The development of an exposure assessment device for primary, secondary and tertiary prevention of work related back and shoulder disabilities

Internship accompanying Seminar Paper

Handed in on 21st of March 2019

Study program of Biomedical Engineering, Linz
University of Applied Sciences Upper Austria

by

Elias Höglinger

Linz, on 21st of March 2019

Evaluator:

FH-Prof. Dipl.-Ing. Dr. Thomas Haslwanter

Executive Summary

The present work aims to allow experts to prevent work related musculoskeletal disorders (WMSDs) by development of an objective measurement tool resulting in prevention of primary, secondary and tertiary injury. First steps were generating a study protocol that allows the later following neural network approach to function to the highest possible accuracy.
Twenty-six activities were handpicked according to previously conducted studies for the prediction algorithm. Additionally, a small survey was created, collecting possible confounders and additional information of the participants. The aim was to record data of 15-20 subjects. Hardware as well as software needed close attention to detail, assuring minimal faults, as this project was still in an early stage, therefore having a higher risk of error.
After collecting subject's data, it had to be preprocessed for the neural network modeling, including cutting, filters and feature generation. The activity recognition for the training dataset was at 97%, whereas test data managed 92%. These numbers sound reassuring, taking the early stage of the project into consideration. A horizontal distance prediction from the hand to the ankle was done for further additions to the system. This data, combined with a pressure insole, would allow predicting force moment. Prediction accuracy was at 6,68cm, being possibly sufficient and having room for improvement. This objective measurement tool provides promising data for its early stage and is likely to help preventing WMSDs in the future.

Table of Contents

1 Introduction

Work-related musculoskeletal disorders (WMSDs) are currently one of the biggest burdens on laborers. These injuries, resulting from overexertion in lifting, accounted for 31 percent (356,910 cases) of the total cases of all workers. The leading type of injury or illness for 2015 was sprains, strains or tears, which accounted for 37 percent of all cases, resulting in approximately 10 days away from work per case. (Bureau of Labor Statistics, 2018a) This statistic shows how important injury prevention is, not only to improve the overall health of workers, but also to decrease costs for employers and the government.

Among construction workers even 40% of all WMSDs are related to lower back injuries resulting in an estimated wage loss for private wage-and-salary construction workers of $46 million in 2014 (Wang et al., 2017).

To quantify the extent of WMSDs among workers, professionals in the ergonomic and epidemiologic field rely on exposure assessment techniques. These techniques include using self-reporting tools, simple and advanced observational techniques, and direct methods. (David, 2005)

Direct methods tend to be more accurate and precise while still quantifying level of exposure to physical workload with respect to intensity/magnitude, reiteration and duration of exposure. These methods surpass self-reporting and observational methods, which only provide a limited snapshot of work conditions at a given point in time. (Nguyen and Arroyo, 2017a)

Prevention of WMSDs can be done in three different stages. Primary prevention aims to avoid the disease or injury even before it occurs, intercepting possible hazards and suggesting to change unsafe behaviors leading to it. Secondary prevention reduces the effect of the occurred WMSD, stopping or decreasing its process and coming up with strategies for further harm with a fitting treatment. Tertiary prevention helps living with an ongoing WMSD that has chronic effects or permanent impairments. It contains damage and avoids or informs about possible newly hazardous situations.

1.1 Problem and motivation

This project focuses on the use of inertial measurement units (IMU) for the detection of different position data from any given worker. An IMU typically combines signals from 3D gyroscopes, accelerometers, barometers and magnetometers. (Bureau of Labor Statistics, 2018a) Currently, the UC Berkeley Ergonomics Research and Graduate Training Program (in coordination with SwiftMotion) have developed a wearable device called the SpineTrack which consists of two IMUs to track spinal motion. However, this system has difficulty predicting tasks like carrying and does not quantify exposure to the shoulders, another common body region injured in manual material handling work. In this project, six sensors will be used to measure the position, acceleration and velocity of various body segments (torso, arms, and thighs) then integrated to estimate the joint angles and movement of the shoulders, spine and hip. Using information from the IMUs, pattern detection algorithms will be used to determine the activity of the worker. Data will be collected to build prediction models and validation of

the prediction models will be performed using video analysis as the gold standard. The information on posture, movement and activity detection will be summarized to provide an overview of the physical demands of a job and the associated risk of MSDs. (Nguyen and Arroyo, 2017a)
A physical demand analysis (PDA) is a systematic procedure to qualify and evaluate the physical and environmental exposures required by employees to perform tasks of their job. It is primarily focused on the physical demands of the job rather than the physical capacities of the worker and describes physical exertion required for performing occupational tasks like sitting, standing, walking, lifting, carrying, reaching, pushing and pulling. A PDA should be conducted by a worker and employer representative, but it would be beneficial if an external consultant conducted the PDA. There are currently no objective measurement tools to quantify and qualify a job or perform a PDA. This lack of an objective measurement tool is of the main reasons the development of this exposure assessment device is greatly needed. (Bureau of Labor Statistics, 2018b; Occupational Health Clinics for Ontario Workers Inc., 2018)

1.2 Aims

The study consists of the following three aims. Aim two and three where carried out in an overlapping timeframe, to see how the artificial neural network (ANN) model would perform.

1. Upgrade the current “SpineTrack” system from two to eight IMUs and calculate the body posture, while ensuring the data to be logged with as little data drops as possible.
2. Collect data with 20 subjects performing different physical activities and tasks.
3. Use artificial neural networks (ANN) to predict mentioned activities including lift/lowering, carrying, push/pulling, walking, crawling, kneeling, sustained forward bend, reaching (overhead & sustained forward reach), etc. Validation of the models will be essential.

2 Background

2.1 Inertial Measurement Units (IMU)

Inertial measurement units (IMU) are usually a combination of three different sensors. Magnetometers measure magnetic fields, gyroscopes orientation and angular velocity and the acceleration is measured by accelerometers. All of the following sensors are widely used in the industry, like in aerospace engineering, research, navigation etc.

2.1.1 Magnetometer

Magnetometers are sensors measuring magnetism. They are split into two groups, laboratory magnetometers where you have to put the sample material inside and survey magnetometers which are divided into two basic types. Scalar magnetometers measure the absolute intensity of a magnetic field they are exposed to and vector magnetometers measure only the direction of the magnetic field, allowing it to measure the spatial orientation of the sensor relative to the field. There are several different approaches of realizing such a magnetometer, which I will not go into. (Ripka (ed), 2002)

2.1.2 Gyroscope

Gyroscopes measure or maintain orientation and angular velocity. They also are used in various roles such as stabilization, autopilot systems and other forms of navigation. One sensor could provide the above data but usually they are used with two or more separate clusters of sensors. The original basic design makes use of inertial property of a wheel (three wheels are used for a three axis approach), newer designs are called unconventional sensors and are for example vibratory gyroscopes, nuclear magnetic resonance (NMR) gyroscopes, micro-machined electromechanical system (MEMS) gyroscopes and several more. [8]

2.1.3 Accelerometer

Accelerometers measure proper acceleration, which is the rate of change of velocity of the object being observed. They are also widely spread in many fields and are even built in our everyday phones used for tilting images and other applications. Most of these small devices use the already mentioned MEMS technology.

2.1.4 Machined Electromechanical Systems (MEMS)

The steady rise in use of IMUs and other sensors pushed the industry to come up with a cheaper, more energy efficient and smaller version of sensors. Early versions of alternative sensors remained rather costly because of the high-precision tolerances, precision assembly techniques and accurate testing required. However MEMS are using silicon based materials in combination with chemical etching and batch processing techniques that are already in use by the electronics integrated circuit industry, reducing cost dramatically. By furthermore reducing the number of parts used MEMS are currently the “Jack-of-all-Trades” of sensors and providing engineers with a level of design flexibility and accuracy beyond anything previously developed. (D. Titterton and J. Weston, 2005)

2.2 Artificial Neural Networks (ANN)

Neural Networks (NN) generally are a network or circuit of neurons. An example of a biological NN would be the human brain. Artificial neural networks (ANN) on the other hand are made up of artificial neurons and are intended to solve complex problems. New advances in that field (LeCun et al., 2015) have already made it possible to solve complex problems in a reasonable amount of time using this form of artificial intelligence (AI). ANNs make use of several neurons as shown in Figure 2-1, cross-linking all of them in each layer. On the left the external input is applied to the system and after working through each layer the output is exported on the right side.

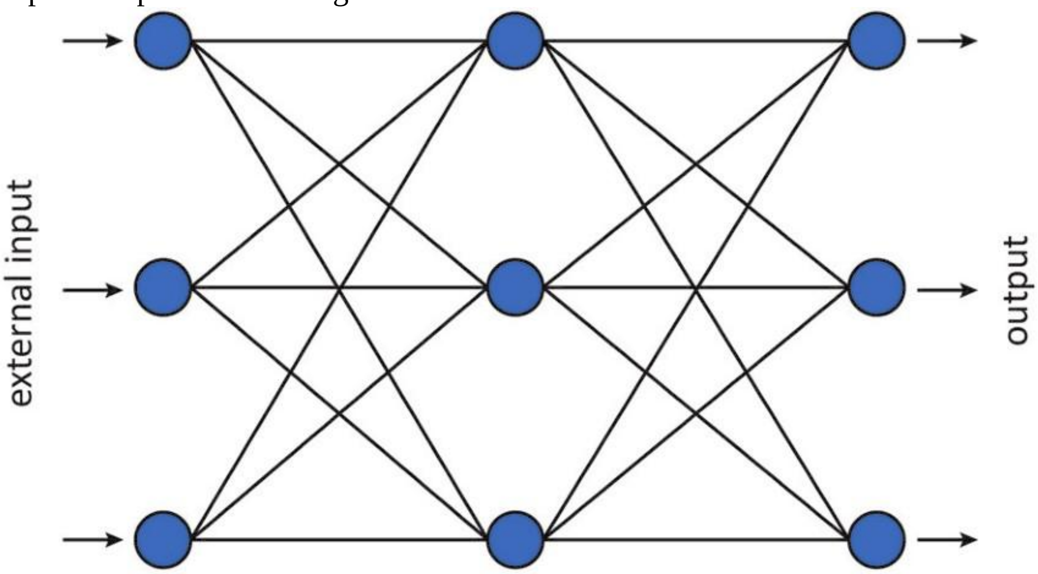

Figure 2-1 Example artificial neural network (Yuste, 2015)

This mentioned systems function in many ways similar to a human brain. They use the same techniques, learning, perceiving, remembering, acting, predicting and reasoning data. (van Gerven and Bohte, 2017)

2.3 Physical Demands Analysis (PDA)

A physical demand analysis (PDA) is a systematic procedure to qualify and evaluate the physical and environmental exposures required by employees to perform tasks of their job as mentioned above. It is an important part of the analytical process of determining compatibility between workers and their specific jobs. The Workplace Safety and Insurance Board has the right to request information on injured worker's remaining functional abilities from their treating health professionals, according to the Bill 99 (Workplace Safety and Insurance Act, 1997) The required Functional Ability Evaluation is performed to determine a workers remaining working capability after an injury to prevent further injury. It furthermore assesses the return to work potential. A PDA does so by breaking the job up into its individual tasks and quantifying environmental conditions, use of machines, equipment, tools, work aids, and specific physical demands of the job. (Bureau of Labor Statistics, 2018b; Occupational Health Clinics for Ontario Workers Inc., 2018)

3 Methods

In the following points I will discuss the recording of the training dataset and the protocol design necessary. I will also address the generation of the artificial neural network (ANN).

