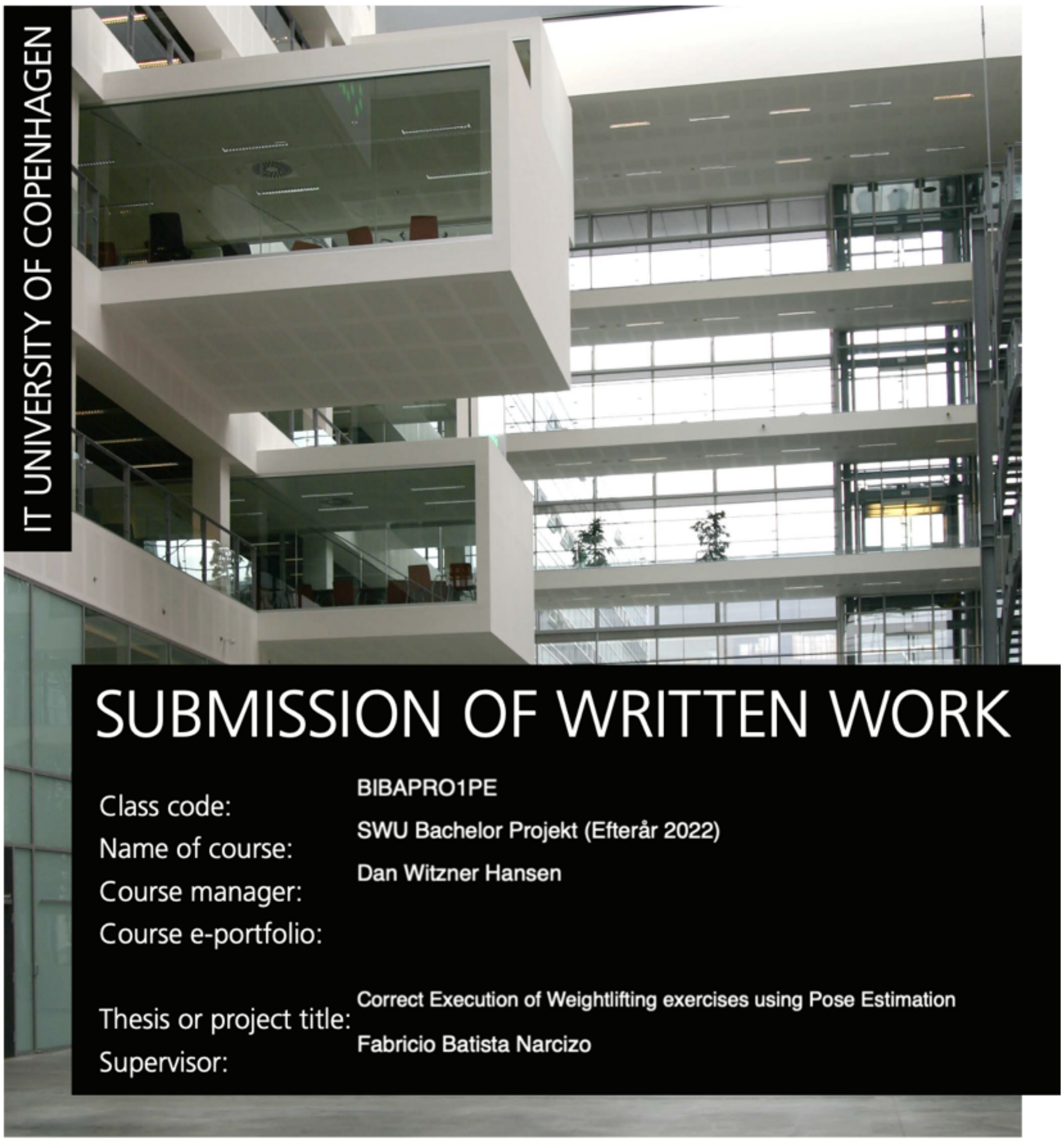


SUBMISSION OF WRITTEN WORK

Class code: BIBAPRO1PE

Name of course: SWU Bachelor Projekt (Efterår 2022)

Course manager: Dan Witzner Hansen

Course e-portfolio:

Thesis or project title: Correct Execution of Weightlifting exercises using Pose Estimation

Supervisor: Fabricio Batista Narcizo

	Full Name:	Birthdate (dd/mm-yyyy):	E-mail:
1.	Benjamin Jarom Lüthje	22/10-1998	bely@itu.dk
2.	Jimmy Ou	16/02-2000	jimy@itu.dk
3.			@itu.dk
4.			@itu.dk
5.			@itu.dk
6.			@itu.dk
7.			@itu.dk

Bachelor Thesis - LiftIT

Benjamin Jarom Lüthje - bely@itu.dk
Jimmy Ou - jimy@itu.dk

15th of December 2022

Abstract

This thesis is a technical analysis of the weightlifting exercise deadlift, using pose detection and machine learning. Weightlifting is an increasingly popular exercise method with substantial benefits. However, if done incorrectly could lead to injuries. The project aims to research whether machine learning technologies help create a solution that is an alternative to hiring a personal trainer. The technical goal is to recognize correct and incorrect movements from video input by running videos through Google MediaPipe pose detection to gather x, y, and z coordinates. The dataset contains either correct or incorrect video labels to feed the machine learning prediction model to predict whether or not a particular video was correct or incorrect. By doing this, the trained model can clear distinct between correct and incorrect movement, resulting in an alternative to hiring a personal trainer.

Table of contents

1 Introduction

Many people put themselves in potential danger by performing weightlifting exercises without the knowledge required to correctly and safely execute the movements with proper form. The current best solution is to hire a personal trainer with the required certifications to provide guidance for completing the exercises with proper form, which most people are unable to afford. The goal is, therefore, to develop a mobile application that functions as an alternative or supplement to a personal trainer when evaluating the execution of weightlifting exercises.

Weightlifting has significant benefits if done correctly, however, if done incorrectly it can lead to various injuries of differing severity. In worst case, it can lead to permanent physical injuries that does not only affect the ability to exercise but also general life quality, such as sleep, doing basic tasks or possibly chronic pain. [1] Our focus in this thesis is on the weight training exercise deadlift. The deadlift is an exercise where a loaded barbell is lifted from the ground to hip level, utilizing a substantial large portion of muscles.

This project aims to use our software development and computer science knowledge to help individuals who weight train and analyze their own execution of the deadlift and the most common mistakes during the lift. By using pose estimation and machine learning we made it possible to visualize different mistakes and determine whether the movement was correct or incorrect.

The vast majority of people working out carry their smartphone or have access to one. Our idea is to utilize the capabilities of a modern smartphone, like the portability and camera sensor, to its greatest extent. Simply uploading a video of the user performing an exercise, mistakes are visualized and categorized in an easy-to-understand manner while also showing how to correct the said mistake.

By utilizing machine learning and pose estimation technologies, our goal is to analyze videos of different people performing the deadlift correctly and incorrectly to see if there is a clear and coherent difference between the two. After that, we created a binary classification in a neural network. With the trained model, the next step is to implement this in an Android application that weightlifters can use to analyze their lifts.

2 Methodology

The first step of any academic research is defining the problem by considering its relevancy and significance. We established the relevancy of the problem by conducting user interviews and a questionnaire to determine whether or not our problem was significant to the users and worthwhile investigating. This made the final product user-oriented by analyzing real-life users' demand for such an application. The questionnaire contained six short yes/no questions, which gave a rough estimate of how relevant the problem was in the daily lives of regular users. This resulted in a quick overview of the current thoughts towards the project itself from a user perspective. The user interviews, however, consisted of 7 in-depth questions, and were answered with sentences and explanations compared to the questionnaire's short yes/no answer format. The participants for the user interviews were ethnically diverse and included multiple genders and people of all ages. We interviews individuals who exercise regularly, who would be interested in the project, and people who never had weight training, to broaden the research pool, resulting in a well-rounded data set. We created a well-rounded research regarding the significance of our problem, by having both a short questionnaire and an explanation based on more detailed user interviews. This is due to both incorporating quantitative research with the questionnaire and qualitative research with user interviews.

Quantitative research is often number focused and helps give an overview of the problem. As the name suggests, this type of research gives a more quantifiable view of the subject matter. However, quantitative research has shortcomings, which is why qualitative research is preferred. Qualitative research consists of a more profound approach. It is beneficial for getting to know the participants' needs, motivations, and rationale [2].

2.1 Questionnaire

The questionnaire provided an overview of the users' current knowledge regarding injuries when executing exercises incorrectly, their standpoint in hiring a personal trainer, and finally, a question of whether or not the user would consider using an app to enhance their experience with weightlifting. From the answers gathered, a clear majority of users were aware of the potential injuries that improper execution of exercises could cause. This was backed by the 91 percent of users who experienced pain after a workout session. These results indicate that most people know the dangers of executing specific exercises incorrectly but need the means to correct them themselves. This fits our goal of providing users who might know the dangers but need more knowledge and guidance to do it correctly.

Furthermore, the answers to whether or not a personal trainer is something the users would consider were distributed 60/40 in favor of the user considering it. However, when we revealed the high price of a personal trainer in the next question, the answers changed. This resulted in only a single person being able to afford to hire a personal trainer. These numbers had a clear connection to the previous questions since most people want to improve their weightlifting form, and the majority considered a personal trainer to be a way to achieve the results they were looking for. However, a personal trainer is not a reliable way to gain knowledge of proper form since only one person in the questionnaire could afford it. At last, most users would consider using an app that helps execute the exercises correctly, which further supports the cause of the project and acts as justification for further research.

By looking at the distribution of yes/no answers to the different questions, it is clear that most users agreed upon the different problems at hand. Only one user could afford a personal trainer, which in turn led to motivation for whether or not our problem was worth looking into and that the possible solution of an application could find a use in the daily lives of people around the world. All of these results from the questionnaire helped confirm the relevancy of our problem statement and served as the basis for later research and development. See Appendix 7.1 for detailed information.

2.2 User interviews

Quantitative research and data from the questionnaire need more for a well-documented and relevant foundation for a research topic. That is why qualitative user interviews were the next step of the process to gain as deep an understanding of the users' needs and opinions as possible. See Appendix 7.2 for detailed information.

The goal of the user interviews was to gain insight into the users' experiences with injuries and possibly longer-lasting problems that comes with them. By looking at the user interviews with Participant 1 and Participant 8 in particular, a clear link between the questionnaire and the user interviews appears. A full-fledged foundation for the rest of the research can be formed by incorporating both the questionnaire and user interview answers.

Participant 1 is new to weightlifting and only exercises once a week due to her busy schedule. She is also well aware of the potential injuries that have kept her from performing specific exercises. In particular, Participant 1 wanted to try doing the deadlift for the first time but ended up skipping the exercise because of the need for more supervision and knowledge regarding the execution. Her exaggerated fear stems from social media and its depiction of serious injuries, which are often very graphic. This has, in turn, resulted in Participant 1 only going to the gym with other more experienced individuals or only performing the few exercises that she is comfortable doing, which do not consist of a wide variety of machines or exercises. The lack of essential compound exercises such as deadlifts, benches, and squats can lead to stagnation in progress. Leaving out these exercises in the user's training program is not recommended for overall strength and progress [3].

However, Participant 1 could not afford a personal trainer, so she would prefer a free alternative to correct her form. She has tried looking at YouTube videos and attempting to mimic the movements, but the lack of visualization of her movements led to her giving up. This is where the proposed app can fill that gap between the lack of visualization of the user doing the exercise in the free alternative with YouTube videos and expensive guidance from a personal trainer. The application will act as a free alternative that bridges the gap between the two weightlifting above assistance ways and provides beginners of weightlifting with an affordable and intuitive application. See participant 1 in Appendix 7.2 for detailed information.

Participant 8 is a male 22-year-old individual whom weight trains five times a week and considers himself a weightlifting enthusiast. He has been working out for two years at the point of the interview and knows how to do each exercise in his training program. However, participant 8 had previously considered incorporating the T-bar row machine in his training regime but needed to figure out how to do the exercise correctly and therefore decided to avoid it altogether. He saw someone more experienced than him doing the T-bar row, which piqued his interest in the exercise, but his fear of getting injured kept him from trying the machine. He never even gave the T-bar row a chance since he knew that the risk of haphazardly jumping into a new exercise outweighed the possible gains from adding the new exercise to his program. Furthermore, participant 8 knew that doing exercises incorrectly could lead to injuries. However, he was unaware of the possible permanent damage caused to the body.

Only recently did he learn his lesson when he tried doing a military press that injured his shoulder and made it impossible for him to sleep on his left side. In the aftermath of the injury, participant 8 heavily considered consulting a personal trainer to avoid future injuries but ultimately could not afford the service. The problem researched in this thesis stems from the fact; that most people can only afford a personal trainer if they have considered hiring one to enhance their performance. The earlier questionnaire also supports this fact, which coincides with other users' opinions toward personal trainers. See Participant 8 in appendix 7.2 for detailed information.

2.3 Interview with certified personal trainer, Alexander Mathiesen

User interviews and questionnaires help establish the groundwork for a well-rounded thesis. However, expert insight can provide a deeper understanding of the topic. It is, therefore, imperative for the thesis to consider the specifics of correct execution and also the dangers of doing the exercises incorrectly. To get the most precise basis, we must define the concrete incorrect executions, before proceeding with the research. An interview with a certified personal trainer gathered the specifics and information about injuries and correct and incorrect execution. The examples below are based on the interview with those mentioned above certified personal trainer Alexander Mathiesen. See Appendix 7.3 for detailed information.

The deadlift is a very beneficial compound exercise focusing on explosiveness and activating the gluteus maximus, hamstrings, lower back, and trapezius muscles. The deadlift is crucial for strengthening these muscles, as well as the stabilizer muscles of the core. If done correctly, the deadlift will contribute to a healthy spine and reduce lower back pain [4].

A large body of research supports that using resistance training, including the deadlift, will help prevent loss of bone mineral density. A progressive loss of bone mineral density can lead to osteoporosis, which significantly increases the risk of fractures among adults. Weight-bearing exercises that load the whole body with external resistance will help increase bone mineral density, and the deadlift is ideal. The exercise will therefore act as an effective way of reducing or reversing the age-related loss of bone mineral density. Therefore, focusing on executing the exercise correctly is crucial as the benefits are many, and the exercise can be highly beneficial for overall health [5].

Based upon knowledge gained from the certified personal trainer Alexander Mathiesen, the correct execution of the deadlift consists of pushing through the floor to lift the bar off the ground to an upright standing position. The position of the hands can either be conventional, with palms facing the user, or reversed, with one of the palms facing away from the user. However, the position of the hands are personal preference and has no effect on the risk of possible injuries and is not a determining factor of whether or not the exercise is done correctly. The distance between the feet is also of personal preference, and there is no one correct way to position the user's feet. The direction of the feet can be angled as the user seems fit. Whether the feet are pointing straight ahead or slightly pointing outwards away from each other is not of concern. It is, however, recommended to at least have a shoulder-width distance between the feet. The most expansive foot stance is called a sumo deadlift, while a narrower stance is called a conventional deadlift. Common injuries happen when hips, arms or the back is moved in an incorrect way. Therefore, our research focuses only on the back, arms, and legs and to correct the execution regarding these body parts exclusively.

The perfect deadlift consists of keeping the center of the bar aligned with the middle of the shoulder blades as shown in Figure 1. This will prevent leaning either forwards or backward and maintain the center of gravity during the exercise. Another important deadlift rule of thumb is to make the arms work as hinges for the weight during the lift. It is, therefore, crucial not to bend the arms at any point during the exercise since it can cause strain on joints in the shoulder.

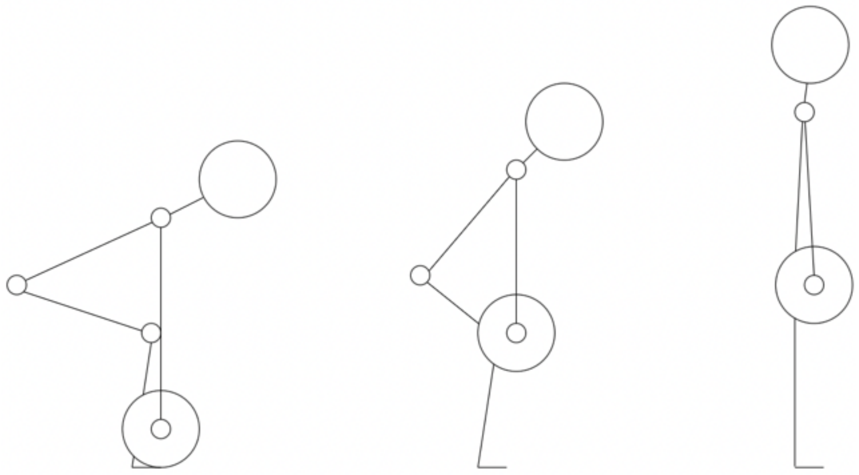


Figure 1: Sequence of correct poses

The perfect deadlift can take much work to get right the first time and is dangerous due to the numerous factors regarding the lift itself. This can lead to several injuries in different body parts. The spine is the most infamous body part prone to injuries since the pressure from the bar's weight can take a toll on the spine if executed incorrectly. The common misdoings can be boiled down to four specific mistakes to combat these injuries and spread awareness on the different ways to do an incorrect deadlift.

1. Executing the deadlift as a squat: Alexander Mathiesen describes that his clients' most common mistake is executing the deadlift like a squat. This involves the wrong starting position of the legs when grabbing the bar. This incorrect starting position can be seen in Figure 2. By having the knees angled at a 90-degree bend, the resulting movement of lifting the bar will not exercise the preferred muscles and instead puts the focus on the quadriceps. This will, in turn, lead to exercising the correct muscle group, which is not necessarily injury-causing but will yield different results. An essential point of weightlifting is activating the correct muscle groups in the body, and exercising said muscles when doing specific movements is imperative for muscular growth. In the deadlift, the gluteus maximus, hamstrings, lower back, and trapezius muscles are the preferred muscles activated during the movement, in decreasing order of focus. Therefore, the activation of the quadriceps during the lift will hinder progress and ultimately result in growth in non-preferred muscle groups.

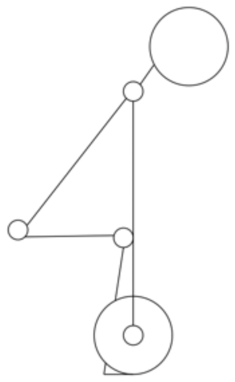


Figure 2: Visualization of incorrect form: Squat

2. Keeping the bar too far away from your shins: Most people are afraid of performing the deadlift with the bar as close to the shins as possible at the start position. The fear comes from being afraid of scratching the bar across the shins, but the proper form consists of keeping the bar as close to the shins as possible without having physical contact. The injury-causing movement stems from the increased distance between the bar and the joint executing the movement. This increased distance between the bar and the body makes the bar feel heavier and can lead to injuries. The possible injuries result from putting an extra strain on the shoulder joints because of the added gravitational pull in an unnatural position. It is unnatural and unhealthy for the shoulders to be extended in front of the body with the added weight of the bar. Furthermore, most of the weight will be focused on the upper back musculature and activation of this muscle group is unnecessary for performing a deadlift. This incorrect starting position is visualized in Figure 3.

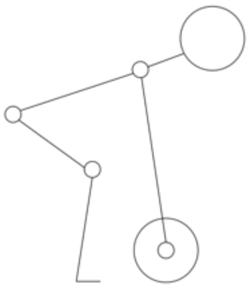


Figure 3: Visualization of incorrect form: Arms

3. Hyperextension of the spine at the end of the lift: The third common mistake is the over-extension of the spine in the upright standing position as shown in Figure 4. The movement should halt when the spine is at a complete vertical standstill and should not overextend to a backward leaning position. This is especially dangerous since the added weight of the bar can lead to various spine injuries due to the unnatural position of the spine bent beyond a vertical standstill. Another possible danger of this particular mistake is the possibility of losing balance due to the hyperextension of the back. It can cause further injuries, such as falling on the floor or the bar falling on top of the user.

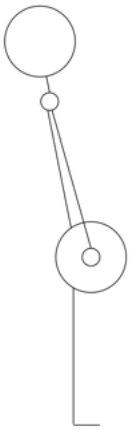


Figure 4: Visualization of incorrect form: Hyperextension

4. Changing the curvature of the back during the lift: The last common mistake to avoid is changing the back's curvature during the lift as shown in Figure 5. Some people have naturally curved spines, which is not inherently an injury-causing condition and a curved back should not be avoided in this case. A common misconception about the deadlift is that a curved spine equals bad and dangerous. The actual concern, however, is the change in back curvature during the lift. Therefore it is not inherently dangerous to have a curved back as long as the back curvature remains static throughout the lift.

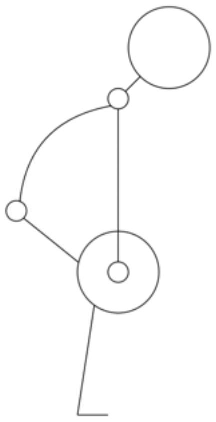


Figure 5: Visualization of incorrect form: Arch

2.4 Spine Mechanics(and the risk of injury)

Tony Leyland [6] states the deadlift's sheer force on the spine. The research focuses on lumbar spinal anatomy and the effects of lumbar curvature during a deadlift. Forces can be applied to the spine in three direction which are compression, tension, and shear, as illustrated in Figure 6.

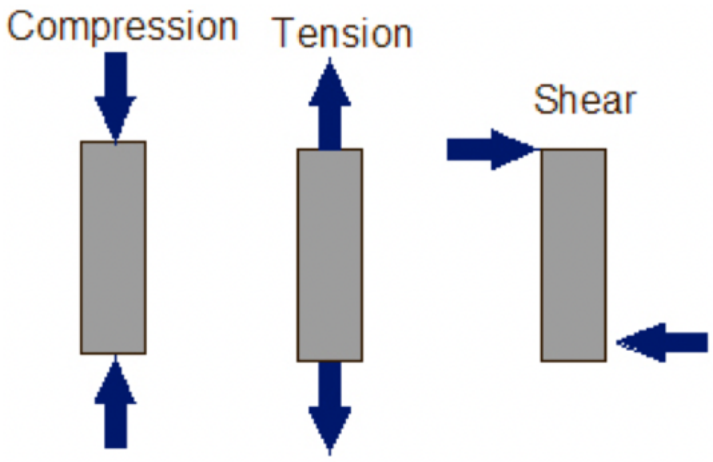


Figure 6: Terminology for directions of force [6].

Significant forces in the form of compression and tear is put on the spine during the execution of a deadlift. Shear is defined as a force that acts parallel to a surface, and when applied to the spine, it can result in the sliding of one vertebra concerning another. The weight being lifted and the center of mass of the arms and upper body is a relatively long way from the spine during the lift, resulting in a high amount of torque on the lumbar vertebrae.

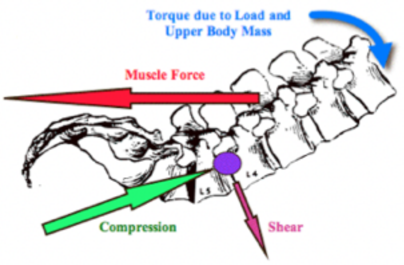


Figure 7: Visualization of forces on the spine during the execution of a deadlift

Figure 7 shows the resulting forces applied to the spine. The vertebrae are a collection of joints each separated by intervertebral discs. The disc between lumbar vertebrae 4 and 5 illustrated with a purple dot is the center of the rotation for these forces and is therefore at the highest risk of a disc slip. Due to the short distance from the joined center of rotation from the line of action of the spinal erector musculature(6-7 centimeters), depending on the weight being lifted, the muscles will pull on the spine with hundreds upwards of thousands of pounds of force. This shear force, combined with the compression force applied to the whole spine due to the stabilization of the lower body against the ground during the lift, results in an exceptionally high amount of stress on the spine. The article further researches

the importance of maintaining normal lumbar lordosis(curvature) by studying the anatomy of the spine and the line of action of the erector muscles and ligaments which connects the vertebra (interspinous ligaments). The spinal biomechanist Dr. Stuart McGill from the University of Waterloo in Ontario identifies the shear force applied to the spine [7]. The force shown in Figure 6 is reaction shear, which results from gravity pulling the load and the upper body downwards. He has proved that the further away the bar is from the legs during a deadlift, the more horizontal the lift will become, the greater the shear force is on the lumbar vertebrae, and the higher the risk of injury is. This supports the danger of the common mistake referred to as “arms” in this thesis, where the bar is positioned away from the body at the start of and/or during the execution of a deadlift.

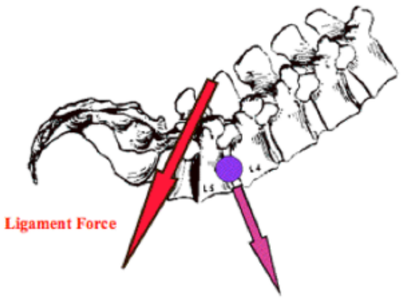


Figure 8: Interspinous ligaments contributes to joint shear in lumbar vertebrae

The joint shear applied to the lumbar vertebrae 4 and 5 is the resultant shear force produced by the muscle/ligament shear. The reaction shear finally determines the risk of injury by executing a deadlift, as illustrated in Figure 8. The ligament force is not illustrated in the figure before it because this force will be mostly neutralized due to the spinal erector musculature, as a result of maintaining a natural curvature of the spine during the whole lift, which creates opposing torque. This supports the danger of the common mistake referred to as “arch” in this thesis, where the spinal curvature is changed during the execution of a deadlift.

