person than the athlete doing the exercise should be visible in video.
- Standard weightlifting bar and weight plates should be used to make the videos as close to reality as possible.
- Subject should evenly divide lifting in shoes and shoeless to cover for different equipment used.

The first requirement ensures that the model created can detect a wider range of body composition and is not fitted to one specific athlete. A small set of athlete that involves both genders will not cover all possible differences between athletes, but will cover enough variations to give the model some flexibility. Thus proving that the model can detect cases in the given scope in addition to related cases where the variations are somehow similar.
The second requirement gives us a sample set of technique issues to test the proposed vector formulas on. This further helps to cover for variations for each technique aspect by having diverse execution and variation from regular movement pattern.

The third requirement deals with how clearly or excessive the technique issues should be performed when an issue is provoked on purpose. An exercise with a distinct technique issue present would be easier for a system to detect, since it is further from the general movement pattern of an optimal execution. However, movement patterns that are too far away from a correct execution are less likely to occur on a regular basis, and thus will not be as useful to detect for most users. On the other hand, the bigger the deviation from an optimal pattern is, the higher the chances are for an serious injury to occur. For this reason we adopt two degrees of severity when performing an technique aspect: moderate and high.

All athletes also performed a correct movement of each exercise from the two specified angles. This is to generate a subset of correctly performed exercises that can be evaluated on the same basis as the the rest of the videos. The main purpose of this is to disclose false positives resulting of inaccurate vector calculations. As well as having a base set for reference, that should not report for any technique issues by the system.

Another prerequisite for the video generation was that the only person visible in the frame should be the one performing the exercise. This choice is not due to possibility but rather simplicity, since all the selected pose estimation systems have multi-person detection. However, ensuring that only one person is visible makes the data processing easier as well as making the pose estimation faster and the resulting dataset smaller.

The last requirements deals with the equipment used by the athlete during the video generation. Making sure to use standardized bars and weight lifting plates provide videos with the same equipment found in most training centers. The standardized weight lifting plates is also of such size that they cover some parts of the subject performing the exercise. It is important to see how the human pose estimation systems respond and behave under these circumstances, since most athletes perform these exercises using weighted barbells. The footwear requirement is simply to test if different footwear result in different outcome from the pose estimation. This is important because athletes perform exercises with everything from lifting shoes to barefoot.

3.4.2 The Filming

The filming process itself was performed at a regular crowded gym with a standard mobile phone camera, so that the preconditions would the same as for possible end-users. A big focus point for the filming was to hit the desired angles as close as possible and to hold the camera steady to avoid to much inaccuracy due to camera movement.

The two angles to be used during filming were taken from a front and a side view. Together this covers all technique issues presented in Table 3.1 and are described below:

- The Front View: Is straight in front of the subject with the athlete in the center of the screen. The whole person should be visible with some space on all edges for good measure.
- The Side View: Is on the right side of the subject and the whole athlete should be visible at the center of the screen. The angle should be such that the right side of the athletes body covers their left side.

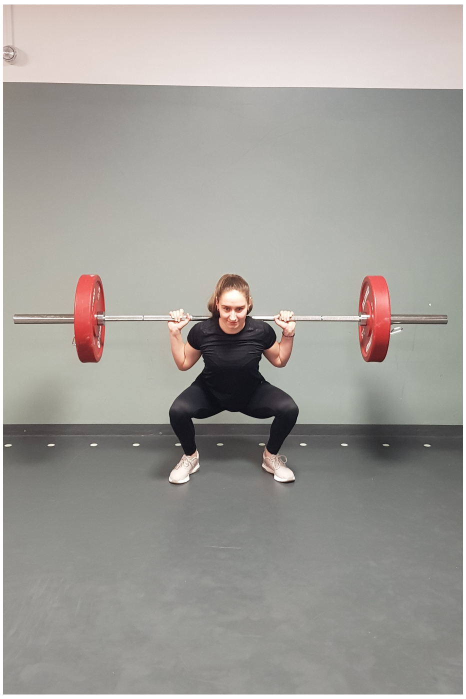


Figure 3.12: Filming Angle: Front

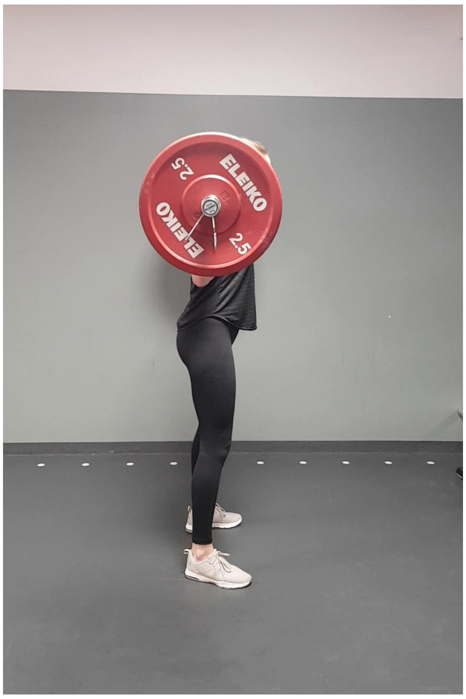


Figure 3.13: Filming Angle: Side

Each video clip is a short snippet where the subject perform the given exercise with or without some technique aspect present from one angle at a time. The video clip begin with the user in the starting position and end when the athlete is back at the starting position after performing one repetition of the exercise.

As mentioned earlier each technique aspect was filmed twice, with two degrees of severity; moderate and high. This choice was made to test the solution on different clarity to see to what extent it would able to detect errors performed by the subjects.

All the videos of the correct exercise execution were performed as precisely as possible and followed the description of the exercises from Section 3.3.1. The videos are without any of the technique issues presented in Table 3.1 earlier on.

3.4.3 Technique Provoking

To gather video samples of each single technique aspect it was necessary to provoke technique issues on purpose, to generate a sample set of each technique aspect. Here we will discuss how each aspect was provoked and to what degree it was done to clearly demonstrate that a technique issue was present. In Table 3.2, the technique aspects are described together with the execution for the different severity degrees. The numbering of entries in Table 3.2 corresponds to their associated column in Table 3.1, where each of the technique aspects were presented.

Table 3.2: Technique Execution

Number	Technique Aspect	Moderate Severity	High Severity
1.1	The knee travel inside of the foot seen from the front.	At least one knee travel inside the foot with about five centimeters.	Both knees travel inside the foot and end up close to each other.
1.2	The feet are pointed inward-toward on another.	The feet are pointing slightly inward with about 10 degrees.	The feet are pointing inward with more than 20 degrees.
1.3	Overextension of the knee in the lock-out phase.	The knees are locked out all the way and the the joint is somewhat overextended.	The knees are locked out as much as the subject are able to, the joint is overextended and the weight are pushed backwards through the knees.
1.4	Asymmetrical rotation of the hips.	The hips are slightly rotated such that the center of the hips are closer to one foot than the other.	The hips are clearly rotated such that the torso is pointing to one of the sides.

Table 3.2: Technique Execution

Number	Technique Aspect	Moderate Severity	High Severity
1.5	Moving center of pressure to the sides as seen from the front.	The hips moves closer to one leg without rotating the hips shifting the pressure slightly more to one leg.	The hips moves closer to one leg without rotating the hips shifting the pressure mostly to be on one of the legs.
2.1	The knee travel inside of the foot seen from the front.	At least one knee travel inside the foot with about five centimeters.	Both knees travel inside the foot and end up close to each other.
2.2	Asymmetrical rotation of the hips.	The hips are slightly rotated such that the center of the hips are closer to one foot than the other.	The hips are clearly rotated such that the torso is pointing to one of the sides.
2.3	Excessive arching of the lower back in the lockout of the lift instead of/in addition to hip straightening.	Lower back is arched at the top of the lift and the upper body is slightly behind the lower body seen from the side.	Lower back is arched at the top of the lift and the upper body is clearly behind the lower body seen from the side.
2.4	Lifting with flexion in the elbow.	Elbows are bent with about 10 degrees in the elbow joint at some point during the lift.	Elbows are bent with more than 20 degrees in the elbow joint during the whole lift.

Deviations

The execution of each technique aspect was followed as closely as described in table 3.2, however some subjects had problems performing some of the technique issues due to poor mobility and body composition restrictions.

The problem we encountered the most during the video generation process had to deal with aspect number 1.3. Two out of four subjects were incapable of overextending their knee in the lock out phase more than barely visible to the human eye. Because of this, only two videos were generated with the high severity instance for this aspect. The medium severity for the same aspect have videos for three out of four subject, even though the aspect is barely visible for some of the athletes.

One of the subject also had to perform the exercises without a regular weight lifting bar and standardized weight plates due to reduced access to public gyms. Instead a broomstick with some smaller radius than a regular lifting bar was used without any plates. It is not expected to affect the movement of the subject in any way since all technique aspects can be perform similarly with any regular stick. In addition to this, the other three subjects will unveil any possible issues related to weight plates covering the body. Which is the main reason behind using regular weight lifting bar and plates.

3.4.4 Resulting Video Foundation

The final video foundation consist of videos from four different subjects, two male and two female. All together there are 86 video snippets, whereas 16 are correct execution of the exercise, while the remanding videos have some technique aspect present. A detailed description of the data foundation and its composition are described below, so that it is clear what data was used to evaluate the solution.

The video foundation has the following composition:

- Eight video snippets of correct execution of the squat and eight correctly executions of the deadlift. The execution was performed by four different subjects. One snippets from each angle for both exercises per subject.
- 86 video snippets where some technique aspects from Table 3.1 are present. 18 snippets for each of the four subjects, one for each severity degree. Resulting in 8 video snippets of each technique aspect, four for each severity degree.

Specifications

Length: Videos ranging from 4 seconds to 10 seconds in length.

Quality: The videos were filmed in either 4k with 60 frames per second or in 1080x1920 pixels with 30 frames per second. All raw, unprocessed and unfiltered.

Size: The size of the videos is between 53 megabytes and 9 megabytes.

3.5 Pose Extraction System

This section will describe the *Pose Extraction System* in detail. Firstly, the human pose estimation implementation will be presented. This will describe what parameters and flags that were used to tune the human pose estimation system to get as high accuracy as possible on the dataset.

Thereafter will the data manipulation tasks performed on the human pose estimation output data be discussed. The section will go through what was done to filter away imprecise estimations, how the data was transformed to be used with the same system and how the dataset was reduced to increase its simplicity and size. The three main data manipulation techniques performed on the dataset were data stripping, data transformation and data filtering and were performed in the order they are presented.

The final output data format to be used as input to the Action Recognition System and the Technique Evaluation System will also be described in detail at the end of this section.

3.5.1 Human Pose Estimation Implementation

The speed and accuracy of existing 2D human pose estimation models vary according to the parameters that are set when training and performing inference on images. We have decided to focus on obtaining the highest accuracy possible and have chosen the parameters accordingly.

1. **AlphaPose:** AlphaPose being a top-down method needs an object detector, where we have chosen YOLOV3 due to speed and high accuracy, as well as using AlphaPose' own pose estimation model called FastPose (DUC) built using ResNet152 deep learning model. Trained on the COCO dataset, this model achieves an AP of 73.3 on this dataset, making it the most precise pose estimation system used in our solution. The –flip parameter is also turned on to maximize accuracy.

2. **OpenPose:** OpenPose uses a bottom-up approach and thus have no need for object detectors. Due to lack of hardware power the parameters used for this model are kept at default to avoid running out of memory and keep speed at the highest. This makes OpenPose the fastest at performing inference on the images.

3. **WrnchAI:** WrnchAI is closed source but has a few parameters that one can turn on and off. We chose to keep head and hands keypoints off because it is not necessary for our calculations. Other parameters such as 3D points and annotated media are turned off as well.

All the generated videos were processed by each of the human pose estimation systems, giving three different pose estimations for each video. This resulted in three different datasets, with three different data formats, since each on the systems handled and stored keypoint data differently. The next job required by the system is therefore to manipulate this data such that both the Action Recognition System and the Technique Evaluation System receive the same data, in one predefined format.

3.5.2 Data Stripping

The first order of business was to strip the files of unnecessary data to speed up the processing and keep the format minimal. The dataset reduction is a fundamental task in data processing to shrink the data load to the minimal to get rid of redundant data. This process was done to all the output material from the human pose estimation systems.

All of the human pose estimation systems being used are multi-person systems with the ability to detect several people in every frame. However, a prerequisite for the solution is that no more than one person should be visible in the frame at any moment. For this reason one can extract all information from the first person detected and disregard any multi-person information that is present in the data bundle. Some of the human pose estimation systems also provides data related to 3D estimation, which can be discarded all together since our system only focuses on 2D pose estimation. There is also a lot of metadata involving body model, frame rate, video details and bounding boxes not relevant to our solution. By removing all of the data mention above, one is left with only the most fundamental data. This includes the relative estimations for each keypoint and their associated confidence score.

3.5.3 Data Transformation

Second task at hand was to transform the data from the human pose estimation systems into the same format, for the Action Recognition System and Technique Evaluation System to receive. The main reason for this was to make sure that the output format from different human pose estimation system would be on the same predefined format and thus could be used on the same system without any discrepancies. This work involves simple data transformations tasks and is important to ensure consistency in the data. By doing so, the output data can be executed on the same program and easily be compared to one another.

The human pose estimation systems incorporates three different kind of body models to track human joints and limbs, all with a different set of keypoints. The different body models for OpenPose, AlphaPose and WrnchAi is showed below in Figure 3.14, 3.15 and 3.16 respectively.

The main transformation task was to remodel the output dataset from the different human pose estimation system so that they follow a common body model for the whole solution. This is important so that all input data to the vector calculation program is identical in structure and the same formulas can be used for all human pose estimation system. The OpenPose body model which is an extension of the COCO model, is a superset for the other two models. This means that all keypoints found in the AlphaPose and WrnchAI models, also are a part of the OpenPose model. This makes it practical to use the OpenPose model as a base and transform the rest of the data to this format.