3.1 Hardware

On the following pages the used hardware and specifics are discussed. This includes the used vest for holding the IMUs in place, the IMUs themselves and the data logging.

3.1.1 Vest

The vest was modified from the previous version containing two sensors, to be able to hold eight. Elastic straps secure the IMUs to the middle of the upper arm, the wrist and above the knees on the thighs. Consistent position of the sensors was a key aspect for the data collection. Any error in placement, like turned upside down sensors, could result in creation of unusable data. A front view of the vest is shown in Figure 3-1, and a back view in Figure 3-2. Currently there is only one general size, with more sizes in production.

Figure 3-1 SpineTrack vest front view

Figure 3-2 SpineTrack vest back view

The vest has already shown not being to invasive for construction workers (Nguyen and Arroyo, 2017b), as it is light, easy to use and not too restrictive.

3.1.2 IMUs

For orientating in a 3D space all sensor data has to be fused together. We used a 3-axis magnetometer, a 3-axis gyro and a 3-axis accelerometer with advanced sensor fusion for orientation all of which are MEMS. Using this amount of sensors meant that streaming the data could pose a reasonable problem, which is why a 2.4 GHz Wi-Fi with two receivers to transmit and obtain data was used. For on board storage a 1GB SPI Flash was used.

3.1.3 Data Logging

After a series of optimizations of the recording algorithm we had to get rid of any unnecessary code for reading the data, as any additional time spent in the while loop in line 19. Listing 1 shows the whole loop, basically only consisting of the actual writing of the data.

```
19 while(1)
20     data = str2num(fscanf(s));
21     data1 = str2num(fscanf(s1));
22
23         telapsed = toc(tstart);
24         dlmwrite(op_fname,[ telapsed (data)],'-append');
25
26         telapsed = toc(tstart);
27         dlmwrite(op_fname,[telapsed (data1)],'-append');
28
29 end
```

Listing 1 While loop for data acquisition [Appendix 1]

The simple jet effective code in Appendix 1 also shows the usage of two COM-ports, representing the two Wi-Fi receivers.

The logging of the data was a unique problem. Because the eight sensors steamed 16 columns of data every 1ms, we had to use two Wi-Fi dongles connected via USB, collecting data from four sensors each. To ensure we did not drop too much data while transferring this amount, we analyzed each of the pilot files using a simple Code (Appendix 5). As you can see in Figure 3-3 the boxplot is very narrow, indicating a consistent transfer of data. Matlab calculates 50% of the data inside the blue box and the whiskers extend to the most extreme data points not considered outliers, which are shown as a red cross.

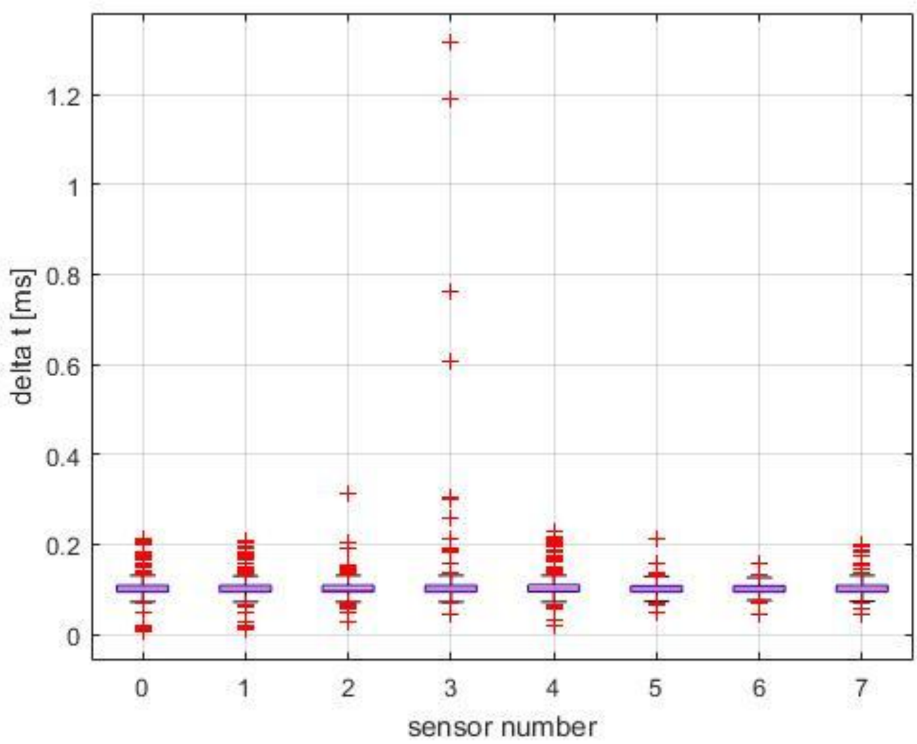

Figure 3-3 data drop analysis

In early stages of the project many transfer problems occurred, most of which could later be tracked to a small error in code. This error was an if- statement for every sensor, only proceeding to save data from the next sensor if it did not receive data from the first. This resulted in a lower likelihood of transmitting for some sensors and unnecessary iterations, thus resulting in data drops.

3.2 Study protocol

The protocol was designed to train the ANN in the most effective way. Different resources where considered, the main tasks where redesigned from the previous study [Spinetrack1], the physical demand analysis [PDA] and meetings with Dr. Harris and Mr. Venkatasubramanian.

3.2.1 Recruitment

Participants were recruited on three different ways, one being a public Facebook page named Free & for Sale, a craigslist ad and fellow students.
Include criteria:

- Familiarity with tools/equipment to be studied
- Between 18 and 65 years of age

Exclude criteria:

- Chronic neck, back, arm, shoulder, or vision conditions

3.2.2 Study participants

Six women and nine man (mean age = 35.6 years, SD = 14.1, range = 20-64) participated in the study. Their mean height was 165,1cm, a SD of 18cm and a range from 130cm to 195,6cm. For weight the mean was 67,6kg, with a SD of 15kg and a range from 48,5kg to 91,2kg. None of the participants reported any chronic diseases (asthma, high blood pressure or diabetes), recent injuries (tendinitis, back problems, fracture or shoulder problems) or discomfort greater than 2 on a one to 10 scale. On the question whether they perform physical activity / exercise such as walking more than 20 minutes, fitness classes, biking, etc. one third answered with 1-2 times a week, one third with 3-5 times a week and one third daily. The majority of the population, seven subjects, reported being students, three full-time employed, another three part-time employed and the remaining two consisted of one unemployed person and a homemaker.

3.2.3 Consent

Consent was obtained of each subject individually using an informed consent document for participating in an ergonomic research study of tools and equipment approved by the IRB. Each participant was also offered and provided with a copy of that document. Questions like purpose, procedures, benefits, risks/discomforts, confidentiality, reimbursement, and contact for questions are answered on this form.

3.2.4 Training protocol

One of the main tasks for this project was to come up with a training or physical activity protocol. The activities have to be representative for various workplace activities to ensure correct evaluation for a PDA and better accuracy in the different NN models. In the development phase the conclusion was drawn that several resources where important for the creation of the protocol, the PDA (an example: (Occupational Health Clinics for Ontario Workers Inc., 2018)) being an obvious one. The previous SpineTrack system already had a protocol tailored to

construction workers, which was used as a baseline. This system used two sensors located on the spine. (Nguyen and Arroyo, 2017a) Because of the increase in sensors to a total number of eight, it was expected that one handed activities, more specific lifting tasks, crawling, crouching and overhead work could be predicted, too. In the early stages of the protocol it also varied the weights between 2,25, 4,5 and 9 kg. It was later decided to drop the highest and lowest weight and take 4,5kg consistently, as the ANN models could not predict differences in the weights without force recordings in the system. Additionally, more lifting data was needed, as the lifting itself is only a short activity ranging from two to four seconds for each lift. The final repetitions for the lifting tasks were fifteen repetitions twice, with a short pause of about two minutes. The previous mentioned specific values of 2.25, 4.5 and 9 kg where chosen correlating to the literature. Research papers usually refer to a maximum cutoff as the weight that 75% of the female population can still handle. (Snook and Ciriello, 1991)
The final protocol consists of the 26 activities listed in Table 1. The initialization for each physical activity was standing still for five seconds with palms facing medial. Lifting tasks had this pause also incorporated between each lifting and lowering and after the last lowering. About one to two minutes of pause was held between the sets of fifteen repetitions to avoid fatigue. All the other activities also had five seconds of standing still at the beginning and end of them.

Label	Weight [kg]:	Time [s]:	Repetitions [#]:
01. Crawling		60	
02. Lifting (Floor to shoulder):	4,5		2x15
03. Lifting (Floor to waist squat):	4,5		2x15
04. Lifting (Shoulder to waist with twist):	4,5		2x15
05. Lifting (Shoulder to waist w/o twist):	4,5		2x15
06. Lifting (Floor to waist stoop):	4,5		2x15
07. One handed Lift (Right to waist):	4,5		2x15
08. One handed Lift (Left to waist):	4,5		2x15
09. Lifting (Floor to 60in):	4,5		2x15
10. Push Clockwise		60	
11. Push Counterclockwise		60	
12. Pull Clockwise		60	
13. Carrying:	4,5	60	
14. Static stoop:		60	
15. Walking:		60	
16. Pulling (One handed, R)		60	
17. Pulling (One handed, L)		60	
18. Sitting		60	
19. Standing		60	
20. Crouching		60	
21. Reaching close (STANDING)		60	Apr. angle [°]: 30
22. Reaching far (STANDING)		60	Apr. angle [°]: 90
23. Reaching high (STANDING)		60	Apr. angle [°]: 135
24. Kneeling		60	
25. Overhead Static		60	
26. Overhead Dynamic		60	

Table 1 All activities performed with specifications

These 26 activities are used to train the different ANN models, explained in more detail in 3.4. The data collection took about three to three and a half hours for this training data.