The cornerstone of our research for further incorporation of machine learning has been the four above-mentioned common mistakes, as well as the research done by Dr. Stuart McGill and Tony Leyland. The sample videos used for the more technical part of the research were based on each of these particular common mistakes. These mistakes were each replicated by the participants who performed the deadlift in the four different incorrect ways, followed by one correct execution of the deadlift. These videos will act as the foundation of our later research.

2.5 Mediapipe

The Google Mediapipe framework was the preferred technology tool to analyze the participants’ videos replicating the incorrect and correct movements. MediaPipe is an open-source framework that builds multimodal, cross-platform applied machine learning pipelines. The framework relies on OpenCV for video processing and FFMPEG for audio handling and has many more dependencies that ultimately result in an easy-to-use, off-the-go framework. Google MediaPipe is a library with many different Machine Learning ready-to-use solutions. One of the ready-to-use MediaPipe solutions is the human pose estimation solution which tracks human body movement with 33 landmarks representing a point of the body outline. These landmarks will have fundamental importance as their position throughout a movement, and their relevance to each other will allow the detection of common mistakes when executing exercises [8].

2.6 Pose Estimation

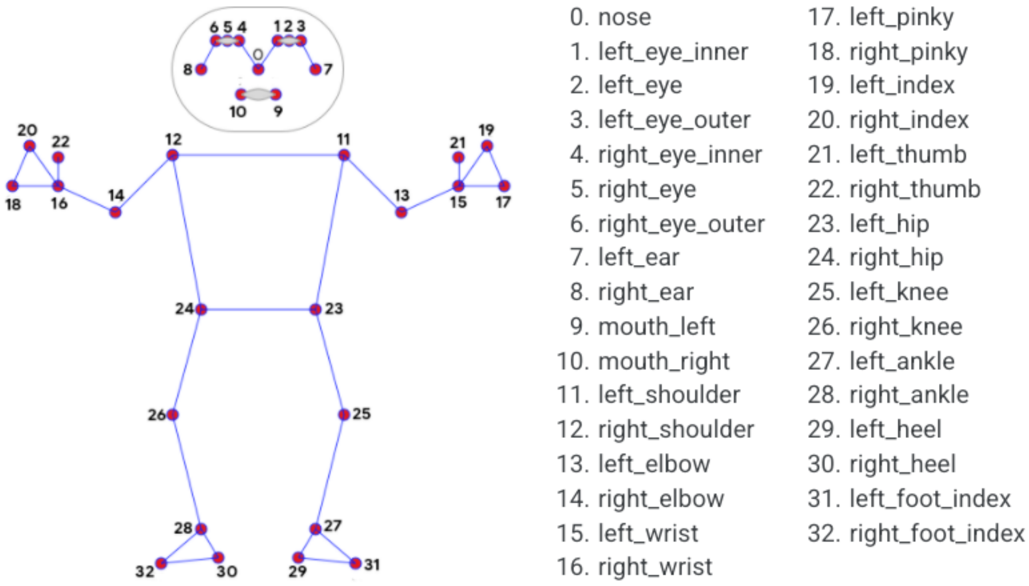


Figure 9: Pose Detection Landmarks

Figure 9 depicts the previously mentioned landmarks utilized in our analysis. This model is the Google MediaPipe BlazePose ideal for our project as it achieved real-time performance on mobile phones with CPU inference [9]. When computed with a video as input, the model will try to place the 33 landmarks on the individual in the video. If the body part is not visible in the video, the model will estimate the landmarks' position. An example of this is seen in Figure 10, where the left hip, landmark 33, is hidden behind the test subject's right arm.

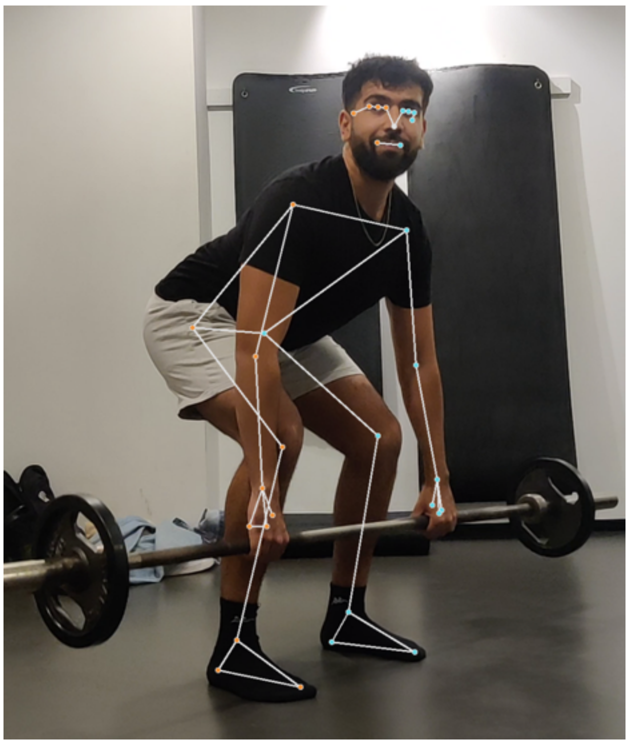


Figure 10: Pose Detection on Test Subject, executing a correct deadlift

2.7 Video Data Collection

A set of individuals were recorded performing the deadlift correctly and incorrectly, according to the findings from our interview with the certified personal trainer Alexander Mathiesen, to collect data for our analysis and binary classification. The data was collected with various cameras and an ethnically diverse group of individuals of all genders and body types, to account for the diversity of people and their different personal smartphones. The devices used to gather our dataset ran an assortment of operating systems, including iOS, Android, Windows, and macOS. This was done to ensure a diverse dataset representing the natural world as closely as possible.

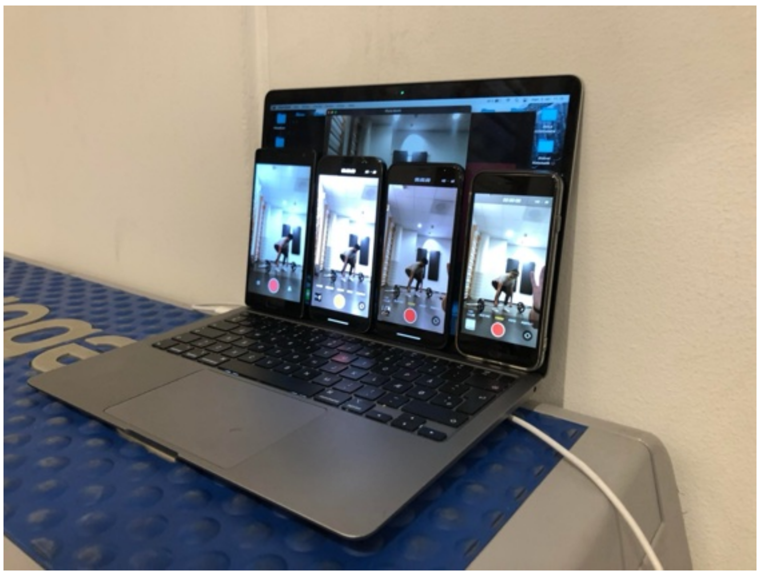
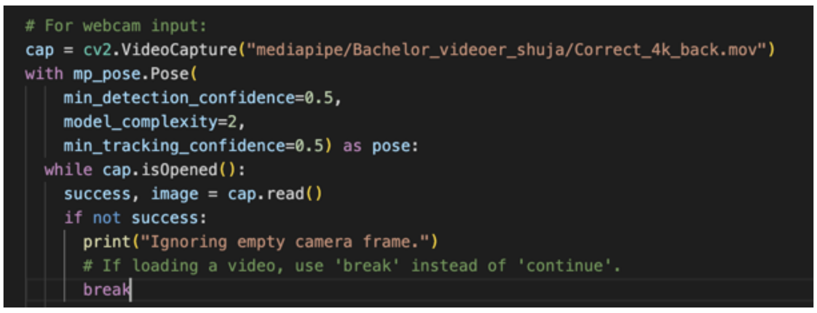


Figure 11: Video Data Collection Setup

The videos were captured by placing different smartphones on a laptop that was also recording and thereby getting videos from almost the same angles but from cameras of different quality as shown in Figure 11. The smartphones varied from the newest iPhone to a seven-year-old Android phone. This was done to test whether the difference between correct and incorrect movements is as clear on videos of worse quality as on videos of higher quality.

2.8 Video Data Processing

To analyze the movements executed in the videos, the videos were shortened to showcase three repetitions of the exercise and then processed through Python code that executed Mediapipe's pose estimation in the video.

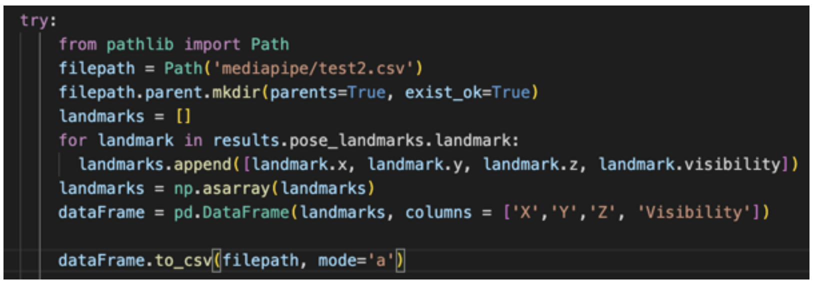


```python
# For webcam input:
cap = cv2.VideoCapture("mediapipe/Bachelor_videoer_shuja/Correct_4k_back.mov")
with mp_pose.Pose(
    min_detection_confidence=0.5,
    model_complexity=2,
    min_tracking_confidence=0.5) as pose:
  while cap.isOpened():
    success, image = cap.read()
    if not success:
      print("Ignoring empty camera frame.")
      # If loading a video, use 'break' instead of 'continue'.
      break
```

Figure 12: Python code to process video files with pose detection

Figure 12 shows a code snippet that uses the OpenCV framework to process the video from a path. Afterward, we defined the model complexity, detection and tracking confidence. Detection confidence ranges from 0.0 - 1.0 and is set to 0.5. It is the minimum confidence value of whether a pose is successfully detected. Tracking confidence also ranges from 0.0 - 1.0 and is also, by default, set to 0.5. This setting detects whether or not the landmarks are tracked confidently. Setting higher values would result in a more robust albeit slower video processing. We kept the default values since lowering the confidence levels proved cumbersome, resulting in landmarks behind the bar disappearing and causing the pose detection to stop working. Furthermore, setting both values higher than default resulted in prolonged processing of the videos, especially those in 4K resolution. The difference in robustness was minimal between default and higher settings, leading to the conclusion of using the default confidence values.

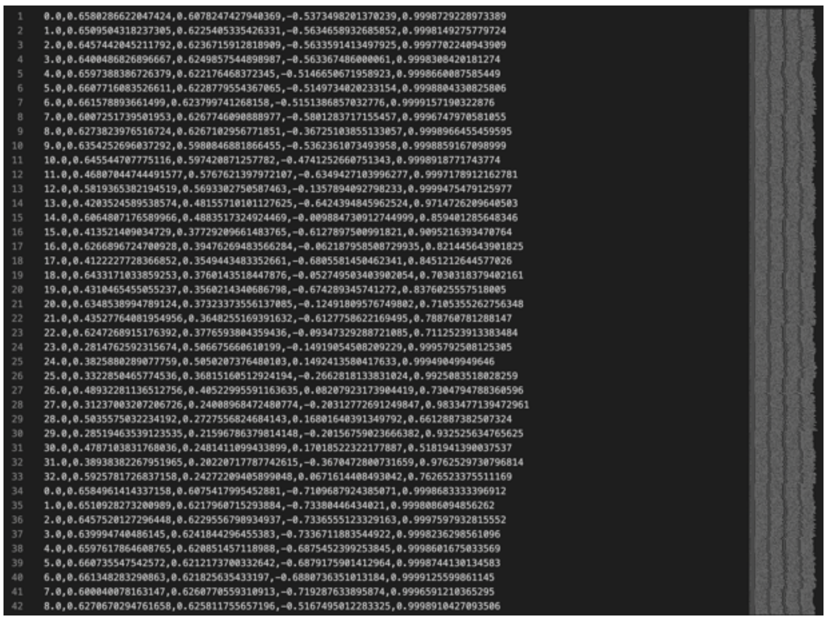
Each landmark will have an X, Y, Z and visibility value for each frame in the video, where X, Y and Z are positional parameters and visibility is a value in percentage of how visible the landmark is. We saved these values to a CSV file and processed it through a Python Script that transforms the data into a usable format for plotting.



```
try:
    from pathlib import Path
    filepath = Path('mediapipe/test2.csv')
    filepath.parent.mkdir(parents=True, exist_ok=True)
    landmarks = []
    for landmark in results.pose_landmarks.landmark:
      landmarks.append([landmark.x, landmark.y, landmark.z, landmark.visibility])
    landmarks = np.asarray(landmarks)
    dataFrame = pd.DataFrame(landmarks, columns = ['X','Y','Z', 'Visibility'])

    dataFrame.to_csv(filepath, mode='a')
```

Figure 13: Python code to create CSV files containing the data from the videos

Figure 13 illustrates a code snippet we used to process data from the landmarks into CSV files which helps plot the landmark values in the next step of the solution. We created a file path for the CSV file, which defines the path the file will be created at. We created the landmarks array as the first thing to store all of the landmarks from the pose detection module. We then converted the array to a NumPy array which we then converted into a Pandas data frame object. The final step is converting the Pandas data frame object to a CSV file with the chosen file path as the destination. The mode parameter is by default set to ‘w’, which is write only, resulting in the landmark features overwriting each other every time in the loop. To fix this, we altered the model to ‘a’, which changes the Python write mode to append to keep it from overwriting itself. See Figure 14 for an example of how the landmark data looks in plain text.



```
0.0,0.6580286622047424,0.6078247427940369,-0.5373498201370239,0.9998729228973389
1.0,0.6509504318237305,0.6225405335426331,-0.5634658932685852,0.9998149275779724
2.0,0.6457442045211792,0.6236715912818909,-0.5633591413497925,0.9997702240943909
3.0,0.6400486826896667,0.6249857544898987,-0.563367486000061,0.9998308420181274
4.0,0.6597388386726379,0.622176468372345,-0.5146650671958923,0.9998660087585449
5.0,0.6607716083526611,0.6228779554367065,-0.5149734020233154,0.9998804330825806
6.0,0.661578893661499,0.623799741268158,-0.5151386857032776,0.9999157190322876
7.0,0.6007251739501953,0.6267746090888977,-0.5801283717155457,0.9996747970581055
8.0,0.6273823976516724,0.6267102956771851,-0.36725103855133057,0.9998966455459595
9.0,0.6354252696037292,0.5980846881866455,-0.5362361073493958,0.9998859167098999
10.0,0.645544707775116,0.597420871257782,-0.4741252660751343,0.9998918771743774
11.0,0.46807044744491577,0.5767621397972107,-0.6349427103996277,0.9997178912162781
12.0,0.5819365382194519,0.5693302750587463,-0.1357894092798233,0.9999475479125977
13.0,0.4203524589538574,0.48155710101127625,-0.6424394845962524,0.9714726209640503
14.0,0.6064807176589966,0.4883517324924469,-0.009884730912744999,0.859401285648346
15.0,0.413521409034729,0.37729209661483765,-0.6127897500991821,0.9095216393470764
16.0,0.6266896724700928,0.39476269483566284,-0.062187958508729935,0.821445643901825
17.0,0.4122227728366852,0.3549443483352661,-0.6805581450462341,0.8451212644577026
18.0,0.6433171033859253,0.3760143518447876,-0.052749503403902054,0.7030318379402161
19.0,0.4310465455055237,0.3560214340686798,-0.674289345741272,0.8376025557518005
20.0,0.6348538994789124,0.37323373556137085,-0.12491809576749802,0.7105355262756348
21.0,0.43527764081954956,0.3648255169391632,-0.6127758622169495,0.788760781288147
22.0,0.6247268915176392,0.3776593804359436,-0.09347329288721085,0.7112523913383484
23.0,0.2814762592315674,0.506675660610199,-0.14919054508209229,0.9995792508125305
24.0,0.3825880289077759,0.5050207376480103,0.1492413580417633,0.99949049949646
25.0,0.3322850465774536,0.36815160512924194,-0.2662818133831024,0.9925083518028259
26.0,0.48932281136512756,0.40522995591163635,0.08207923173904419,0.7304794788360596
27.0,0.31237003207206726,0.24008968472480774,-0.20312772691249847,0.9833477139472961
28.0,0.5035575032234192,0.2727556824684143,0.16801640391349792,0.6612887382507324
29.0,0.28519463539123535,0.21596786379814148,-0.20156759023666382,0.932525634765625
30.0,0.4787103831768036,0.2481411099433899,0.17018522322177887,0.5181941390037537
31.0,0.38938382267951965,0.20220717787742615,-0.3670472800731659,0.9762529730796814
32.0,0.5925781726837158,0.24272209405899048,0.0671614408493042,0.7626523375511169
0.0,0.6584961414337158,0.6075417995452881,-0.7109687924385071,0.9998683333396912
1.0,0.6510928273200989,0.6217960715293884,-0.73380446434021,0.9998086094856262
2.0,0.6457520127296448,0.6229556798934937,-0.7336555123329163,0.9997597932815552
3.0,0.639994740486145,0.6241844296455383,-0.7336711883544922,0.9998236298561096
4.0,0.6597617864608765,0.620851457118988,-0.6875452399253845,0.9998601675033569
5.0,0.660735547542572,0.6212173700332642,-0.6879175901412964,0.9998744130134583
6.0,0.661348283290863,0.621825635433197,-0.6880736351013184,0.9999125599861145
7.0,0.600040078163147,0.6260770559310913,-0.719287633895874,0.9996591210365295
8.0,0.6270670294761658,0.625811755657196,-0.5167495012283325,0.9998910427093506
```

Figure 14: Landmark Data

2.9 Principal Components Analysis

Principal component analysis (PCA) reduces dimensionality of datasets. PCA identifies a reduced set of features representing the original dataset in a lower-dimensional subspace. PCA creates new variables, which maximizes variance. This is achieved with a minimal loss of information and is particularly useful for the dataset of this thesis by increasing interpretability for our later analysis.

PCA summarizes data and extracts information regarding individual differences. We obtain several advantages when dealing with a reduced dataset instead of high-dimensional data, like the ability to visualize the data in 2 or 3 dimensions, less storage space, and noise reduction. The dataset analyzed in this thesis consists of many dimensions/features. Each frame of the videos processed contains information about 33 landmarks that each have 4 variables, resulting in 132 dimensions. Furthermore, we analyzed 15 and 30 frames at a time, resulting in a dimensionality of upwards of 1080 dimensions per computation, and this is naturally impossible to plot.

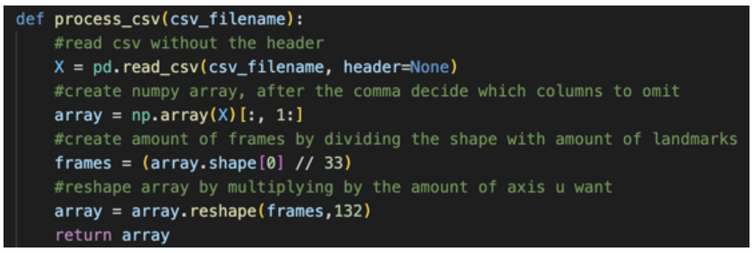
By incorporating PCA, all these variables are computed into 2-dimensional data, which allows us to visualize the data in a graph. This will be highly beneficial for analyzing the data from the different deadlifts to visualize whether or not we can identify a clear distinction between the data derived from them [10]. PCA uses variance as an objective and mathematical way to quantify the amount of information in our data. It is a linear transformation that transforms data to a new coordinate system, where the data's most significant variance lies on the first coordinate. This is called the first principal component.

The second principle component is the projection of the second most significant variance, and so on. In this way, PCA relies on the information from the data that varies the most from each other and distinguishes them from each other from that. PCA creates new variables from a combination of the existing variables, and uses these variables to distinguish the data. The new variables derived from the original data set are called PC_1, PC_2 up to PC_n, where n is the number of variables in the original dataset.

We only used the first two PC variables in our plots, meaning some information was lost. The point of PCA is to reduce this information loss and keep as much distinguishable information in these first few PCA variables. The data is plotted in 2 dimensions, to visualize how PC_1 is calculated from a 2-dimensional dataset. The data will be shifted so that the center is on top of the origin (0,0). After that, a line is drawn so that it goes through the origin, and its collected distance to the plotted data is minimized. PCA does, however, project the data onto this line and then maximize the sum of the squared distances from the projected points to the origin.

The sums are squared, so negative values do not cancel out positive ones. This line is principal component 1. Its slope tells of the importance of the variables on the X- and Y-axis. Hence a PC_1 line that tilts closer to the X-axis than the Y-axis, indicates that the variable on the X-axis, has greater importance for the value of PC1. This is called linear combination. A unit that describes the ratio of the importance of the two variables when creating PC_1 is calculated and is called the Eigenvalue [11].

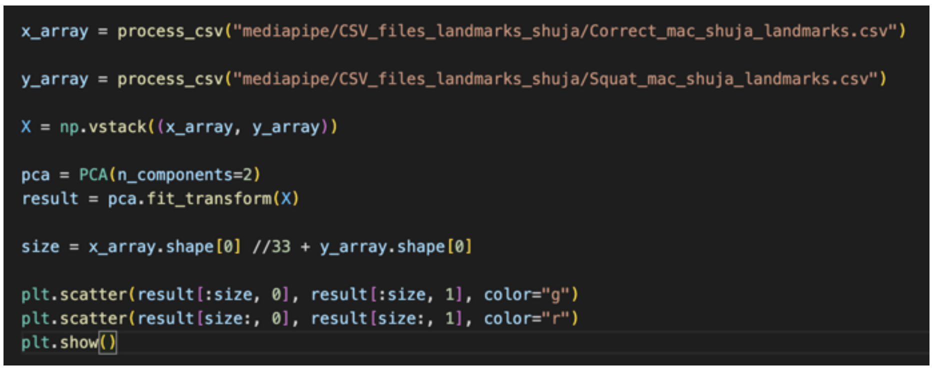
We applied the concept of PCA to the CSV files with Python code that processes one correct execution and one incorrect execution, followed by applying PCA on the two datasets and then plotting the data in a coordinate system. A couple of imports were necessary, such as Pandas and NumPy.



```python
def process_csv(csv_filename):
    #read csv without the header
    X = pd.read_csv(csv_filename, header=None)
    #create numpy array, after the comma decide which columns to omit
    array = np.array(X)[:, 1:]
    #create amount of frames by dividing the shape with amount of landmarks
    frames = (array.shape[0] // 33)
    #reshape array by multiplying by the amount of axis u want
    array = array.reshape(frames,132)
    return array
```

Figure 15: Python code to process csv files into numpy array

Figure 15 shows code that takes a CSV file as a parameter and reads it without the header. To apply PCA, a NumPy array is preferred and not the standard Python array, so we converted the array into a NumPy array with column 1 omitted, which is the index column. For future-proofing of the code, more columns can be omitted or included depending on what data is chosen to be included from the CSV files. Afterward, we calculate the number of frames for each CSV file by dividing the size of the array with the number of landmarks included since each frame contains the total number of landmarks which in the case of the picture is 33. The array is then finally reshaped to be the size of the frames variable calculated above and amount of landmarks multiplied by the amount of axis to be included. This step is important so the arrays can be stacked on top of each other into a final array later on since they both are reshaped into (frames, 132). This code snippet features the first step of processing the CSV files and is followed by plotting the data.



```
x_array = process_csv("mediapipe/CSV_files_landmarks_shuja/Correct_mac_shuja_landmarks.csv")

y_array = process_csv("mediapipe/CSV_files_landmarks_shuja/Squat_mac_shuja_landmarks.csv")

X = np.vstack((x_array, y_array))

pca = PCA(n_components=2)
result = pca.fit_transform(X)

size = x_array.shape[0] //33 + y_array.shape[0]

plt.scatter(result[:size, 0], result[:size, 1], color="g")
plt.scatter(result[size:, 0], result[size:, 1], color="r")
plt.show()
```

Figure 16: Python code to plot the correct and incorrect execution

To make the code in Figure 16 possible, we required the Scikit-learn import to apply the PCA with an already implemented solution that transforms the final array using PCA. Furthermore, the Matplotlib library also played a big part in enabling plotting in Python.

The Python code example in Figure 16 depicts the processing of the correct and incorrect datasets, which are then stacked to make a final array (X) consisting of both the x and y arrays. We created the PCA variable with components set to 2, which reduces the dimensions of the result to 2 dimensions. We did this to enable plotting the CSV files into a 2-dimensional graph. Afterward, the result variable is created by applying the fit transform to the final array. We calculated the size of where the split between the correct and incorrect amount of dots, and the graph is plotted with green representing correct dots and red representing incorrect dots.