This transformation was performed by shifting the numbering for all common keypoint to match the numbering seen in the OpenPose model. An example of a common keypoint is the right knee keypoint in both the OpenPose and the AlphaPose model. The numbering however, is different and the keypoints will be at different positions in the data sequence and therefore need to be swapped around to match

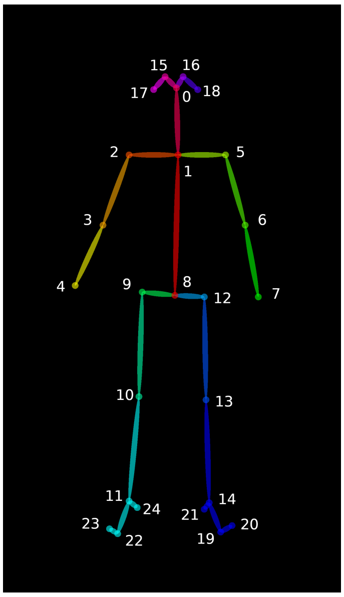


Figure 3.14: Keypoints: OpenPose

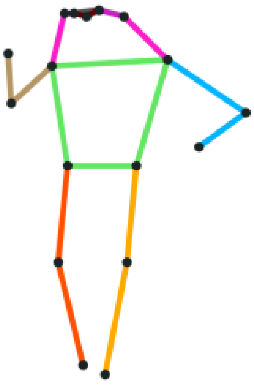


Figure 3.15: Keypoints: AlphaPose

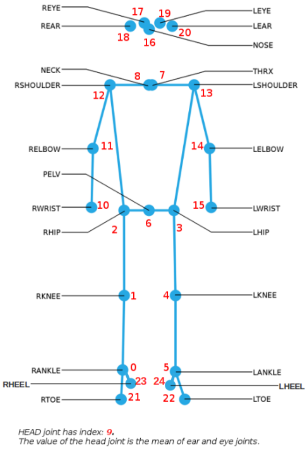


Figure 3.16: Keypoints: WrnchAI

the data sequence of the OpenPose body model. This process also requires the possibility for keypoints not present in the subset to be null. For example will the tuples for keypoint 19, 20, 21, 22, 23 and 24 all be null for all instances of the AlphaPose model after the transformation to the OpenPose model base. This is because these keypoints does not exist on the AlphaPose model at all.

Further, OpenPose produces a single json-file with keypoint for each single frame of the video while AlphaPose and wrnchAi outputs a single json-file with data from each frame merged. For this discrepancy the single file implementation was found to be the best solution when each video should be processed as a whole and result in a single output file. Another option is to treat each video as a stream of keypoints from each frame was discussed. This alternative would require a non-optimal splitting of both the AlphaPose and the WrnchAI dataset and reduce the simplicity of the data input to the other systems. The choice was thus disregarded and a single input and a single output file was settled as the format for each video.

3.5.4 Data Filtering

The final and considerably most important data processing task to be performed on the material was the data filtering. This process filter away data points with low probability to be accurate. This is done by removing points expected to be inaccurate by the human pose estimation systems itself as well as filter away trailing points with unnatural high variability. This process is absolutely crucial for the accuracy of the solution. By removing probable inaccurate estimated points, the chance of detecting technique issues that do not exists decreases as well. Thus being one of the most important steps to avoid false positives.

All the three human pose estimation system provide a confidence score of how likely each estimation are to be correct along with the data points. The simplest and most effective way of filtering to remove inaccurate estimations was to disregard keypoints with a low confidence score. Since the videos are captured in a controlled environment with only one person visible in the center of the screen they are more likely to estimate well. But to safe guard against completely wrong estimations especially from the side view, a confidence score threshold was decided on. All confidence scores with lower than 70% probability were discarded and not used in the final solution for the action recognition and technique evaluation tasks. The choice of 70% was to make sure most keypoints would pass through the filter but at the same time filter away some estimated points with insecurity. The dataset contains keypoints for every frame of the video and is therefore detailed enough to detect technique aspects even if keypoints for some frames are disregarded.

However, all rules have exceptions. Some calculations required a confidence score threshold up to 90% due to noise that might cause false positives and were therefore adjusted accordingly. The side view keypoints had a significantly lower confidence score average than the rest of the data. For this reason, it was necessary to lower the confidence score threshold down to 60% for all videos produced with a filming angle from the side..

Due to innate inaccuracies in human pose estimation systems and computer vision systems in general it is necessary to filter out keypoints that scored high on probability but had too great of a distance difference from the previous frame. These inaccuracies may occur due to noise in filming or other distortions and may trigger a false positive in our results. To avoid this inaccurate data, we filter out the keypoints that highly deviate from points close to itself.

3.5.5 Final Input Format

After going through the data processing pipeline just presented, all data from the different human pose estimation systems had the minimal format listed below. This is the format for all data passed on to the Action Recognition System and the Technique Evaluation System and is therefore the final format that will be used with the rest of the system.

The data is formatted into a dictionary where the key is the keypoint name, and the value contains the x coordinate, y coordinate or the probability of it being true. The index of this array is treated as our frame and if a value is missing it is filled in as a null. An example of the format is shown in Listing 1.

x : X coordinate normalized to the range [0,1]

y : Y coordinate normalized to the range [0,1]

s : Confidence score in the range [0,1]

```
{
    "kneeR_x": [x1, x2, ... ],
    "kneeR_y": [y1, y2, ... ],
    "kneeR_s": [s1, s2, ... ],
    "heelL_x": null,
    ...
}
```

Listing 1: Data Format: Input to Vector Calculation Program

3.6 Action Recognition System

This section will present the *Action Recognition* part of the system, which detect the angle and performed exercise for a given video. It will first go through the data preprocessing steps like normalization and noise filtering performed on the time series. Then it will describe the implementation of action recognition and how *dynamic time warping* and *k-nearest neighbors* were implemented to detect exercise and filming angle.

3.6.1 Data Preprocessing

This part of the system treated each dataset as time series data, where the keypoint positions over time are used to capture the relationship between frames. The data imported to this component is a direct result of human pose estimation systems which have a varying degree of accuracy and noise. In addition, each dataset is different as a result of body variations and the subjects position to the frame. Therefore, noise filtering and normalization were performed as a preprocessing step to make the time series comparable to one another.

Normalization

The keypoints retrieved from the human pose estimation systems were normalized image coordinates. This means that the keypoints are dependent on the subjects position relative to the camera. To account for this problem we had to achieve translational invariance before passing the data on to the classifier. The way this was achieved was to make the neck keypoint center of the coordinate system. This was done by subtracting the neck (x,y) pair from all other keypoint coordinates in the dataset. For data models that did not contain a neck ke point, the center point of the shoulders were simply used instead.

An optimal solution would also try to achieve scale invariance. However, this would require making the distance between the left and the right shoulder 1 by dividing all other keypoints by this distance. But the distance between the shoulders is not the actual distance but rather an 2D projection onto the image plane. This makes it prone to failures when the subject's body is not facing directly towards front of the camera as with the side view detection. For this reason it was not possible to achieve scale invariance for all data in the dataset. The benefit of using the shoulders on a front viewing angle is that the shoulder distance remains relatively constant throughout the videos.

Noise Filtering

Another issue with the dataset retrieved from the human pose estimation system is its natural noisiness. When comparing two time series, the optimal result would be that two time series for the same exercise would have the same curve over time, without any noise that lead to inaccurate predictions. To achieve this, we applied the LOESS (Locally Weighted Scatterplot Smoothing) filter. It works by taking each data point and derive a better estimate for it by taking the weighted average of neighbouring points. The closest neighbouring points will have a higher weight and thus have more effect on the average. This method smooth out the data and extract the general concept of the movement pattern of each body part for a given exercise, making the exercises easier to compare to one another.

3.6.2 Time Series Analysis

The ultimate goal for the Action Recognition System was to detect both the given exercise and the filming angle for a given set of time series. To achieve this, dynamic time warping (DTW) was applied as a method to measure similarity between to time series of different length. It is a non-linear alignment strategy using dynamic programming that account for phase shift in the data.

However, the fact that the solution uses multiple filming angles complicates the situation when the keypoints are coordinates relative to the image itself. A keypoint moving on the x-axis seen from the front can not be compared to the same keypoint moving on the x-axis seen from the side. This is because the two keypoints moves on different planes of each other, which actually gives three different planes to consider when evaluating the data.

For this reason a simple, yet effective angle detection algorithm was implemented that could detect the filming angel before dynamic time warping was used to detect the exercise independently for each case. The formula takes the distance between the shoulders on the x-axis as the only attribute and then classify the filming angle as either front or side view, based on the distance between the left and the right shoulder. Recall that the keypoints are image coordinates and the x-axis maps to different planes of reality for each of the filming angles. Thus making the shoulders appear close together on the side view, while being naturally far apart on the front view, even when the subject's pose is the same. The shoulders are static keypoints and cannot move further apart or closer together such as feet or hands. In addition, they are naturally further apart than other static keypoints like hips or ears, thus making them the best single feature to detect the given angle of a video.

Dynamic Time Warping

After detecting the correct angle, the dynamic time warping algorithm was used to compare the similarity between time series to classify the given exercise. More precisely, fastDTW, an approximate Dynamic Time Warping algorithm with linear time and space complexity was applied. This technique essentially aligns two time series by iterative warping the time axis to find an optimal alignment. The Euclidean distance function was used to find the distance between x[i] and y[j]. Two different approaches to distance matching are shown i Figure 3.17 and 3.18.

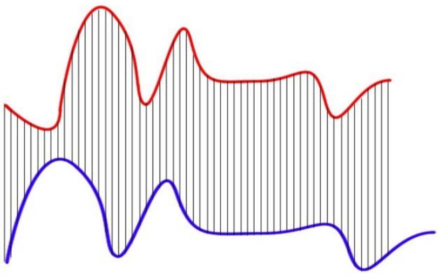


Figure 3.17: Euclidean Distance Matching

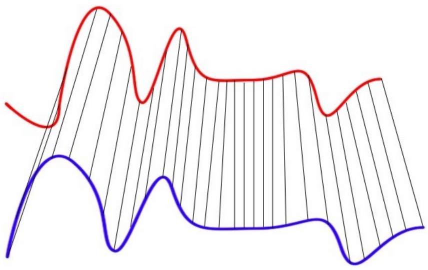


Figure 3.18: Dynamic Time Warping Matching

For the dynamic time warping to be effective, it is necessary to select input data that accurately describes the pattern in hand. Out of the seventeen available keypoints, the left shoulder y-axis was chosen as it is always visible in both exercises and viewing angles. It is also a keypoint that describes the behavior of the movements accurately as when performing squats the shoulders are always lower than deadlift at mid-exercise.

This approach makes for a more scalable solution, so that more exercises can be added to the solution later on without much tweaking. A much simpler solution involving measuring the hand position in proposition with the shoulders was discussed. This would probably classify very well since the hand position is very different for the exercises. However, this would create trouble when new exercises like bench press is involved and would require new and more separating formulas for each new exercise added.

K-Nearest Neighbour

For the classification problem a 1-nearest neighbour approach was used to classify the exercise as either a squat or a deadlift, based on the similarity score provided by fastDTW. For this solution one single base was selected for each exercise, so that all new classifications only need a minimum number of comparisons when calculated. This decision makes the solution both faster and more scalable if more exercises gets added later on. In addition, by choosing only a single base, the solution gets tested against multiple body composition, which provides a clearer picture of how the solution will perform on a bigger dataset with more subjects.

The base representing each exercise was a correct performance of the given exercise, since most flaws only deviated so much from a correct execution, while two different flaws can be far apart from each other. Next up was to decide which person represent the general movement pattern the best. This was done by comparing the similarity between all correct executions of a given exercise with dynamic time warping. By doing so, the exercise that is closest to all other correct execution of the same exercise are chosen. For this case, the same person was considered to be closest to all the other subject for both views and both exercises.

All other videos of a given exercise in the dataset were then executed on the 1-Nearest Neighbour algorithm to find the single closest match.

3.7 Technique Evaluation System

This section will present the *Technique Evaluation System* in its entirety. The system takes the input from the two previously presented systems and outputs a table where all the different technique aspects is either present or absent for a given video or data sequence. The section itself will first present the data model and keypoints used to present a basis for talking about formulas. Consequently it will present all formulas used to calculated and evaluate each technique aspect.

3.7.1 Data Model and Keypoints

The data model used in the solution is an extension of the COCO body model that includes three extra keypoints for each foot and contains 25 keypoints all together. Because some datasets have been transformed into this model from less detailed models, a total of 8 keypoints have the possibility of containing the value *null* in the cases where the data for a given keypoint does not exist. An overview of each index and their associated body part is found in Table 3.3. The table also showcase which keypoints that has the possibility of being *null*.

Index	Joint	Null	Index	Joint	Null
0	Nose	Not null	13	LKnee	Not null
1	Neck	null	14	LAnkle	Not null
2	Rshoulder	Not null	15	REye	Not null
3	RElbow	Not null	16	LEye	Not null
4	RWrist	Not null	17	REar	Not null
5	LShoulder	Not null	18	LEar	Not null
6	LElbow	Not null	19	LBigToe	null
7	LWrist	Not null	20	LSmallToe	null
8	MidHip	null	21	LHeel	null
9	RHip	Not null	22	RBigToe	null
10	RKnee	Not null	23	RSmallToe	null
11	RAnkle	Not null	24	RHeel	null
12	LHip	Not null			

Table 3.3: Body Model: Index Table

3.7.2 Calculations

All vector formulas developed and used for the calculation will be presented along with each technique aspect from Table 3.1 in its own subsection. The subsection will also contain the associated description of the technique issue and the given detection view. Thereafter the method for calculation will be discussed to showcase what was done to uniquely detect a given technique aspect and justify the choices made when developing the formula. All sections will also have two related plots, showing the time series for different keypoints over time. One for a correct execution of the exercise and one where the technique aspect is present. This is done to better visualize how the formula separate the two instances from each other.

3.7.3 Squat: Inward Knee

Description: The knee travel inside of the foot seen from the front.

Detection View: Front View.

The relative height of the subject was used as a measure of the progress of the exercise repetition. For this, the neck keypoints were used as a representation of the person relative height at a given frame. The low values in relative height indicate that the person is standing stationary, while the peak is in the middle of the exercise, when the subject is at the bottom position. At the non-stationary part of the exercise the knees should be outside the ankles, as seen in Figure 3.19. However, cases where the knees do not cross the ankles on the x-axis, thus indicating that the knees are inside of the ankles, as seen in Figure 3.20 can occur. This is a clear indication of a technical issue with inward knees and an error flag will therefore be raised.

As mentioned, it is natural for the knees to be on the inside of the ankles at the start and end of the exercise when the subject is standing stationary or starting/ending their descent/ascent. That is why it is important to compare the keypoints position to one another in relation to the relative height of the user at that specific frame.