3.2.5 Test protocol

The test or task protocol was used to validate the generated models on a separate dataset. Three main fields of work have been chosen (carpentry, bottle packing and drilling). Each of the chosen tasks had three different variants. Carpentry and bottle packing only vary the lift height between floor, waist and shoulder, drilling consists of overhead drilling, stooped drilling and overhead painting. For the carpentry tasks subjects first had to lift carpet from a shelf onto a cart and then push and pull the cart for about two meters. After placing the carpet on the floor, they were asked to lay the carpet in a pre-defined rectangle. Bottle packing started with opening the box and furthermore putting each of the twelve bottles, four of them on the same distance (close, medium far and far reach), into the box and closing it. Finally, the box had to be carried to an about two meters away shelf on different heights. Drilling involved picking up a drill, or paint roller with one hand, walking to the designated area about one meter away and drilling overhead, drilling stooped and overhead dynamic painting. Afterwards the tool was returned

to the original spot.
Initialization for each of the tasks was five seconds of standing still with the arms to the side, palms facing medial. This was also done at the end of each task.
The final task tries to incorporate all previously performed physical activities into one combined task, with each of the activities lasting for about 15 seconds. This final task data was crucial for determining the accuracy of the ANN models.

3.2.6 Horizontal distance prediction

We were curious whether we could predict how far away a subject was holding an object with the generated data. To validate the horizontal distance prediction of the distance D, we came up with eight different positions (Figure 3-4) we put each subject in and measured the true distance with a tape measure.

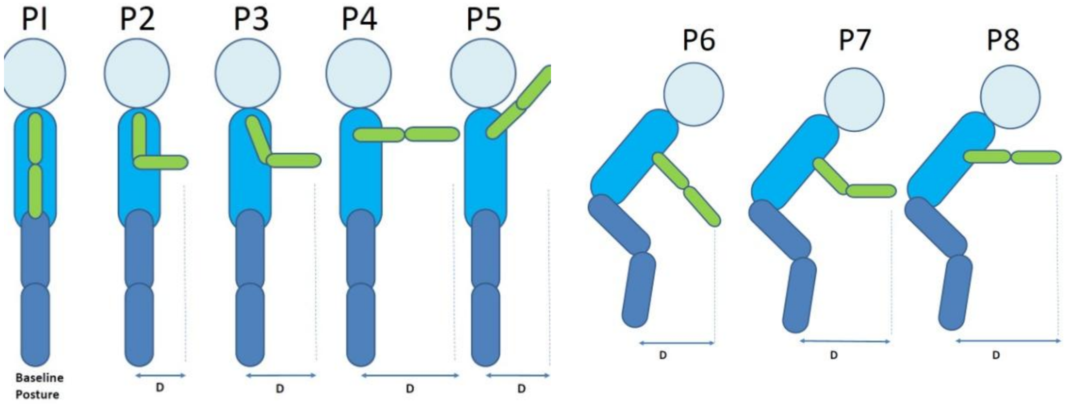

Figure 3-4 Horizontal distance prediction positions with distance D

The eventual goal of this prediction is to estimate the load moment once load estimates are available.

3.3 Survey

All subjects were asked to fill out a baseline survey at the beginning of the study. There was no subject having recent injuries (tendinitis, back problems, fracture, and shoulder problems) or chronic diseases like e.g. Asthma, High Blood Pressure or Diabetes that could have affected their ability to exercise.

3.4 Artificial Neural Network Modeling

One of the main objectives of this work is to generate an ANN model that can predict different activities, all 26 of which are mentioned in 3.2.4. In order to train an ANN there is a lot of preprocessing necessary, ranging from cutting data to generating so called super-features. These tasks greatly influence the accuracy of the model.

3.4.1 Preprocessing

Several steps had to be performed in order to prepare the data for the ANN processing. The first step was to cut the standing still portion out of the data, as mentioned in 3.2.4. This was performed using a simple script with the function ginput (graphical input). Lines 15-17 in Listing 2 read the data from the files, line 20 plots the data, line 21 lets the user choose the cutting points and lines 22-24 cut the data and save it in the “Processed” folder.

```matlab
for ii = 1:length(files)
    input_filename = files(ii).name;
    X = ['Now processing...', input_filename]; disp(X)
    cal_raw_data = csvread(input_filename,2);

    if (strcmp(input_filename,'Carrying.csv'))
        plot(cal_raw_data(:,14:19)); title(input_filename, 'Interpreter', 'none')
        [x,y]=ginput(2);
        data_need = cal_raw_data(floor(x(1)):floor(x(2)),1:19);
        data_need(:,1) = data_need(:,1)-data_need(1,1);
        csvwrite('Processed\Carrying.csv',data_need);
        clear cal_raw_data x y data_need
```

Listing 2 Preprocessing training dataset (final protocol) [Appendix 4]

The cutting of the data is essential for the performance of the ANN, therefore it had to be done carefully. The cutoff points had to be as close to the data as possible, as the first ANN had problems confusing lifting activities and standing idle. This was due to not cutting the standing in between enough, therefore leaving standing data in the training dataset. An example of these cuts with a raw data file is given in Figure 3-5.

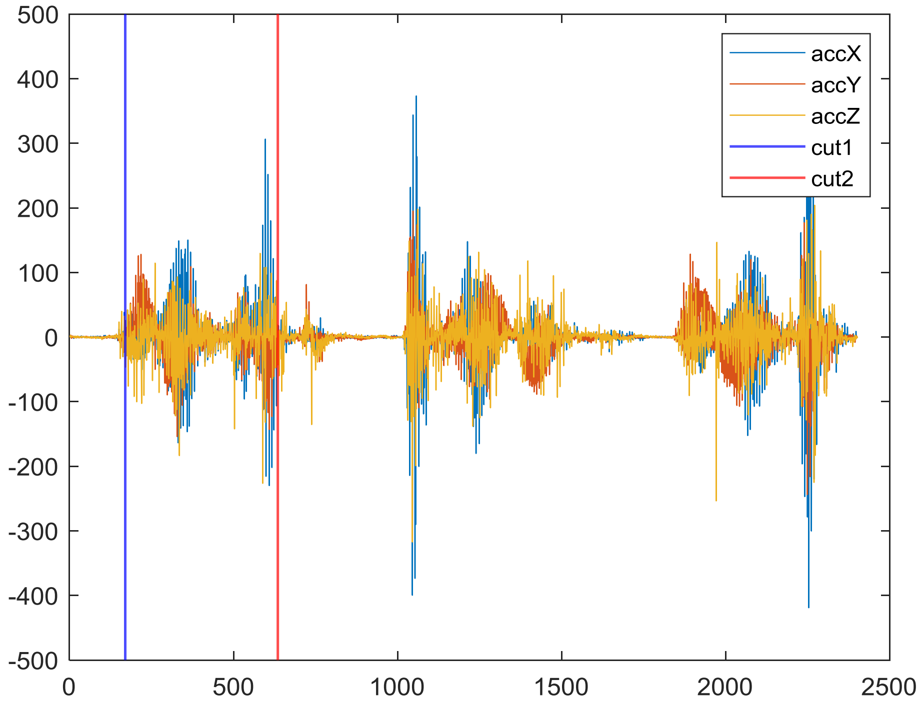

Figure 3-5 Example of floor to shoulder lift preprocessing with cutting points

In the next step a so called "super feature" file had to be generated by using a method called feature extraction using the previously cut data. This file basically translates the data of the original data files into a form that the ANN can use and make predictions with. Line 18-70 (Appendix 2) extract each sensors data individually. The following lines 85, 86 and 89 build a new time index based on the minimal and maximal timestamp values of all sensors. This new index is used for interpolating the quaternion, gyroscope, linear and raw acceleration data. Using Euler equations a matrix is generated, consisting of trunk angles, right and left shoulder elevation, right and left elbow angle and right and left knee inclination (line 123-203, Appendix 2). These extracted features get merged with the quaternion, linear acceleration, gyroscope and raw acceleration data into one matrix. To complete the matrix labels from 0-24 where added. The correlation of the labels is summarized in lines 233-257. The same was done for the test dataset in Appendix 3 , so that the ANN could predict activities on the task data.

4 Results

Main results of this study were the activity recognition and the horizontal distance prediction. Other results and conclusions could be drawn with the generated dataset, too, but time limited it to these two results.

4.1 Activity recognition

The activity recognition was done using the described ANN approach mentioned in 3.4. The confusion matrix in Figure 4-1 shows how accurately which state is predicted on itself. The overall accuracy of 97% is only representative for the training dataset, as it is compared to itself. Number 13, which represents pulling has the lowest training accuracy over all other tasks, still being 92%. More iterations of the training would improve that but would likely result in an over-fit, thus making the model perform worse in the actual prediction.