The goal of plotting the datasets with PCA applied consists of determning whether or not there is a clear difference between correctly executed exercises and incorrectly executed exercises. To fully grasp the dataset, we applied PCA to reduce the dimensions, followed by a visual representation showing the difference in the plots created. Before applying PCA and plotting in a 2-dimensional graph, the sheer amount of dimensions in the dataset was almost impossible to comprehend and compare without any visual representation or dimension reduction.

3 Analysis

3.1 Model Complexity

An essential part of the video analysis was the model complexity. This determines the precision of the Mediapipe Pose Estimation tracking of landmarks and has three different settings; Model complexity 0, 1 and 2. The accuracy of the landmarks increases as the model complexity goes up. Furthermore, when we increased the model complexity, the inference latency is also increased. This in turn led to a slower computation and conversion to the CSV files with the data but the improved landmark accuracy was imperative to our research. However, to get the largest amount of starting data, all of the videos were processed with all of the 3 different complexities. After processing the video data and plotting the results with different model complexities, it was clear that plots of higher complexity resulted in a more condensed cluster. This led to the decision, that only model complexity 2 for all the videos would be useful for the rest of the project.

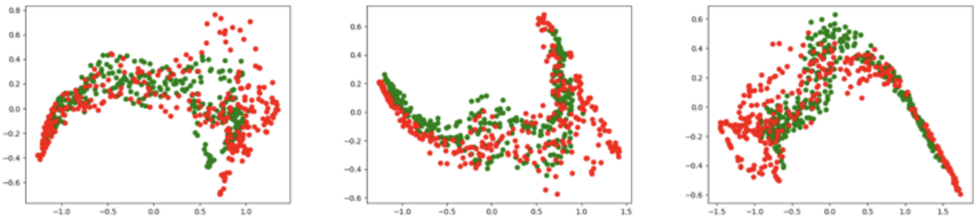


Figure 17: Model Complexities (0,1,2)

3.2 Dimension reduction

The first step in analyzing the pose estimation, applied to the different videos, was plotting all data. The first plot consists of a scatter of green and red dots, where each green dot represents a single frame from a correct execution of a deadlift, and each red dot represents a single frame of an incorrect execution deadlift. We analyzed if there is a clear difference or variation in the clusters of the plots. The first plot has all 132 dimensions consisting of the 33 landmarks that each have 4 features. It is then reduced into two dimensions and plotted on the graph. For each incorrect movement, there exists a corresponding correct movement from the same camera and angle of the same individual. This is visualized in Figure 18.

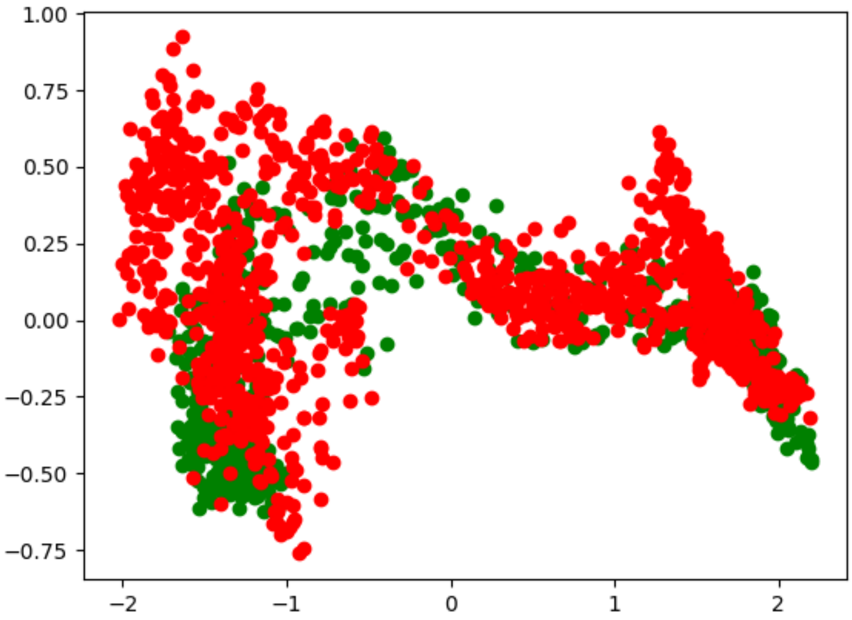


Figure 18: Plot arch, module complexity 2

It was imperative to remove unnecessary data from the CSV files to reduce the number of dimensions which reduces the input size for the machine learning model. We had a biased perception that the visibility values would have minimal influence on the plot. We made new plots to test this statement, but this time without the visibility values. The resulting plots look strikingly similar to the original plots, leading us to conclude that dropping visibility indeed makes no difference. The removal of visibility decreases the number of dimensions per frame from 132 to 99, making a significant difference in computation speed and accuracy. To implement this in the Python code, the modular process CSV method provided an easy way to omit columns such as visibility by simply changing the value after the comma.

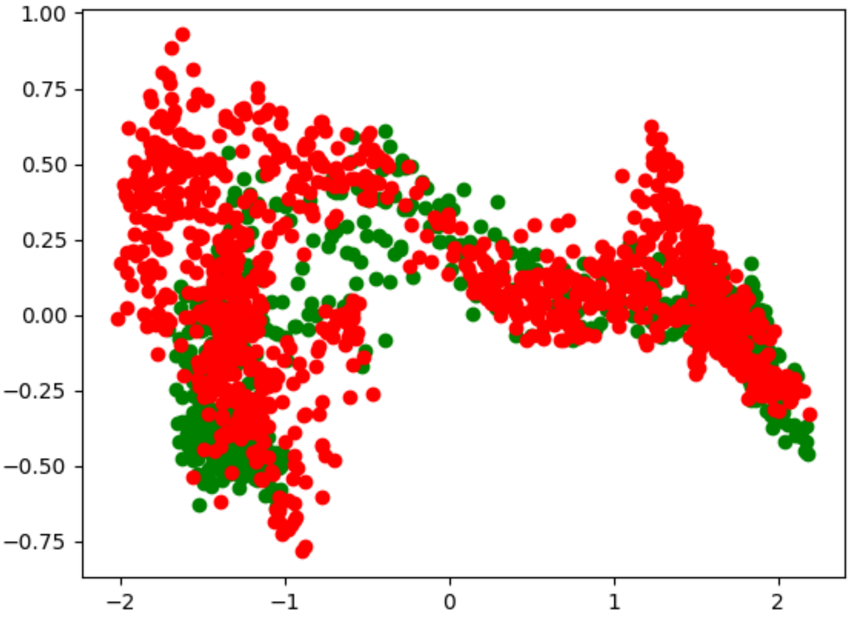


Figure 19: Plot arch, without visibility

Since removing the visibility values had such an insignificant impact on the plots, further enhancement to the dimension reduction was possible. The information gathered from the interview with the personal trainer led to the conclusion that several landmarks are without importance when analyzing a deadlift. When observing the movements presented by the personal trainer we interviewed, it is clear that only the torso, legs, and arms are interested when considering which landmarks to include. This is because the feet are planted on the ground throughout the movement, and the hands will be on the bar. They remain static throughout the execution, and it is impossible to tell if a participant is making mistakes simply by observing the feet or hands. The same goes for the head and its many landmarks, such as eyes, nose, and mouth. By reducing the number of landmarks, each frame would only consist of concise data, resulting in a faster execution of the machine learning model. This was the final amount of dimension reduction conducted on the data set. Therefore we created a new plot, where all landmarks except the torso, arms, and legs were left out, depicting the final plot for the machine learning model.

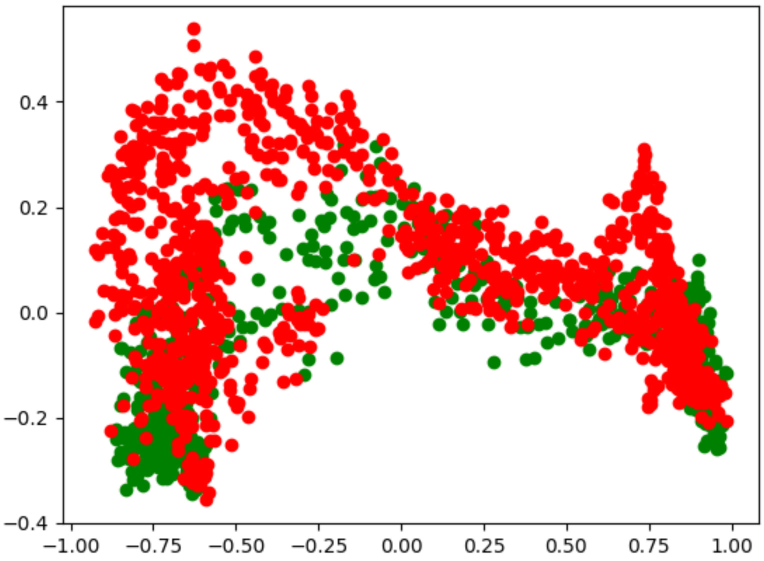
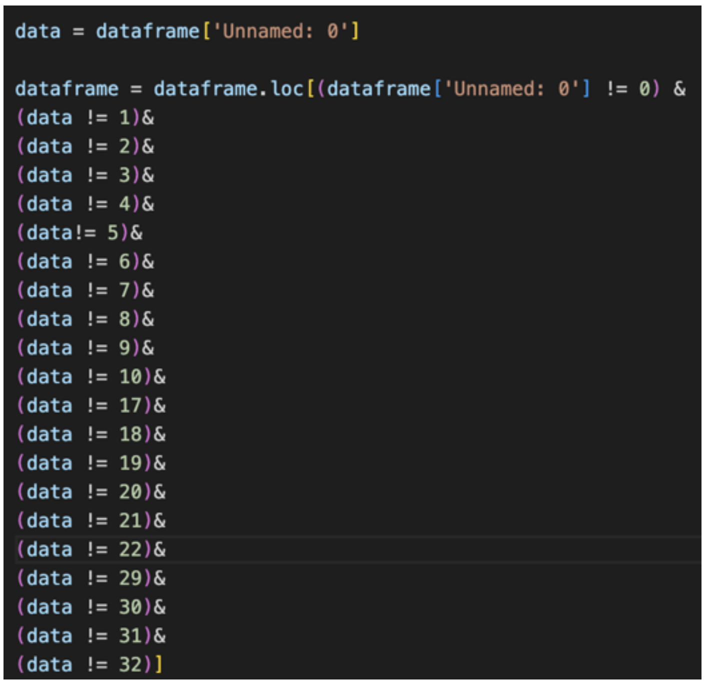


Figure 20: Plot arch, without visibility and insignificant landmarks

Limiting the landmarks to consist only of the 12 most essential landmarks was implemented in the Python code in Figure 21.



```
data = dataframe['Unnamed: 0']

dataframe = dataframe.loc[(dataframe['Unnamed: 0'] != 0) &
(data != 1)&
(data != 2)&
(data != 3)&
(data != 4)&
(data!= 5)&
(data != 6)&
(data != 7)&
(data != 8)&
(data != 9)&
(data != 10)&
(data != 17)&
(data != 18)&
(data != 19)&
(data != 20)&
(data != 21)&
(data != 22)&
(data != 29)&
(data != 30)&
(data != 31)&
(data != 32)]
```

Figure 21: implementation in python to remove unnecessary landmarks

By using the built-in data frame, we removed all the landmarks which were not necessary by finding each landmark under the column ‘Unnamed: 0‘ which was the index column and then removing the landmarks that were not needed. This resulted in a CSV file with only the landmarks 11-16 and 23-28, which were the crucial landmarks to analyze in the final step of the data processing.

3.3 Analysis of plots

3.3.1 Common Mistake 1: Squat

This plot contains a correct execution of a deadlift plotted in green, against an incorrect execution where the participant purposely made the mistake of executing the movement as a squat, plotted in red.

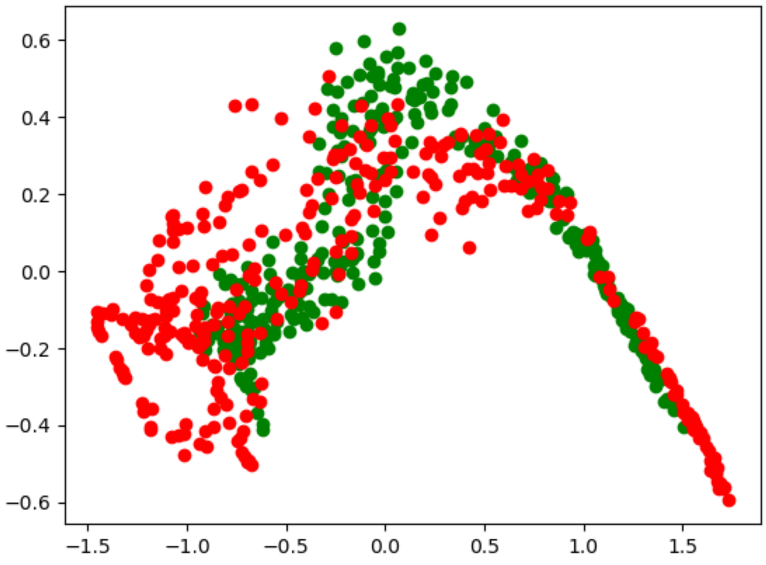


Figure 22: Final plot of arms mistake with visibility and unimportant landmarks removed

With this mistake, the hips maintained a vertical angle at the start and middle section of the lift, but the movement ended the same way. This can be seen on the graph, as the red and green dots differ at the left and middle but are very similar in the end.

3.3.2 Common Mistake 2: Arms

Figure 23 illustrates a correct execution of a deadlift plotted in green against an incorrect execution where the participant purposely made the mistake of keeping the bar away from the body throughout the movement, plotted in red.

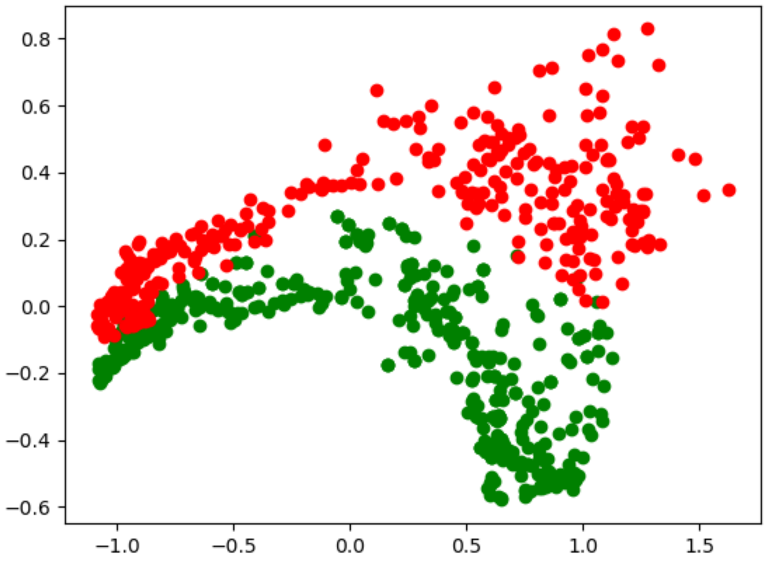


Figure 23: Final plot of arms mistake with visibility and unimportant landmarks removed

As expected, the plots are very similar, as the torso and legs are positioned similarly throughout the movement. However, the different arms position might be the explanation for the red outlying cluster of dots seen in the top right of the graph.

3.3.3 Common Mistake 3: Hyperextension

Figure 24 shows a correct execution of a deadlift plotted in green against an incorrect execution where the participant purposely made the mistake of hyper-extending the back at the end of the movement, plotted in red.

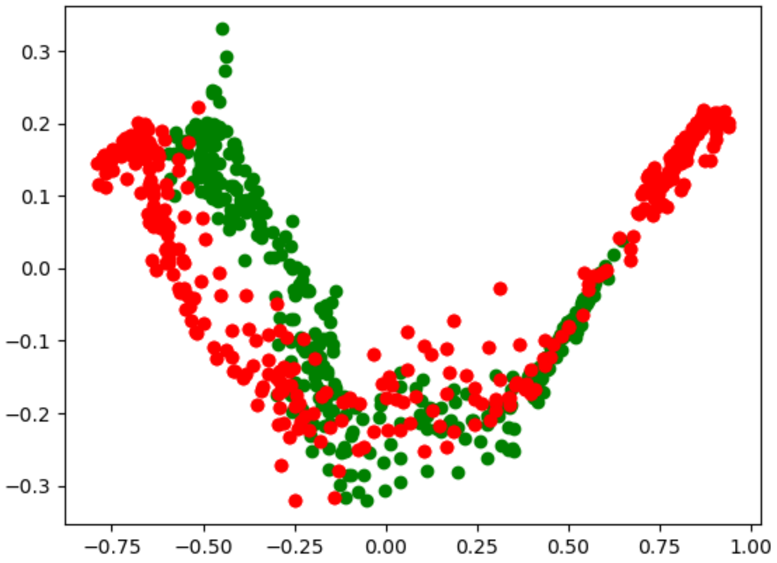


Figure 24: Final plot of arms mistake with visibility and unimportant landmarks removed

The hyper-extension can be visualized to the right of the graph, as the individual will continue the exercise, by extending the back at the end of the lift, instead of ending it. There is a red cluster of dots on the right that shows this correlation.

3.3.4 Common Mistake 4: Arch

Finally, Figure 25 presents a correct execution of a deadlift plotted in green against an incorrect execution where the participant purposely made the mistake of arching the lower back throughout the movement, plotted in red.

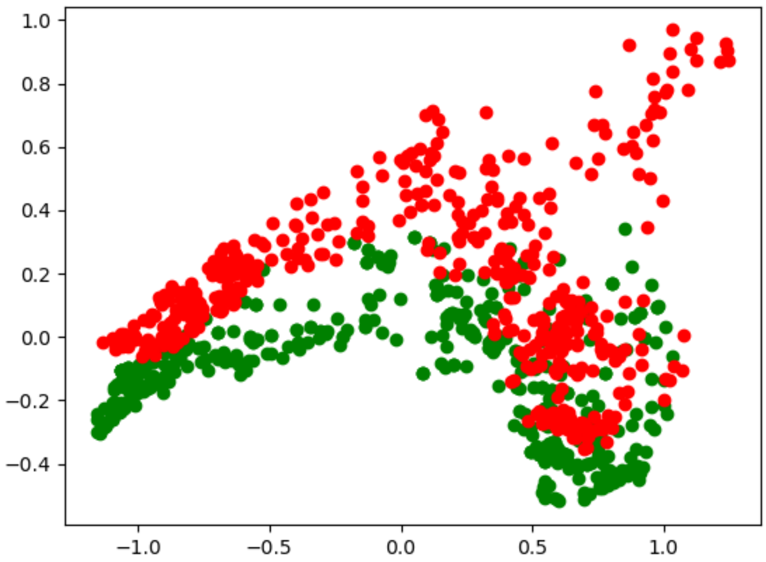


Figure 25: Plot of arch mistake with visibility and unimportant landmarks removed

The composition of red and green dots are very similar, as the movement is so similar, however there is a cluster of red dots in the top right corner, which are not there for the correct movement. The arching happens in the mid position of the movement, which matches the cluster quite well.

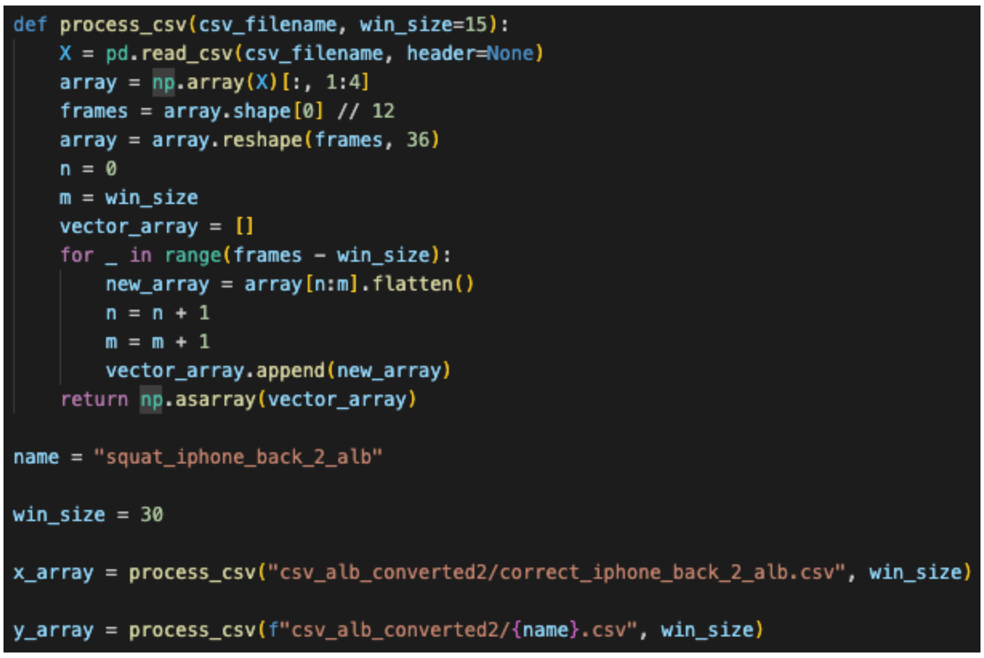
3.4 Window of frames

The first step was creating plots where each dot represented one frame of video plotted on the graph. However, most movements require a more extended frame window to analyze adequately. To see the difference between the lifting movement of the incorrect and the correct, it is imperative to analyze the videos with a bigger window of frames to get a coherent vector of dots representing a fluid body movement compared to the static 1 dot = 1 frame procedure done before.

The code to produce the PCA plots were altered to depict each window of frames increasing by one each dot, to create a fluid vector that represents a movement rather than still frames. The first dot represented frame 1-15 and the next dot represented frame 2-16 and so on, which in turn led to a much more clear distinction between incorrect and correct movement. By incorporating a window of frames instead of single frames the plots will increase in readability in terms of precisely finding the difference in right or wrong execution. Furthermore, the newly created plot, shows the movement as a vector over time rather than a still frame plotted on the graph.

We altered the code to produce the PCA plots to depict each window of frames increasing by one each dot, to create a fluid vector that represents a movement rather than still frames. The first dot represented frames 1-15, and the following dot represented frames 2-16, and so on, which led to a more apparent distinction between incorrect and correct movement. By incorporating a window of frames instead of single frames, the plots will increase in readability to precisely find the difference in right or wrong execution. Furthermore, the newly created plot shows the movement as a vector over time rather than a still frame plotted on the graph. The code is shown in figure 26.

Figure 26 shows our code to create these windows of frames. It takes as input a csv file and window frame size, which is 15 in our case. It creates an array containing pose detection information from all frames of the csv file being read. Then it makes two variables, n which is 0 and m which is the size of the window of frames. Then it loops through the entire dataset, and creates arrays containing information from frames n-m which is 1-15, and then increments the variables to 2-16, creates a new array and so on. Thereby, all these arrays containing pose detection information are appended into the final array.



```python
def process_csv(csv_filename, win_size=15):
    X = pd.read_csv(csv_filename, header=None)
    array = np.array(X)[:, 1:4]
    frames = array.shape[0] // 12
    array = array.reshape(frames, 36)
    n = 0
    m = win_size
    vector_array = []
    for _ in range(frames - win_size):
        new_array = array[n:m].flatten()
        n = n + 1
        m = m + 1
        vector_array.append(new_array)
    return np.asarray(vector_array)

name = "squat_iphone_back_2_alb"

win_size = 30

x_array = process_csv("csv_alb_converted2/correct_iphone_back_2_alb.csv", win_size)

y_array = process_csv(f"csv_alb_converted2/{name}.csv", win_size)
```

Figure 26: Python code to create plots with a window of frames

The first analysis step was creating a window of frames of size 30, which is a 1-second video for each incorrect movement. The plot consists of the correct movement in green and the incorrect movement in red on the same camera and test subject.

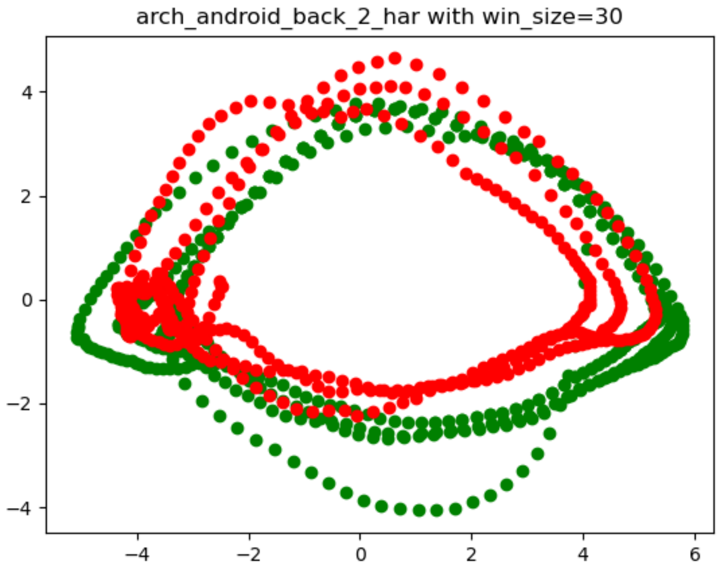


Figure 27: Plot arch, window of frames = 30

This led to inconclusive PCA plots because the window of frames was 1 second each and the shape of the two movements seemed very similar, thereby depicting a minimal distinction between the correct and incorrect movement. We changed the window to 15 frames to further enhance the coherence of the incorrect and correct movements, which led to much more conclusive results. It is clear to see the sequence of incorrect movements and how it diverges from the correct movements when the window of frames is shortened and leads to a more precise depiction, as shown in Figure 28.