One tricky aspect with this technique issue is that inward knees do not always result in both knees caving in, which is why we check for these patterns for both knees independently. One knee might go inward at the start of the movement, long before what is good practice, indicating an inward knee. This aspect is captured by comparing how many frames one of the knees stays crossed over its corresponding ankle compared to same numbers for the other knee.

3.7.4 Squat: Inward Feet

Description: The feet are pointed inward-toward on another.

Detection View: Front View.

Inward feet is a much simpler problem to detect than inward knees, since it limits the user from pushing their knees outside their ankles on the x-axis anatomically. This results in the knees never getting outside of the ankles. On time series, this is

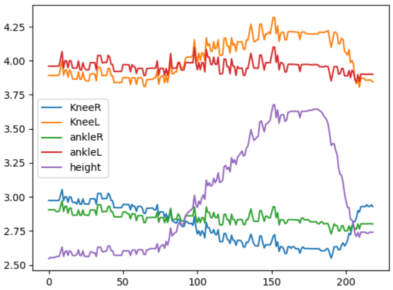


Figure 3.19: Time Series: Squat - Correct

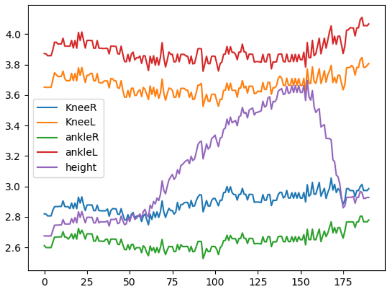


Figure 3.20: Time Series: Squat - Inward Knees

shown by the knee keypoints never crossing the paths of their related ankle keypoint, as shown in Figure 3.22. A correct execution of the squat, where the knee keypoints crosses their respective ankle keypoint is shown in Figure 3.21. An easy check for this was sufficient to detect the lifting aspect. This removes the need to compare feet keypoints, which are only available on the BODY-25 model and OpenPose.

Some inward feet instances are very subtle, making the knees pass the ankles on the x-axis ever so slightly. Because of this, a threshold was added to the calculation that limits how close the knees can be throughout the repetition without triggering a detected inward feet result. This limit was set so that it triggered most instances of inward knee without getting false positives on the correct executed movements.

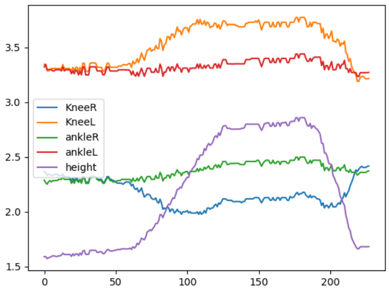


Figure 3.21: Time Series: Squat - Correct

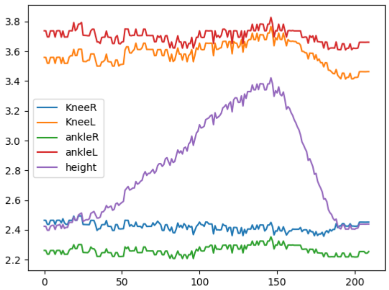


Figure 3.22: Time Series: Squat - Inward feet

3.7.5 Squat: Overextension Knees

Description: Overextension of the knee in the lock-out phase.

Detection View: Side View.

The starting and end stance should be the same when performing a squat. Knee keypoint behind the ankle keypoints indicate an overextension of the knees in the lock-out phase, which put a lot of pressure on the knee joint. A straightforward method was implemented to try to catch some of the cases. The method compared the first and last frames of the movement to detect differences in the distance between ankles and knees. If the knee keypoints were further behind the ankles in the lockout phase than in the starting stance, this indicated an overextension in the knees as seen in Figure 3.24. Figure 3.23 shows a correct execution of the same exercise.

The keypoints suffered from extremely noise data, as a result of the filming being done from the side. This lead to the decision of a simpler approach for detection this issue. As seen in the time series visualization, it is even hard to recognise this as being the same exercise.

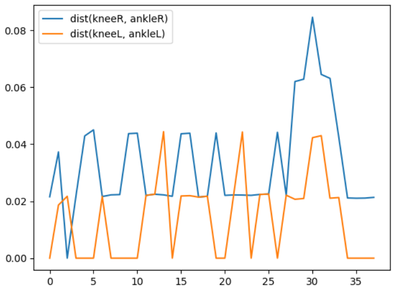


Figure 3.23: Time Series: Squat - Correct

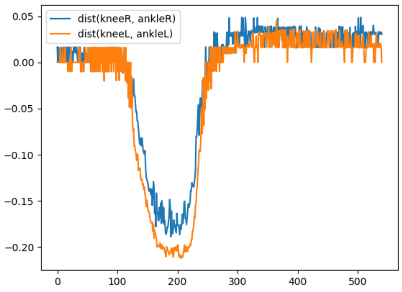


Figure 3.24: Time Series: Squat - Overextension Knees

3.7.6 Squat: Hip Rotation

Description: Asymmetrical rotation of the hips.

Detection View: Front View.

Rotation is difficult to estimate through coordinates on a 2D plane. Thus, an assumption made was that the elbows remain at a relatively constant distance from each other and any major shortening of this distance is an indication of rotation of the subject. This is a reasonable assumption to make when elbows do not move considerably, or not at all on the x-axis for a correct performed squat. Thus a shortening of the distance would indicate that the person has rotated, making the squat asymmetrical.

The distance between the elbows along with the height is shown for a correct performed squat in Figure 3.25 and for a squat with asymmetrical hip rotation in Figure 3.26. The subject relative height is shown to demonstrate the subjects position when the rotation occur. However, the shortening in distance between elbows is subtle and hard to track, which which makes this method prone to errors and false positives.

3.7.7 Squat: Hip Shift

Description: Moving center of pressure to the sides as seen from the front.

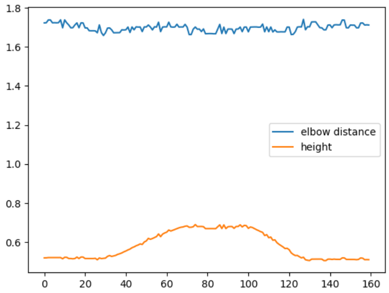


Figure 3.25: Time Series: Squat - Correct

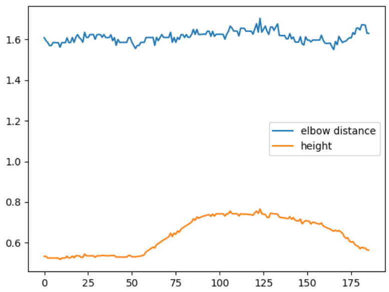


Figure 3.26: Time Series: Squat - Hip Rotation

Detection View: Front View.

Shifting of the hips has the clear aspect that the hips move to either the right or left on the x-axis seen from the front. In a correct performance of the squat, the hips should move as little as possible. A pattern that is observed when the hips shifts to one side, is that the distance between hips and ankle keypoints for the same side changes as well. The hips moves closer to the ankle for one of the sides, whereas the the hip moves farther away from the ankle for the other side.

This can be seen in Figure 3.28, and is a clear indication that the center of pressure has moved closer to one leg than the other. A correct performance of the squat, where the distance between hips and ankle keypoints remains much more constant can be seen in Figure 3.27. In this method a threshold value is used to compare how the max and minimum distance of ankles and hips differ from each other. If the difference is large enough, as seen in the hip shift example, it will trigger a hip shift flag.

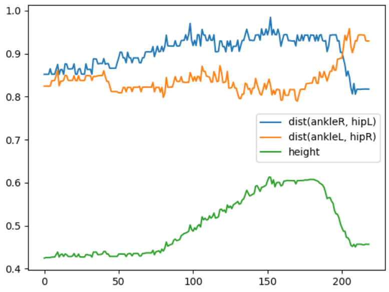


Figure 3.27: Time Series: Squat - Correct

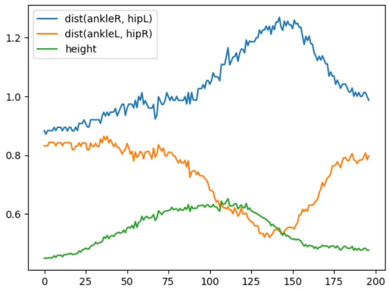


Figure 3.28: Time Series: Squat - Hip Shift

3.7.8 Deadlift: Inward Knees

Description: The knee travel inside of the foot seen from the front.

Detection View: Front View.

For a regular deadlift it is normal for the knees to be very close to the ankles on the x-axis, making this pattern much harder to detect than the squat inward knees problem. The approach are to look at the knee keypoints' movement at the bottom of the exercise movement. This is the hardest part of the lift and where the body compensate by shifting the knees towards one another if the technique is poor. Though if one of the knees suddenly shifts in the x-axis, then it might suggest an inward knee problem. This inward shift is measured by comparing the distance between ankles and knee keypoints for both legs.

As with the inward knee problem for squat, each knee are tested independently to detect instances where one knee moves considerably in comparison to the other.

Figure 3.29 shows the time series for a correct execution of the deadlift, while Figure 3.30 shows the time series for a deadlift with inward knees.

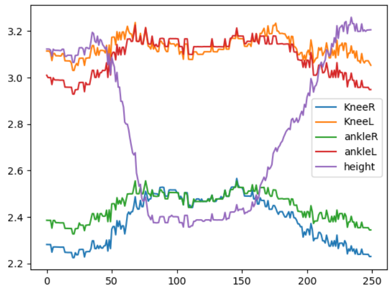


Figure 3.29: Time Series: Deadlift - Correct

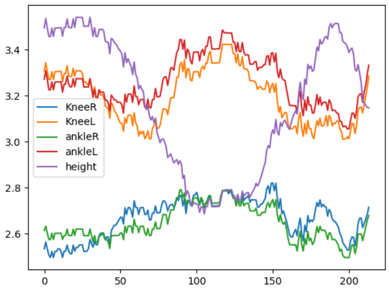


Figure 3.30: Time Series: Deadlift - Inward Knees

3.7.9 Deadlift: Hip Rotation

Description: Asymmetrical rotation of the hips.

Detection View: Front View.

Many of the same aspects as with hip rotation in a squat applies to deadlift as well. It is hard to detect a rotation on a 2D plane with the given data. But we can still track the absolute distance between left and right sides of the body and use this to measure rotation the same way as with hip rotation for squat. In this case the distance between shoulders and opposite hips were used to infer a rotation of the subject. Shoulders and hip keypoints had an overall higher confidence score from the pose estimation systems, making them a better choice than elbows or wrists. In addition shoulder distance had a value of zero after the normalization and could not be used as a single measure of rotation.

The distance from one of the shoulders and the opposite hip will shorten while the opposite pair will lengthen when rotating the hips, as seen in Figure 3.32, thus indicating a detected issue. A correct movement of the same exercise is shown in Figure 3.31.

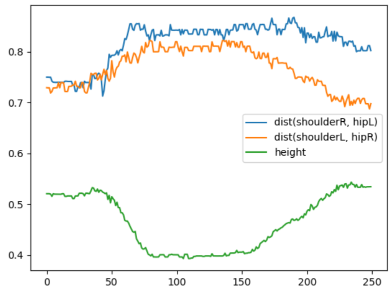


Figure 3.31: Time Series: Deadlift - Correct

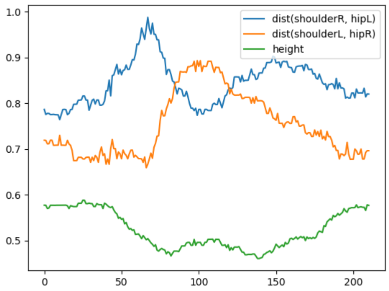


Figure 3.32: Time Series: Deadlift - Hip Rotation

3.7.10 Deadlift: Arching Lower Back

Description: Excessive arching of the lower back in the lockout of the lift instead of/in addition to hip straightening.

Detection View: Front View.

Arching of the back may be detected by tracking if the shoulders pass the hips or ankles on the x-axis at the peak height of the deadlift. The difficulty here is that tracking of the shoulders, hips or ankles from the side provides keypoints with a very low average confidence score. This is leading to spikes and other noise in the time series graphs. Because of the low confidence scores, the confidence score threshold was lowered to avoid getting a substantial amount of errors.

This method uses a simple calculation that compares the first and the last frames of a video. The distance between the shoulder keypoints and the ankle keypoints on the x-axis is first measured. Then if the right shoulder is closer to the ankles in the last frames of the repetition in comparison with the first frames, a technique issue is detected. If the keypoints are detected correctly, this will indicate a severely arching of the lower back as seen in Figure 3.34. A correct execution of the deadlift without arching in the lower back is shown in Figure 3.33.

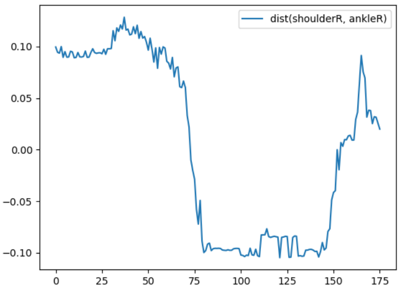


Figure 3.33: Time Series: Deadlift - Correct

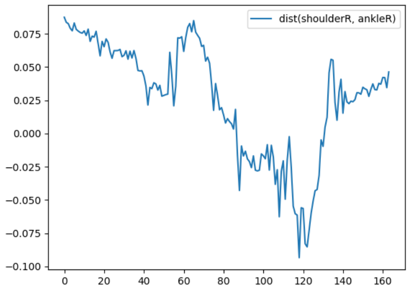


Figure 3.34: Time Series: Deadlift - Arching Lower Back

3.7.11 Deadlift: Elbow Flexion

Description: Lifting with flexion in the elbow.

Detection View: Front View.

An elbow flexion might happen in any part of a deadlift repetition, since the arms should be straight the entire duration of the repetition. The elbows and wrist keypoints should therefore not move in any significant amount on the x-axis seen from the front.

The method that was used, compared the difference between maximum and minimum distance in elbow keypoints to track any significant change in distance throughout the repetition. A big change in the distance indicate a flexion in the elbow as shown in Figure 3.36. An execution of the deadlift without elbow flexion is shown in Figure 3.35.