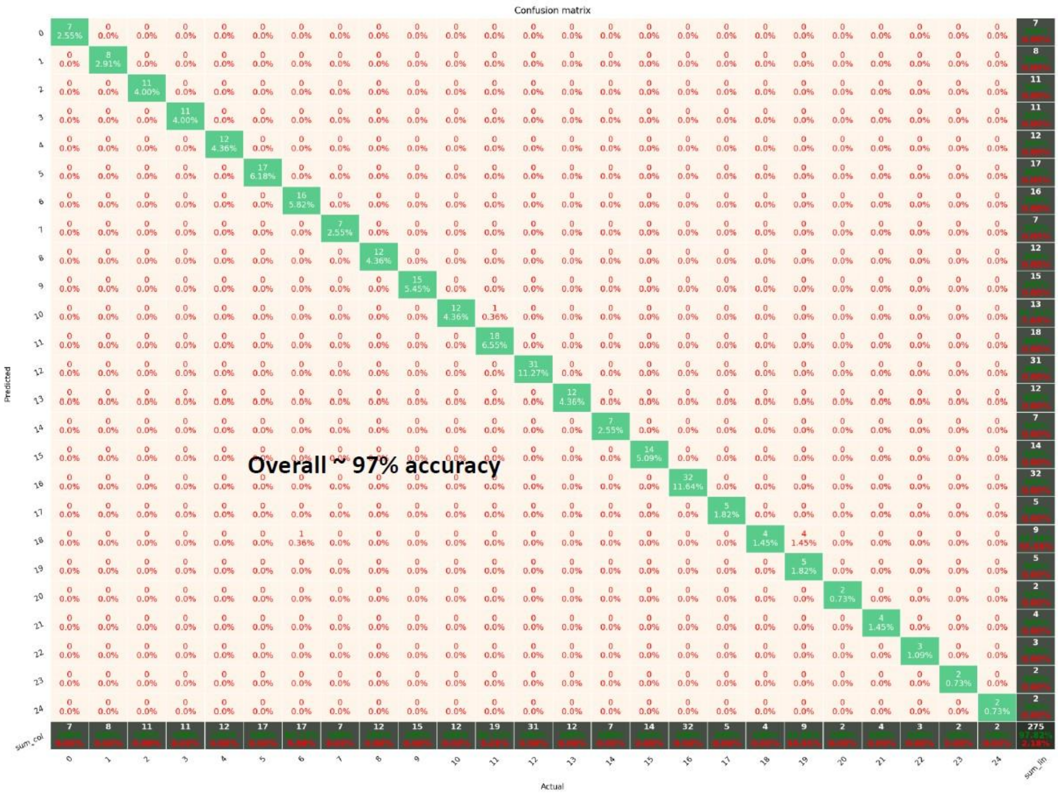

Figure 4-1 Confusion matrix

For the purpose of activity recognition, the in 2.2.4 mentioned test protocol data was used. This was harder to predict, because it was not the exact same movements. Unfortunately, no exact accuracy can be given yet, as the frame by frame analysis of the video is still being done. An early comparison is shown in Figure 3 2. On the left side the algorithm gives a best estimate of current activity and on the right is the video at the same point in time. A rough estimate of the accuracy is 92%, which would be sufficient for the intended purposes.

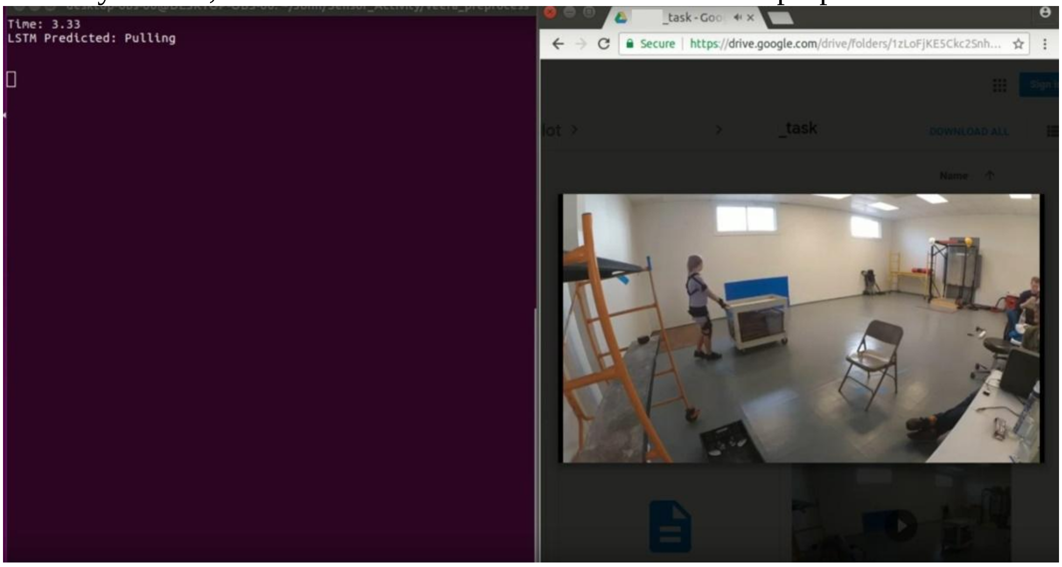

Figure 4-2 Side by side prediction

4.2 Horizontal distance prediction

Figure 4-3 shows a direct comparison of the true distance (red) in each posture for each subject, whilst also showing the estimated distance (green) of the model. Errors for each posture are shown below in blue.

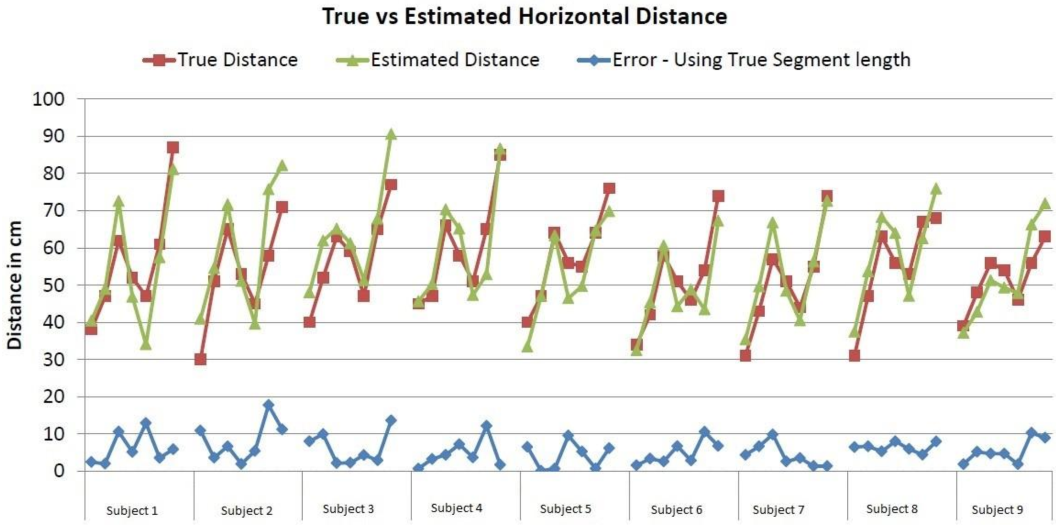

Figure 4-3 Comparison of true and estimated horizontal distance with error rate

In Figure 4-3 target distances are plotted against the actual outputs. The resulting RMS error was 6,68cm.

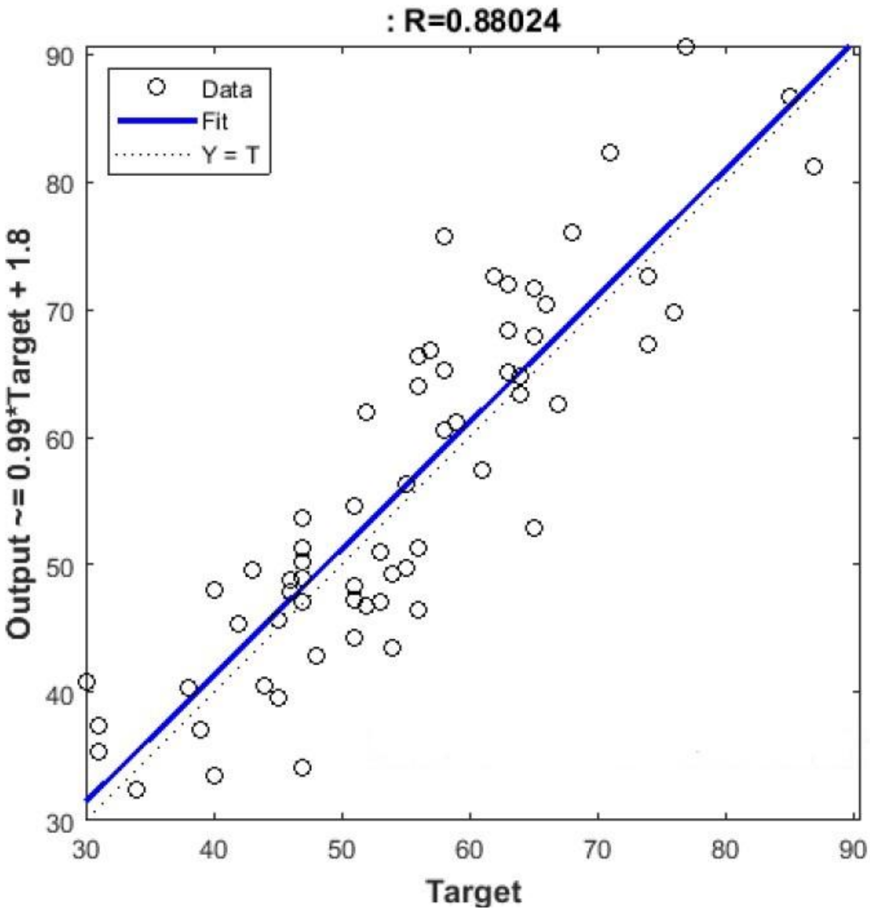

Figure 4-4 RMS Error

When using estimated segment lengths, which can be calculated from the body height, error increases significantly. A comparison of the true segment length vs. the estimated segment length is shown in Figure 4-5.

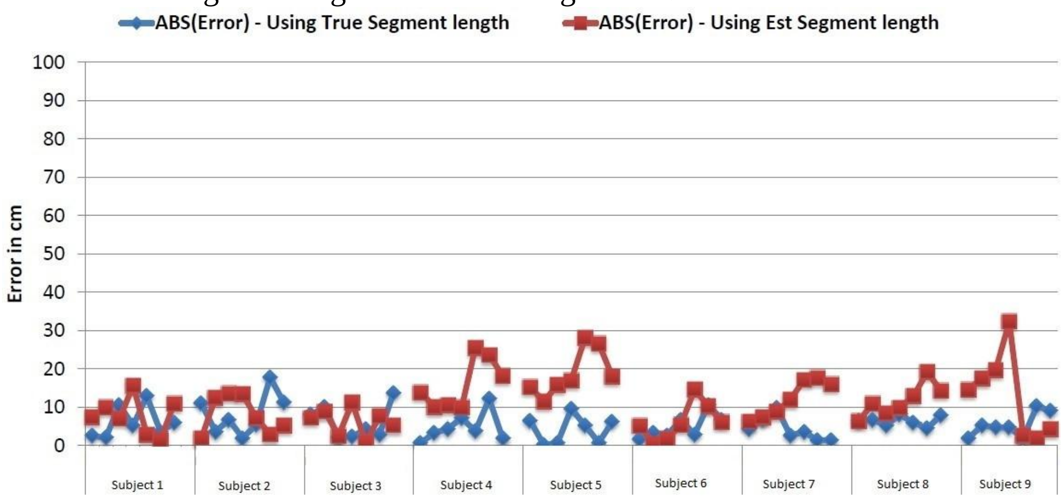

Figure 4-5 True vs. estimated segment length

5 Discussion

This early study showed how much valuable information the SpineTrack system provides although being fairly accurate. There are still certain restrictions, errors or room for upgrades.