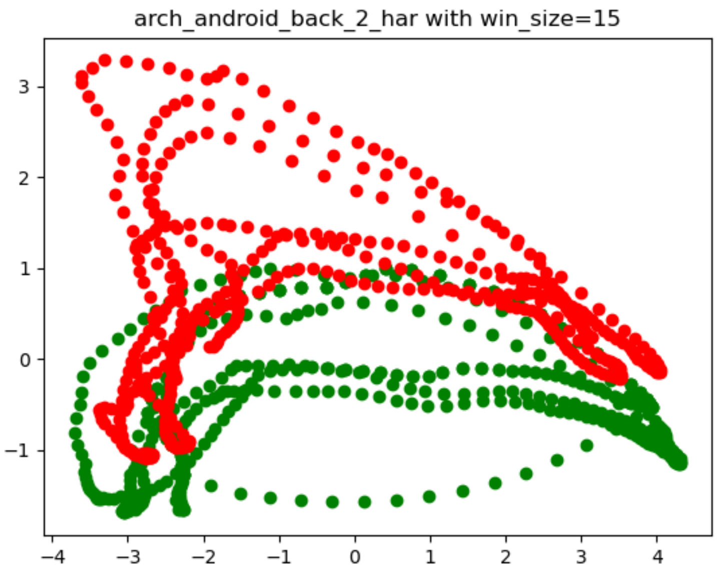


Figure 28: Plot arch, window of frames = 15

3.5 Normalization

By looking at the plots for the window of frames in Figure 29, it is clear to see the offset of dots in the graphs. The overall shape difference in terms of the vectors are portrayed correctly but can be shifted upwards of each other, as shown in the Figure 29.

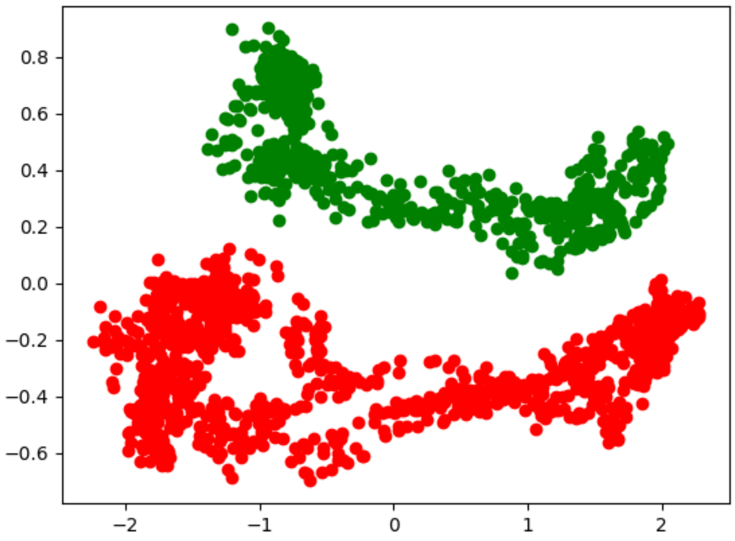


Figure 29: Plot arch, offset plotting

The data, which are coordinate values of x, y, and z, is processed into PCA plots and creates this offset. Therefore, the plots of the correct and incorrect frames can have the correct shape in terms of difference, but be offset. We used different camera angles for the different test subjects, which can alter the positional coordinate values and result in an offset. The subjects were filmed and the plots portray the data of the video by position. The different camera angles, and the distance between the camera and test subject also plays a part in this offset. The offset between correct and incorrect makes the visualization harder to grasp and compare since the green and red dots are not intertwining.

3.6 Machine Learning

Machine learning is a branch of artificial intelligence focusing on using data and algorithms to imitate learning, gradually improving accuracy. Machine learning relies on input, such as training data, to gradually attain knowledge about entities, domains, and their connections.

With machine learning, data is observed to recognize patterns. The goal of machine learning is to allow computers to learn autonomously without the need for human assistance and intervention. A well-trained machine learning algorithm can predict data based on the tendencies it has picked up from its input. The quality and quantity of the data will be essential for the accuracy of these predictions. Machine learning is valuable because the speed and scale of its predictions will surpass that of any human mind. A neural network is the name of the software solution that leverages machine learning algorithms to make predictions on data. A neural network takes data as input, processes it through layers of nodes, and gives recognition as an output. The traditional neural network was the first type [12].

3.6.1 LSTM

A traditional neural network does not allow information to persist, which is why a recurrent neural network is preferred. The neural network can use loops to pertain information in the next state. Recurrent networks are helpful when the gap between the relevant information and the result of looping through that information to produce a prediction is small. However, recurrent neural networks cannot predict, for example, auto-completed sentences that require long-term memory. An example could be if the following suggestion of a word prediction requires context from the previous sentence, which renders recurrent neural networks useless when the gap grows [13].

The inability to learn long-term dependencies caused the introduction of LSTM (Long Short Term Memory) networks in 1997 by Hochreiter Schmidhuber. This type of neural network handles long-term dependency by default and is an overhaul to the dated recurrent neural network form. The output of an LSTM is dependent on three things: The cell state, which is the current long-term memory of the network. The hidden state is the output at the previous point in time and the input data at the current time. A series of gates are used to control the sequence of data and its flow of information. A typical LSTM has three gates; forget gate, input gate, and output gate. They are neural networks and work more like a filter for the data processed through the network [14].

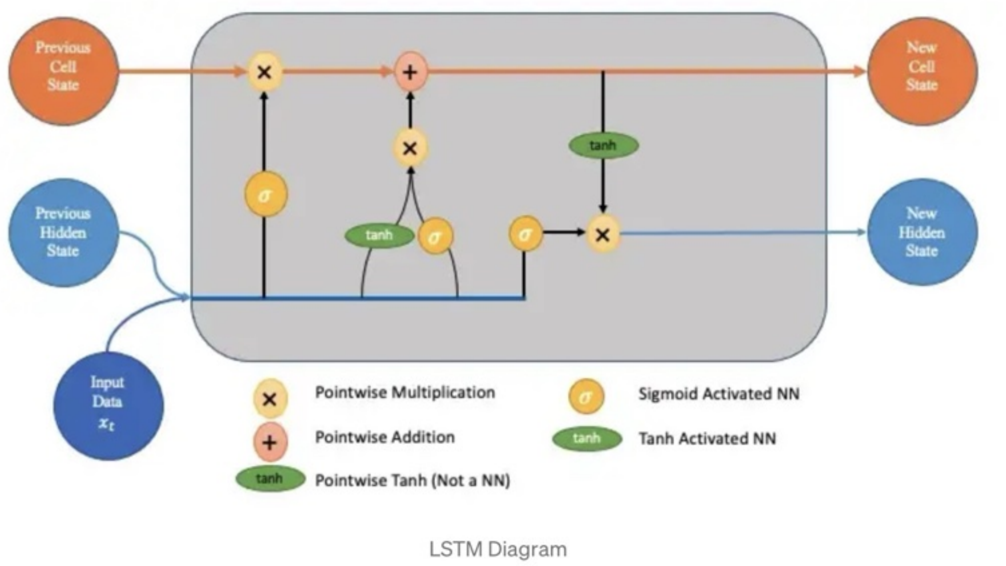


Figure 30: An LSTM diagram

Figure 30 shows an overview of how the LSTM network functions.

Step 1 of the process is the forget gate. This gate represents the decision on which bits of the network's cell state/long-term memory are helpful with the given previous hidden state and new input data. The neural network is then fed the new input data and the previous hidden state. Using the sigmoid activation, the network will then generate a vector for each element between the interval [0,1]. This interval represents the relevancy of the element's input where a value is more relevant the closer to 1 it gets. This is why it acts as a filter, allowing more information to pass through this gate if the value is closer to 1. The values are then pointwise multiplied with the previous cell state, which results in multiplying by a number close to 0 and keeps the value of irrelevant components close to 0. This in turn leads to less influence in the following gates. To summarize, this gate decides which pieces of long-term memory have the most weight. It also retains the most weight while forgetting the pieces with the least weight.

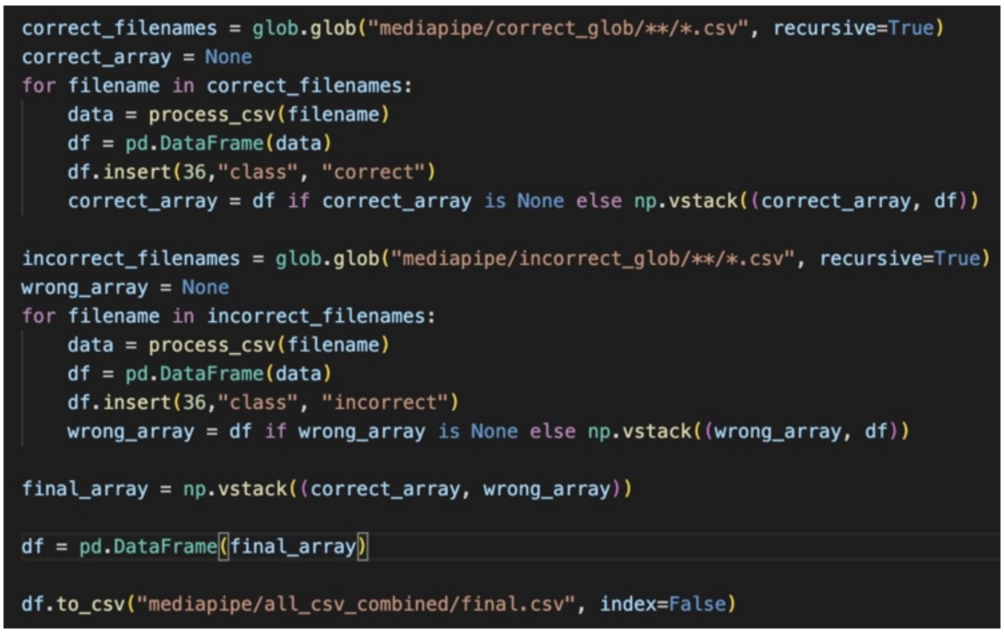
Step 2 is the input gate and involves the new memory network. This step is essential in deciding what new information is added to the network's long-term memory, given the previous hidden state and new input data. The inputs are the same as in the forget gate. However, the process differs by determining which new information should be added to long-term memory rather than what information is relevant. The new memory network consists of a tanh activated neural network that combines the previous hidden state and the new input data to create a new memory update vector. The Tanh function is relevant here because the values lie in $[-1, 1]$ because the negative values are necessary to reduce an element in the cell state. The vector created uses the new input to update each component in the cell state with the vector value created.

The creation of the new memory vector has a big problem, however. The vector needs to consider whether or not the new input data is worth about and transferred to the next step. The input gate is then activated, which acts like the forget gate to filter which components of the new memory vector are worth passing on to the next step. It uses pointwise multiplication to determine whether or not the output of the new memory vector through the input gate is near zero, which results in not updating that element in the cell state. Both the outputs of tanh and sigmoid activated parts are pointwise multiplied and again regulate the output closer to 0 if deemed irrelevant. The final step of updating is using pointwise multiplication on the combined vector and adding it to the cell state, which updates the long-term memory.

The output gate is the final gate of the neural network, which decides the new hidden state. Three things are used to compute this state, the updated cell state, the previous hidden state, and the new input data. The output gate uses sigmoid activation function. This gate is applied to the newly updated cell state, ensuring that all necessary information is saved to the new hidden state. The cell state is passed through a tanh function, forcing the values into the interval [-1,1]. The sigmoid activation function takes any x-axis coordinate and turns it into a y-axis coordinate between 0 and 1. The tanh, or hyperbolic tangent, activation function takes any x-axis coordinate and turns it into a y-axis coordinate between -1 and 1.

3.6.2 Creating the training data

The training data for the LSTM prediction model consisted of the previously created CSV files. This included both the correct and incorrect execution data, and we created a Python script to prepare the input data for the model.



```
correct_filenames = glob.glob("mediapipe/correct_glob/**/*.csv", recursive=True)
correct_array = None
for filename in correct_filenames:
    data = process_csv(filename)
    df = pd.DataFrame(data)
    df.insert(36,"class", "correct")
    correct_array = df if correct_array is None else np.vstack((correct_array, df))

incorrect_filenames = glob.glob("mediapipe/incorrect_glob/**/*.csv", recursive=True)
wrong_array = None
for filename in incorrect_filenames:
    data = process_csv(filename)
    df = pd.DataFrame(data)
    df.insert(36,"class", "incorrect")
    wrong_array = df if wrong_array is None else np.vstack((wrong_array, df))

final_array = np.vstack((correct_array, wrong_array))

df = pd.DataFrame(final_array)

df.to_csv("mediapipe/all_csv_combined/final.csv", index=False)
```

Figure 31: Python script to prepare csv files for the prediction model

The python code in Figure 31 shows functions by adding a 36^{th} column to every CSV file. The correct CSV files will have the "correct" class, and every incorrect CSV file will have the "incorrect" class as their 36^{th} column. This is due to the binary distribution of the prediction model since the model is supposed only to distinguish correct from incorrect. The correct and incorrect arrays are then stacked on top of each other with np.vstack and converted to a DataFrame. The final step consists of converting the DataFrame into a CSV file, the final file used to train the prediction model.

We then create the final.csv file, but more preprocessing is still required before the training of the model can begin. This consisted of creating a row called series_id which determines the interval of frames used in the model itself. The first interval chosen was 30 frames which then resulted in the combined30.csv file which looks like the picture below in Figure 32.

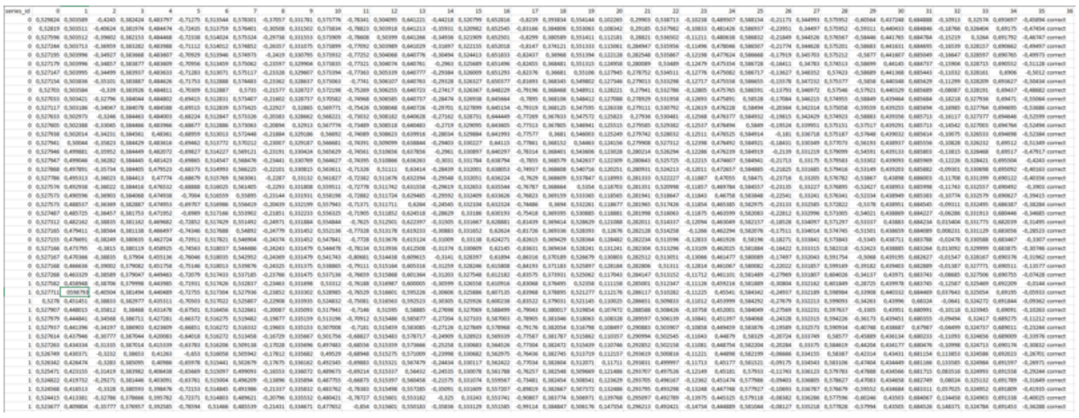
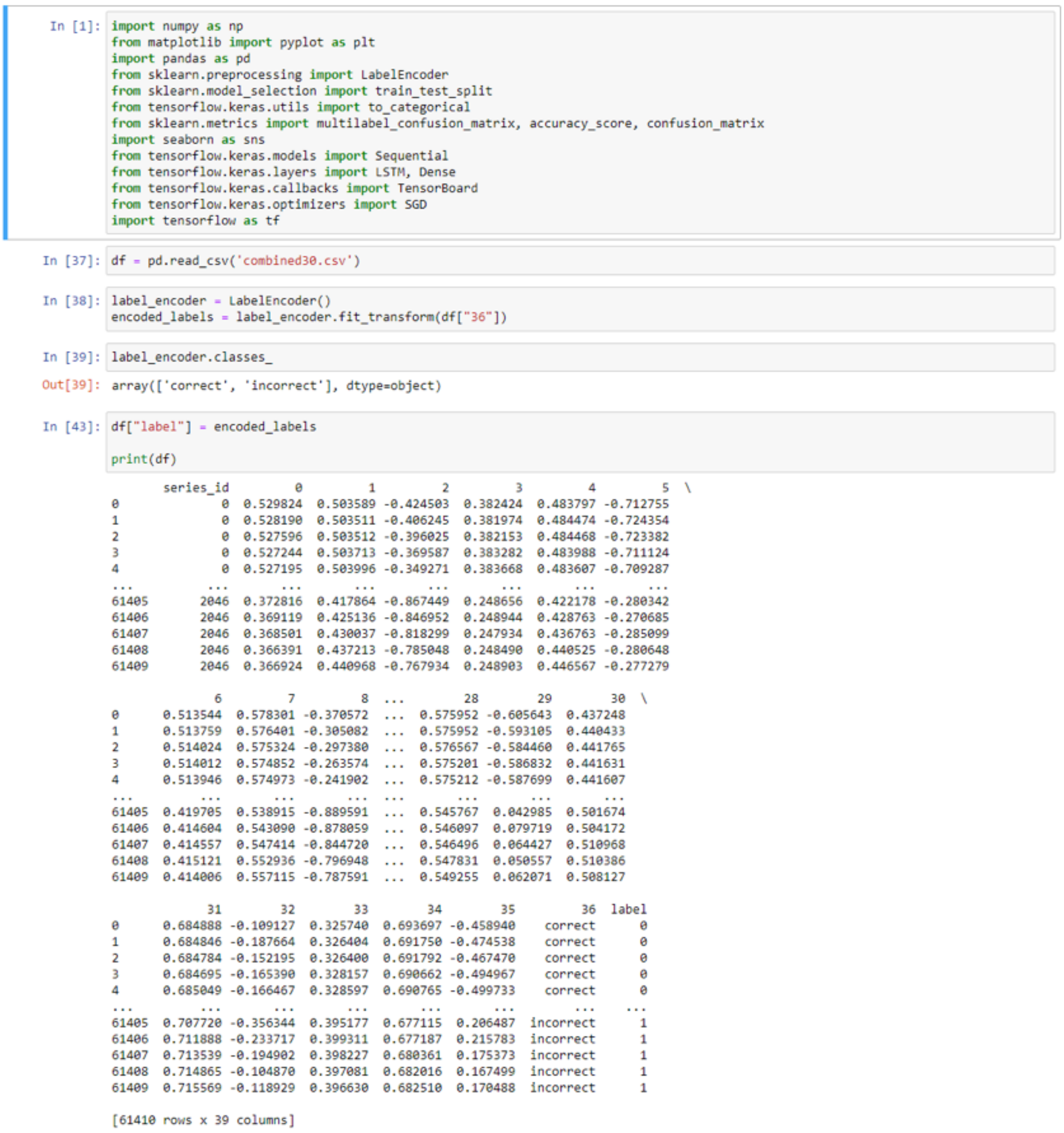


Figure 32: combined30.csv example

Figure 32 visualizes how the data looks with the interval being set to 30 frames. The data preprocessing can now begin.

3.6.3 Preparing the training data

Jupyter Notebook is a web-based interactive computing platform that provides live python code execution. It was the preferred platform for preparing, training, and showing the model results because of its interactive nature.[15] First, we turn the the previously created combined30.csv file into a pandas DataFrame object, providing easier manipulation than a raw CSV file. The first step of preparing the data is creating a label for each correct and incorrect class, with 0's representing the correct class and 1's representing the incorrect class.



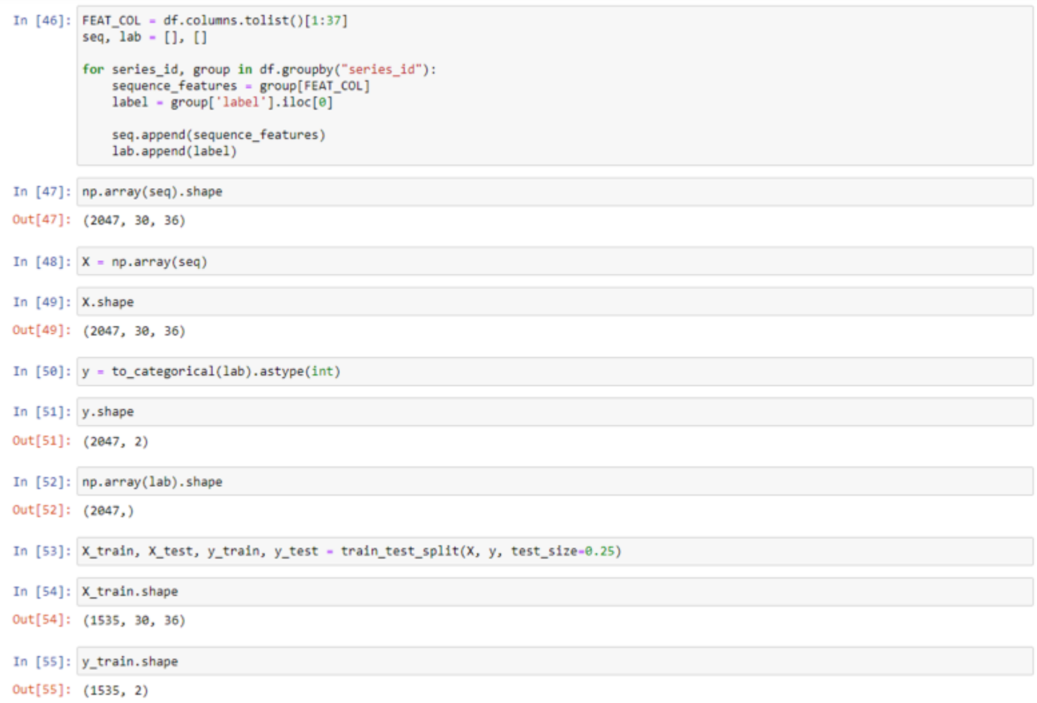
```
In [1]:  import numpy as np
         from matplotlib import pyplot as plt
         import pandas as pd
         from sklearn.preprocessing import LabelEncoder
         from sklearn.model_selection import train_test_split
         from tensorflow.keras.utils import to_categorical
         from sklearn.metrics import multilabel_confusion_matrix, accuracy_score, confusion_matrix
         import seaborn as sns
         from tensorflow.keras.models import Sequential
         from tensorflow.keras.layers import LSTM, Dense
         from tensorflow.keras.callbacks import TensorBoard
         from tensorflow.keras.optimizers import SGD
         import tensorflow as tf

In [37]: df = pd.read_csv('combined30.csv')

In [38]: label_encoder = LabelEncoder()
         encoded_labels = label_encoder.fit_transform(df["36"])

In [39]: label_encoder.classes_

Out[39]: array(['correct', 'incorrect'], dtype=object)

In [43]: df["label"] = encoded_labels

         print(df)
```

```
       series_id         0         1         2         3         4         5  \
0              0  0.529824  0.503589 -0.424503  0.382424  0.483797 -0.712755   
1              0  0.528190  0.503511 -0.406245  0.381974  0.484474 -0.724354   
2              0  0.527596  0.503512 -0.396025  0.382153  0.484468 -0.723382   
3              0  0.527244  0.503713 -0.369587  0.383282  0.483988 -0.711124   
4              0  0.527195  0.503996 -0.349271  0.383668  0.483607 -0.709287   
...          ...       ...       ...       ...       ...       ...       ...   
61405       2046  0.372816  0.417864 -0.867449  0.248656  0.422178 -0.280342   
61406       2046  0.369119  0.425136 -0.846952  0.248944  0.428763 -0.270685   
61407       2046  0.368501  0.430037 -0.818299  0.247934  0.436763 -0.285099   
61408       2046  0.366391  0.437213 -0.785048  0.248490  0.440525 -0.280648   
61409       2046  0.366924  0.440968 -0.767934  0.248903  0.446567 -0.277279   

              6         7         8  ...        28        29        30  \
0      0.513544  0.578301 -0.370572  ...  0.575952 -0.605643  0.437248   
1      0.513759  0.576401 -0.305082  ...  0.575952 -0.593105  0.440433   
2      0.514024  0.575324 -0.297380  ...  0.576567 -0.584460  0.441765   
3      0.514012  0.574852 -0.263574  ...  0.575201 -0.586832  0.441631   
4      0.513946  0.574973 -0.241902  ...  0.575212 -0.587699  0.441607   
...         ...       ...       ...  ...       ...       ...       ...   
61405  0.419705  0.538915 -0.889591  ...  0.545767  0.042985  0.501674   
61406  0.414604  0.543090 -0.878059  ...  0.546097  0.079719  0.504172   
61407  0.414557  0.547414 -0.844720  ...  0.546496  0.064427  0.510968   
61408  0.415121  0.552936 -0.796948  ...  0.547831  0.050557  0.510386   
61409  0.414006  0.557115 -0.787591  ...  0.549255  0.062071  0.508127   

             31        32        33        34        35         36  label  
0      0.684888 -0.109127  0.325740  0.693697 -0.458940    correct      0  
1      0.684846 -0.187664  0.326404  0.691750 -0.474538    correct      0  
2      0.684784 -0.152195  0.326400  0.691792 -0.467470    correct      0  
3      0.684695 -0.165390  0.328157  0.690662 -0.494967    correct      0  
4      0.685049 -0.166467  0.328597  0.690765 -0.499733    correct      0  
...         ...       ...       ...       ...       ...        ...    ...  
61405  0.707720 -0.356344  0.395177  0.677115  0.206487  incorrect      1  
61406  0.711888 -0.233717  0.399311  0.677187  0.215783  incorrect      1  
61407  0.713539 -0.194902  0.398227  0.680361  0.175373  incorrect      1  
61408  0.714865 -0.104870  0.397081  0.682016  0.167499  incorrect      1  
61409  0.715569 -0.118929  0.396630  0.682510  0.170488  incorrect      1  

[61410 rows x 39 columns]
```

Figure 33: Labelling code with example

As shown in Figure 33, we added all of the necessary imports to the CSV file and then we create the Pandas DataFrame object from the previously mentioned CSV file. The labels column are correctly asserted with 0's and 1's in the corresponding class.