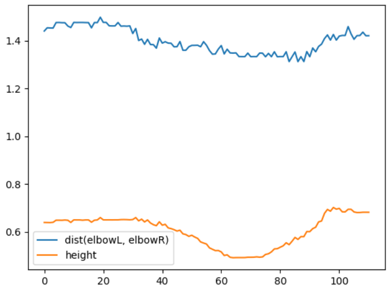


Figure 3.35: Time Series: Deadlift - Correct

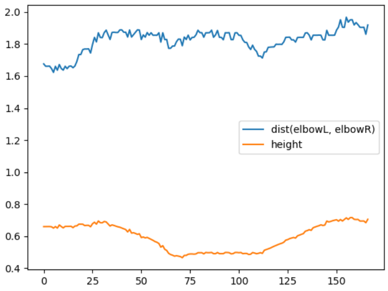


Figure 3.36: Time Series: Deadlift - Elbow Flexion

Chapter 4

Evaluation & Results

This chapter will present the evaluation matrices and evaluation dataset for the system along with the data result for the Action Recognition System and the Technique Evaluation System independently.

First the quantitative and qualitative evaluation metricises are presented separately before the dataset used for evaluation are described. Then the results of each subsystem is presented one at a time.

For the Action Recognition System, the data for angle and exercise detection will be presented separately, before the output result for the Technique Evaluation System will be introduced, one technique aspect at a time. Lastly, the result for different human pose estimation systems will be displayed in relation to one another.

4.1 Data Quality Assessment

This section will present the data analyzation methods used on the output data to evaluate the systems data quality. Here we will look at both quantitative and qualitative methods to analyze the data quality. The quantitative methods involves calculating precision and recall to measure percentage of correct estimations as well as percentage of technique issues actually found by the system. The qualitative analyses engage in talks with an expert to ensure the technique aspect in the video foundation is correct and to compare the findings with a qualitative review of the same videos. At the end of the section, the evaluation dataset used for the quantitative data assessment will be presented.

4.1.1 Quantitative Data Analysis

The quantitative data analysis will be the main quality assessment activity for evaluating the systems overall ability to correctly identify technique issues related to risk of injury. Here we define a true technique issue (ground truth) to be the one we have categorized in our videos. The formulas calculated here are based on the videos of the subjects created in this thesis.

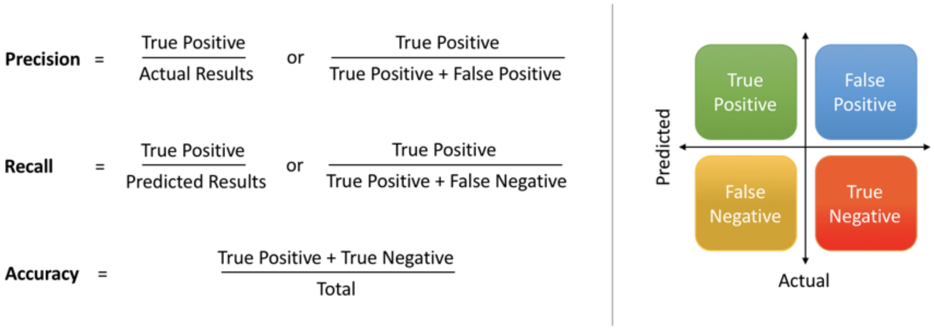

Precision = True Positive / Actual Results or True Positive / True Positive + False Positive
Recall = True Positive / Predicted Results or True Positive / True Positive + False Negative
Accuracy = True Positive + True Negative / Total
Predicted
True Positive
False Positive
False Negative
True Negative
Actual

Figure 4.1: Precision and Recall

Recall

$$\frac{Number\ of\ \textbf{retrieved true}\ technique\ errors}{Number\ of\ \textbf{true}\ technique\ errors}$$

The goal of this system is to catch as many true form technique errors as possible and is geared towards inexperienced lifters. Finding these errors is crucial to accurately give feedback on the exercise movement. A high recall leads to a higher number of found technique errors, and a low recall will lead to a lower number of found technique errors. One can adjust this by changing how much data that is being filter during the input and evaluation phase. Though increasing the recall will inevitably decrease the accuracy as we gain more and more hits on technique errors. A balance here is needed, as too many false positives will lead to the user being confused and focusing on improving aspects that do not need improvement. By tuning down the filtering of the data, one might get a high recall score, but also risk gaining too many false positives. Adjusting the recall to low, by filtering out many datapoints from our input, one risk receiving too few true positives, leading the user to never receive knowledge on their critical flaws in their movements.

Precision

$$\frac{Number\ of\ \textbf{retrieved true}\ technique\ errors}{Number\ of\ \textbf{retrieved}\ technique\ errors}$$

As mentioned earlier, since our system is focused on inexperienced lifters with above average errors in their techniques, precision will be a lower priority than recall. An experienced lifter might have few errors in their lifting making it harder to pinpoint these. While an inexperienced lifter might have several errors that are important to address, and will be less affected by a lower precision. Though a too low precision will lead to a system that is unreliable and useless in its task, it will be kept at a minimum to avoid this. Filtering more data will lead to a higher precision, but as mentioned, also lead to a lower recall score.

F1 Score

$$2 * \frac{Precision\ *\ Recall}{Precision\ +\ Recall}$$

F1 Score, also called F-measure uses precision and recall to calculate the combined accuracy of the two. The value ranges from 0 to 1, where 1 being a perfect score and 0 being the worst. It is often used as a measure to find a good balance between recall and precision.

Accuracy

$$\frac{Number\ of\ \textbf{true}\ technique\ errors}{All\ technique\ errors}$$

Accuracy is a measure of how well the system performs overall when making a correct prediction. An important note on this measure is that a high score will not inevitably lead to a valuable system. If the system never finds technique errors and the chance of a user performing a technique error is low, then it will score high, but at the same time never catch situations where a fault has occurred. This would lead to the system being useless while scoring high at accuracy. Since this will be used on inexperienced lifters the number of technique errors will be above average making the accuracy measure relevant.

True vs. False and Positive vs. Negative

To discuss and evaluate the predicted results in regards to actual results we will use the terminologies **true positive**, **false positive**, **true negative** and **false negative** as represented in Figure 4.1.

For this solution, a **true positive** would mean that the system has correctly predicted a technique error for a video containing the actual technique error. In other words, the system detect a technique error on a video containing the technique error.

A **false positive** implies that the system predicted a technique issue in a video, when the video did not actual have this technique issue present. Thus, a correct execution of an exercise is predicted as a technique error.

A **true negative** would occur if a video without any technique issue present is predicted as having no technique issues present. Meaning that a correct execution of an exercise would be predicted as a correct execution.

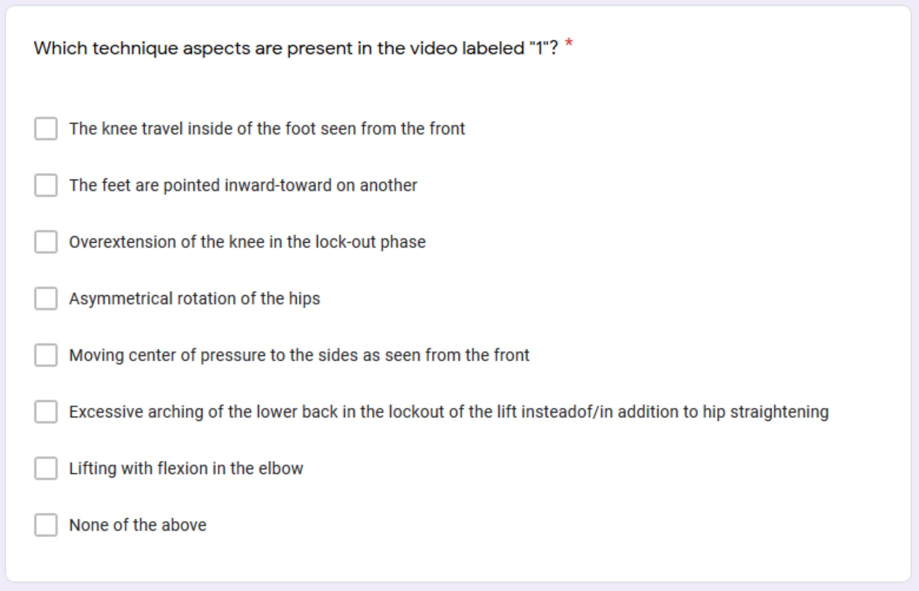
A **false negative** would be if the system predicts a video containing a technique issue to be without the technique issue. In other words, a technique error video would be predicted as a correct execution of the exercise.

For this feedback system, the **true positive** and **false positive** are the most critical instances. This is because the systems concern themselves with predicting technique aspect present in a video and are not actually predicting which technique aspects that are absent. Thus making the two predictive terms more relevant.

4.1.2 Qualitative Data Analysis

The qualitative data analysis is used as a supporting mean to ensure the reliability and correctness of the system. By involving an expert with experience within the field to evaluate the video foundation, the quality assessment gets an extra layer of validity. It is hard to pinpoint the flaws in the exercises, and experts represent the very best validators. The main reason for this process is to assure that the video foundation used in the evaluation is a correct representation of the desired technique aspect. This process assures that a video of a given technique aspect detected to be an issue represents an actual technique issue with a risk of injury in real life.

This quality assessment was performed by presenting an expert with two videos of each technique aspect and a list of all technique issues listed in Table 3.1. The videos were picked randomly from among the four suspects and severity degrees. The expert was then asked to mark off for all technique issues noticeable in a given video. The results were first compared to what technique issue the video originally was meant to show. With the goal of detecting any videos that did not capture the technique aspect properly. Later, the same list was evaluated against the output from the vector calculation system to evaluate the detection ability of the system. An example question from the questionnaire form presented to the expert is shown in Figure 4.2.


Which technique aspects are present in the video labeled "1"? *

□ The knee travel inside of the foot seen from the front

□ The feet are pointed inward-toward on another

□ Overextension of the knee in the lock-out phase

□ Asymmetrical rotation of the hips

□ Moving center of pressure to the sides as seen from the front

□ Excessive arching of the lower back in the lockout of the lift insteadof/in addition to hip straightening

□ Lifting with flexion in the elbow

□ None of the above

Figure 4.2: Screenshot: Questionnaire Form

4.1.3 Evaluation Dataset

All data result tables presented in the second part of this section will have a unique row for each human pose estimation system, as well as a row for the accumulation of all scores across different systems. Each row will have a *precision*, *recall*, *F1* and *accuracy* score as described in Section 4.1.1. In addition there will be an *error* score, representing cases where the confident score of the pose estimation system is too low. Thus filtering away points such that the calculation can not be completed properly and the program returns an error instead. The error cases are neglected when calculating for the four other metrics.

The evaluation dataset consisted of keypoint data generated from all three pose estimation systems using all 82 videos. For each view and each exercise there is one corresponding video of correct execution of the videos for each user. However, as mention, not all the users were able to deliberately perform a squat with a knee extension, leading to only 5 videos created in this category. The total number of files being used in this evaluation is 252, for each pose estimation system this is 82 files.

Table 4.1: Evaluation Dataset

Pose Estimator	Exercise	View	Flaw	Videos
WrnchAI	Squat	Front	Correct	4
OpenPose	Squat	Front	Correct	4
AlphaPose	Squat	Front	Correct	4
WrnchAI	Squat	Front	Inward Knees	8
OpenPose	Squat	Front	Inward Knees	8
AlphaPose	Squat	Front	Inward Knees	8
WrnchAI	Squat	Front	Inward Feet	8
OpenPose	Squat	Front	Inward Feet	8
AlphaPose	Squat	Front	Inward Feet	8
WrnchAI	Squat	Front	Hip Shift	8
OpenPose	Squat	Front	Hip Shift	8
AlphaPose	Squat	Front	Hip Shift	8
WrnchAI	Squat	Front	Hip Rotation	8
OpenPose	Squat	Front	Hip Rotation	8
AlphaPose	Squat	Front	Hip Rotation	8
WrnchAI	Deadlift	Front	Correct	4
OpenPose	Deadlift	Front	Correct	4
AlphaPose	Deadlift	Front	Correct	4
WrnchAI	Deadlift	Front	Inward Knees	8
OpenPose	Deadlift	Front	Inward Knees	8
AlphaPose	Deadlift	Front	Inward Knees	8
WrnchAI	Deadlift	Front	Elbow Flex	8
OpenPose	Deadlift	Front	Elbow Flex	8
AlphaPose	Deadlift	Front	Elbow Flex	8
WrnchAI	Deadlift	Front	Hip Rotation	7
OpenPose	Deadlift	Front	Hip Rotation	7
AlphaPose	Deadlift	Front	Hip Rotation	7
WrnchAI	Squat	Side	Correct	4
OpenPose	Squat	Side	Correct	4
AlphaPose	Squat	Side	Correct	4
WrnchAI	Squat	Side	Knee Extension	5
OpenPose	Squat	Side	Knee Extension	5
AlphaPose	Squat	Side	Knee Extension	5
WrnchAI	Deadlift	Side	Correct	4
OpenPose	Deadlift	Side	Correct	4
AlphaPose	Deadlift	Side	Correct	4
WrnchAI	Deadlift	Side	Arch	8
OpenPose	Deadlift	Side	Arch	8
AlphaPose	Deadlift	Side	Arch	8

4.2 Action Recognition System

The goal of the *Action Recognition System* was to accurately predict the filming angle and performed exercise. This consisted of the two sub-tasks, angle detection and exercise detection. The result of each one are presented independently in its own section.

4.2.1 Angle Detection

The implemented angle detection algorithm showed great result on the test data, with a perfect classification of filming angle for all videos. This gives a *precision* and *recall* score of 1.0 and 0 errors across all human pose estimation systems. The total number of videos tested on the system were 252.

Table 4.2: Result: Angle Detection

Estimator	Prec.	Rec.	F1	Acc.	Errors
Avg/Total	1.00	1.00	1.00	1.00	0
AlphaPose	1.00	1.00	1.00	1.00	0
OpenPose	1.00	1.00	1.00	1.00	0
WrnchAI	1.00	1.00	1.00	1.00	0

4.2.2 Exercise Detection

After determining the viewing angle, the exercise detection algorithm was used to determine what vector calculations to use on the video. The exercise detection was able to detect all true deadlifts, but also predicted 5 false positives, where videos of a squat were wrongfully predicted to be deadlifts. These were squats filmed from the side view, which inherently consisted of keypoints with fairly low confidence scores. The rest of the 247 files were predicted correctly, resulting in a fairly high accuracy. OpenPose was the only candidate with correct predictions for all of the 84 videos.