5.1.1 Activity recognition

While the learning algorithm has a high accuracy on the training dataset, consistency and accuracy of actual data predictions are only fairly accurate. In addition to that many activities have not been covered by this study, like varying heights of one-armed lifts, walking backwards, jumping, climbing a ladder, etc. This would cause the system to perform poorly in an actual working environment while still giving a good first analysis. Information on the exposure of a worker to manual material handling (MMH) tasks though, is a crucial first step towards evaluating workplace settings objectively. MMH tasks have been linked to low-back disorders in many studies like (Marras et al., 2000) while ergonomic interventions are becoming more effective and more available. Knowing the exposure of workers would not only reduce overall injury rates, but also ease return to work for the previously injured.

5.1.2 Horizontal distance prediction

An important step for further improvements in the SpineTrack project is the horizontal distance prediction. A low error rate would ensure better predictions for load moment predictions, covered in the following study. The RMS error of 6,68cm still needs improvement for accurate predictions.

5.1.3 Limitations

The study is an important step in the right direction, but there is room for improvement in different aspects. The study population and therefore the training database for the ANN approach is possibly to low for reliable predictions in a normal working environment. The average weight as well as height of the subjects involved was also relatively low compared to the average of California citizens, especially considering 66% of the subjects where male and roughly 50% students. A larger study group could reduce that bias significantly. Consistent placement of the sensors was a constant problem, likely causing some of the accuracy to drop. This could be improved by a sturdier design of the vest and different sizes, like small, medium and X-large.

6 Future work

There are several possible improvements for the system. The mentioned larger population could help the ANN significantly. Furthermore, different sizes for the vest are already in production. For an accurate validation of the activity recognition the captured video has to go through a frame by frame analysis and then be compared to the actual predictions. Transition states are nearly impossible to predict as a specific activity. A possible way to overcome this would be to train the model with a transition activity or label confusing input for the system as transition. The prediction algorithm could be split into two layers, predicting outer activities first, like walking, standing or lifting and only differentiating lifting into specific lifts once the general lifting activity has been recognized. The in Figure 6-1 shown dashboard is currently under development. It shows percentages spent in active, idle and MMH time helping the wearer and others analyze the routine workday and work to rest ratios, which are likely to be a risk factor for low back disorder (Sbriccoli et al., 2007). Time spent in activities is plotted in the bar diagram below (sample data). With this tool work day assessment and PDAs would be highly objective and easy to review.

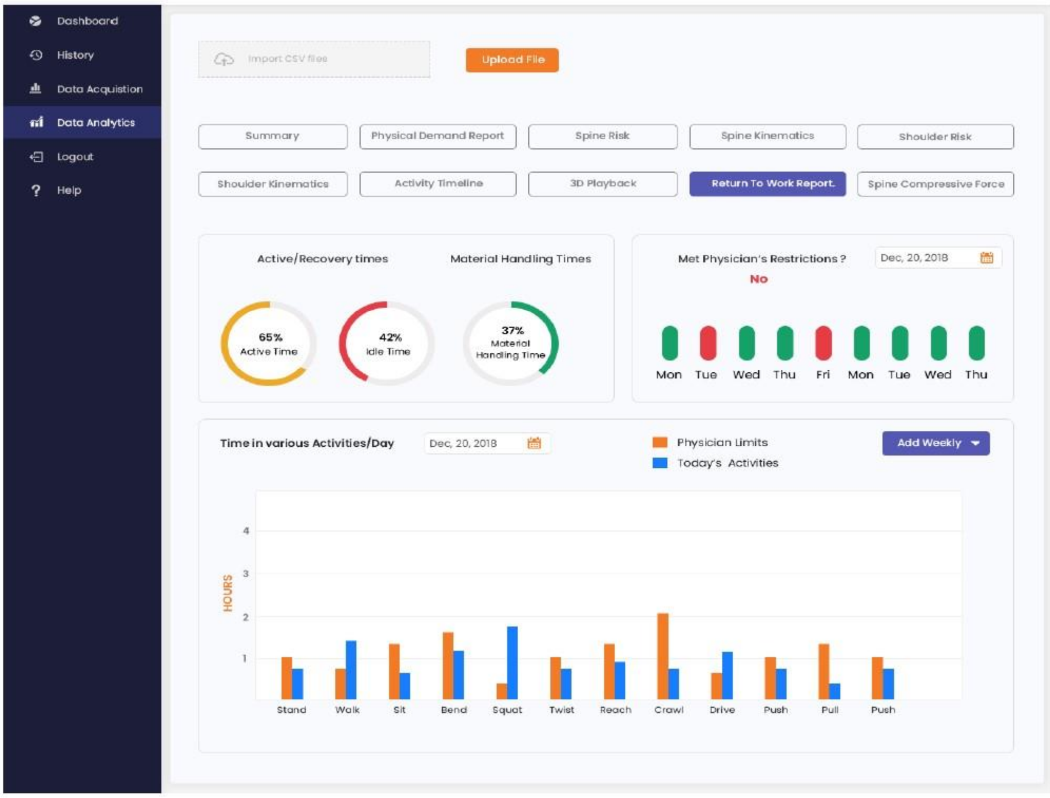

Figure 6-1 dashboard integration

Taking a closer look at the physical demands report in Figure 6-2 we are trying to summarize physical demands and splitting them up in time spent in each category.

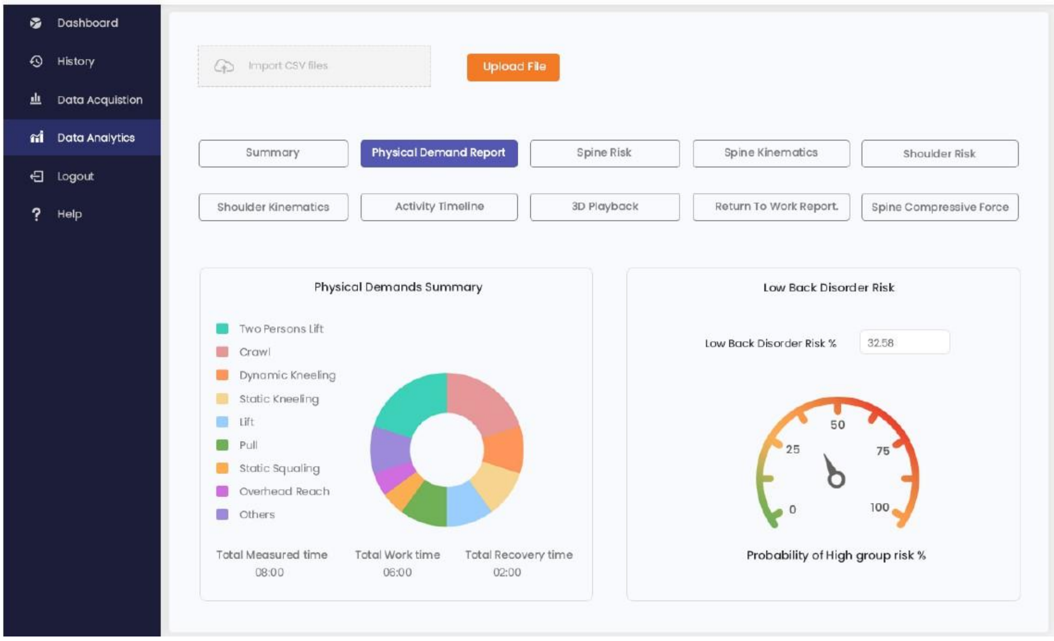

Figure 6-2 physical demand analysis

The lower back disorder risk estimation could be done with help of the Bloswick model (Merryweather et al., 2009) shown in Equation 1. In this model BW is representative for the subject’s weight in pounds, L represents the applied load in also in pounds, HB the horizontal distance from the hands to the L5/S1 of the lower back in inches and Θ the torso angle to the horizontal plain.

$$F_c = .045(\mathrm{BW})(H)\cos\Theta + \frac{L(\mathrm{HB})}{2} + 0.8\left(\frac{\mathrm{BW}}{2} + L\right)$$

Equation 1 Bloswick model (Merryweather et al., 2009)

An additional system would be a pressure insole, adding a force measurement to the system. This additional system would require more data being transmitted but could be used for load moment and load factors in different parts of the body. Spinal load factors are known to be a high-risk probability of occupational low-back disorder (Granata and Marras, 1999), therefore being a vital parameter in preventing WMSDs.

References

Bureau of Labor Statistics, 2018a. Nonfatal Occupational Injuries and Illnesses Requiring Days Away From Work, 2015.

Bureau of Labor Statistics, 2018b. PHYSICAL DEMANDS.

D. Titterton, J. Weston, 2005. Strapdown inertial navigation technology - 2nd edition - [Book review]. IEEE Aerosp. Electron. Syst. Mag. 20, 33–34. https://doi.org/10.1109/MAES.2005.1499250

David, G.C., 2005. Ergonomic methods for assessing exposure to risk factors for work-related musculoskeletal disorders. Occup. Med. Oxf. Engl. 55, 190–199. https://doi.org/10.1093/occmed/kqi082

Granata, K.P., Marras, W.S., 1999. Relation between spinal load factors and the high-risk probability of occupational low-back disorder. Ergonomics 42, 1187–1199. https://doi.org/10.1080/001401399185072

LeCun, Y., Bengio, Y., Hinton, G., 2015. Deep Learning. Nature 521, 436–44. https://doi.org/10.1038/nature14539

Marras, W.S., Allread, W.G., Burr, D.L., Fathallah, F. a, 2000. Prospective validation of a low-back disorder risk model and assessment of ergonomic interventions associated with manual materials handling tasks. Ergonomics 43, 1866–1886. https://doi.org/10.1080/00140130050174518

Merryweather, A., C Loertscher, M., Bloswick, D., 2009. A revised back compressive force estimation model for ergonomic evaluation of lifting tasks. https://doi.org/10.3233/WOR-2009-0924

Nguyen, P., Arroyo, J., 2017a. The effectiveness and usability of a personal device for the prevention of musculoskeletal disorders in construction.

Nguyen, P., Arroyo, J., 2017b. The effectiveness and usability of a personal device for the prevention of musculoskeletal disorders in construction.

Occupational Health Clinics for Ontario Workers Inc., 2018. PHYSICAL DEMANDS ANALYSIS (PDA).