The final step of preparing the data is splitting the X and y arrays into X_train, X_test, y_train and y_test. This is the data used in the actual training of the model.



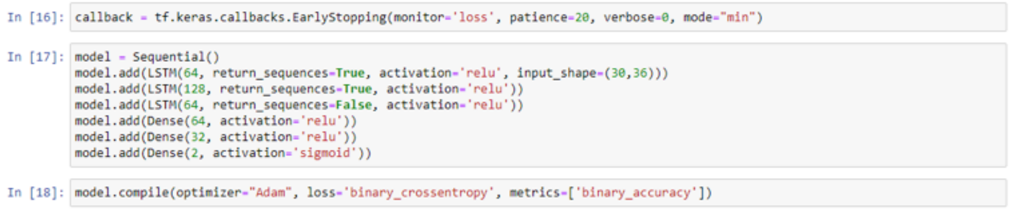
```
In [46]: FEAT_COL = df.columns.tolist()[1:37]
         seq, lab = [], []

         for series_id, group in df.groupby("series_id"):
             sequence_features = group[FEAT_COL]
             label = group['label'].iloc[0]

             seq.append(sequence_features)
             lab.append(label)

In [47]: np.array(seq).shape

Out[47]: (2047, 30, 36)

In [48]: X = np.array(seq)

In [49]: X.shape

Out[49]: (2047, 30, 36)

In [50]: y = to_categorical(lab).astype(int)

In [51]: y.shape

Out[51]: (2047, 2)

In [52]: np.array(lab).shape

Out[52]: (2047,)

In [53]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25)

In [54]: X_train.shape

Out[54]: (1535, 30, 36)

In [55]: y_train.shape

Out[55]: (1535, 2)
```

Figure 34: final step of preprocessing the data

In Figure 34, we wrote the code for the X and y in the model. First, we assigned the variable FEAT COL to all the landmark data columns. Two empty arrays, seq, and lab, were created as X and y, respectively. Then the for loop was grouped by series_id to get the frame interval instead of single frames. The sequence features variable was assigned all the landmark data, while the label variable took the first value of the label row and appended it to the empty array seq and lab. To visualize the data, we output the shapes of seq and x, which correctly shows that they are identical. Afterward, we created the by using the to categorical.astype(int) method, which returned a 2-dimensional array that complies with the model. At last, the X and y was split into the test size 0.25 which splits the X and y array into 75% of train and 25% of test.

3.6.4 Creating the model

After the preprocssing of the data, it was ready to be inserted into the training model, many factors play a role in creating it.



```
In [16]: callback = tf.keras.callbacks.EarlyStopping(monitor='loss', patience=20, verbose=0, mode="min")

In [17]: model = Sequential()
         model.add(LSTM(64, return_sequences=True, activation='relu', input_shape=(30,36)))
         model.add(LSTM(128, return_sequences=True, activation='relu'))
         model.add(LSTM(64, return_sequences=False, activation='relu'))
         model.add(Dense(64, activation='relu'))
         model.add(Dense(32, activation='relu'))
         model.add(Dense(2, activation='sigmoid'))

In [18]: model.compile(optimizer="Adam", loss='binary_crossentropy', metrics=['binary_accuracy'])
```

Figure 35: Creating the LSTM model

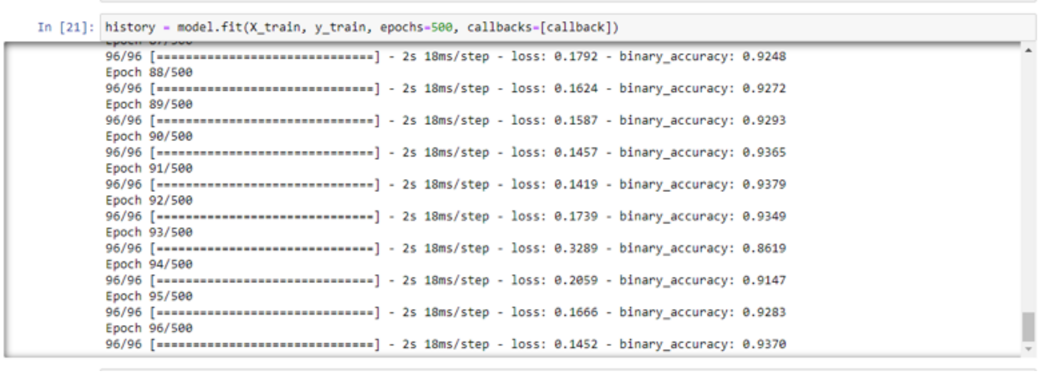
First, we define the callback method chosen for the specific model. In the case of this project, the TensorFlow EarlyStopping callback was the preferred option since it allows the monitoring of a specific Quantity [16]. We chose the monitoring of loss as the quantity in the LiftIT project with accompanying patience that determines the number of epochs with no improvement to the monitor before it stops training. In the case of this project, the patience parameter was 20. The usual interval is between 10-100 in most projects, but running the model with lower patience resulted in unusable results while increasing the patience kept the model training forever. The verbose option is 0, which is silent, while set to 1 will display when the callback takes action. Lastly, the mode is set to min since the model should stop training when the Quantity monitor stops decreasing, or in other terms, when there is no improvement to loss.

The next step includes defining the model itself by first defining it as a sequential model that enables quickly adding a stack of linear layers to the prediction model. The model API is defined, and it is then time to add the model layers one by one. The first layer adds an LSTM layer with 64 units which define the dimension of hidden states in the layer. Since the LSTM layers have to pass the data on to the next LSTM layer in the sequential model, return sequences have to be set to true to enable the next stacked LSTM to function correctly. Then an activation function ReLu is defined, since it processes the positive weighted inputs directly or otherwise it will output zero. The last thing in this particular layer includes the input shape, which is the number of frames, in this case, 30, per prediction, and then the number of features per frame, which is 12 landmarks times 3 dimensions (x,y,z) that equals 36. The next layer added is another LSTM layer but with 128 units instead of 64, which also has return sequences set to true. The last LSTM layer, however, is not obliged to return sequences since the next layer is not an LSTM but a Dense layer. A Dense layer is a layer that is deeply connected with its preceding layer, which means the neurons of this layer are all deeply connected to every neuron of the preceding LSTM layer. The last layer of the model is another Dense layer. However, the amount of categories are specified as the unit parameter, which in this case is 2 since the project uses a binary classification for the prediction model. Furthermore, the last activation model used is sigmoid since the prediction model is a binary classification of correct/incorrect, which is suited for the sigmoid activation function.

The final step in creating the model is compiling it with an optimizer, Adam. Adam is recommended as the default optimizer for most applications and is fast and requires fewer parameters for tuning. Afterward, we define the loss function, which in the case of this project, is a binary classification, and is defined as the binary cross-entropy loss function. The last parameter is the metrics that are monitored during the training of the model itself, which is the binary accuracy displayed after each epoch.

3.6.5 Training the model - iterative approach

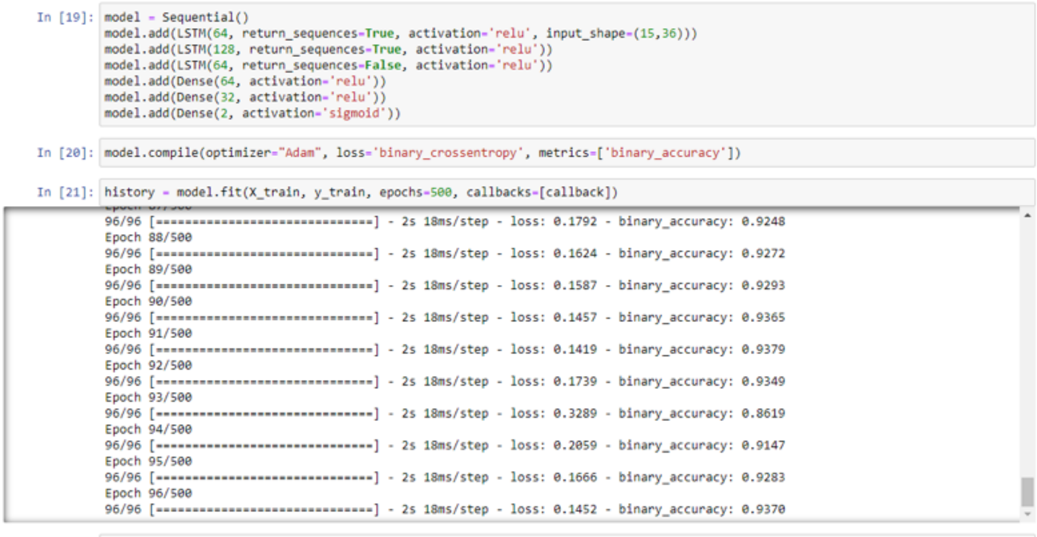
When it is time to train the model with the preprocessed data, the model.fit method includes the X_train and y_train data. The epochs are the maximum amount of iterations over the data, and is set to 500, which will stop the training after 500 iterations, however since the callback from figure 35 is the parameter, the Earlystopping callback will kick in before the maximum amount of epochs have been reached.



```
In [21]: history = model.fit(X_train, y_train, epochs=500, callbacks=[callback])
         96/96 [==============================] - 2s 18ms/step - loss: 0.1792 - binary_accuracy: 0.9248
         Epoch 88/500
         96/96 [==============================] - 2s 18ms/step - loss: 0.1624 - binary_accuracy: 0.9272
         Epoch 89/500
         96/96 [==============================] - 2s 18ms/step - loss: 0.1587 - binary_accuracy: 0.9293
         Epoch 90/500
         96/96 [==============================] - 2s 18ms/step - loss: 0.1457 - binary_accuracy: 0.9365
         Epoch 91/500
         96/96 [==============================] - 2s 18ms/step - loss: 0.1419 - binary_accuracy: 0.9379
         Epoch 92/500
         96/96 [==============================] - 2s 18ms/step - loss: 0.1739 - binary_accuracy: 0.9349
         Epoch 93/500
         96/96 [==============================] - 2s 18ms/step - loss: 0.3289 - binary_accuracy: 0.8619
         Epoch 94/500
         96/96 [==============================] - 2s 18ms/step - loss: 0.2059 - binary_accuracy: 0.9147
         Epoch 95/500
         96/96 [==============================] - 2s 18ms/step - loss: 0.1666 - binary_accuracy: 0.9283
         Epoch 96/500
         96/96 [==============================] - 2s 18ms/step - loss: 0.1452 - binary_accuracy: 0.9370
```

Figure 36: Example of training the model with 30 frames interval

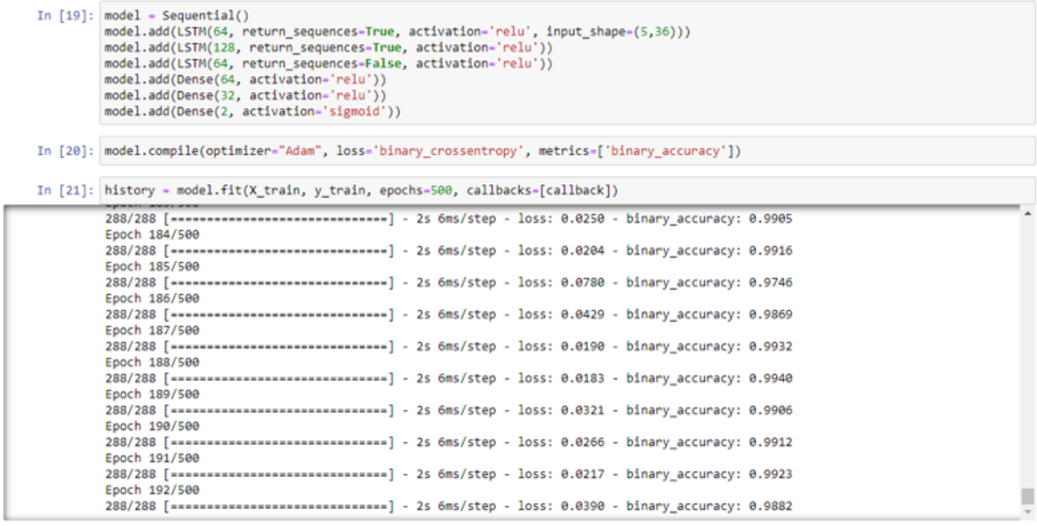
Figure 36 shows the EarlyStopping callback kicks in when the loss stops decreasing and ends up with a binary accuracy of 0,8388, which is not optimal and led to us thinking iteratively, just like when the plotting of the CSV files was done earlier in the process. Therefore, the CSV files were changed to an interval of 15 frames instead of 30 as the first iterative step. The only thing to change in the code was the reading of the CSV file and input shape specified in the first LSTM layer when creating the model.



```
In [19]: model = Sequential()
         model.add(LSTM(64, return_sequences=True, activation='relu', input_shape=(15,36)))
         model.add(LSTM(128, return_sequences=True, activation='relu'))
         model.add(LSTM(64, return_sequences=False, activation='relu'))
         model.add(Dense(64, activation='relu'))
         model.add(Dense(32, activation='relu'))
         model.add(Dense(2, activation='sigmoid'))

In [20]: model.compile(optimizer="Adam", loss='binary_crossentropy', metrics=['binary_accuracy'])

In [21]: history = model.fit(X_train, y_train, epochs=500, callbacks=[callback])
         96/96 [==============================] - 2s 18ms/step - loss: 0.1792 - binary_accuracy: 0.9248
         Epoch 88/500
         96/96 [==============================] - 2s 18ms/step - loss: 0.1624 - binary_accuracy: 0.9272
         Epoch 89/500
         96/96 [==============================] - 2s 18ms/step - loss: 0.1587 - binary_accuracy: 0.9293
         Epoch 90/500
         96/96 [==============================] - 2s 18ms/step - loss: 0.1457 - binary_accuracy: 0.9365
         Epoch 91/500
         96/96 [==============================] - 2s 18ms/step - loss: 0.1419 - binary_accuracy: 0.9379
         Epoch 92/500
         96/96 [==============================] - 2s 18ms/step - loss: 0.1739 - binary_accuracy: 0.9349
         Epoch 93/500
         96/96 [==============================] - 2s 18ms/step - loss: 0.3289 - binary_accuracy: 0.8619
         Epoch 94/500
         96/96 [==============================] - 2s 18ms/step - loss: 0.2059 - binary_accuracy: 0.9147
         Epoch 95/500
         96/96 [==============================] - 2s 18ms/step - loss: 0.1666 - binary_accuracy: 0.9283
         Epoch 96/500
         96/96 [==============================] - 2s 18ms/step - loss: 0.1452 - binary_accuracy: 0.9370
```

Figure 37: Example of training the model with 15 frames interval

As seen in figure 37, the improvements were already apparent, but it was still not fulfilling enough. We then decided to add another iteration with 5 frames intervals instead of 15 to fully realize the model's potential in terms of accuracy.

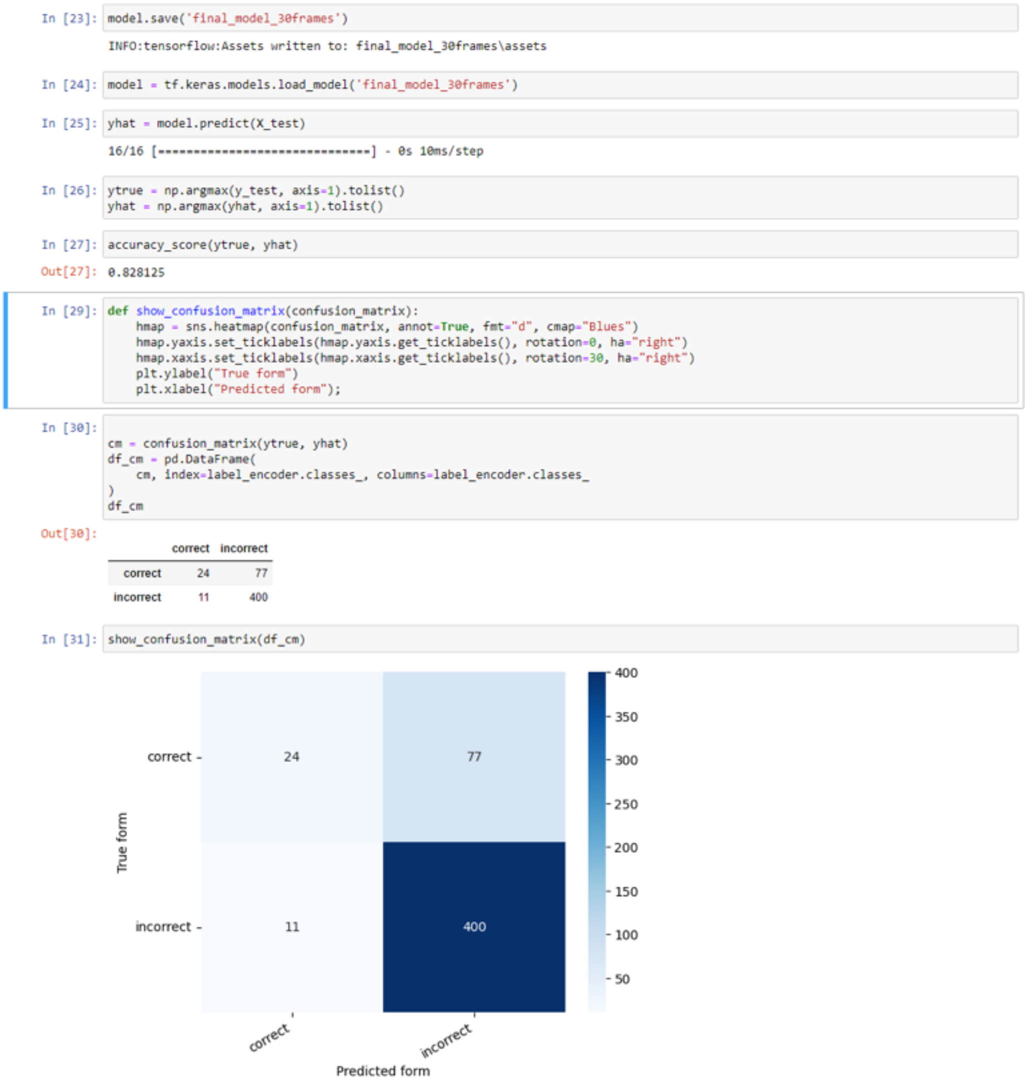


```
In [19]: model = Sequential()
         model.add(LSTM(64, return_sequences=True, activation='relu', input_shape=(5,36)))
         model.add(LSTM(128, return_sequences=True, activation='relu'))
         model.add(LSTM(64, return_sequences=False, activation='relu'))
         model.add(Dense(64, activation='relu'))
         model.add(Dense(32, activation='relu'))
         model.add(Dense(2, activation='sigmoid'))

In [20]: model.compile(optimizer="Adam", loss='binary_crossentropy', metrics=['binary_accuracy'])

In [21]: history = model.fit(X_train, y_train, epochs=500, callbacks=[callback])
         288/288 [==============================] - 2s 6ms/step - loss: 0.0250 - binary_accuracy: 0.9905
         Epoch 184/500
         288/288 [==============================] - 2s 6ms/step - loss: 0.0204 - binary_accuracy: 0.9916
         Epoch 185/500
         288/288 [==============================] - 2s 6ms/step - loss: 0.0780 - binary_accuracy: 0.9746
         Epoch 186/500
         288/288 [==============================] - 2s 6ms/step - loss: 0.0429 - binary_accuracy: 0.9869
         Epoch 187/500
         288/288 [==============================] - 2s 6ms/step - loss: 0.0190 - binary_accuracy: 0.9932
         Epoch 188/500
         288/288 [==============================] - 2s 6ms/step - loss: 0.0183 - binary_accuracy: 0.9940
         Epoch 189/500
         288/288 [==============================] - 2s 6ms/step - loss: 0.0321 - binary_accuracy: 0.9906
         Epoch 190/500
         288/288 [==============================] - 2s 6ms/step - loss: 0.0266 - binary_accuracy: 0.9912
         Epoch 191/500
         288/288 [==============================] - 2s 6ms/step - loss: 0.0217 - binary_accuracy: 0.9923
         Epoch 192/500
         288/288 [==============================] - 2s 6ms/step - loss: 0.0390 - binary_accuracy: 0.9882
```

Figure 38: Example of training the model with 5 frames interval

3.6.6 Results

To evaluate the results in a more organized and easy-to-observe manner, the model can be saved and then loaded into Jupyter Notebook to analyze and create different ways to visualize the model results.

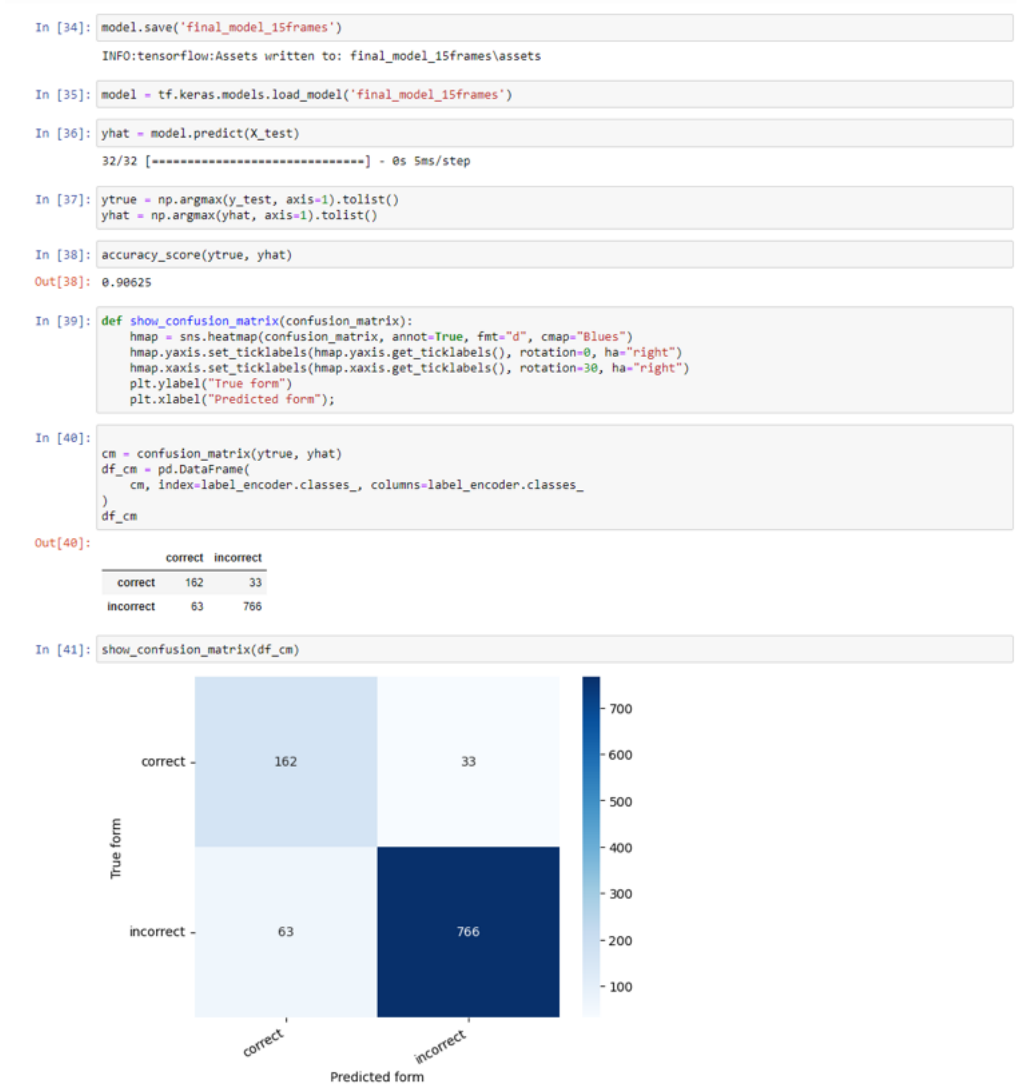


```
In [23]: model.save('final_model_30frames')

         INFO:tensorflow:Assets written to: final_model_30frames\assets

In [24]: model = tf.keras.models.load_model('final_model_30frames')

In [25]: yhat = model.predict(X_test)

         16/16 [==============================] - 0s 10ms/step

In [26]: ytrue = np.argmax(y_test, axis=1).tolist()
         yhat = np.argmax(yhat, axis=1).tolist()

In [27]: accuracy_score(ytrue, yhat)

Out[27]: 0.828125

In [29]: def show_confusion_matrix(confusion_matrix):
             hmap = sns.heatmap(confusion_matrix, annot=True, fmt="d", cmap="Blues")
             hmap.yaxis.set_ticklabels(hmap.yaxis.get_ticklabels(), rotation=0, ha="right")
             hmap.xaxis.set_ticklabels(hmap.xaxis.get_ticklabels(), rotation=30, ha="right")
             plt.ylabel("True form")
             plt.xlabel("Predicted form");

In [30]:
         cm = confusion_matrix(ytrue, yhat)
         df_cm = pd.DataFrame(
             cm, index=label_encoder.classes_, columns=label_encoder.classes_
         )
         df_cm

Out[30]:
```

	correct	incorrect
correct	24	77
incorrect	11	400

```
In [31]: show_confusion_matrix(df_cm)
```