Table 4.3: Result: Exercise Detection - Squat

Estimator	Prec.	Rec.	F1	Acc.
Avg/Total	0.96	1.00	0.98	0.98
AlphaPose	0.98	1.00	0.99	0.99
OpenPose	1.00	1.00	1.00	1.00
WrnchAI	0.91	1.00	0.95	0.95

Table 4.4: Result: Exercise Detection - Deadlift

Estimator	Prec.	Rec.	F1	Acc.
Avg/Total	1.00	0.96	0.98	0.98
AlphaPose	1.00	0.98	0.99	0.99
OpenPose	1.00	1.00	1.00	1.00
WrnchAI	1.00	0.91	0.95	0.95

4.3 Technique Evaluation System

The goal of the *Technique Evaluation System* was to correctly detect individual technique errors for a given exercise video. This was done across 2 exercises and 9 technique aspects in total. Here the data for each technique issue will be presented individually in its own section. The high scores on many of the technique issues in this section may be partially due to the size of the evaluation dataset.

4.3.1 Squat: Inward Knees

The dataset used for evaluating this technique issue contained all correct front squats, inward feet for the squat and inward knees for the squat. The reason for adding inward feet to the evaluation dataset as well, is that all generated videos of inward knees also contain inward knees by default. Thus providing a bigger dataset to evaluate on, which is favorable.

This detection method scored higher on precision than recall, meaning that most of the returned cases represent an actual inward knee problem. However, some of the lower recall scores indicates that not all cases of inward knees were correctly classified as a technique issue. The overall score of this technique detection algorithm were high for most metrics, implying that this technique issue may be possible to detect for general cases.

Table 4.5: Result: Squat - Inward Knees

Estimator	Prec.	Rec.	F1	Acc.	Errors
AlphaPose	1.00	0.87	0.93	0.90	0
OpenPose	1.00	0.83	0.90	0.86	5
WrnchAI	0.94	1.00	0.97	0.95	0
Avg/Total	0.98	0.91	0.94	0.91	5

4.3.2 Squat: Inward Feet

The dataset used for evaluating this technique issue contained all correct front squats and inward feet videos for the squat.

The detection method scored high on both precision and recall giving a high F1 score for all human pose estimation systems. A few videos tested on OpenPose resulted in an error, while the other two systems had no errors at all. The overall score for this detection algorithm were among the best, indicating that this were one of the easier technique aspects to detect using this method.

Table 4.6: Result: Squat - Inward Feet

Estimator	Prec.	Rec.	F1	Acc.	Errors
AlphaPose	1.00	1.00	1.00	1.00	0
OpenPose	1.00	0.83	0.90	0.89	3
WrnchAI	0.89	1.00	0.94	0.92	0
Avg/Total	0.95	0.95	0.95	0.93	3

4.3.3 Squat: Hip Rotation

The dataset used for evaluating this technique issue contained all correct front squats and hip rotation videos for the squat.

Rotation is difficult to track using points on a 2D plane, but one can gain some results by comparing the distance between the elbows as done here. The problem with this technique aspect is that it is very subtle thus making it hard to differentiate between the inaccuracy in the output of the pose estimation systems and the ground truth movement. The high score in recall and lower in precision suggests that the threshold is too low, and that the system detects hip rotation even in videos that do not contain any technique issues.

Table 4.7: Result: Squat - Hip Rotation

Estimator	Prec.	Rec.	F1	Acc.	Errors
AlphaPose	0.86	0.75	0.80	0.75	0
OpenPose	0.70	1.00	0.82	0.70	2
WrnchAI	0.75	1.00	0.86	0.80	2
Avg/Total	0.76	0.90	0.83	0.75	4

4.3.4 Squat: Hip Shift

The dataset used for evaluating this technique issue contained all correct front squats and hip shift videos for the squat.

The hip shift results were promising, and all the pose estimation methods seemed to be consistent with the results. In addition only one error were generated from OpenPose. Some higher score in recall than precision indicates that the system is providing some false positives, while detecting most of the true technique errors.

However, a flaw in one of the subjects correct execution of the squat was detected during evaluation. The calculation method uses the ankles of the user as a measure of where the hips move, but this assumes that the ankles are remained stationary. One of the users move their ankle position while performing the exercise, which throws off the calculation, producing an extra false positive in all the estimators for this technique aspect.

Table 4.8: Result: Squat - Hip Shift

Estimator	Prec.	Rec.	F1	Acc.	Errors
AlphaPose	0.89	1.00	0.94	0.92	0
OpenPose	0.88	0.88	0.88	0.81	1
WrnchAI	0.89	1.00	0.94	0.92	0
Avg/Total	0.88	0.96	0.92	0.89	1

4.3.5 Squat: Overextension of the Knees

The dataset used for evaluating this technique issue contained all correct front squats and overextension of the knee videos for the squat.

Overextension of the knees seems to be a hard problem to detect. Very low precision scores and some higher recall scores indicate that too many videos are reported as a detected technique issue. The somewhat high recall score may be due the small dataset, making it easy to return all videos as a detected technique aspect along with several false positives.

Table 4.9: Result: Squat - Knee Extention

Estimator	Prec.	Rec.	F1	Acc.	Errors
AlphaPose	0.55	1.00	0.71	0.56	0
OpenPose	0.57	0.80	0.67	0.56	0
WrnchAI	0.50	0.8	0.62	0.44	0
Avg/Total	0.54	0.86	0.67	0.52	0

4.3.6 Deadlift: Elbow Flex

The dataset used for evaluating this technique issue contained all correct deadlifts and elbow flex videos for the deadlift.

The elbows are difficult keypoints to track for the human pose estimation systems, as reflected by the high number of errors produced by OpenPose and overall. Though when detected, the evaluation system detects this lifting aspect with very high success. The precision is prefect, meaning that only true technique errors get returned. In addition, the system detect almost all the actual technique issues, except some videos for the AlphaPose estimator, as shown by the high recall scores.

Table 4.10: Result: Deadlift - Elbow Flex

Estimator	Prec.	Rec.	F1	Acc.	Errors
AlphaPose	1.00	0.88	0.93	0.92	0
OpenPose	1.00	1.00	1.00	1.00	11
WrnchAI	1.00	1.00	1.00	1.00	3
Avg/Total	1.00	0.93	0.97	0.96	14

4.3.7 Deadlift: Inward Knees

The dataset used for evaluating this technique issue contained all correct deadlifts and inward knees videos for the deadlift.

Inward knees for the deadlift also showed promising results with all true technique errors detected, as shown by an overall recall score of 1. The precision score is also rather good, but the some lower numbers indicates that the system detect a few correct execution as an inward knee issue, resulting in false positives. Since both ankles and knees were being used with a minimum confidence score of 0.7, the system also returned a relatively high amount of errors.

Table 4.11: Result: Deadlift - Inward Knees

Estimator	Prec.	Rec.	F1	Acc.	Errors
AlphaPose	0.80	1.00	0.89	0.83	0
OpenPose	0.86	1.00	0.93	0.89	3
WrnchAI	1.00	1.00	1.00	1.00	3
Avg/Total	0.87	1.00	0.93	0.9	6

4.3.8 Deadlift: Hip Rotation

The dataset used for evaluating this technique issue contained all correct deadlifts and hip rotation videos for the deadlift.

This method showed very variable result based on the human pose estimation system used. WrnchAI scored very good overall, while the two other systems had moderate to low scores. The recall score is overall good, but as mentioned earlier this may be due to the small dataset which in turn results to many detected technique issues. The same can be said to be the reason for the some lower precision score as well. In addition, the system resulted in a couple of errors.

Table 4.12: Result: Deadlift - Hip Rotation

Estimator	Prec.	Rec.	F1	Acc.	Errors
AlphaPose	0.86	0.86	0.86	0.82	0
OpenPose	0.57	0.80	0.67	0.50	3
WrnchAI	1.00	1.00	1.00	1.00	2
Avg/Total	0.79	0.88	0.83	0.79	5

4.3.9 Deadlift: Arching Lower Back

The dataset used for evaluating this technique issue contained all correct deadlifts and arching of lower back videos for the deadlift.

Detecting arching of the lower back returned undoubtedly the poorest results of the evaluation system. A simple check for arching seems to be insufficient in detecting an arch. Especially AlphaPose had problems with this method, scoring terribly for both precision and recall. The other two systems did a better job for both precision and recall, but are still no able to detect the technique issue correctly, showed by the low F1 and accuracy scores.

Table 4.13: Result: Deadlift - Arching Lower Back

Estimator	Prec.	Rec.	F1	Acc.	Errors
AlphaPose	0.25	0.13	0.17	0.17	0
OpenPose	0.60	0.38	0.47	0.42	0
WrnchAI	0.75	0.38	0.50	0.50	0
Avg/Total	0.54	0.29	0.38	0.36	0

4.3.10 System Comparison

The system performed very differently depending on the filming angle of the processed video. Front view, generated very good result for most metrics, while side view showed sub-par performance.

In addition, the human pose estimation systems also performed differently even though they used the same data and formulas. WrnchAI had the best results, with AlphaPose close behind. OpenPose, while showing good results, had the lowest scores out of the three systems.

Front View

All of the pose estimation systems performed good with an average precision, recall, F1 and accuracy score around 0.9 for front view videos. One thing to note, is the amount of errors generated while performing the calculations. These are probably due to the high confidence scores used for the front view detection.

Table 4.14: Result: Estimator Comparison - Front

Estimator	Prec.	Rec.	F1	Acc.	Errors
AlphaPose	0.91	0.91	0.91	0.88	0
OpenPose	0.86	0.91	0.87	0.81	28
WrnchAI	0.92	1.00	0.96	0.94	10
Avg/Total	0.89	0.93	0.91	0.88	38

Side View

On the side view videos, all of the pose estimation systems resulted in sub-par performance with average scores around 0.5 for most metrics. No errors were observed while checking for technique aspects using side view keypoints, this may partially be due to the lowered confidence threshold that was necessary to be able to run the files. The lowered confidence threshold may also have played a role in the poor results that were obtained.

Table 4.15: Result: Estimator Comparison - Side

Estimator	Prec.	Rec.	F1	Acc.	Errors
AlphaPose	0.40	0.56	0.47	0.36	0
OpenPose	0.59	0.59	0.59	0.49	0
WrnchAI	0.62	0.59	0.60	0.47	0
Avg/Total	0.54	0.58	0.56	0.44	0

Overall

While OpenPose and AlphaPose have similar results with regards to precision, recall, F1 and accuracy score, AlphaPose produced points with high confidence scores, resulting in zero evaluation errors. WrnchAI had a higher score overall than the other two systems, while producing ten errors.

Table 4.16: Result: Estimator Comparison

Estimator	Prec.	Rec.	F1	Acc.	Errors
AlphaPose	0.80	0.83	0.82	0.76	0
OpenPose	0.80	0.84	0.81	0.74	28
WrnchAI	0.86	0.91	0.87	0.84	10
Avg/Total	0.82	0.86	0.83	0.78	38

4.4 Qualitative Results

The qualitative results show that an expert is able to detect most technique aspects on the sample video dataset, with only three incorrect predictions. All incorrect prediction were technique aspects from the side view, which correlates well with the results from the feedback system developed.

All incorrect predictions from the experts were false negatives, meaning that the expert predicted that the video did not contain any technique issues while the video actually were meant to indicate a technique issue. This has skewed the result for the evaluation on side view somehow, by trying to detect technique issues in videos, that according to the expert, actually is not there.

The rest of the videos in the sample set were predicted correctly. The sample set, prediction and result are presented in Figure 4.17.

Table 4.17: Qualitative Results

Exercise	Angle	Technique Aspect	Predicted	Result
Squat	Front	Correct	Correct	True
Squat	Front	Inward Knees	Inward Knees	True
Squat	Front	Inward Knees	Inward Knees	True
Squat	Front	Inward Feet	Inward Feet	True
Squat	Front	Inward Feet	Inward Feet	True
Squat	Front	Hip Shift	Hip Shift	True
Squat	Front	Hip Shift	Hip Shift	True
Squat	Front	Hip Rotation	Hip Rotation	True
Squat	Front	Hip Rotation	Hip Rotation	True

Table 4.17: Qualitative Results

Deadlift	Front	Correct	Correct	True
Deadlift	Front	Inward Knees	Inward Knees	True
Deadlift	Front	Inward Knees	Inward Knees	True
Deadlift	Front	Elbow Flex	Elbow Flex	True
Deadlift	Front	Elbow Flex	Elbow Flex	True
Deadlift	Front	Hip Rotation	Hip Rotation	True
Deadlift	Front	Hip Rotation	Hip Rotation	True
Squat	Side	Correct	Correct	True
Squat	Side	Knee-extension	Correct	False
Squat	Side	Knee-extension	Correct	False
Deadlift	Side	Correct	Correct	True
Deadlift	Side	Arch	Correct	False
Deadlift	Side	Arch	Arch	True

4.5 Comparison to the State-of-the-Art

To see how the system actually performs and how the technology can be applied to the technique evaluation application, it is important the view the research in light of similar research in the same area. For this application, the Pose Trainer developed by Steven Chen and Richard Yang were the closest with regards to evaluation method approach and end goal. Thus, setting the bar for research with similar approaches in the same area.

When comparing the two solutions, it is important to measure them according to multiple factors such as end result of the system, dataset used for evaluation and technique detection approach.

Exercises

Pose Trainer uses a total of four exercise, namely *bicep curl*, *front raise*, *shoulder shrug* and *shoulder press*. For these movements, only the arms and shoulders are involved in performing the exercise, giving a smaller subset of relevant keypoints. In addition these exercise are considered to be isolation exercises, meaning that one muscle group is used to push or pull the weight. This makes the exercises easier to perform, and thus decreases the number of technique issues that naturally occur while lifting.

In comparison, this solution only uses two exercises. However, these exercises are compound exercise were the whole body is part of the movement and more muscles are used on each repetition. These are also very technical movements that are harder for the user to learn, making them very prone to injuries. Because of this, this thesis needed to track the whole body continuously and look for multiple issues at the same time.

Dataset

Pose Trainer used a dataset of 112 videos in total, divided between the four exercises evaluated. This gave an average of 28 videos per each exercise. The dataset were based on two different male subjects executing each exercise. All human pose estimation data were generated by running all videos on OpenPose.

For this solution, a total of 84 videos were created, giving an average of 42 videos for each of the two exercises. The dataset are based on four different subjects executing each movement, two male and two female. All videos were processed by three different human pose estimation systems, namely OpenPose, AlphaPose and WrnchAI. This process tripled the amount of output data.