Ripka (ed), P., 2002. Magnetic Sensors and Magnetometers. Meas. Sci. Technol. 13, 645–645. https://doi.org/10.1088/0957-0233/13/4/707

Sbriccoli, P., Solomonow, M., Zhou, B.-H., Lu, Y., 2007. Work to rest durations ratios exceeding unity are a risk factor for low back disorder; a feline model. J. Electromyogr. Kinesiol. Off. J. Int. Soc. Electrophysiol. Kinesiol. 17, 142–152. https://doi.org/10.1016/j.jelekin.2006.01.014

Snook, S.H., Ciriello, V.M., 1991. The design of manual handling tasks: revised tables of maximum acceptable weights and forces. Ergonomics 34, 1197–1213. https://doi.org/10.1080/00140139108964855

van Gerven, M., Bohte, S., 2017. Editorial: Artificial Neural Networks as Models of Neural Information Processing. Front. Comput. Neurosci. 11, 114–114. https://doi.org/10.3389/fncom.2017.00114

Wang, X., Dong, X.S., Choi, S.D., Dement, J., 2017. Work-related musculoskeletal disorders among construction workers in the United States from 1992 to 2014. Occup. Environ. Med. 74, 374–380. https://doi.org/10.1136/oemed-2016-103943

Yuste, R., 2015. From the neuron doctrine to neural networks. Nat. Rev. Neurosci. 16, 487–497. https://doi.org/10.1038/nrn3962

Appendix

```
clear
close all

delete(instrfindall);
s = serial('COM13');
s.BaudRate = 115200;
fopen(s);

s1 = serial('COM5');
s1.BaudRate = 115200;
fopen(s1);

ii = 1;
% create a CSV file with name from dateTime
op_fname = strcat(num2str(now),'.csv');
t = cputime;
tstart = tic;

while(1)
    data = str2num(fscanf(s));
    data1 = str2num(fscanf(s1));

        telapsed = toc(tstart);
        dlmwrite(op_fname,[ telapsed (data)],'-append');

        telapsed = toc(tstart);
        dlmwrite(op_fname,[telapsed (data1)],'-append');

end
fclose(s);
```

Appendix 1 Full code for data acquisition of the sensors (data_acq4)

```
%close all
clear
% Input Stand Data. Read the Back, Shoulder, Elbow, Knee angles
% input_filename =
'C:\ergo_related\SpineTrack\DeepLearning\Charlotte_data\Walking.csv';
% [filepath,name,ext] = fileparts(input_filename);

cd 'E:\Programme\GoogleDrive\SpineTrack_activityData\Non Pilot\Elias_12-7-
18\Elias_data\Processed'
files = dir('*.csv');
super_features = []; % matrix that will hold all training data set along with labels

for ii = 1:length(files)
    input_filename = files(ii).name;
    X = ['Now processing...', input_filename]; disp(X)
    cal_raw_data = csvread(input_filename,1);

    [cal_rows_1]=find(cal_raw_data(:,2)==1); % L5-S1 rows
    [cal_rows_0]=find(cal_raw_data(:,2)==0); % T3-T4 rows

    [cal_rows_2]=find(cal_raw_data(:,2)==2); % L5-S1 rows
    [cal_rows_3]=find(cal_raw_data(:,2)==3); % T3-T4 rows

    [cal_rows_4]=find(cal_raw_data(:,2)==4); % L5-S1 rows
```

```
[cal_rows_5]=find(cal_raw_data(:,2)==5); % T3-T4 rows

[cal_rows_6]=find(cal_raw_data(:,2)==6); % L5-S1 rows
[cal_rows_7]=find(cal_raw_data(:,2)==7); % T3-T4 rows

sensor_list = unique(cal_raw_data(:,2));
if(length(sensor_list) < 7)
    continue;
end

if(find(sensor_list==3))
    cal_rows_3 = cal_rows_3;
else
    cal_rows_3 = cal_rows_2;
end

if(find(sensor_list==2))
    cal_rows_2 = cal_rows_2;
else
    cal_rows_2 = cal_rows_3;
end

if(find(sensor_list==4))
    cal_rows_4 = cal_rows_4;
else
    cal_rows_4 = cal_rows_5;
end

if(find(sensor_list==5))
    cal_rows_5 = cal_rows_4;
else
    cal_rows_5 = cal_rows_4;
end

if(find(sensor_list==6))
    cal_rows_6 = cal_rows_7;
else
    cal_rows_6 = cal_rows_7;
end

if(find(sensor_list==7))
    cal_rows_7 = cal_rows_6;
else
    cal_rows_7 = cal_rows_6;
end

ts_1 = cal_raw_data(cal_rows_1,1);
ts_0 = cal_raw_data(cal_rows_0,1);

ts_2 = cal_raw_data(cal_rows_2,1);
ts_3 = cal_raw_data(cal_rows_3,1);

ts_4 = cal_raw_data(cal_rows_4,1);
ts_5 = cal_raw_data(cal_rows_5,1);

ts_6 = cal_raw_data(cal_rows_6,1);
ts_7 = cal_raw_data(cal_rows_7,1);

% Get max and min of all timestamps across all sensors
```

```
max_ts = max([max(ts_0) max(ts_1) max(ts_2) max(ts_3) max(ts_4) max(ts_5) max(ts_6) max(ts_7)]);
min_ts = min([min(ts_0) min(ts_1) min(ts_2) min(ts_3) min(ts_4) min(ts_5) min(ts_6) min(ts_7)]);

% now build a new time index
new_ts = [ceil(min_ts):0.1:floor(max_ts)];

% Interpolate Quat, gyro, linear and raw accelerations for the new_ts

%     data_ts = cal_raw_data(cal_rows_0,1);
%     data_newts = [data_ts(1):ts:data_ts(datalength)];

%     for ii=1:3
%         [x, index] = unique(data_ts(1:datalength));
%         Eul(:,ii) = interp1(x, eul(index,ii), data_newts);
%     end
[x_0,index_0] = unique(ts_0);
[x_1,index_1] = unique(ts_1);
[x_2,index_2] = unique(ts_2);
[x_3,index_3] = unique(ts_3);

[x_4,index_4] = unique(ts_4);
[x_5,index_5] = unique(ts_5);
[x_6,index_6] = unique(ts_6);
[x_7,index_7] = unique(ts_7);

Data0 = interp1(x_0,cal_raw_data(cal_rows_0(index_0),3:end),new_ts);
Data1 = interp1(x_1,cal_raw_data(cal_rows_1(index_1),3:end),new_ts);
Data2 = interp1(x_2,cal_raw_data(cal_rows_2(index_2),3:end),new_ts);
Data3 = interp1(x_3,cal_raw_data(cal_rows_3(index_3),3:end),new_ts);

Data4 = interp1(x_4,cal_raw_data(cal_rows_4(index_4),3:end),new_ts);
Data5 = interp1(x_5,cal_raw_data(cal_rows_5(index_5),3:end),new_ts);
Data6 = interp1(x_6,cal_raw_data(cal_rows_6(index_6),3:end),new_ts);
Data7 = interp1(x_7,cal_raw_data(cal_rows_7(index_7),3:end),new_ts);

% calculate Trunk Angles
sensor_0_quat = Data0(:,1:4); % T3-T4
sensor_1_quat = Data1(:,1:4); % L5-S1

eul_back = zeros(min(size(sensor_0_quat,1),size(sensor_1_quat,1)),3);
for ii = 1:min(size(sensor_0_quat,1),size(sensor_1_quat,1))
    inv_0 = quatinv(sensor_0_quat(ii,:));
    qD = quatmultiply(inv_0,sensor_1_quat(ii,:));
    rotm = quat2rotm(qD);
    eul_back(ii,:) = rotm2eul(rotm)*180/pi;
end

% calculate Right shoulder elevation
sensor_2_quat = Data2(:,1:4); % Right Upper Arm
eul_r_sh = zeros(min(size(sensor_0_quat,1),size(sensor_1_quat,1)),3);

for ii = 1:min(size(sensor_0_quat,1),size(sensor_2_quat,1))
    inv_1 = quatinv(sensor_2_quat(ii,:));
    qD = quatmultiply(inv_1,sensor_0_quat(ii,:));
    rotm = quat2rotm(qD);
    %     eul(ii,:) = rotm2eul(rotm)*180/pi;
    eul_r_sh(ii,:) = rotm2eul(rotm,'ZYZ')*180/pi;
```

```
end

% calculate Left shoulder elevation
sensor_4_quat = Data4(:,1:4); % Left Upper Arm
eul_l_sh = zeros(min(size(sensor_0_quat,1),size(sensor_1_quat,1)),3);
for ii = 1:min(size(sensor_0_quat,1),size(sensor_4_quat,1))
    inv_1 = quatinv(sensor_4_quat(ii,:));
    qD = quatmultiply(inv_1,sensor_0_quat(ii,:));
    rotm = quat2rotm(qD);
    %     eul(ii,:) = rotm2eul(rotm)*180/pi;
    eul_l_sh(ii,:) = rotm2eul(rotm,'ZYZ')*180/pi;
end

% calculate Right elbow angle
sensor_3_quat = Data3(:,1:4); % Right Lower Arm
eul_r_el = zeros(min(size(sensor_0_quat,1),size(sensor_1_quat,1)),3);

for ii = 1:min(size(sensor_2_quat,1),size(sensor_3_quat,1))
    inv_1 = quatinv(sensor_3_quat(ii,:));
    qD = quatmultiply(inv_1,sensor_2_quat(ii,:));
    rotm = quat2rotm(qD);
    %     eul(ii,:) = rotm2eul(rotm)*180/pi;
    eul_r_el(ii,:) = rotm2eul(rotm,'ZYZ')*180/pi;
end

% calculate Left Elbow angle
sensor_5_quat = Data5(:,1:4); % Left Lower Arm
eul_l_el = zeros(min(size(sensor_0_quat,1),size(sensor_1_quat,1)),3);

for ii = 1:min(size(sensor_4_quat,1),size(sensor_5_quat,1))
    inv_1 = quatinv(sensor_5_quat(ii,:));
    qD = quatmultiply(inv_1,sensor_4_quat(ii,:));
    rotm = quat2rotm(qD);
    %     eul(ii,:) = rotm2eul(rotm)*180/pi;
    eul_l_el(ii,:) = rotm2eul(rotm,'ZYZ')*180/pi;
end

% calculate Right knee incline angle
sensor_6_quat = Data6(:,1:4); % Right Lower Arm
eul_r_knee = zeros(min(size(sensor_0_quat,1),size(sensor_1_quat,1)),3);

for ii = 1:min(size(sensor_1_quat,1),size(sensor_6_quat,1))
    inv_1 = quatinv(sensor_6_quat(ii,:));
    qD = quatmultiply(inv_1,sensor_1_quat(ii,:));
    rotm = quat2rotm(qD);
    %     eul(ii,:) = rotm2eul(rotm)*180/pi;
    eul_r_knee(ii,:) = rotm2eul(rotm,'ZYZ')*180/pi;
end

% calculate Left knee inclination
sensor_7_quat = Data7(:,1:4); % Left Lower Arm
eul_l_knee = zeros(min(size(sensor_0_quat,1),size(sensor_1_quat,1)),3);

for ii = 1:min(size(sensor_1_quat,1),size(sensor_7_quat,1))
    inv_1 = quatinv(sensor_7_quat(ii,:));
    qD = quatmultiply(inv_1,sensor_1_quat(ii,:));
    rotm = quat2rotm(qD);
    %     eul(ii,:) = rotm2eul(rotm)*180/pi;
    eul_l_knee(ii,:) = rotm2eul(rotm,'ZYZ')*180/pi;
```

```
end

% call script to plot the Euler Angles
% plot_fig;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Construct a matrix with features extracted in addition to raw data
% [Spine_Twist Spine_Bend Spine_Roll Right_shoulder Left_shoulder
% .........    Right_elbow Left_elbow Right_knee_pitch Left_knee_pitch
% .........    Right_knee_roll Left_knee_roll]
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
features_extract = [eul_back(:,1) eul_back(:,2) eul_back(:,3) eul_r_sh(:,2) ...
    eul_l_sh(:,2) eul_r_el(:,2) eul_l_el(:,2) eul_r_knee(:,2:3) ...
    eul_l_knee(:,2:3)];
%%%% Now add Quat, Linear Acc, Gyro, raw Acc and extracted features from all 8 sensors
features = [Data0(:,1:4) Data0(:,8:10) Data0(:,12:17) Data1(:,1:4) Data1(:,8:10) Data1(:,12:17) ...
    Data2(:,1:4) Data2(:,8:10) Data2(:,12:17) Data3(:,1:4) Data3(:,8:10) Data3(:,12:17) ...
    Data4(:,1:4) Data3(:,8:10) Data4(:,12:17) Data5(:,1:4) Data5(:,8:10) Data5(:,12:17) ...
    Data6(:,1:4) Data6(:,8:10) Data6(:,12:17) Data7(:,1:4) Data7(:,8:10) Data7(:,12:17) ...
    features_extract];

% Resize this to include time chunks of 1 sec
flatten = [];
for ii = 1:floor(size(features,1)/10)
    temp = features((ii-1)*10+1:ii*10,:);
    temp = reshape(temp',1,size(temp,1)*size(temp,2));
    flatten = [flatten;temp];
end

%%%% Now add label
% 0     Sit
% 1     Stand
% 2     Kneel
% 3     Static_Stoop
% 4     Crouch
% 5     Walk
% 6     Crawl
% 7     Reach_Close
% 8     Reach_Far
% 9     Reach_close
% 10        Overhead Static
% 11        Overhead Dynamic
% 12        Carrying
% 13        Pull
% 14        Pull_Onehanded_left
% 15        Pull_Onehanded_Right
% 16        Push
% 17        Lift_straight
% 18        Lift_floor_2_shoulder
% 19        Lift_Floor_2_Waist_squat
% 20        Lift_Floor_2_Waist_stoop
```

```
% 21          Lift OneHanded_left
% 22          Lift_OneHanded_Right
% 23          Lift_shoulder_to_waist_Twist
% 24          Lift_shoulder_to_waist_woTwist

if (strcmp(input_filename,'Carrying.csv'))
    label = 12;
elseif (strcmp(input_filename,'Crawling.csv'))
    label = 6;
elseif (strcmp(input_filename,'Crouching.csv'))
    label = 4;
elseif (strcmp(input_filename,'Kneeling.csv'))
    label = 2;
elseif (strcmp(input_filename,'L_Floor_60in_1.csv'))
    label = 17;
 elseif (strcmp(input_filename,'L_Floor_60in_2.csv'))
     label = 17;
elseif (strcmp(input_filename,'L_Floor_Shoulder_1.csv'))
    label = 18;
elseif (strcmp(input_filename,'L_Floor_Shoulder_2.csv'))
    label = 18;
elseif (strcmp(input_filename,'L_Floor_Waist_Squat_1.csv'))
    label = 19;
elseif (strcmp(input_filename,'L_Floor_Waist_Squat_2.csv'))
    label = 19;
elseif (strcmp(input_filename,'L_Floor_Waist_Stoop_1.csv'))
    label = 20;
elseif (strcmp(input_filename,'L_Floor_Waist_Stoop_2.csv'))
    label = 20;
elseif (strcmp(input_filename,'L_OneH_L_1.csv'))
    label = 21;
elseif (strcmp(input_filename,'L_OneH_L_2.csv'))
    label = 21;
elseif (strcmp(input_filename,'L_OneH_R_1.csv'))
    label = 22;
elseif (strcmp(input_filename,'L_OneH_R_2.csv'))
    label = 22;
elseif (strcmp(input_filename,'Pushing_Clockwise.csv'))
    label = 16;
elseif (strcmp(input_filename,'Pushing_Counterclockwise.csv'))
    label = 16;
elseif (strcmp(input_filename,'Pulling_Clockwise.csv'))
    label = 13;
elseif (strcmp(input_filename,'Pulling_OneH_L.csv'))
    label = 14;
elseif (strcmp(input_filename,'Pulling_OneH_R.csv'))
    label = 15;
elseif (strcmp(input_filename,'Reaching_Close.csv'))
    label = 7;
elseif (strcmp(input_filename,'Reaching_Far.csv'))
    label = 8;
elseif (strcmp(input_filename,'Reaching_High.csv'))
    label = 9;
elseif (strcmp(input_filename,'Sitting.csv'))
    label = 0;
elseif (strcmp(input_filename,'Standing.csv'))
    label = 1;
elseif (strcmp(input_filename,'Static_Stoop.csv'))
    label = 3;
```

```
    elseif (strcmp(input_filename,'Walking.csv'))
        label = 5;
    elseif (strcmp(input_filename,'L_Shoulder_Waist_Twist_1.csv'))
        label = 23;
    elseif (strcmp(input_filename,'L_Shoulder_Waist_Twist_2.csv'))
        label = 23;
    elseif (strcmp(input_filename,'L_Shoulder_Waist_wo_1.csv'))
        label = 24;
    elseif (strcmp(input_filename,'L_Shoulder_Waist_wo_2.csv'))
        label = 24;
    elseif (strcmp(input_filename,'Overhead_Static.csv'))
        label = 10;
    elseif (strcmp(input_filename,'Overhead_Dynamic.csv'))
        label = 11;
    else
        % do nothing here
    end

    % Update features with label as Last Column
    flatten = [flatten ones(1,size(flatten,1))'*label];
    % Update the Super features matrix that includes features for all
    % activities per Subject
    super_features =[super_features;flatten];

    % Now clear all the variables before reading next file
    clear flatten features features_extract cal_raw_data Data0 Data7 eul_back eul_r_sh eul_l_sh eul_r_el eul_l_el eul_r_knee eul_l_knee
    clear sensor_0_quat sensor_1_quat sensor_2_quat sensor_3_quat sensor_4_quat sensor_5_quat sensor_6_quat sensor_7_quat
end % end of loop for 1 subject data

dlmwrite('super_features.csv',super_features,'-append')
```