Figure 39: Test results of the model with 30 frames interval

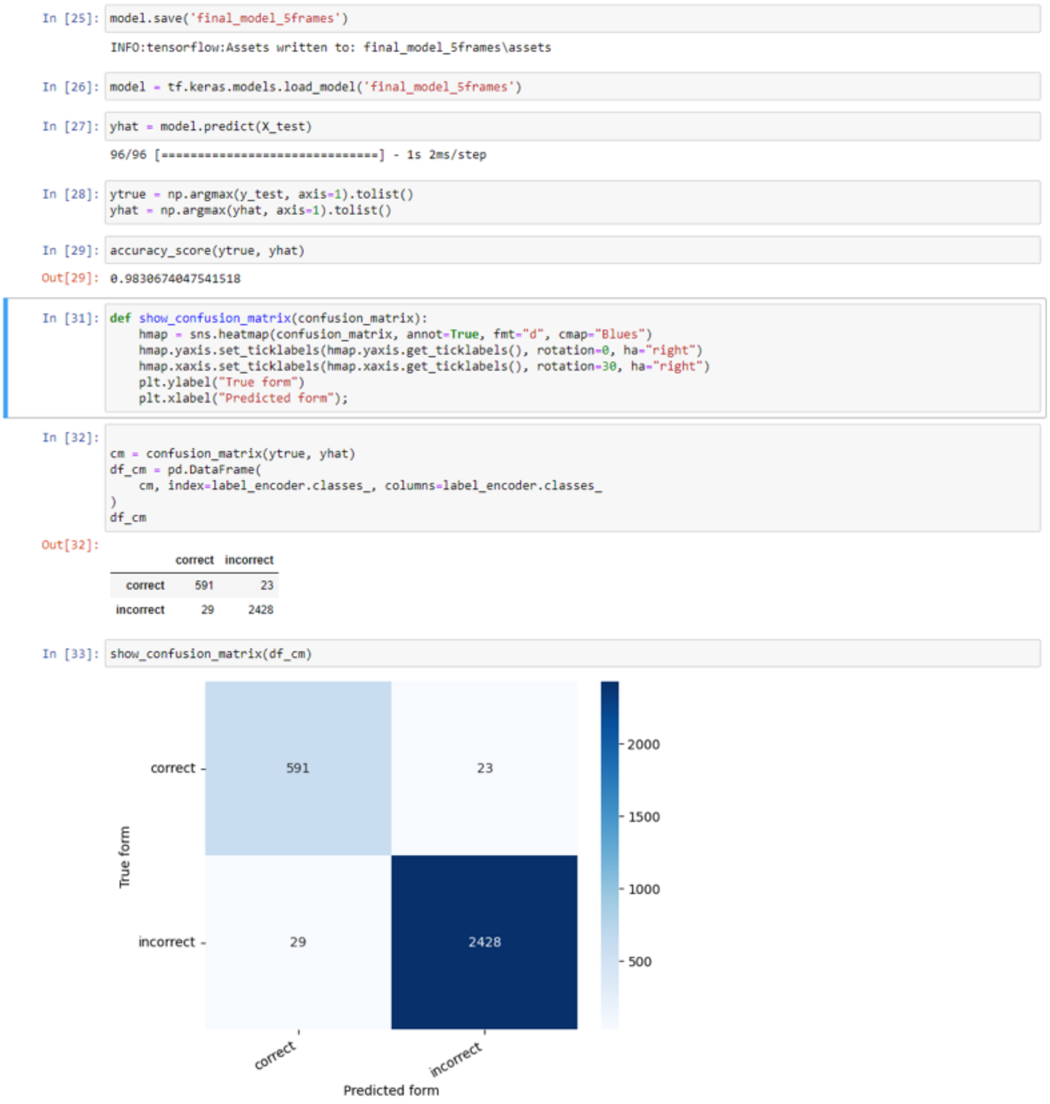


```
In [34]: model.save('final_model_15frames')

         INFO:tensorflow:Assets written to: final_model_15frames\assets

In [35]: model = tf.keras.models.load_model('final_model_15frames')

In [36]: yhat = model.predict(X_test)

         32/32 [==============================] - 0s 5ms/step

In [37]: ytrue = np.argmax(y_test, axis=1).tolist()
         yhat = np.argmax(yhat, axis=1).tolist()

In [38]: accuracy_score(ytrue, yhat)

Out[38]: 0.90625

In [39]: def show_confusion_matrix(confusion_matrix):
             hmap = sns.heatmap(confusion_matrix, annot=True, fmt="d", cmap="Blues")
             hmap.yaxis.set_ticklabels(hmap.yaxis.get_ticklabels(), rotation=0, ha="right")
             hmap.xaxis.set_ticklabels(hmap.xaxis.get_ticklabels(), rotation=30, ha="right")
             plt.ylabel("True form")
             plt.xlabel("Predicted form");

In [40]:
         cm = confusion_matrix(ytrue, yhat)
         df_cm = pd.DataFrame(
             cm, index=label_encoder.classes_, columns=label_encoder.classes_
         )
         df_cm

Out[40]:
```

	correct	incorrect
correct	162	33
incorrect	63	766

```
In [41]: show_confusion_matrix(df_cm)
```

Figure 40: Test results of the model with 15 frames interval



```
In [25]: model.save('final_model_5frames')

         INFO:tensorflow:Assets written to: final_model_5frames\assets

In [26]: model = tf.keras.models.load_model('final_model_5frames')

In [27]: yhat = model.predict(X_test)

         96/96 [==============================] - 1s 2ms/step

In [28]: ytrue = np.argmax(y_test, axis=1).tolist()
         yhat = np.argmax(yhat, axis=1).tolist()

In [29]: accuracy_score(ytrue, yhat)

Out[29]: 0.9830674047541518

In [31]: def show_confusion_matrix(confusion_matrix):
             hmap = sns.heatmap(confusion_matrix, annot=True, fmt="d", cmap="Blues")
             hmap.yaxis.set_ticklabels(hmap.yaxis.get_ticklabels(), rotation=0, ha="right")
             hmap.xaxis.set_ticklabels(hmap.xaxis.get_ticklabels(), rotation=30, ha="right")
             plt.ylabel("True form")
             plt.xlabel("Predicted form");

In [32]:
         cm = confusion_matrix(ytrue, yhat)
         df_cm = pd.DataFrame(
             cm, index=label_encoder.classes_, columns=label_encoder.classes_
         )
         df_cm

Out[32]:
```

	correct	incorrect
correct	591	23
incorrect	29	2428

```
In [33]: show_confusion_matrix(df_cm)
```

Figure 41: Test results of the model with 5 frames interval

The results in Figure 39,40 and 41 show a clear improvement when using an iterative approach to the training. This is especially apparent when looking at the predicted accuracy score computed for each interval, which improved from 82.81% accuracy for the 30 frames interval to 90.62% accuracy for the 15-frame interval to 98.30% accuracy for the 5 frame interval. Furthermore, the confusion matrices all visualize the model's results by depicting the darkness of the blue as to how accurate the model has become after training it.

4 Reflection & Discussion

4.1 Normalization

A big part of the research was gathering data from the videos and using it to feed the model. However, the accuracy was skewed due to the different cameras and camera angles which led to the need for normalization. We did not have the appropriate time frame to complete the normalization of our datasets, but this could have improved both the visualization of our graphs and the following data used in the model itself.

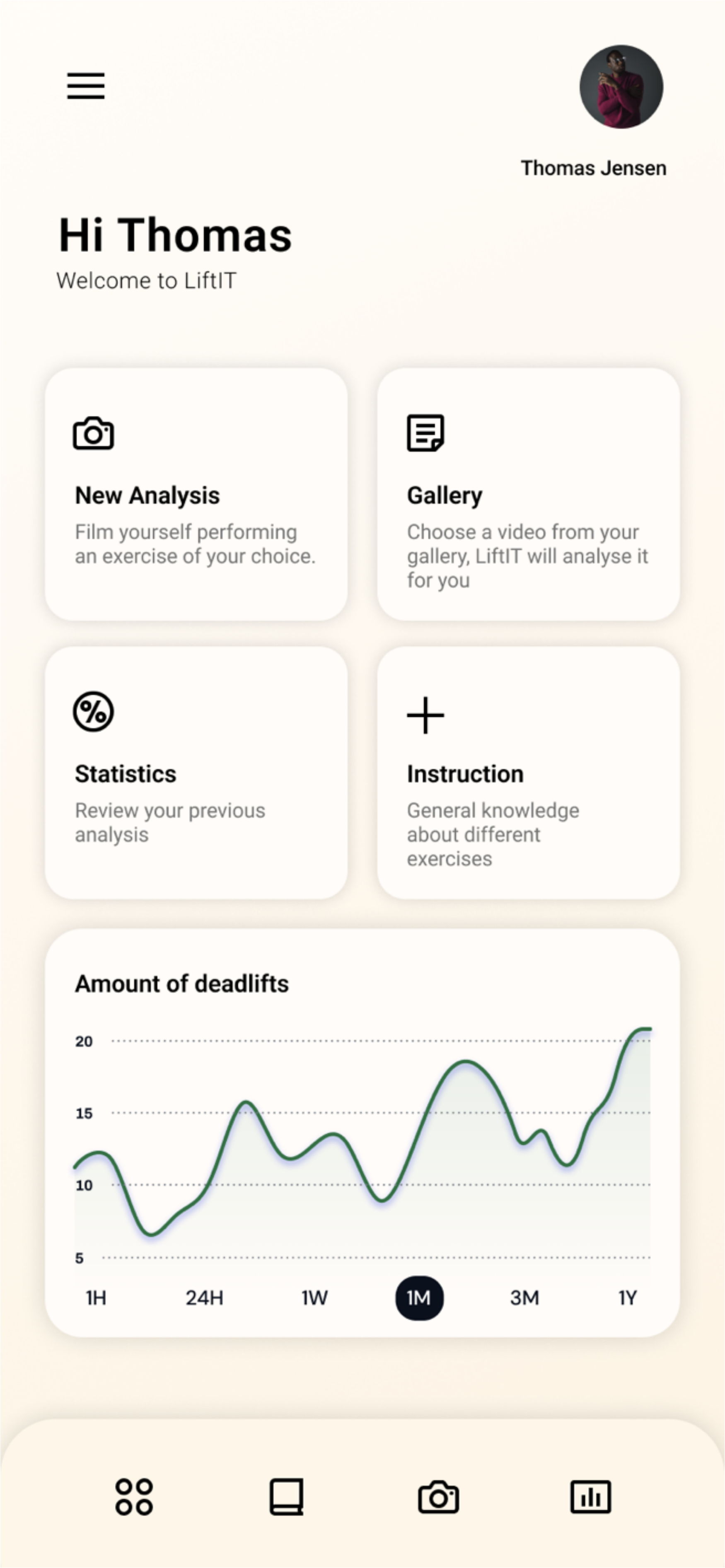
4.2 Data collection

One of the essential parts of any machine learning project is the data. The quality and quantity of the data are significant factors when it comes to depicting the accuracy of the model. The dataset used for our thesis could have improved in several ways. Our goal for the data was to represent the deadlifts of the natural world, where the diversity of lifts are incredibly significant. This is important for the neural network model to be usable for anyone who wants to improve their life and correct their mistakes. We did succeed in filming a diverse group of people. However, the diversity and number of people could have been much better than the four people filmed. However, given the time frame of the thesis and the time consumption of processing the data, this was not possible. When filming these deadlifts, we used the same angle each time and made the participant do exactly three repetitions of the exercise. This had tremendous benefits for the comparability of the plots. However, there needs to be a better representation of deadlifts in the real world. To get a greater insight into the differences between a correct and incorrect execution of the exercise, we could have filmed the exercise from multiple angles, and either had the participants do a variety of repetitions or always a single repetition. This would also have been a significant obstacle when developing an app that utilizes the neural network model created in this thesis. It might only correctly recognize a correct or incorrect deadlift when filmed from the right angle with the correct number of repetitions. Our model lacks a diverse input dataset. The videos were filmed in low lighting, which, especially for the older smartphones, was a hindering factor when the pose estimation had to precisely recognize the movements of the limbs. This represents various phones and locations people can be deadlifting in. However, a variety of lighting and environments could have been optimal.

Another hindering factor of our dataset was that most of our participants had no experience with the deadlift before filming them. This was okay since representing people of varying experiences in our dataset was one of our goals, so anyone could use the neural network. However, instructing the participants to perform the deadlift correctly or incorrectly in a particular matter, based on the previous findings from research and the talk with the personal trainer deemed quite tricky. This was simply due to our lack of experience with fitness instruction and resulted in some variations in the performance of the different participants. A solution to this could have been to get help from the certified personal trainer Alexander Mathiesen whom we previously interviewed since he is educated in fitness instruction. This would have led to a dataset of higher quality and, finally are more precise neural network.

4.3 Development of the application

One of our goals for this thesis was to develop an application and implement a neural network model. The idea would be that users could film themselves doing an execution of the deadlift. The app would process the video through our model, which would recognize any of the four mistakes analyzed in this thesis if present. However, due to time restrictions, this still needs to be achieved. We chose to put all of our focus on data processing and to train the neural network instead, but we could have developed an application, if a larger time frame had been at our disposal. However, we have developed and prototype in Figma.



Thomas Jensen

Hi Thomas

Welcome to LiftIT

New Analysis

Film yourself performing an exercise of your choice.

Gallery

Choose a video from your gallery, LiftIT will analyse it for you

Statistics

Review your previous analysis

Instruction

General knowledge about different exercises

Amount of deadlifts

Figure 42: The LiftIT home screen

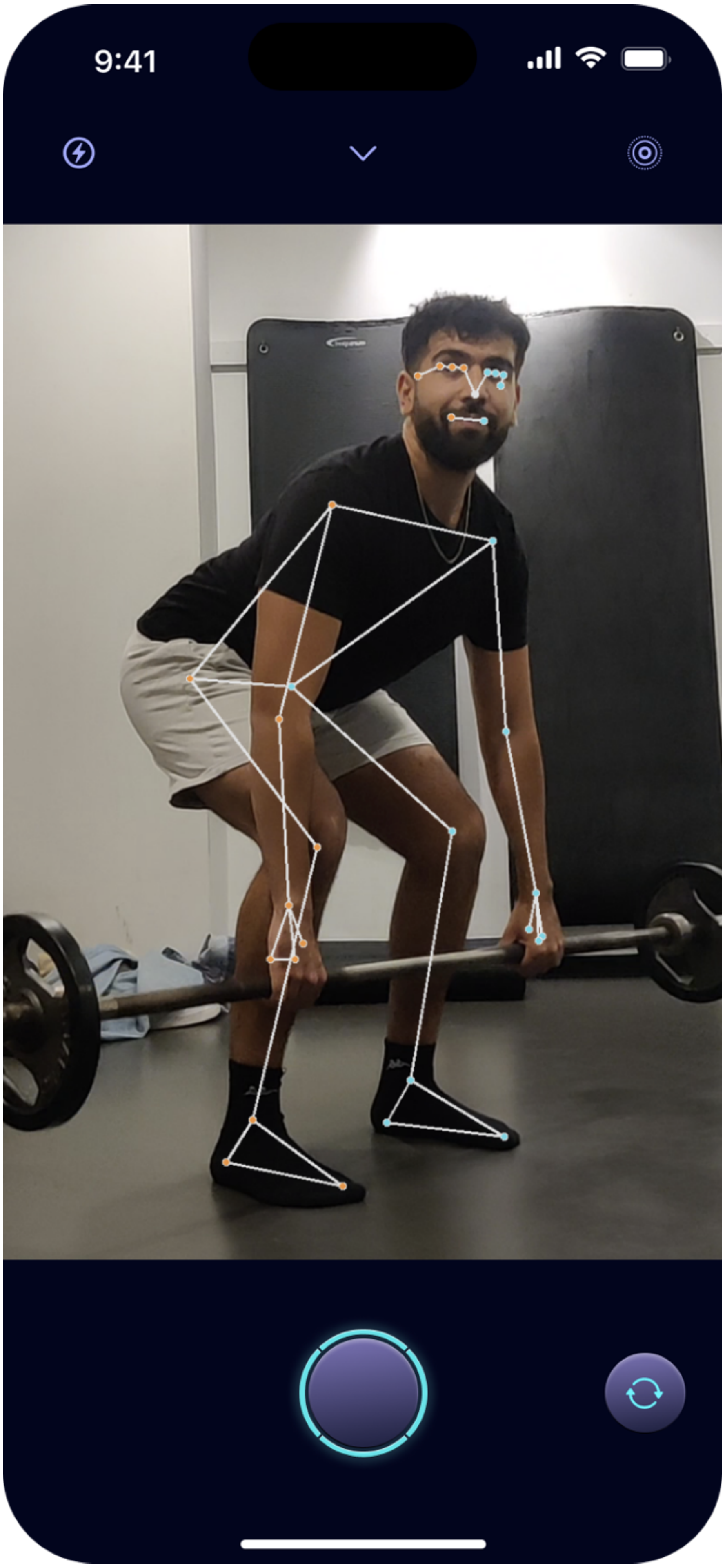


Figure 43: The new analysis button opens up the camera to film the user doing the lift with real time Pose Estimation applied

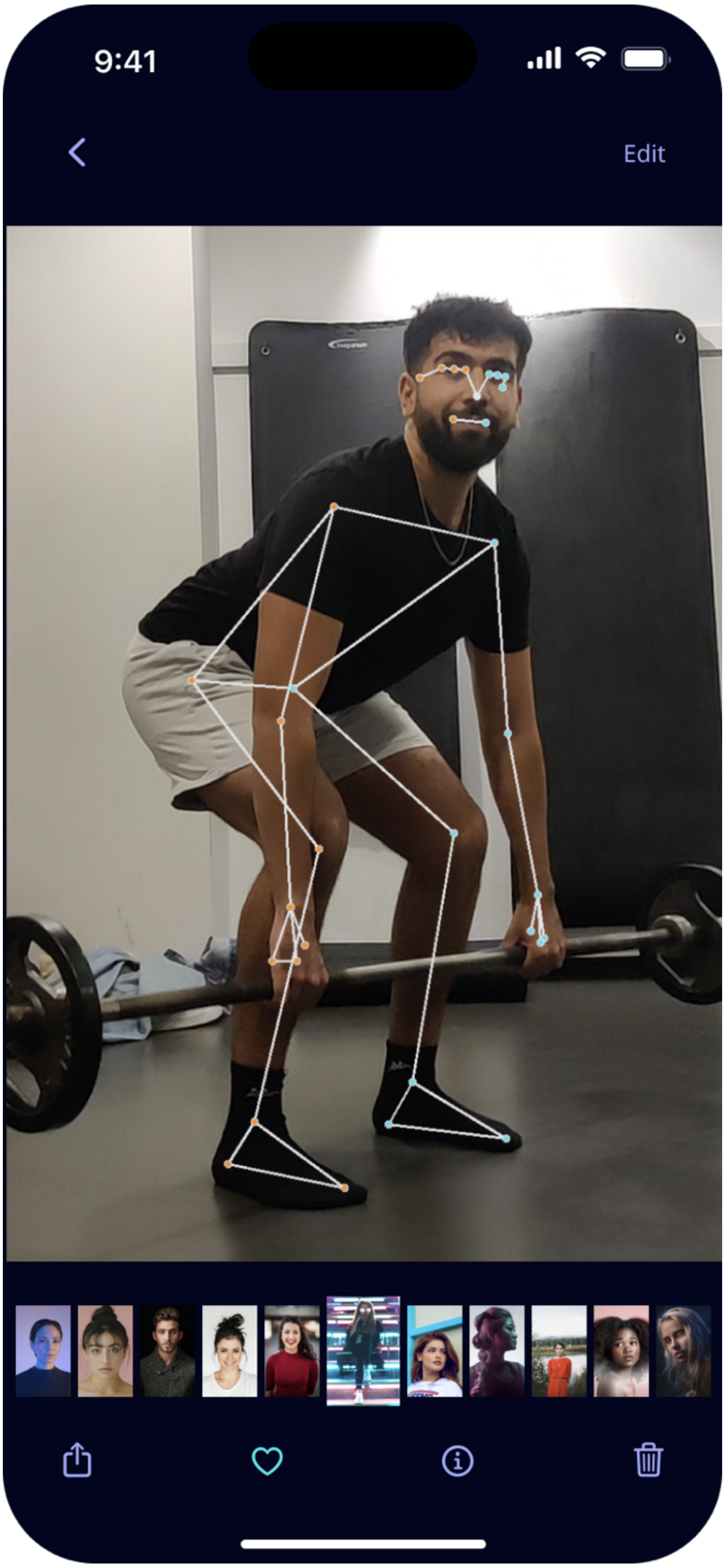
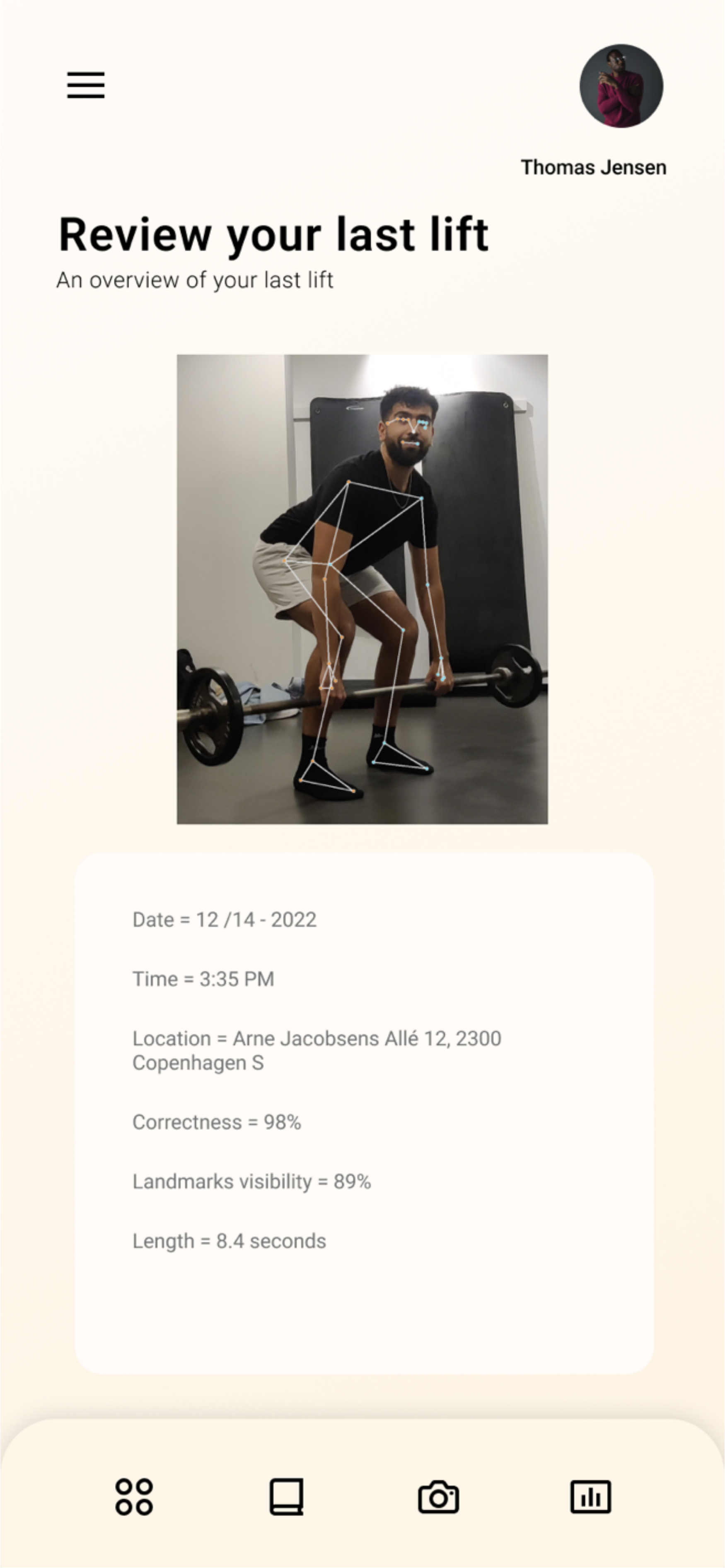


Figure 44: The Gallery button opens up the phone's gallery to choose a video to be analyzed with Pose Estimation