Detection Method

Pose Trainer used two different approaches to evaluate the technique, one heuristic-based and one machine learning based approach. The two methods were used independently of each other. The heuristic-based method used vector geometry to evaluate the exercise, while the machine learning approach used dynamic time warping. For detecting viewing angle, Pose Trainer compares the visibility of the individual keypoints to determine whether it was being filmed from the side or front.

Our approach only used a single vector geometric approach to evaluate the exercise movement. However, a dynamic time warping approach was also applied a to automatically detect both the exercise and viewing angle. It also focused on specific technique related issues and tested for each one independently to get a better understanding of what the user is doing wrong.

Results

While using the same metrics to evaluate the results, Pose Trainer presented the data with regards to correct or incorrect execution of the movements as shown in Table 4.18. While this thesis present the result for each given technique aspect separately, Which in turn makes the results challenging to compare to one another.

Pose Trainer achieved good results using their machine learning approach, with front raise being correctly detected as either correct or incorrect for all exercises. The other exercises also showed good results with a F1 score around the 0.8 mark. This data however, is only the result of the machine learning approach. As for the geometric algorithm used, none of the results were mentioned except for the bicep curl detector which was able to detect 80% of the bad executions.

The *bicep curl*, *front rise* and *shoulder press* were all detected from a side view perspective, and gave considerable higher scores than the side view evaluation in this thesis. This reveals that dynamic time warping may be a better choice for this scenario.

However, comparing the front view results, this thesis had a F1 score of around 0.9 based on the pose estimator used. Indicating that it can detect unique technique aspect just as good, if not better than the machine learning approach used by Pose

Trainer. In addition, the OpenPose result for this thesis had an average F1 score of 0.82, which is close to the average score of Pose Trainer, which also uses OpenPose for the pose estimation task. Though, this does not take into account the amount of errors generated by the OpenPose approach, at about 28 errors.

Both Pose Trainer and this system were 100% accurate when detecting viewing angle of the film despite using different a approach for detection.

Table 4.18: Pose Trainer: Machine Learning Results

Exercise	Prec.	Rec.	F1	Videos
Bicep Curl:				
Correct	0.80	1.00	0.89	4
Incorrect	1.00	0.67	0.80	3
Avg/Total	0.89	0.86	0.85	7
Front Raise:				
Correct	1.00	1.00	1.00	6
Incorrect	1.00	1.00	1.00	6
Avg/Total	1.00	1.00	1.00	12
Shoulder Shrug:				
Correct	1.00	0.75	0.86	8
Incorrect	0.71	1.00	0.83	5
Avg/Total	0.89	0.85	0.85	13
Shoulder Press:				
Correct	0.67	0.86	0.75	7
Incorrect	0.83	0.62	0.71	8
Avg/Total	0.89	0.73	0.73	15

Chapter 5

Discussion

This chapter will discuss the different results and limitations of the system and its associated components. Each sub-system will first be discussed individually before the system as a whole will be reviewed.

Thesis Goal

The development of Human Pose Estimation has come a long way and the table is now set for a more widespread application of the technology in the sport and fitness industry. That is why the goal of this thesis was to explore the opportunities in one of these areas by answering the following question:

RQ: To which extent can 2D Human Pose Estimation be used as a tool to give valuable feedback on weight training technique to minimize risk of injuries?

In addition to gaining insight into different Human Pose Estimation Systems and researching risk related to weight training, two physical systems were decided on to try answering the research question. A *Action Recognition System* with the goal of automatically detecting the exercise and filming angle of a video, such that the technique aspects related to the given exercise and filming angle can be automatically tested for. And a *Technique Evaluation System* with the goal of detecting technique aspects with a high association to risk of injury. Thereby informing the users when they are performing an exercise in such manner that they risk injuring them self.

5.1 Action Recognition System

This system had two main objectives: detect the angle that the video was filmed from and detect the exercise performed in the video. Both were equally important to correctly classify which sub-set of technique aspects that should be tested for on the specific video.

5.1.1 Angle Detection

To be able to give feedback on weight training technique, it is crucial to know which angle the exercise is being filmed from. This is the first decisive step in analyzing the exercise videos, a wrong angle prediction would lead to wrong results throughout the rest of the system. By using a distance vector between both shoulders the system was able to accurately predict all viewing angles without any error. The distance between the shoulders of a person is very consistent, independent of what movement the user is doing. This suggests that larger datasets of the same composition of front and side viewing angles will achieve similar results.

Limitations

The angle detector has not been tested for angles that are not necessarily categorized as being strictly front or side. The system cannot rule out angles that are not supported by the vector calculations.

Since the system is using the average non-normalized distance between the shoulders, any movement that involves excessive rotations of the user will increase the likelihood of false predictions. The squat and the deadlift restricts the user from rotating excessively, but other complex exercises might prove to be challenging.

Since the shoulders are not normalized for this detection method, a person very far away from the camera doing a front viewing exercise, will possibly have a perceived shoulder distance small enough to be predicted by the system as being filmed from the side. The average shoulder distance was compared against a constant threshold value, which is independent of how much space the user is taking up from the image.

5.1.2 Exercise Detection

Knowing that all the angle detection tests were successful, the next crucial step in the system was to detect what exercise the user was performing. The system is able to detect all deadlifts as deadlifts, meaning that when the user was performing a deadlift the system was sure to predict this correctly. The same could be said for squats filmed from a front viewing angle. The right shoulder y-axis used as input for the Dynamic Time Warping algorithm seemed to be a good choice for these two types of exercises. Though the system struggled to predict the side viewing angles for the squat.

The errors made from predicting side viewing squats as deadlifts creates an uncertainty for this exercise category. An error at this stage of the system leads to wrong vector calculations and from there wrong user feedback. This challenges the research question this thesis is reaching to answer, can this system give valuable feedback to the user? Only 4-5 out of 82 videos per user were taken from the side angle, making the results questionable as if the system is able to accurately detect the exercise from a side view perspective.

Limitations

The right shoulder y-axis was used as input for the Dynamic Time Warping algorithm and is not necessarily expandable to other exercises. In some exercises the user does not move their shoulders on the y-axis, such as a shoulder press or a bicep curl. This suggests that larger changes to the exercise detection are needed to accommodate for other exercises. As a result, using additional keypoints to deal with the added complexity in other exercises is needed.

As with angle detection, the system was unable to rule out exercises that are not strictly a deadlift or a squat, leading to a wrong evaluation. The Exercise Detection system expected either a squat or a deadlift and would categorize the video in either of those two independent of what the user is doing in the video.

5.2 Technique Evaluation System

The goal of this part of the system was to detect technique aspects and wrong lifting forms while already knowing the keypoints, viewing angle and exercise at hand. This is where the accuracy of the Pose Estimation systems would strongly determine how well one is able to correctly predict the technique aspects that are apparent in a video.

5.2.1 Technique Detection

Hip Rotation was hard to detect, as rotation leads to a shorter difference in the length on x-axis. It might also make it harder for the pose estimators to detect the keypoints when the shoulders go more out of the camera view. Despite of this, the method used was able to detect this technique aspect to a noticeable degree. Comparing this to the results from hip shift, which were much more accurate and fewer errors, it may strengthen the suspicion that rotation is a movement that is innately hard to detect on a two-dimensional plane.

Inward feet and inward knees had one of the most promising results, with the method for detecting inward knees being rather complex for the squat, but simpler for the inward knees deadlift and inward feet squat. These promising results observed from inward knees were surprising as there were several patterns for this technique aspect, as mentioned in the section Squat: Inward Knee. Inward feet had a very specific pattern, making it easier to detect. Though, due to some noise in the keypoint data, the results were not perfect while using OpenPose and WrnchAI. The consistent pattern between the ankles and knees suggests that this method will be successful in detecting inward feet in larger dataset with an accuracy analogous to the accuracy of these findings.

The elbow flex also had a rather simple method of calculating keypoint distances on the x-axis. This simple method lead to very good results, perfect using WrnchAI and OpenPose, with AlphaPose close behind. The overall precision was 1.00, meaning that all detected elbow flex were true. As with the inward feet, having a simple method while scoring high suggests that the results are scalable to larger datasets.

Overextension of the knees and arch were the hardest to detect, indicated by the low results shown in Figure 4.9 and 4.13. Both of these technique aspects were filmed from a side view, making the side view results the worst in the evaluation. The technique detection was unable to give any valuable feedback on whether the user was performing a squat with overextension of the knees or arching of the lower back on a deadlift. Several of the users had difficulties in provoking an overextension, leading to a lower dataset, but there is doubt whether this impacted the overall performance of the method. Considering that the expert that evaluated samples of the videos had most trouble with detecting the side viewing technique aspects, it might suggest that a more precise and complex equipment is needed here.

Overall the technique detectors were able to detect the front view exercise technique aspects with significant results, while the side view technique aspects were less successful. This is undoubtedly related to the Human Pose Estimators differences in accuracy for the two angles. Poor accuracy for the side view translated into uncertainty in the time series and complex calculations. While high accuracy from the front perspective gave clear patterns and easier partition of correct and incorrect dataset.

Limitations

The dataset had a correct/flawed file ratio of about 1 2 meaning that there were about two times more videos with a technique aspect than with a correct execution for the exercise. This means that if the methods detected all videos to contain the technique aspect, then precision would default to 0.66 and recall to 1.00, which may partially explain the very positive results. There might also have been some bias from the researchers while instructing the users to initiate an exercise with a specific technique aspect, leading to a dataset that does not capture keypoint patterns that occur with other untested users.

Some technique aspects inherit other technique aspects, such as inward feet resulting in inward knees as well. The system does not take into account that a hip shift or hip rotation might for also include inward knees. This increases the complexity of the system, but it necessary to really explain the movement in its entirety. Often, more technique aspect are prominent while performing an exercise, due to low flexibility or too heavy weights.

The side view keypoints were very inaccurate, making it tough to see any patterns in the movement from a visual standpoint on the time series graphs. This also led to difficulties when creating methods for the side view data, since there were few similarities between the graphs. The keypoint data for side view, also had large enough value spikes to trigger any threshold value that were set for the movement, making it even harder to pinpoint patterns in the data.

The methods used frame count as an index for the threshold values, a better metric would may be to use the length as a percentage of the video. A user might perform the exercise quite fast, or use a camera that has a high frames per second count, thereby leading to more false results. Although, the data tested on also had a high diversity in frames per second, showing that this metric is applicable as well.

5.2.2 Human Pose Estimation System

From the preliminary studies it was known that AlphaPose had the highest mean average precision out of the candidates, and that OpenPose and WrnchAI had similar scores. AlphaPose had zero errors when detecting technique aspects, which was contributed by the high precision and high confidence score in the keypoints. WrnchAI had about 10 errors, with 8 of them coming from the deadlift exercise. OpenPose had surprisingly 28 errors, and was the most unstable out of all the candidates.

Overall, WrnchAI scored the highest on accuracy, but with more errors than AlphaPose, which scored the second highest. Focusing on the front view scores shown in Table 4.14, the difference between WrnchAI and AlphaPose shortens, though WrnchAI was able to detect all the relevant technique aspects with a recall of 1.00. Considering that the keypoints from the side view were far from optimal, the front view scores might suggest a better comparison of the Human Pose Estimators. OpenPose had the lowest accuracy out of all the candidates, but still showed noticeable result.

When determining which Pose Estimation System that performed best at detection technique aspects it is important to take both the number of errors, recall and precision into consideration. WrnchAI had a perfect recall score on the front viewing exercises and a high precision with a score of 0.92, but due to the high error count, it would not do well in a real life scenario unless the confidence threshold was lowered, thus lowering the evaluation scores. With an error percentage of 34% and the lowest scores on the evaluation dataset, OpenPose proved to be the worst candidate. The errors generated from OpenPose could be mitigated by lowering the confidence threshold, but this would further worsen the accuracy score. Having zero errors and second highest precision score, AlphaPose seems to be the most reliable Human Pose Estimation system to generate keypoints that gives the most accurate feedback to the user.

Limitations

The errors, as mentioned in the methodology section, were a result of too low confidence score leading to inaccurate calculations. Even though the confidence thresholds were the same for every pose estimation system, some lead to more errors than others. The confidence output of every system might be using a different scale, meaning that a confidence score of 0.9 on AlphaPose might not be the same as a 0.9 score on WrnchAI. This might also explain the higher scores on WrnchAI, as they may be stricter on what a confidence score of 0.9 is, leading too more errors under calculation. It would be interesting to see if AlphaPose and WrnchAI had similar results if the confidence threshold was lowered for WrnchAI, thereby removing the errors.

The dataset consisted of a very narrow userbase, consisting only of healthy young individuals with some amateur lifting experience. This might discriminate towards users with physical disabilities or body types that largely diverge from the dataset generated in this thesis. Also, due to most training facilities being closed in 2020 as a result of the global COVID-19 pandemic, filming of some users performing the exercises had to be done in less realistic scenarios. This may have partially impacted the performance of the Human Pose Estimators.

5.3 General Discussion

There are a few things that could of been done differently too better evaluate the system. Considering that there are a lot of methods being tested, the dataset would benefit from being considerably larger, by adding more users and even more videos, to better answer the research question of this thesis. Fortunately the results still seem to point towards a satisfactory conclusion. Also regarding the goal of thesis, if users are to use this software, there needs to be an acceptance in the market. Would anyone film themselves and serve the video to the application? An assumption has been made that the answer is yes, but further research into this topic is needed to accurately answer this question. Lastly, keypoints on a 2-dimensional video seems to give valuable information on whether or not the user is performing their exercises with correct form. Though an important aspect of weight lifting is also knowing which muscles to engage, not only if the person is using a correct form. A correct form may increase the likelihood that the correct muscles are being used, but this can only be tracked by other more sophisticated hardware or through user feedback.

The findings indicate that 2D Human Pose Estimation may be successful in giving feedback on weight training technique to minimize risk of injuries from a front viewing angle on healthy individuals. Further research and improvement to the technique detection is needed to better answer if the success of front view is transferable to side viewing angles. The favorable results from the Pose Trainer on side viewing angles might also suggest that dynamic time warping is a better choice for these types of technique aspects. Exercises that require rotation seem to be harder to detect, but the system is able to generate partially successful results here. Comparing these findings to the Pose Trainer, the system at hand seems to generate similar results, yet with a larger dataset and more precise feedback on more complex exercises. This research extends what has been done in Pose Trainer and confidently shows how well simple techniques used with 2D Human Pose Estimation keypoints can give feedback on weight lifting form.