Appendix 2 Full code for feature extraction of the training dataset (data_preprocessv3.m)

```

%close all
clear

% Input Stand Data. Read the Back, Shoulder, Elbow, Knee angles
% input_filename = 'C:\ergo_related\SpineTrack\DeepLearning\Charlotte_data\Walking.csv';
% [filepath,name,ext] = fileparts(input_filename);

cd 'E:\Programme\GoogleDrive\SpineTrack_activityData\Non Pilot\Gerald_12-18-18\Gerald_task'
files = dir('*.csv');
super_features = []; % matrix that will hold all training data set along with labels

for ii = 1:length(files)
    input_filename = files(ii).name;
    X = ['Now processing...', input_filename]; disp(X)
    cal_raw_data = csvread(input_filename,1);

    [cal_rows_1]=find(cal_raw_data(:,2)==1); % L5-S1 rows
    [cal_rows_0]=find(cal_raw_data(:,2)==0); % T3-T4 rows

    [cal_rows_2]=find(cal_raw_data(:,2)==2); % L5-S1 rows
    [cal_rows_3]=find(cal_raw_data(:,2)==3); % T3-T4 rows
```

```

[cal_rows_4]=find(cal_raw_data(:,2)==4); % L5-S1 rows
[cal_rows_5]=find(cal_raw_data(:,2)==4); % T3-T4 rows

[cal_rows_6]=find(cal_raw_data(:,2)==6); % L5-S1 rows
[cal_rows_7]=find(cal_raw_data(:,2)==7); % T3-T4 rows

ts_1 = cal_raw_data(cal_rows_1,1);
ts_0 = cal_raw_data(cal_rows_0,1);

ts_2 = cal_raw_data(cal_rows_2,1);
ts_3 = cal_raw_data(cal_rows_3,1);

ts_4 = cal_raw_data(cal_rows_4,1);
ts_5 = cal_raw_data(cal_rows_5,1);

ts_6 = cal_raw_data(cal_rows_6,1);
ts_7 = cal_raw_data(cal_rows_7,1);

% Get max and min of all timestamps across all sensors
max_ts = max([max(ts_0) max(ts_1) max(ts_2) max(ts_3) max(ts_4) max(ts_5) max(ts_6) max(ts_7)]);
min_ts = min([min(ts_0) min(ts_1) min(ts_2) min(ts_3) min(ts_4) min(ts_5) min(ts_6) min(ts_7)]);

% now build a new time index
new_ts = [ceil(min_ts):0.1:floor(max_ts)];

% Interpolate Quat, gyro, linear and raw accelerations for the new_ts

%     data_ts = cal_raw_data(cal_rows_0,1);
%     data_newts = [data_ts(1):ts:data_ts(datalength)];

%     for ii=1:3
%         [x, index] = unique(data_ts(1:datalength));
%         Eul(:,ii) = interp1(x, eul(index,ii), data_newts);
%     end
[x_0,index_0] = unique(ts_0);
[x_1,index_1] = unique(ts_1);
[x_2,index_2] = unique(ts_2);

[x_3,index_3] = unique(ts_3);
[x_4,index_4] = unique(ts_4);
[x_5,index_5] = unique(ts_5);

[x_6,index_6] = unique(ts_6);
[x_7,index_7] = unique(ts_7);


Data0 = interp1(x_0,cal_raw_data(cal_rows_0(index_0),3:end),new_ts);
Data1 = interp1(x_1,cal_raw_data(cal_rows_1(index_1),3:end),new_ts);
Data2 = interp1(x_2,cal_raw_data(cal_rows_2(index_2),3:end),new_ts);
Data3 = interp1(x_3,cal_raw_data(cal_rows_3(index_3),3:end),new_ts);

Data4 = interp1(x_4,cal_raw_data(cal_rows_4(index_4),3:end),new_ts);
Data5 = interp1(x_5,cal_raw_data(cal_rows_5(index_5),3:end),new_ts);
Data6 = interp1(x_6,cal_raw_data(cal_rows_6(index_6),3:end),new_ts);
Data7 = interp1(x_7,cal_raw_data(cal_rows_7(index_7),3:end),new_ts);