Review your last lift

An overview of your last lift

Date = 12 /14 - 2022

Time = 3:35 PM

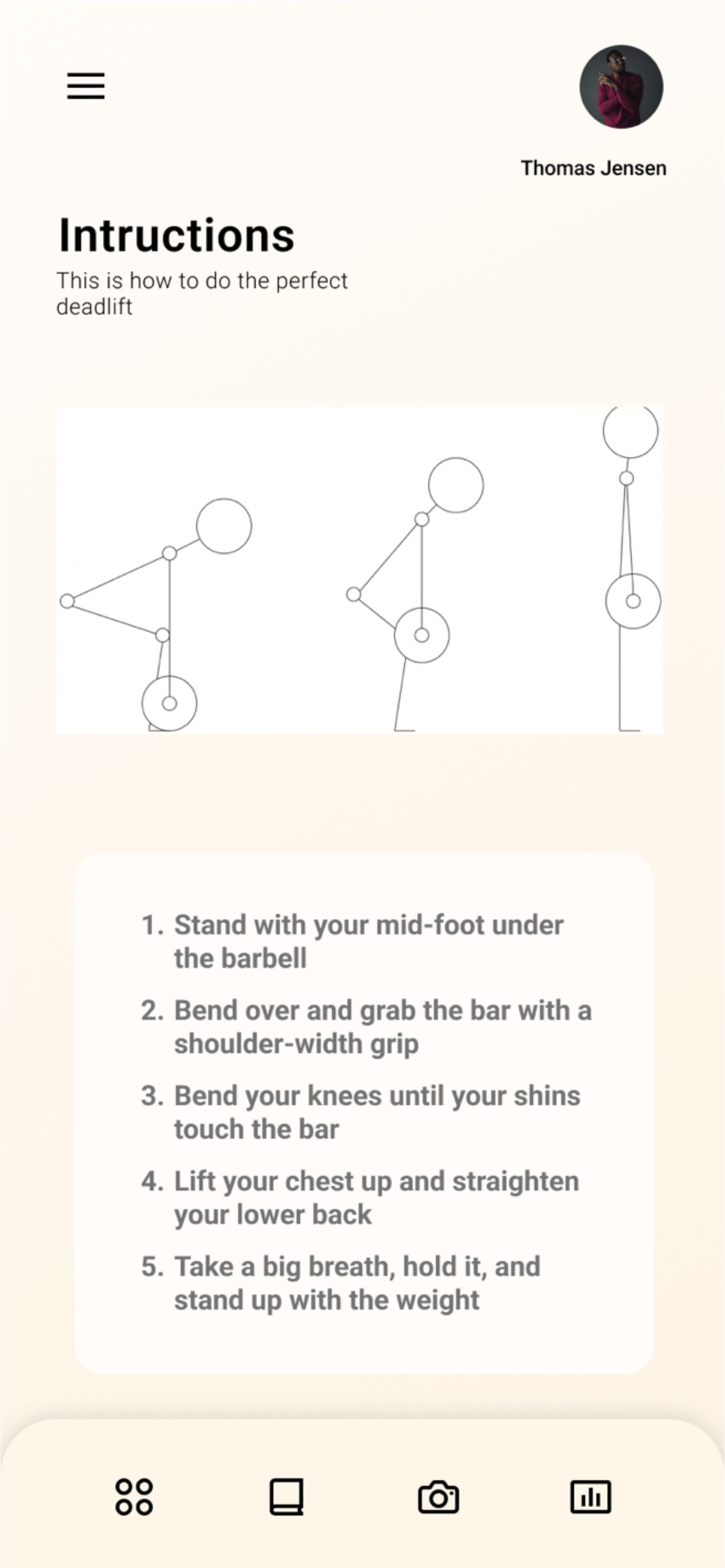
Location = Arne Jacobsens Allé 12, 2300
Copenhagen S

Correctness = 98%

Landmarks visibility = 89%

Length = 8.4 seconds

Figure 45: The Statistics button opens up the review of the last lift with the Machine learning model results as well



Thomas Jensen

Intructions

This is how to do the perfect deadlift

1. Stand with your mid-foot under the barbell
2. Bend over and grab the bar with a shoulder-width grip
3. Bend your knees until your shins touch the bar
4. Lift your chest up and straighten your lower back
5. Take a big breath, hold it, and stand up with the weight

Figure 46: The Instructions button opens up the instructions on how to do the perfect deadlift in plain text

4.4 The model

Early in the research, we decided to use the binary distribution in the model as the preferred option instead of a categorical distribution. This was due to the black-and-white nature of our research containing either correct or incorrect video recordings of the participants. The goal was to create as big an incorrect data pool as possible to have more training data for the model. This approach, in hindsight, was not the most effective way of optimizing the data for the prediction model. The main reason is the relatively low accuracy when doing the 30-frame interval because of the model's inability to distinguish between correct and incorrect when the frame interval was this big. This could be due to the 4 different mistakes not having their categorical class, which theoretically would result in much better processing and accompanying accuracy increases. The approach in this research was not entirely wrong but could benefit from defining the 4 different mistakes, each with their class instead.

The average accuracy of the 3 models was 0.90354, which is 90.354% which would result in the model having a wrong prediction roughly 1/10 times. In the real world, this could mean people who have never done the deadlift before being fed the wrong information. The lowest accuracy score was the 30 frames interval, which was 82% correct prediction chance. In this case, the long-lasting effects of the misinformation regarding whether or not the user of the model did the lift correctly or incorrectly would have too much of an impact. It would be on the verge of being ethically incorrect to publish something that only has 90% accuracy in a situation where every wrong prediction could lead to long-lasting injuries. This possibility of injuries goes against the core of this project, preventing injuries with a free alternative rather than paying for a personal trainer.

However, to combat this model inaccuracy, rethinking the whole concept of processing the videos with the correct or incorrect class could be a good idea. In the LiftIT implementation, the whole of the incorrect video data was put as incorrect, when some of the movement inside of the incorrect movement actually could be correct. This black-and-white approach to incorrect and correct can lead to worse accuracy in the model since the correct movements during the lift are labeled as incorrect, as shown in Figure 47.

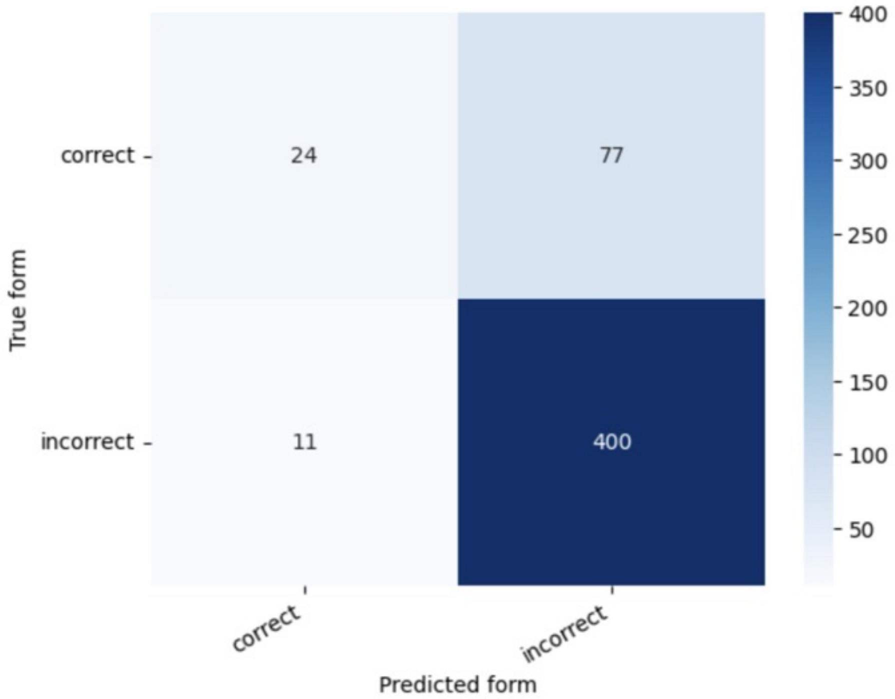


Figure 47: Confusion matrix for 30 frames interval

The confusion matrix is mistakenly predicting the movement to be incorrect while being incorrect significantly more than the other way around. This is due to the incorrect videos containing correct movement, leading to worse results.

For example, in the squat incorrect execution, it is only in the starting position, and in the next 1-2 seconds, the landmarks are incorrect. After exiting the squat position, the rest of the movement of that repetition is correct. A fix to this problem would be going over the videos in video editing software, pinpointing what exact frames the correct and incorrect movement occurs, and defining the correct or incorrect class for the intervals. To further improve upon this, we could have implemented a categorical distribution. This could be done by labeling the different frames with the exact mistake and changing the model to a categorically distributed prediction model instead. This implementation would lessen the number of times the prediction model would wrongly predict a video to be incorrect while it was correct.

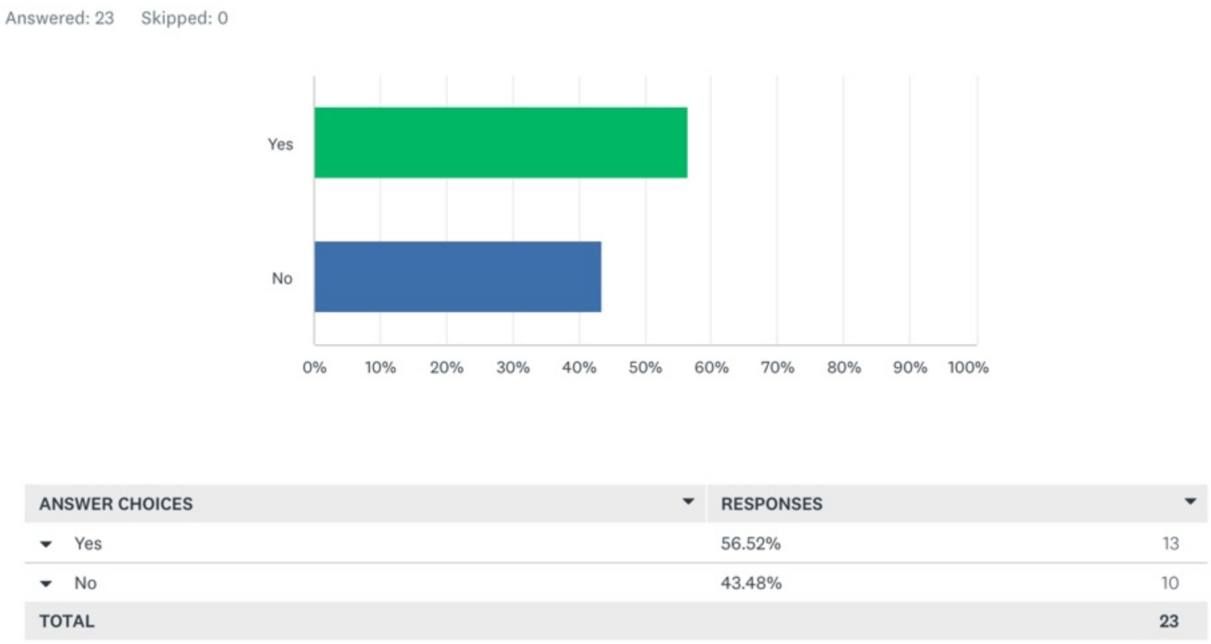
5 Conclusion

A personal trainer can help with avoiding injuries, but the prices can be too high for most people. That is the basis for our thesis, which explores whether or not a mobile application can function as a cheaper alternative to a personal trainer to execute the deadlift correctly. By incorporating the users' needs and expert opinions on the matter, we established the basis for further research. Another vital part of the research was incorporating PCA to reduce dimensions in the data to plot it in a graph. The graphing of the CSV files were not imperative in the creation of the Machine Learning model but rather functioned as a way to visualize and define whether or not the differences in correct and incorrect were significant. The CSV files created from the videos were then input into the binary distribution of correct/incorrect. We implemented this by creating an LSTM model to predict whether or not the movement was labeled as incorrect or correct. An analysis of the model results and the plotted PCA graphs showed a clear distinction between incorrect and correct movements. The results of the model were of high enough accuracy to be considered relevant when we defined the window of frames to be 5. The reflection and discussion section of the paper gave us an insight into possible future improvements and processes that we could have executed differently to achieve better end results. Furthermore, the lack of time also gave us some challenges regarding building the application itself. However, the ground works for developing said application was done in this paper. The findings in the research led to the conclusion that incorporating the recorded video data intertwined with some Machine Learning is a valid alternative to a personal trainer.

6 Appendix

6.1 Questionnaire

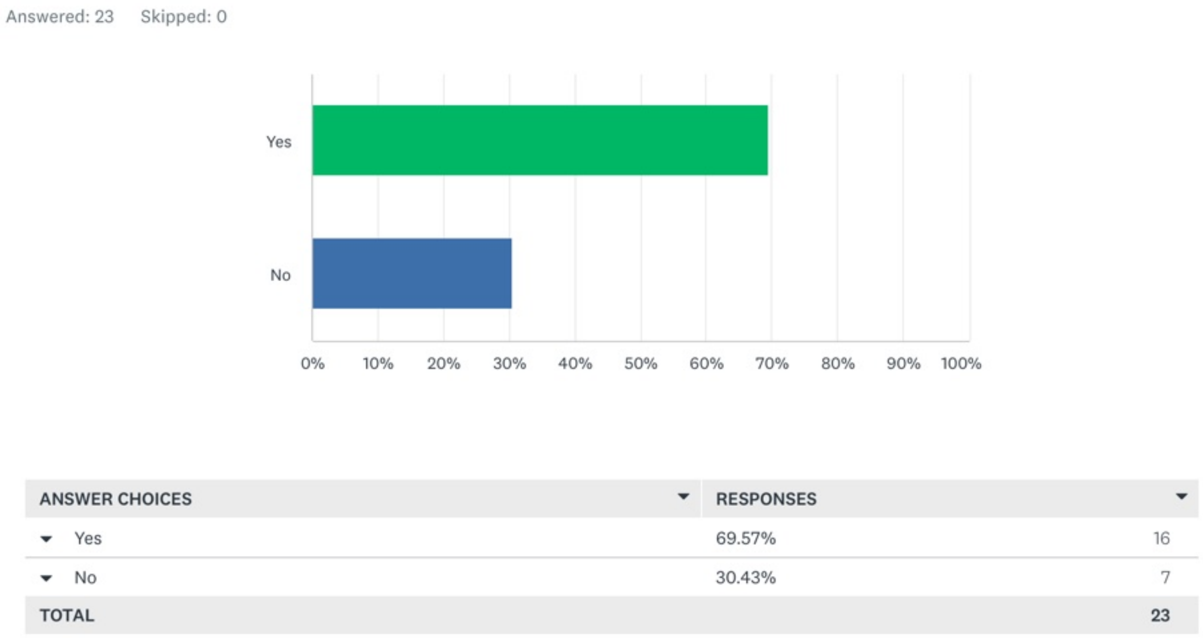
Do you often experience problems with executing exercises in a trouble free manner?



Answered: 23 Skipped: 0

ANSWER CHOICES	RESPONSES	
Yes	56.52%	13
No	43.48%	10
TOTAL		23

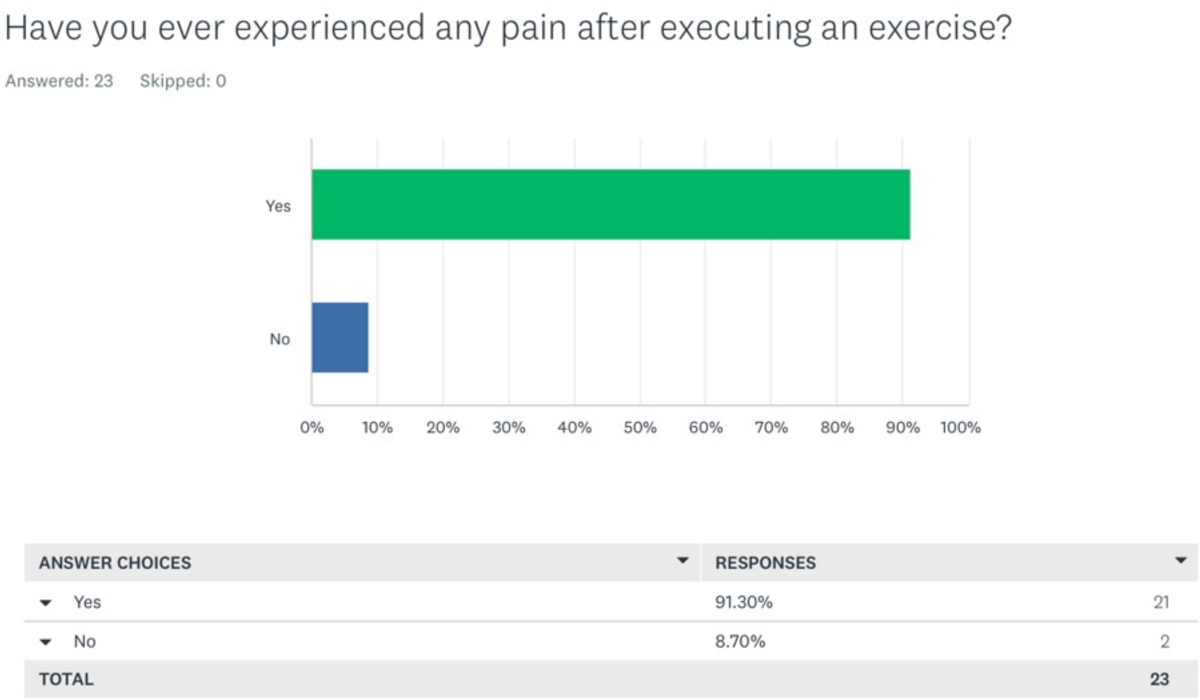
Are you aware of the dangers of improper execution of certain exercises?



Answered: 23 Skipped: 0

ANSWER CHOICES	RESPONSES	
Yes	69.57%	16
No	30.43%	7
TOTAL		23

Have you ever experienced any pain after executing an exercise?

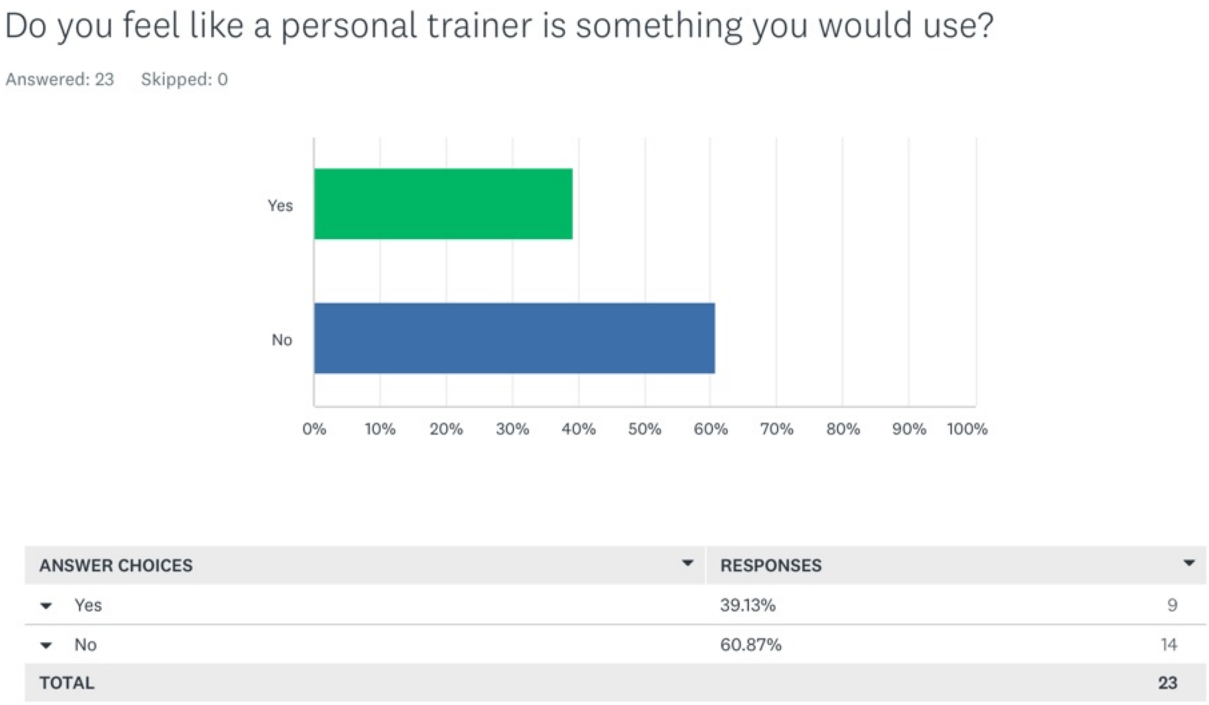


Have you ever experienced any pain after executing an exercise?

Answered: 23 Skipped: 0

ANSWER CHOICES	RESPONSES	
Yes	91.30%	21
No	8.70%	2
TOTAL		23

Do you feel like a personal trainer is something you would use?

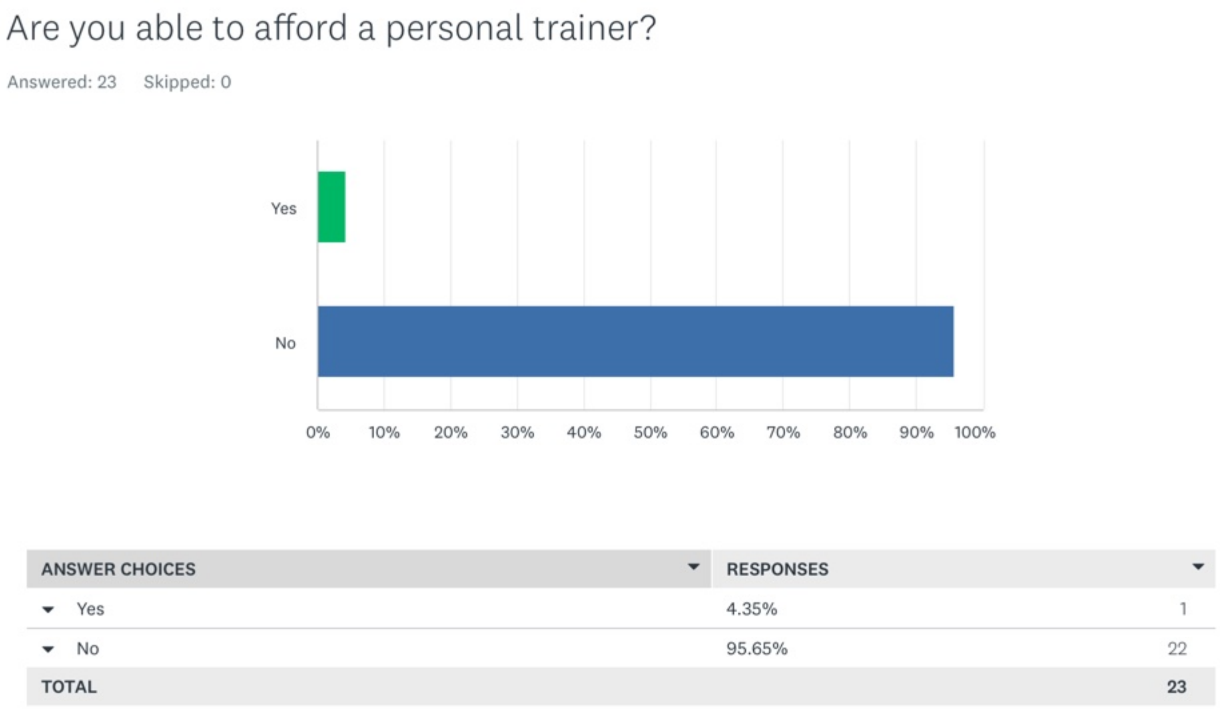


Do you feel like a personal trainer is something you would use?

Answered: 23 Skipped: 0

ANSWER CHOICES	RESPONSES	
Yes	39.13%	9
No	60.87%	14
TOTAL		23

Are you able to afford a personal trainer?

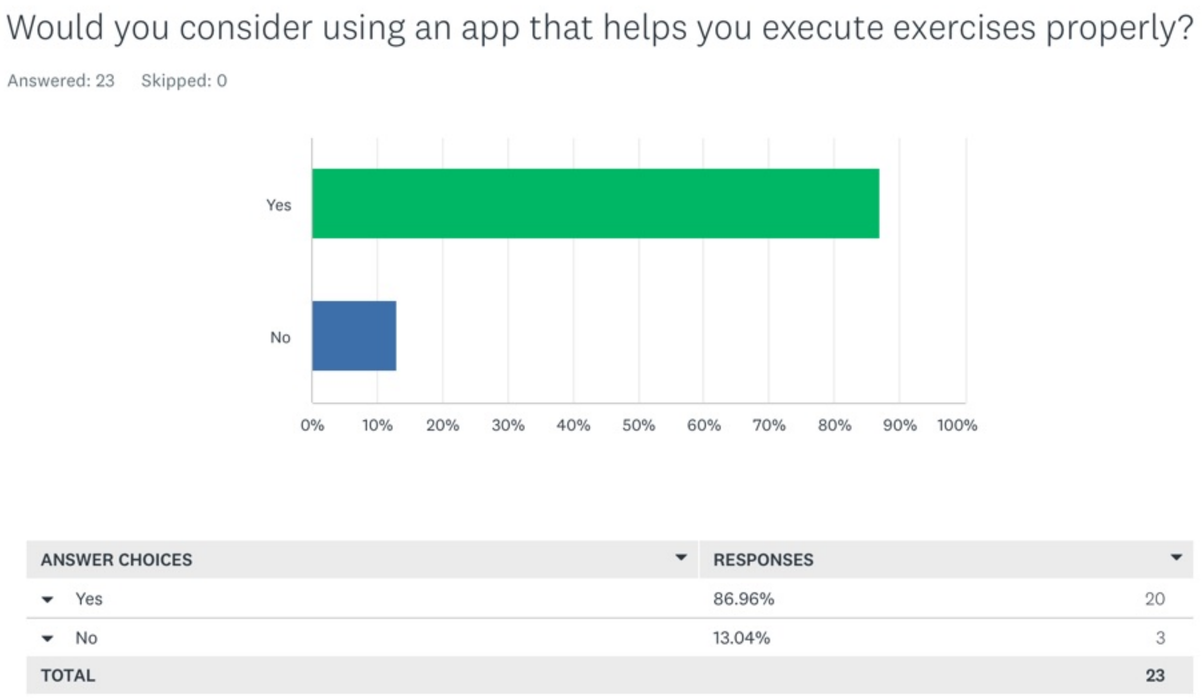


Are you able to afford a personal trainer?