Chapter 6

Conclusion & Future Work

This research aimed at using Human Pose Estimation to capture and detect specific technique related issues in weight training. By testing multiple Human Pose Estimation systems on a small set of users, the thesis has shown that the technology can be applied to create a working fitness technique evaluation system for multiple users, exercises and technique issues.

The research clearly illustrates that Pose Estimation technology can be accurate enough to detect technique issues in weight lifting, but it also raises the question about its possibility to detect issues only visible from the side. The inaccuracy of the Pose Estimation tools on side view videos impacts the rest of the solution to perform significantly better on videos seen from the front than the side.

While the small user set limits the generalizability of the results, this approach provides new insight by testing the same solution for different body compositions and multiple pose extraction systems. Proving that variations can be accounted for and generalized such that inferences about weight lifting technique can be performed correctly.

Earlier work in the area has mainly focused on depth cameras or multiple sensors to gain information in the three-dimensional space. This thesis explored the two-dimensional space by only using a single RGB camera to capture the pose of the subject. This resulted in a more accessibility service, that theoretically requires no more than a mobile camera of the users themselves.

This thesis also separates from earlier work by detecting specific technique issues instead of exclusively distinguishing between correct and incorrect execution of an exercise. By attacking the problem in this manner you have more information on precisely what the user are doing wrong during an exercise. Thus giving yourself the opportunity to provide the user with feedback on what they have to change in order to fix their form for that exercise. This is an important aspect to minimize the risk of injury, which this system is all about.

Future Work

Based on these conclusions, practitioners should consider testing the same approach on a bigger user set, with higher variability between experience and body composition to confirm the findings of this thesis. An approach where the users themselves film and provide footage for testing could reveal issues not observed in a controlled environment, as well as help maximize the diversity in the dataset. Further work should also include adding to the exercise and technique pool to explore the transferability of the solution to other exercises.

Since the specific technique issues are known, it is possible to build on the solution to provide specific feedback for improvement to the user. This could simply be the detected aspect or more detailed information as which knee moves inward or which way the hips rotate. A description on how to improve the technique issue could also be presented as feedback, provided the domain knowledge is present.

Another improvement is to use the relative height of the person in time series to track the subjects movements during an exercise. By doing so, the start and stop position of a repetition can be automatically defined and repetitions counted. This dismisses the need to manually define the start and end position of a movement.

The improvements mention above are not only interesting topics for further research, but added all together, they have the opportunity to form a complete application with interface, technique evaluation and feedback to the user. An application to guide the users towards an injury free lifting experience, all with the *Human Pose Estimation Assisted Fitness Technique Evaluation System* at the core of the application.

Bibliography

[1] Dario G. Liebermann et al. "Advances in the application of information technology to sport performance". In: *Journal of Sports Sciences* 20.10 (2002). PMID: 12363293, pp. 755–769. DOI: 10.1080/026404102320675611. eprint: https://doi.org/10.1080/026404102320675611. URL: https://doi.org/10.1080/026404102320675611.

[2] Darren E.R. Warburton, Crystal Whitney Nicol, and Shannon S.D. Bredin. "Health benefits of physical activity: the evidence". In: *CMAJ* 174.6 (2006), pp. 801–809. ISSN: 0820-3946. DOI: 10.1503/cmaj.051351. eprint: https://www.cmaj.ca/content/174/6/801.full.pdf. URL: https://www.cmaj.ca/content/174/6/801.

[3] Frank Penedo and Jason Dahn. "Exercise and well-being: A review of mental and physical health benefits associated with physical activity". In: *Current opinion in psychiatry* 18 (Apr. 2005), pp. 189–93. DOI: 10.1097/00001504-200503000-00013.

[4] Chester S. Jones, Carin Christensen, and Michael Young. "Weight Training Injury Trends". In: *The Physician and Sportsmedicine* 28.7 (2000). PMID: 20086651, pp. 61–72. DOI: 10.3810/psm.2000.07.1086. eprint: https://doi.org/10.3810/psm.2000.07.1086. URL: https://doi.org/10.3810/psm.2000.07.1086.

[5] Justin Keogh and Paul Winwood. "The Epidemiology of Injuries Across the Weight Training Sports: A Systematic Review". In: *Sports Medicine* (June 2016). DOI: 10.1007/s40279-016-0575-0.

[6] *Idrett og friluftsliv, levekårsundersøkelsen.* https://www.ssb.no/kultur-og-fritid/statistikker/fritid/hvert-3-aar. Accessed: 2020-01-15.

[7] *WHO obesity among adults.* https://www.who.int/data/gho/data/indicators/indicator-details/GHO/prevalence-of-overweight-among-adults-bmi-greaterequal-25-(crude-estimate)-(-). Accessed: 2020-01-15.

[8] Alexander Toshev and Christian Szegedy. "DeepPose: Human Pose Estimation via Deep Neural Networks". In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (Dec. 2013). DOI: 10.1109/CVPR.2014.214.

[9] Imam Riadi, Sunardi Sunardi, and Arizona Firdonsyah. "Forensic Investigation Technique on Android's Blackberry Messenger using NIST Framework". In: *International Journal of Cyber-Security and Digital Forensics* 6 (Oct. 2017), pp. 198–205.

[10] Henrik Sjöberg et al. "Content Validity Index and Reliability of a New Protocol for Evaluation of Lifting Technique in the Powerlifting Squat and Deadlift". In: *Journal of Strength and Conditioning Research* (Sept. 2018). DOI: `10.1519/JSC.0000000000002791`.

[11] Zhe Cao et al. "OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields". In: (Dec. 2018).

[12] Anton Milan et al. *MOT16: A Benchmark for Multi-Object Tracking*. 2016. arXiv: `1603.00831 [cs.CV]`.

[13] *Pose Detection comparison : wrnchAI vs OpenPose*. `https://www.learnopencv.com/pose-detection-comparison-wrnchai-vs-openpose/`. Accessed: 2020-01-29.

[14] Dushyant Mehta et al. *XNect: Real-time Multi-person 3D Human Pose Estimation with a Single RGB Camera*. July 2019.

[15] Ching-Hang Chen and Deva Ramanan. *3D Human Pose Estimation = 2D Pose Estimation + Matching*. 2016. arXiv: `1612.06524 [cs.CV]`.

[16] Julieta Martinez et al. *A simple yet effective baseline for 3d human pose estimation*. 2017. arXiv: `1705.03098 [cs.CV]`.

[17] Dushyant Mehta et al. "VNect". In: *ACM Transactions on Graphics* 36.4 (July 2017), pp. 1–14. ISSN: 0730-0301. DOI: `10.1145/3072959.3073596`. URL: `http://dx.doi.org/10.1145/3072959.3073596`.

[18] Laxman Kumarapu and Prerana Mukherjee. *AnimePose: Multi-person 3D pose estimation and animation*. 2020. arXiv: `2002.02792 [cs.GR]`.

[19] Christian Zimmermann et al. *3D Human Pose Estimation in RGBD Images for Robotic Task Learning*. 2018. arXiv: `1803.02622 [cs.CV]`.

[20] E. Marchand, H. Uchiyama, and F. Spindler. "Pose Estimation for Augmented Reality: A Hands-On Survey". In: *IEEE Transactions on Visualization and Computer Graphics* 22.12 (Dec. 2016), pp. 2633–2651. ISSN: 2160-9306. DOI: `10.1109/TVCG.2015.2513408`.

[21] Adrià Arbués-Sangüesa, Coloma Ballester, and Gloria Haro. *Single-Camera Basketball Tracker through Pose and Semantic Feature Fusion*. 2019. arXiv: `1906.02042 [cs.CV]`.

[22] Kirk Goldsberry. "CourtVision : New Visual and Spatial Analytics for the NBA". In: 2012.

[23] Lewis Bridgeman et al. "Multi-Person 3D Pose Estimation and Tracking in Sports". In: *CVPR Workshops*. 2019.

[24] M. A. Fischler and R. A. Elschlager. "The Representation and Matching of Pictorial Structures". In: *IEEE Trans. Comput.* 22.1 (Jan. 1973), pp. 67–92. ISSN: 0018-9340. DOI: `10.1109/T-C.1973.223602`. URL: `https://doi.org/10.1109/T-C.1973.223602`.

[25] Kaiming He et al. "Mask R-CNN". In: *2017 IEEE International Conference on Computer Vision (ICCV)* (2017), pp. 2980–2988.

[26] George Papandreou et al. *Towards Accurate Multi-person Pose Estimation in the Wild.* 2017. arXiv: `1701.01779 [cs.CV]`.

[27] Sun Ke et al. “Deep High-Resolution Representation Learning for Human Pose Estimation”. In: (Feb. 2019).

[28] Yilun Chen et al. “Cascaded Pyramid Network for Multi-person Pose Estimation”. In: June 2018, pp. 7103–7112. DOI: `10.1109/CVPR.2018.00742`.

[29] Hao-Shu Fang et al. *RMPE: Regional Multi-person Pose Estimation.* 2016. arXiv: `1612.00137 [cs.CV]`.

[30] Alejandro Newell, Zhiao Huang, and Jia Deng. “Associative Embedding: End-to-End Learning for Joint Detection and Grouping”. In: *NIPS.* 2016.

[31] George Papandreou et al. “PersonLab: Person Pose Estimation and Instance Segmentation with a Bottom-Up, Part-Based, Geometric Embedding Model”. In: *ECCV.* 2018.

[32] Muhammed Kocabas, Salih Karagoz, and Emre Akbas. “MultiPoseNet: Fast Multi-Person Pose Estimation Using Pose Residual Network: 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part XI”. In: Sept. 2018, pp. 437–453. ISBN: 978-3-030-01251-9. DOI: `10.1007/978-3-030-01252-6_26`.

[33] Leonid Pishchulin et al. *DeepCut: Joint Subset Partition and Labeling for Multi Person Pose Estimation.* 2015. arXiv: `1511.06645 [cs.CV]`.

[34] Joseph Redmon and Ali Farhadi. *YOLOv3: An Incremental Improvement.* 2018. arXiv: `1804.02767 [cs.CV]`.

[35] Wei Liu et al. “SSD: Single Shot MultiBox Detector”. In: *Lecture Notes in Computer Science* (2016), pp. 21–37. ISSN: 1611-3349. DOI: `10.1007/978-3-319-46448-0_2`. URL: `http://dx.doi.org/10.1007/978-3-319-46448-0_2`.

[36] Tsung-Yi Lin et al. *Microsoft COCO: Common Objects in Context.* 2014. arXiv: `1405.0312 [cs.CV]`.

[37] R. A. Güler, N. Neverova, and I. Kokkinos. “DensePose: Dense Human Pose Estimation in the Wild”. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition.* June 2018, pp. 7297–7306. DOI: `10.1109/CVPR.2018.00762`.

[38] Mykhaylo Andriluka et al. *PoseTrack: A Benchmark for Human Pose Estimation and Tracking.* 2017. arXiv: `1710.10000 [cs.CV]`.

[39] Carl Vondrick, Donald Patterson, and Deva Ramanan. “Efficiently Scaling up Crowdsourced Video Annotation”. In: *International Journal of Computer Vision* 101 (Jan. 2012). DOI: `10.1007/s11263-012-0564-1`.

[40] *How our tech works.* `https://wrnch.ai/technology/`. Accessed: 2020-01-29.

[41] J.K. Aggarwal and M.S. Ryoo. “Human Activity Analysis: A Review”. In: *ACM Comput. Surv.* 43.3 (Apr. 2011). ISSN: 0360-0300. DOI: `10.1145/1922649.1922653`. URL: `https://doi.org/10.1145/1922649.1922653`.

[42] M. Asadi-Aghbolaghi et al. “A Survey on Deep Learning Based Approaches for Action and Gesture Recognition in Image Sequences”. In: *2017 12th IEEE International Conference on Automatic Face Gesture Recognition (FG 2017).* 2017, pp. 476–483.

[43] Víctor Ponce-López et al. “Gesture and Action Recognition by Evolved Dynamic Subgestures”. In: (Sept. 2015). DOI: 10.5244/C.29.129.

[44] L. Chen et al. “Sensor-Based Activity Recognition”. In: *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 42.6 (2012), pp. 790–808.

[45] Zawar Hussain, Michael Sheng, and Wei Emma Zhang. “Different Approaches for Human Activity Recognition: A Survey”. In: (2019). arXiv: 1906.05074 [cs.CV].

[46] Imen Jegham et al. “Vision-based human action recognition: An overview and real world challenges”. In: *Forensic Science International: Digital Investigation* 32 (2020), p. 200901. ISSN: 2666-2817. DOI: https://doi.org/10.1016/j.fsidi.2019.200901. URL: http://www.sciencedirect.com/science/article/pii/S174228761930283X.

[47] Shugang Zhang et al. “A Review on Human Activity Recognition Using Vision-Based Method”. In: *Journal of Healthcare Engineering* 2017 (July 2017), p. 3090343. ISSN: 2040-2295. DOI: 10.1155/2017/3090343. URL: https://doi.org/10.1155/2017/3090343.

[48] S. Mitra and T. Acharya. “Gesture Recognition: A Survey”. In: *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 37.3 (2007), pp. 311–324.

[49] Mais Yasen and Shaidah Jusoh. “A systematic review on hand gesture recognition techniques, challenges and applications”. In: *PeerJ Computer Science* 5 (2019), e218.

[50] G.R.S. Murthy and R.s Jadon. “A review of vision based hand gesture recognition”. In: *International Journal of Information Technology and Knowledge Management* 2 (Aug. 2), pp. 405–410.

[51] Santiago Riofrio et al. “Gesture Recognition Using Dynamic Time Warping and Kinect: A Practical Approach”. In: Nov. 2017, pp. 302–308. DOI: 10.1109/INCISCOS.2017.36.

[52] Eamonn Keogh and Michael Pazzani. “Derivative Dynamic Time Warping”. In: *First SIAM International Conference on Data Mining* 1 (Jan. 2002). DOI: 10.1137/1.9781611972719.1.

[53] Stan Salvador and Philip Chan. “Toward Accurate Dynamic Time Warping in Linear Time and Space”. In: vol. 11. Jan. 2004, pp. 70–80.

[54] S. Sempena, Nur Ulfa Maulidevi, and Peb Ruswono Aryan. “Human action recognition using Dynamic Time Warping”. In: *Proceedings of the 2011 International Conference on Electrical Engineering and Informatics*. 2011, pp. 1–5.