Answered: 23 Skipped: 0

ANSWER CHOICES	RESPONSES	
Yes	4.35%	1
No	95.65%	22
TOTAL		23

Would you consider using an app that helps you execute exercises properly?



Would you consider using an app that helps you execute exercises properly?

Answered: 23 Skipped: 0

ANSWER CHOICES	RESPONSES	
Yes	86.96%	20
No	13.04%	3
TOTAL		23

6.2 User Interviews

Participant 1

1 - Do you experience problems with executing exercises for the first time (Trying a machine you don't know how to properly use or watching another person doing an exercise you have never tried doing)

Yes, I am afraid of having bad form, as this can cause injuries. That's why i usually refrain from executing exercises for the first time on my own as I am scared of doing it wrong. I would then wait for a more experienced gym goer to guide me. A specific instance was when I wanted to do the deadlift but I had no idea how to do the movement and therefore tried googling the correct form, but I ultimately gave up and did another exercise instead.

2- Are you aware of the dangers of improper execution of certain exercises, if so are you worried about potential injuries?

Yes I am, I am a beginner at going to the gym, so therefore I have to be extra aware of correcting my form in order to avoid potential injuries. I have also seen a lot of Tiktoks where very serious injuries are shown and that have scared me to do certain exercises such as leg press where I have seen some legs snap. I also saw a tiktok where a woman was doing the Squat and fall backwards and died. This has scarred me and made me extra cautious and somewhat scared of doing certain exercises on my own.

3 - Have you ever experienced any pain after executing an exercise, has it ever held you back from working out or doing other activities?

No, I haven't experienced any pains other than from dropping a weight on my finger. This is possibly due to me being too scared to go up in weight in the exercises I am not 100% confident in. This could be the reason for me staying in the comfortable weight range and not seeing many results of my workouts.

4 - Would you consider using a personal trainer? Why or why not?

No, because I cannot afford one, and I do not have time to work out consistently.

5 - Are you able to afford a personal trainer, if yes, is the high price (5000 a month) of a personal trainer worth it?

No, I think the price is way too high for me and I would not be able to keep up with the workout programme since I have school to worry about.

6 - Would you consider using an app that helps you execute exercises properly, if no, why not?

Yes, if it was free. If it is expensive, I would rather resort to getting help from people I know.

7 - If you would consider using an app to help you correctly execute your workouts. What would you say would be the most important aspects of the app?
Visualizing the correct form for me. It does not help looking at youtube videos for example since I have a hard time seeing if I myself am doing the movement correctly.

Participant 2
1 - Do you experience problems with executing exercises for the first time (Trying a machine you don’t know how to properly use or watching another person doing an exercise you have never tried doing)
This does happen sometimes. When I first started going to the gym, I often found myself having no clue about executing exercises, which held me back from going at all. But after I gained some experience, it got better. But there are still a lot of exercises that I do not perform, as I am unsure about the form of these.

2- Are you aware of the dangers of improper execution of certain exercises, if so are you worried about potential injuries?
I am aware as I have suffered an injury before. I tend to stay away from exercises that have caused me pain.

3 - Have you ever experienced any pain after executing an exercise, has it ever held you back from working out or doing other activities?
Yes. I have often experienced pain in my lower back due to deadlifting.

4 - Would you consider using a personal trainer? Why or why not?
I would consider a personal trainer to help me make sure I do not get injured.

5 - Are you able to afford a personal trainer, if yes, is the high price (5000 a month) of a personal trainer worth it?
I am unable to afford this. Even if I were, I think the price is a bit too high.

6 - Would you consider using an app that helps you execute exercises properly, if no, why not?
If it is easy to use, and helps me perform exercises better, then yes.

7 - If you would consider using an app to help you correctly execute your workouts. What would you say would be the most important aspects of the app?

Ease of use and focus on proper form.

Participant 3

1 - Do you experience problems with executing exercises for the first time (Trying a machine you don't know how to properly use or watching another person doing an exercise you have never tried doing)

When I first started working out, I would experience this often, but as I've gotten more experienced it's more variations to exercises that I find myself being confused about.

2- Are you aware of the dangers of improper execution of certain exercises, if so are you worried about potential injuries?

I was always worried about any free weight exercise, since they involved the most weight. I spent a lot of time practicing my form on deadlifts especially to avoid lasting back-pain.

3 - Have you ever experienced any pain after executing an exercise, has it ever held you back from working out or doing other activities?

When I first started, before my joints had adjusted to lifting weights, I experienced a fair amount of pain that initially had me considering not working out with weights at all. Later it would only be in relation to pushing myself too hard, which I've completely stopped doing. I focus on how each lift feels in relation to whether I should try lifting more.

4 - Would you consider using a personal trainer? Why or why not?

I would not at this time, since I feel like I now have a decent grasp of what I'm doing in the gym. If I had the money for it when I started working out I would have definitely done a couple of sessions to get into it.

5 - Are you able to afford a personal trainer, if yes, is the high price (5000 a month) of a personal trainer worth it?

I would be able to afford it, but I don't think it's worth it with the price point.

6 - Would you consider using an app that helps you execute exercises properly, if no, why not?

For new exercises I would love some pointers before I try to wing it myself. Having that on my phone and not having to look at a youtube video would be an improvement over how i currently work these things out.

7 - If you would consider using an app to help you correctly execute your workouts. What would you say would be the most important aspects of the app?

I would need to have a flow that quickly can get me to the specific exercise if I'm at the gym and in doubt about how to execute something specific. A quick search function or something like that. Additionally, I would personally have liked a workout planner where you could compose your own workout plan or try a pre-composed one while I'm at the gym. I would also need video demonstration as well as muscle activation, optimal break timers.

Participant 4

1 - Do you experience problems with executing exercises for the first time (Trying a machine you don't know how to properly use or watching another person doing an exercise you have never tried doing)

Wouldn't say that I've had problems, but I have definitely experienced some difficulties in understanding how certain machines worked, and/or found the small instructions on them confusing.

2- Are you aware of the dangers of improper execution of certain exercises, if so are you worried about potential injuries?

Personally I have learned how many of the machines work from a trainer and are very aware of the dangers of using a machine incorrectly. Due to that I am not worried about injuries today, but if I had not learned to use them properly beforehand I would be very concerned with possible injuries.

3 - Have you ever experienced any pain after executing an exercise, has it ever held you back from working out or doing other activities?

For several years, when I was younger, I used too much weight on many machines (even though I used them correctly), which caused a lot of pain and overexertion. This has not led to injuries per say, but has held me from doing strength-training and using machines when I am at the gym today.

4 - Would you consider using a personal trainer? Why or why not?

I would consider it, for a more consistent and correct training, but I also prefer to be on my own when i work out, and not having to talk or listen to anyone whilst doing so- so in total i would properly not want to use a personal trainer.

5 - Are you able to afford a personal trainer, if yes, is the high price (5000 a month) of a personal trainer worth it?

I am, but I think the price is too high and definitely not worth it.

6 - Would you consider using an app that helps you execute exercises properly, if no, why not?

Definitely, it would be a great way to make sure you are doing things correctly, but you can still work out on your own, and I am guessing it would be a lot cheaper than a personal trainer.

7 - If you would consider using an app to help you correctly execute your workouts. What would you say would be the most important aspects of the app?

Small guides (videos) showing how to properly do the exercises, some explanations of the dangers of using machines wrong or using too much weight etc. Maybe a way to check that I am doing the exercise correctly. A feature to track your training would be nice as well.

Participant 5

1 - Do you experience problems with executing exercises for the first time (Trying a machine you don't know how to properly use or watching another person doing an exercise you have never tried doing)

Yes. If I see a person doing an exercise that I want to try, I would never ask for help. I'm too shy. I would just try it out and see if it feels right, maybe i will watch a youtube video or ask a friend.

2- Are you aware of the dangers of improper execution of certain exercises, if so are you worried about potential injuries?

Yes. I have never deadlifted as I am too afraid of the consequences of doing it wrong. I would like to have a personal trainer or an experienced person to guide me.

3 - Have you ever experienced any pain after executing an exercise, has it ever held you back from working out or doing other activities?

I have tried getting a major injury from doing an exercise wrong. I performed a heavy pulldown that resulted in a tendon tear in both my forearms. It hurt for months and still hurts. I can feel when doing regular everyday tasks like opening doors. After researching I found out that I have been doing it wrong for years.

4 - Would you consider using a personal trainer? Why or why not?

I would like to, but it is too expensive. I feel like doing it properly is very important, especially when lifting heavy. If i get rich one day, i would reconsider.

5 - Are you able to afford a personal trainer, if yes, is the high price (5000 a month) of a personal trainer worth it?
No, as a student I would not be able to afford it.

6 - Would you consider using an app that helps you execute exercises properly, if no, why not?
yeah, if it could help me avoid injuries i would use it. I often ask my friend to check if I exercise properly. This is not possible when exercising alone, and I would not ask a stranger. So an app could be very helpful.

7 - If you would consider using an app to help you correctly execute your workouts. What would you say would be the most important aspects of the app?
That i can learn proper technique of exercises to avoid injury.

Participant 6
1 - Do you experience problems with executing exercises for the first time (Trying a machine you don't know how to properly use or watching another person doing an exercise you have never tried doing)

Yes, however I often search the web on how to execute the technique. After that the problem is solved. One time I tried a machine for the first time and was in doubt whether or not I was doing it right. I then googled it and the problem was fixed.

2- Are you aware of the dangers of improper execution of certain exercises, if so are you worried about potential injuries?

Yes I am aware. However starting with lower amount of weight allows me to not be worried about potential injuries. This could be a problem if I eventually want to increase the weight to see better results in my training. However my focus is not on lifting too heavy, but lifting light and for a lot of repetitions.

3 - Have you ever experienced any pain after executing an exercise, has it ever held you back from working out or doing other activities?

Not pain, however soreness is common for me with new exercises which usually goes away after doing that particular exercises a second time.

4 - Would you consider using a personal trainer? Why or why not?

No, I like to work out by myself so I can decide for myself what exercises I wanna do. I like the freedom of going to the gym by myself, which is why I usually work out alone.

5 - Are you able to afford a personal trainer, if yes, is the high price (5000 a month) of a personal trainer worth it?

No, I can’t afford a personal trainer and in my opinion a personal trainer is not worth the money. It is too expensive

6 - Would you consider using an app that helps you execute exercises properly, if no, why not?

Yes, I have already browsed the Fitness World app which has some tutorials on exercises. However the app didn't seem to cover the exercises that I am currently doing. I would consider using another app if that app covers the exercises that I need help with.

7 - If you would consider using an app to help you correctly execute your workouts. What would you say would be the most important aspects of the app?

Most importants aspects for me is that such an App has a big variety of exercises. Furthermore, the breakdown of exercises shouldn't take too long and be easy to understand.

Participant 7

1 - Do you experience problems with executing exercises for the first time (Trying a machine you don’t know how to properly use or watching another person doing an exercise you have never tried doing)

Yes, the first few times I do an exercise, I'm always unsure if I'm doing it right. However, since I have been going to the gym for 3-4 years consistently, I have “mastered” most of, if not all of the exercises I do. But I can see the troubles of doing an exercise for the first time for newcomers.

2- Are you aware of the dangers of improper execution of certain exercises, if so are you worried about potential injuries?

Yes, but I don't always think about whether I will get an injury. It is not my top priority to avoid injuries.

3 - Have you ever experienced any pain after executing an exercise, has it ever held you back from working out or doing other activities?

Even though I am not worried and am not thinking too much about injuries, I still had a lot in the beginning of my fitness journey. It has held me back for other activities such as cooking and gaming.

4 - Would you consider using a personal trainer? Why or why not?

No, I would rather use youtube. Unless I want to become a pro bodybuilder. A personal trainer would only be useful or worthwhile for me if I wanted to take it to the next level of fitness, which is competing.

5 - Are you able to afford a personal trainer, if yes, is the high price (5000 a month) of a personal trainer worth it?

No, that is too expensive for me since I am studying.

6 - Would you consider using an app that helps you execute exercises properly, if no, why not?

Yes, in the beginning of a person’s fitness journey an app would be very useful to set the groundworks for that particular exercise.

7 - If you would consider using an app to help you correctly execute your workouts. What would you say would be the most important aspects of the app?

Easy to use and quick to benefit from. That is in my opinion the most important aspects of such an app.

Participant 8

1 - Do you experience problems with executing exercises for the first time (Trying a machine you don't know how to properly use or watching another person doing an exercise you have never tried doing)

Actually, today I saw a machine(T-bar row) that I have never used that I wanted to try, but because I was unsure about how to use it properly I ended up skipping it.

2- Are you aware of the dangers of improper execution of certain exercises, if so are you worried about potential injuries?

I am aware of the potential dangers but I am not really worried about it. I did not know that i could actually cause permanent injuries or pain lasting a long time.

3 - Have you ever experienced any pain after executing an exercise, has it ever held you back from working out or doing other activities?

After doing a military press, I suddenly experienced a lot of pain in my left shoulder. After that, I never did that exercise again since I was afraid that it would happen again. I was unable to sleep on that side, and I stopped doing any exercise for a couple of weeks.

4 - Would you consider using a personal trainer? Why or why not?

I have considered a personal trainer to help me avoid the injuries.

5 - Are you able to afford a personal trainer, if yes, is the high price (5000 a month) of a personal trainer worth it?

I am unable to afford a personal trainer.

6 - Would you consider using an app that helps you execute exercises properly, if no, why not?

Yes especially if it would visually show me how to execute an exercise that I was unsure about.

7 - If you would consider using an app to help you correctly execute your workouts. What would you say would be the most important aspects of the app?

Participant 9

1 - Do you experience problems with executing exercises for the first time (Trying a machine you don't know how to properly use or watching another person doing an exercise you have never tried doing)

Not for the time being, because I am so experienced, but if I would have problems, I could search on google on how to execute it.

2- Are you aware of the dangers of improper execution of certain exercises, if so are you worried about potential injuries?
Yes. I am worried about injuries, and that's why I will always think about my form and how to execute the exercise in the safest way.

3 - Have you ever experienced any pain after executing an exercise, has it ever held you back from working out or doing other activities?
When I was a newbie, I didn't know how to bench press and I would feel a sharp pain in my shoulder and I had to stop working out at that moment and go home from the gym. Furthermore, in the beginning I also tried to do the deadlift and wound up having the same issue but instead with my lower back with a sharp pinching pain.

4 - Would you consider using a personal trainer? Why or why not?
I have used a personal trainer before, and it's good since they know exactly what to do to get you to your goal, even if it's only about diet or your training program.

5 - Are you able to afford a personal trainer, if yes, is the high price (5000 a month) of a personal trainer worth it?
I used to be able to afford a personal trainer, but not now since I study the things myself and don't need a personal trainer since the years of going to the gym and getting more gym knowledge from friends and articles have helped me improve my form.

6 - Would you consider using an app that helps you execute exercises properly, if no, why not?
I don't use an app to help me with my workout - I just write it down in my notes on my phone. I trust myself enough to do the exercises well enough to avoid injury.

7 - If you would consider using an app to help you correctly execute your workouts. What would you say would be the most important aspects of the app?
For my use case I would not consider using an app to help me with my for since I am confident that I would maintain good form throughout the exercise.

Participant 10

1 - Do you experience problems with executing exercises for the first time (Trying a machine you don’t know how to properly use or watching another person doing an exercise you have never tried doing)

I have experienced having trouble with form, when executing a specific new exercise. I have had trouble activating or utilizing the proper muscles when performing different compound(Exercises where multiple muscles are used) movements like bench press and deadlift. I haven't always braced my muscles before executing the movement, which has led to pain. With the deadlift, I have had trouble caused by rounding my back which I definitely could feel the following days.

2- Are you aware of the dangers of improper execution of certain exercises, if so are you worried about potential injuries?

Yes I am. Especially with the access to the internet, which has given me a lot of information about the potential dangers of weight lifting. I am very aware of the fact that I have to use my body my whole life, and I want to take extra good care of it. I have been taught all my life that improper execution is dangerous. Muscle pain is okay but joint pain is quite bad.

3 - Have you ever experienced any pain after executing an exercise, has it ever held you back from working out or doing other activities?

Once after a bench press session I experienced pain in my shoulder which lasted for months. I had trouble sleeping on that side, and I often woke up in the middle of the night because of the pain. I could feel the pain throughout the day, especially when driving a car. I said no to a lot social activities because of the pain, so it definitely had an effect on my life.

4 - Would you consider using a personal trainer? Why or why not?

I can see the positives of a personal trainer, but because of the price I have always just asked my friends who have trained for longer or are more knowledgeable on the area.

5 - Are you able to afford a personal trainer, if yes, is the high price (5000 a month) of a personal trainer worth it?

Not at all

6 - Would you consider using an app that helps you execute exercises properly, if no, why not?

Yes. I often see myself searching up exercises in the gym, so my phone is an integrated part of my workouts. I could easily see myself using an app too.

7 - If you would consider using an app to help you correctly execute your workouts. What would you say would be the most important aspects of the app?

I would like some correct and well communicated information in text or video. No filler information and only the most importants aspects that would help me have a pain free gym session.

Participant 11

1 - Do you experience problems with executing exercises for the first time (Trying a machine you don't know how to properly use or watching another person doing an exercise you have never tried doing)

I find it generally much easier executing exercises for the first time with a machine since there is a lot more support and isolation (path of motion).

Trying free weight exercises for the first time is much harder since balance and the inclusion of other muscles plays a much larger role in performing the exercise. If done incorrectly, it is much easier to hit another muscle group more than the one you are focusing on or injuring yourself. This is apparent when I did the deadlift for the first time and wound up working out my traps too much instead of focusing on the right muscles to flex during the motion.

2- Are you aware of the dangers of improper execution of certain exercises, if so are you worried about potential injuries?

I am aware of the dangers of improper execution and worried about potential injuries. That's why I always listen to my body and am willing to cut a session short or change exercises if I feel that something Is wrong.

3 - Have you ever experienced any pain after executing an exercise, has it ever held you back from working out or doing other activities?

Yes. I have had shoulder, elbow, back and knee pain before. The elbow pain has held me back before and caused me to both pause my training for an extended amount of time and also affected my ability to carry my groceries for example.

4 - Would you consider using a personal trainer? Why or why not?

No, I am not Bill Gates.

5 - Are you able to afford a personal trainer, if yes, is the high price (5000 a month) of a personal trainer worth it?

There is no way I would pay a whole month of SU for a service that can be replaced by articles on the internet and information from my gym bros.

6 - Would you consider using an app that helps you execute exercises properly, if no, why not?

Sounds like a good idea for some. I got the homies to teach me.

7 - If you would consider using an app to help you correctly execute your workouts. What would you say would be the most important aspects of the app?

Give a visualization of the exact animation or video of the correct execution of the movement.

6.3 Interview with personal trainer

Questions and answers from Certified personal trainer Alexander Mathiesen

When executing a deadlift. What are some common mistakes that you often witness your clients make?

The most common mistake that I see my clients do, is executing the deadlift like a squat. This results in activation of improper muscles which means that the client will not train the desired muscles which are the glutes, hamstrings and lower back. When doing the exercise, it is important that the back starts as horizontal as possible so that exercise becomes a proper hinge movement.

Another common mistake was being afraid of keeping the bar close enough to the shins. This in turn leads to an increased distance between the bar and the joint executing the movement. This can result in both injuries and making the bar feel heavier than if the bar is kept close to the shin when executing the deadlift.

Another important body part to focus on during the deadlift is the back. It is ultimately not an issue if you have an arched or straight back, as long as the back position remains static throughout the whole movement. Therefore, it is a misconception that an arched back is bad when doing the deadlift. However if the back position changes during the movement, that is where the danger lies. The back is prone to disc slip which is a condition where the tissue between the bones in your spine pokes out and can leave you injured for 6-12 weeks. To conclude: keep the back in a static position throughout the whole movement.

Another spinal problem with deadlift is called hyperextension, which is when the upright position of the deadlift would be overextended leading to a backwards leaning standing position. This is very unhealthy for the spine which is not supposed to be under such weighted stress in a bended position.

Can you define the perfect deadlift?

A good rule of thumb is to keep the center of the bar aligned with the middle of the shoulderblades. This in turn leads to not leaning too far forwards or backwards and also helps balancing out your center of gravity during the exercise.

It is also important that the arms only work as hinges holding the weight, and that the muscles in the arms are not used to lift the weight. This means that the arms should be completely straight throughout the whole movement, as well as completely vertical.

We showed Alexander the pose estimation landmark framework to explain how the app was intended to function. This led to him proposing the idea of aligning the shoulder and hips in a parallel straight vertical line and analyzing whether or not the user would be able to keep it straight throughout the movement.

Do you have any sources that can support your statements? (Articles/Vidoesfr)Har du noget teori der kan understøtte det (artikler/videoer/bøger)?
maxer.dk - Jacob Beermann
mass - meta analysis of strength sports

References

[1] Snap Fitness. *Advantages Of Going To The Gym Everyday: Everything You Need To Know*. accessed: 12-05-2022. URL: https://www.snapfitness.com/ae/blog/advantages-of-going-to-the-gym-everyday-everything-you-need-to-know/.

[2] Bryn Farnsworth. *Qualitative vs Quantitative Research – What Is the difference?* accessed: 12-05-2022. URL: https://imotions.com/blog/learning/research-fundamentals/qualitative-vs-quantitative-research/.

[3] Pete McCall. *5 Benefits of Compound Exercises*. accessed: 12-05-2022. URL: https://www.acefitness.org/resources/pros/expert-articles/5811/5-benefits-of-compound-exercises/.

[4] Tyler Read. *The top 8 benefits of deadlifts*. accessed: 12-05-2022. URL: https://www.healthline.com/health/fitness/deadlift-benefits#benefits.

[5] Sang Wan Kim A Ram Hong. *Effects of Resistance Exercise on Bone Health*. accessed: 10-17-2022. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6279907/.

[6] Tony Leyland. "Spine Mechanics for Lifters". In: *the Crossfit Journal Articles* 63 (2007). DOI: https://lbs.co.il/data/attachment-files/2012/02/6975_63_07_Spine_Mechanics.pdf.

[7] S.M. McGill. "Low Back Disorders: Evidence-Based Prevention and Rehabilitation". In: *Champaign, IL: Human Kinetics* ().

[8] Google. *Mediapipe github repo*. accessed: 12-05-2022. URL: https://google.github.io/mediapipe/.

[9] Valentin Bazarevsky and Ivan Grishchenko. *On-device, Real-time Body Pose Tracking with MediaPipe BlazePose*. accessed: 12-05-2022. URL: https://ai.googleblog.com/2020/08/on-device-real-time-body-pose-tracking.html.

[10] Adeliya Latypova Ferath Kherif. *Machine Learning, Methods and Applications to Brain Disorders*. Laboratory of Research in Neuroimaging (LREN), Department of Clinical Neurosciences, CHUV, University of Lausanne, Lausanne, Switzerland, 2020.

[11] StatQuest with Josh Starmer. *StatQuest: Principal Component Analysis (PCA), Step-by-Step*. accessed: 12-05-2022. URL: https://www.youtube.com/watch?v=FgakZw6K1QQ.

[12] Jay Selig. *What Is Machine Learning? A Definition*. accessed: 12-05-2022. URL: https://www.expert.ai/blog/machine-learning-definition/.

[13] colah. *Understanding LSTM Networks*. accessed: 12-05-2022. URL: https://colah.github.io/posts/2015-08-Understanding-LSTMs/.

[14] Rian Dolphin. *LSTM Networks — A Detailed Explanation*. accessed: 12-05-2022. URL: https://towardsdatascience.com/lstm-networks-a-detailed-explanation-8fae6aefc7f9.

[15] Jupyter. *Jupyter Notebook documentation*. accessed: 12-05-2022. URL: https://docs.jupyter.org/en/latest/.

[16] Tensorflow. *tf.keras.callback.EarlyStopping documentation*. accessed: 12-05-2022. URL: https://www.tensorflow.org/api_docs/python/tf/keras/callbacks/EarlyStopping.