[55] S. Celebi et al. “Gesture recognition using skeleton data with weighted dynamic time warping”. In: *VISAPP 2013 - Proceedings of the International Conference on Computer Vision Theory and Applications* 1 (Jan. 2013), pp. 620–625.

[56] James Rwigema, Hyo-rim Choi, and Taeyong Kim. “A Differential Evolution Approach to Optimize Weights of Dynamic Time Warping for Multi-Sensor Based Gesture Recognition”. In: *Sensors* 19 (Feb. 2019), p. 1007. DOI: 10.3390/s19051007.

[57] Pascal Schneider et al. *Gesture Recognition in RGB Videos Using Human Body Keypoints and Dynamic Time Warping*. 2019. arXiv: 1906.12171 [cs.CV].

[58] Stephen M. Roth et al. "Muscle Size Responses to Strength Training in Young and Older Men and Women". In: *Journal of the American Geriatrics Society* 49.11 (2001), pp. 1428–1433. DOI: 10.1046/j.1532-5415.2001.4911233.x. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1046/j.1532-5415.2001.4911233.x. URL: https://onlinelibrary.wiley.com/doi/abs/10.1046/j.1532-5415.2001.4911233.x.

[59] R. Seguin and M. E. Nelson. "The benefits of strength training for older adults". In: *Am J Prev Med* 25.3 Suppl 2 (Oct. 2003), pp. 141–149.

[60] L Sewall and LJ Micheli. "Strength training for children". In: *Journal of pediatric orthopedics* 6.2 (1986), pp. 143–146. ISSN: 0271-6798. DOI: 10.1097/01241398-198603000-00004. URL: https://doi.org/10.1097/01241398-198603000-00004.

[61] JA Guy and Lyle Micheli. "Strength Training for Children and Adolescents". In: *The Journal of the American Academy of Orthopaedic Surgeons* 9 (Jan. 2001), pp. 29–36. DOI: 10.5435/00124635-200101000-00004.

[62] Mark E. Lavallee and Tucker Balam. "An overview of strength training injuries: acute and chronic." In: *Current sports medicine reports* 9 5 (2010), pp. 307–13.

[63] LJ Mazur, RJ Yetman, and WL Risser. "Weight-training injuries. Common injuries and preventative methods". In: *Sports medicine (Auckland, N.Z.)* 16.1 (July 1993), pp. 57–63. ISSN: 0112-1642. DOI: 10.2165/00007256-199316010-00005. URL: https://doi.org/10.2165/00007256-199316010-00005.

[64] *Homecourt*. https://www.homecourt.ai/. Accessed: 2020-01-15.

[65] Steven Chen and Richard Yang. "Pose Trainer: Correcting Exercise Posture using Pose Estimation". In: (Mar. 2018). DOI: 10.13140/RG.2.2.29224.47367.

[66] *Smartsuit Pro*. https://www.rokoko.com/en/products/smartsuit-pro. Accessed: 2020-02-17.

[67] Š. Obdržálek et al. "Accuracy and robustness of Kinect pose estimation in the context of coaching of elderly population". In: (Aug. 2012), pp. 1188–1193. ISSN: 1557-170X. DOI: 10.1109/EMBC.2012.6346149.

[68] Joe Sarsfield et al. "Clinical assessment of depth sensor based pose estimation algorithms for technology supervised rehabilitation applications". In: *International Journal of Medical Informatics* 121 (2019), pp. 30–38. ISSN: 1386-5056. DOI: https://doi.org/10.1016/j.ijmedinf.2018.11.001. URL: http://www.sciencedirect.com/science/article/pii/S1386505618312759.

[69] Frank Zijlstra. "Silhouette-based human pose analysis for feedback during physical exercises". In: (Jan. 2007).

[70] H. Chen et al. "Computer-assisted self-training system for sports exercise using kinects". In: (July 2013), pp. 1–4. ISSN: null. DOI: 10.1109/ICMEW.2013.6618307.

[71] Hua-Tsung Chen, Yu-Zhen He, and Chun-Chieh Hsu. "Computer-assisted yoga training system". In: *Multimedia Tools and Applications* 77.18 (Sept.

2018), pp. 23969–23991. ISSN: 1573-7721. DOI: 10.1007/s11042-018-5721-2. URL: https://doi.org/10.1007/s11042-018-5721-2.

[72] H. Xie, A. Watatani, and K. Miyata. “Visual Feedback for Core Training with 3D Human Shape and Pose”. In: (July 2019), pp. 49–56. ISSN: null. DOI: 10.1109/NICOInt.2019.00017.

[73] Jiaqi Zou et al. “Intelligent Fitness Trainer System Based on Human Pose Estimation”. In: (2019). Ed. by Songlin Sun, Meixia Fu, and Lexi Xu, pp. 593–599.

[74] Rushil Khurana et al. “GymCam: Detecting, Recognizing and Tracking Simultaneous Exercises in Unconstrained Scenes”. In: *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 2 (Dec. 2018), pp. 1–17. DOI: 10.1145/3287063.

[75] *NoSQL Databases Explained*. https://www.mongodb.com/nosql-explained. Accessed: 2020-03-15.

[76] *Docker overview*. https://docs.docker.com/get-started/overview/. Accessed: 2020-03-20.

[77] *matplotlib*. https://matplotlib.org/. Accessed: 2020-03-20.

[78] *statsmodels*. https://www.statsmodels.org/. Accessed: 2020-03-20.

[79] *numpy*. https://numpy.org/. Accessed: 2020-03-20.

[80] *MODEL ZOO*. https://github.com/MVIG-SJTU/AlphaPose/blob/master/docs/MODEL_ZOO.md. Accessed: 2020-02-26.

Appendix A

Source Code

A.1 Pose Extraction System

Some code snippets demonstrating dataformat and keypoint information stored in the database.

A.2 Technique Evaluation System

All formulas produced in connection with the Technique Evaluation System to individually detect a given technique aspect.

```
points.append({
    'file': exercise.split('.')[0],
    'user': user,
    'degree': degree,
    'estimator': 'alphapose',
    'view': exercise.split('.')[0].split('_')[2],
    'exercise': exercise.split('.')[0].split('_')[1],
    'flaw': exercise.split('.')[0].split('_')[3],
    'keypoints': {
            'nose_y' : nose_y,
            'nose_x' : nose_x,
            'nose_s' : nose_s,
            'eyeL_y' : eyeL_y,
            'eyeL_x' : eyeL_x,
            'eyeL_s' : eyeL_s,
            'eyeR_y' : eyeR_y,
            'eyeR_x' : eyeR_x,
            'eyeR_s' : eyeR_s,
            'earL_y' : earL_y,
            'earL_x' : earL_x,
            'earL_s' : earL_s,
            'earR_y' : earR_y,
            'earR_x' : earR_x,
            'earR_s' : earR_s,
            'wristL_y' : wristL_y,
            'wristL_x' : wristL_x,
            'wristL_s' : wristL_s,
            'wristR_y' : wristR_y,
            'wristR_x' : wristR_x,
            'wristR_s' : wristR_s,
            'shoulderL_y': shoulderL_y,
            'shoulderL_x': shoulderL_x,
            'shoulderL_s': shoulderL_s,
            'shoulderR_y': shoulderR_y,
            'shoulderR_x': shoulderR_x,
            'shoulderR_s':shoulderR_s,
            'elbowL_y': elbowL_y,
            'elbowL_x': elbowL_x,
            'elbowL_s': elbowL_s,
            'elbowR_y': elbowR_y,
            'elbowR_x': elbowR_x,
            'elbowR_s': elbowR_s,
            'kneeL_y': kneeL_y,
            'kneeL_x': kneeL_x,
            'kneeL_s': kneeL_s,
            'kneeR_y': kneeR_y,
            'kneeR_x': kneeR_x,
            'kneeR_s': kneeR_s,
            ...
```

Listing 2: Source Code: Squat - Inward Knees

```
             ...
            'hipL_y': hipL_y,
            'hipL_x': hipL_x,
            'hipL_s': hipL_s,
            'hipR_y': hipR_y,
            'hipR_x': hipR_x,
            'hipR_s': hipR_s,
            'ankleL_y': ankleL_y,
            'ankleL_x': ankleL_x,
            'ankleL_s': ankleL_s,
            'ankleR_y': ankleR_y,
            'ankleR_x': ankleR_x,
            'ankleR_s': ankleR_s,
            'neck_x': neck_x,
            'neck_y': neck_y
    }})
```

Listing 3: Source Code: Squat - Inward Knees

```python
maxR = 0
maxL = 0
start_pos = True
one_knee_over_time = 0
if (len(neck) == 0):
    minh = maxh = 0
minh = neck[1]
maxh = (neck[1] - minh)
for i in range(len(kneesL)):
    distanceR = abs(kneesL[i] - ankleL[i])
    distanceL = abs(kneesR[i] - ankleR[i])
    height = (neck[i] - minh)
    if maxR < distanceR:
        maxR = distanceR
    if maxL < distanceL:
        maxL = distanceL
    if maxh < height:
        maxh = height
    # Knees outside ankles
    if start_pos and ((kneesL[i] > ankleL[i])
            and (kneesR[i] < ankleR[i])):
        start_pos = False
    # One knee goes over ankle long before the other
    elif (not start_pos)
        and (kneesL[i] < ankleL[i] and kneesR[i] < ankleR[i]
        or kneesR[i] > ankleR[i] and kneesL[i] > ankleL[i]):
        if one_knee_over_time < 15:
            one_knee_over_time = one_knee_over_time + 1
        else:
            print('knee over ankle too long')
            return 'InwardKnees'
    # Knees go in before starting height
    elif (not start_pos and ((kneesL[i] < ankleL[i])
            and (kneesR[i] > ankleR[i]))):
        if (height > maxh*0.7):
            print('knees behind ankles too early')
            return 'InwardKnees'
# Knees never go outside ankles
if start_pos:
    print('knees never outside ankle')
    return 'InwardKnees'
if maxR < 0.2 or maxL < 0.2:
    print('Knees too near ankles')
    return 'InwardKnees'
return 'Correct'
```

Listing 4: Source Code: Squat - Inward Knees

```
start_pos = True
maxR = 0
maxL = 0
for i in range(len(kneesL)):
    distanceR = abs(kneesL[i] - ankleL[i])
    distanceL = abs(kneesR[i] - ankleR[i])

    if maxR < distanceR:
        maxR = distanceR
    if maxL < distanceL:
        maxL = distanceL
    # Knees outside ankles
    if start_pos and ((kneesL[i] > ankleL[i])
                        and (kneesR[i] < ankleR[i])):
        start_pos = False
if start_pos:
    print('knees never outside ankle')
    return 'InwardFeet'
if maxR < 0.15 or maxL < 0.15:
    print('Knees too near ankles')
    return 'InwardFeet'
return 'Correct'
```

Listing 5: Source Code: Squat - Inward Feet

```
start_pos = True
maxL = elbows[1]
minL = elbows[1]
minH = neck[1]
maxH = neck[1]
index = 0
# Does not accommadate for different holding positions
# Shoulder scaling fixes person size problem.
# Unstable values makes for innacurate
# Joachim holds the bar very close,
# making it hard to predict
for i in range(len(elbows)):
    if maxL < elbows[i]:
        maxL = elbows[i]
    if minL > elbows[i]:
        minL = elbows[i]
    if maxH < neck[i]:
        maxH = neck[i]
        index = i
    if minH > neck[i]:
        minH = neck[i]
if ((minL < t)):
    return 'HipRotation'
return 'Correct'
```

Listing 6: Source Code: Squat - Hip Rotation

```
maxL = kneesL[1]
minL = kneesL[1]
maxR = kneesR[1]
minR = kneesR[1]
# Distance from hip and ankle shortens
# during lowest height depending on which side that shifts
for i in range(len(kneesL)):
    if maxL < kneesL[i]:
        maxL = kneesL[i]
    if minL > kneesL[i]:
        minL = kneesL[i]
    if maxR < kneesR[i]:
        maxR = kneesR[i]
    if minR > kneesR[i]:
        minR = kneesR[i]
if ((maxL*t > minL) or (maxR*t > minR)):
    return 'HipShift'
return 'Correct'
```

Listing 7: Source Code: Squat - Hip Shift

```
maxR = 0
maxL = 0
knee_inward = False
for i in range(len(kneesL)):
    if (kneesL[i] < ankleL[i] or kneesR[i] > ankleR[i]):
        distanceR = abs(kneesL[i] - ankleL[i])
        distanceL = abs(kneesR[i] - ankleR[i])
        knee_inward = True
        if maxR < distanceR:
            maxR = distanceR
        if maxL < distanceL:
            maxL = distanceL
# Normal to see inwardknee in deadlift, but no excessively.
# Inward knee over threshold
if knee_inward and maxR > t or maxL > t:
    return 'InwardKnees'
return 'Correct'
```

Listing 8: Source Code: Deadlift - Inward Knees

```
maxLD = shouldersL[1]
minLD = shouldersL[1]
maxRD = shouldersR[1]
minRD = shouldersR[1]
for i in range(len(shouldersL)):
    if maxLD < shouldersL[i]:
        maxLD = shouldersL[i]
    if maxRD < shouldersR[i]:
        maxRD = shouldersR[i]
# Need to accommodate for starting width held
if (abs(minRD - maxRD) > t or abs(minLD - maxLD) > t):
    return 'HipRotation'
return 'Correct'
```

Listing 9: Source Code: Deadlift - Hip Rotation

```
    if (shouldersL[0] < shouldersL[-1]):
        return 'Arch'
    return 'Correct
```

Listing 10: Source Code: Deadlift - Arching Lower Back

```
start_pos = True
maxD = shouldersL[1]
minD = shouldersL[1]
for i in range(len(shouldersL)):
    if maxD < shoulders[i]:
        maxD = shoulders[i]
    if minD > shoulders[i]:
        minD = shoulders[i]
if (abs(minD - maxD) > t):
    return 'ElbowFlex'
return 'Correct'
```

Listing 11: Source Code: Deadlift - Elbow Flexion

```
if not(shouldersL[1] - shouldersL[-1] > t
        or shouldersR[1] - shouldersR[-1] > t):
    return 'KneeExtention
```

Listing 12: Source Code: Deadlift - Knee Extension

Joachim Eivindsen & Brede Kristensen

Human Pose Estimation Assisted Fitness Technique Evaluation System

NTNU
Norwegian University of
Science and Technology