```

```
% calculate Trunk Angles
sensor_0_quat = Data0(:,1:4); % T3-T4
sensor_1_quat = Data1(:,1:4); % L5-S1

eul_back = zeros(min(size(sensor_0_quat,1),size(sensor_1_quat,1)),3);
for ii = 1:min(size(sensor_0_quat,1),size(sensor_1_quat,1))
    inv_0 = quatinv(sensor_0_quat(ii,:));
    qD = quatmultiply(inv_0,sensor_1_quat(ii,:));
    rotm = quat2rotm(qD);
    eul_back(ii,:) = rotm2eul(rotm)*180/pi;
end

% calculate Right shoulder elevation
sensor_2_quat = Data2(:,1:4); % Right Upper Arm
eul_r_sh = zeros(min(size(sensor_0_quat,1),size(sensor_1_quat,1)),3);

for ii = 1:min(size(sensor_0_quat,1),size(sensor_2_quat,1))
    inv_1 = quatinv(sensor_2_quat(ii,:));
    qD = quatmultiply(inv_1,sensor_0_quat(ii,:));
    rotm = quat2rotm(qD);
     %     eul(ii,:) = rotm2eul(rotm)*180/pi;
     eul_r_sh(ii,:) = rotm2eul(rotm,'ZYZ')*180/pi;
 end

 % calculate Left shoulder elevation
 sensor_4_quat = Data4(:,1:4); % Left Upper Arm
 eul_l_sh = zeros(min(size(sensor_0_quat,1),size(sensor_1_quat,1)),3);
 for ii = 1:min(size(sensor_0_quat,1),size(sensor_4_quat,1))
     inv_1 = quatinv(sensor_4_quat(ii,:));
     qD = quatmultiply(inv_1,sensor_0_quat(ii,:));
     rotm = quat2rotm(qD);
     %     eul(ii,:) = rotm2eul(rotm)*180/pi;
     eul_l_sh(ii,:) = rotm2eul(rotm,'ZYZ')*180/pi;
 end

 % calculate Right elbow angle
 sensor_3_quat = Data3(:,1:4); % Right Lower Arm
 eul_r_el = zeros(min(size(sensor_0_quat,1),size(sensor_1_quat,1)),3);

 for ii = 1:min(size(sensor_2_quat,1),size(sensor_3_quat,1))
     inv_1 = quatinv(sensor_3_quat(ii,:));
     qD = quatmultiply(inv_1,sensor_2_quat(ii,:));
     rotm = quat2rotm(qD);
     %     eul(ii,:) = rotm2eul(rotm)*180/pi;
     eul_r_el(ii,:) = rotm2eul(rotm,'ZYZ')*180/pi;
 end

 % calculate Left Elbow angle
 sensor_5_quat = Data5(:,1:4); % Left Lower Arm
 eul_l_el = zeros(min(size(sensor_0_quat,1),size(sensor_1_quat,1)),3);

 for ii = 1:min(size(sensor_4_quat,1),size(sensor_5_quat,1))
     inv_1 = quatinv(sensor_5_quat(ii,:));
     qD = quatmultiply(inv_1,sensor_4_quat(ii,:));
     rotm = quat2rotm(qD);
     %     eul(ii,:) = rotm2eul(rotm)*180/pi;
     eul_l_el(ii,:) = rotm2eul(rotm,'ZYZ')*180/pi;
 end
```

```
% calculate Right knee incline angle
sensor_6_quat = Data6(:,1:4); % Right Lower Arm
eul_r_knee = zeros(min(size(sensor_0_quat,1),size(sensor_1_quat,1)),3);

for ii = 1:min(size(sensor_1_quat,1),size(sensor_6_quat,1))
    inv_1 = quatinv(sensor_6_quat(ii,:));
    qD = quatmultiply(inv_1,sensor_1_quat(ii,:));
    rotm = quat2rotm(qD);
    %     eul(ii,:) = rotm2eul(rotm)*180/pi;
    eul_r_knee(ii,:) = rotm2eul(rotm,'ZYZ')*180/pi;
end

% calculate Left knee inclination
sensor_7_quat = Data7(:,1:4); % Left Lower Arm
eul_l_knee = zeros(min(size(sensor_0_quat,1),size(sensor_1_quat,1)),3);

for ii = 1:min(size(sensor_1_quat,1),size(sensor_7_quat,1))
    inv_1 = quatinv(sensor_7_quat(ii,:));
    qD = quatmultiply(inv_1,sensor_1_quat(ii,:));
    rotm = quat2rotm(qD);
    %     eul(ii,:) = rotm2eul(rotm)*180/pi;
    eul_l_knee(ii,:) = rotm2eul(rotm,'ZYZ')*180/pi;
end

% call script to plot the Euler Angles
% plot_fig;


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Construct a matrix with features extracted in addition to raw data
% [Spine_Twist Spine_Bend Spine_Roll Right_shoulder Left_shoulder
% .........    Right_elbow Left_elbow Right_knee_pitch Left_knee_pitch
% .........    Right_knee_roll Left_knee_roll]

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
features_extract = [eul_back(:,1) eul_back(:,2) eul_back(:,3) eul_r_sh(:,2) ...
    eul_l_sh(:,2) eul_r_el(:,2) eul_l_el(:,2) eul_r_knee(:,2:3) ...
    eul_l_knee(:,2:3)];
%%%% Now add Quat, Linear Acc, Gyro, raw Acc and extracted features from all 8 sensors
features = [Data0(:,1:4) Data0(:,8:10) Data0(:,12:17) Data1(:,1:4) Data1(:,8:10) Data1(:,12:17) ...
    Data2(:,1:4) Data2(:,8:10) Data2(:,12:17) Data3(:,1:4) Data3(:,8:10) Data3(:,12:17) ...
    Data4(:,1:4) Data3(:,8:10) Data4(:,12:17) Data5(:,1:4) Data5(:,8:10) Data5(:,12:17) ...
    Data6(:,1:4) Data6(:,8:10) Data6(:,12:17) Data7(:,1:4) Data7(:,8:10) Data7(:,12:17) ...
    features_extract];

% Resize this to include time chunks of 1 sec
flatten = [];
for ii = 1:floor(size(features,1)/10)
    temp = features((ii-1)*10+1:ii*10,:);
    temp = reshape(temp',1,size(temp,1)*size(temp,2));
    flatten = [flatten;temp];
end
```

```
%%%% Now add label
% 0		Sit
% 1	Stand
% 2	Kneel
% 3	Static_Stoop
% 4	Crouch
% 5	Walk
% 6	Crawl
% 7	Reach_Close
% 8	Reach_Far
% 9	Reach_close
% 10		Overhead Static
% 11		Overhead Dynamic
% 12		Carry
% 13		Pulling
% 14		Pulling_Onehanded_left
% 15		Pulling_Onehanded_Right
% 16		Pushing
% 17		Lift_straight
% 18		Lift_floor_2_shoulder
% 19		Lift_Floor_2_Waist_squat
% 20		Lift_Floor_2_Waist_stoop
% 21		Lift OneHanded_left
% 22		Lift_OneHanded_Right
% 23		Lift_shoulder_to_waist_Twist
% 24		Lift_shoulder_to_waist_woTwist

%     if (strcmp(input_filename,'Carrying_5.csv'))
%         label = 12;
%     elseif (strcmp(input_filename,'Carrying_10.csv'))
%         label = 12;
%     elseif (strcmp(input_filename,'Carrying_20.csv'))
%         label = 12;
%     elseif (strcmp(input_filename,'Crawling.csv'))
%         label = 6;
%     elseif (strcmp(input_filename,'Crouching.csv'))
%         label = 4;
%     elseif (strcmp(input_filename,'Kneeling.csv'))
%         label = 2;
%     elseif (strcmp(input_filename,'L_Floor_60in_5.csv'))
%         label = 17;
%     elseif (strcmp(input_filename,'L_Floor_60in_10.csv'))
%         label = 17;
%     elseif (strcmp(input_filename,'L_Floor_60in_20.csv'))
%         label = 17;
%     elseif (strcmp(input_filename,'L_Floor_Shoulder_5.csv'))
%         label = 18;
%     elseif (strcmp(input_filename,'L_Floor_Shoulder_10.csv'))
%         label = 18;
%     elseif (strcmp(input_filename,'L_Floor_Shoulder_20.csv'))
%         label = 18;
%     elseif (strcmp(input_filename,'L_Floor_Waist_Squat_5.csv'))
%         label = 19;
%     elseif (strcmp(input_filename,'L_Floor_Waist_Squat_10.csv'))
%         label = 19;
%     elseif (strcmp(input_filename,'L_Floor_Waist_Squat_20.csv'))
%         label = 19;
%     elseif (strcmp(input_filename,'L_Floor_Waist_Stoop_5.csv'))
```

```
%         label = 20;
%     elseif (strcmp(input_filename,'L_Floor_Waist_Stoop_10.csv'))
%         label = 20;
%     elseif (strcmp(input_filename,'L_Floor_Waist_Stoop_20.csv'))
%         label = 20;
%     elseif (strcmp(input_filename,'L_OneH_L_5.csv'))
%         label = 21;
%     elseif (strcmp(input_filename,'L_OneH_L_10.csv'))
%         label = 21;
%     elseif (strcmp(input_filename,'L_OneH_L_15.csv'))
%         label = 21;
%     elseif (strcmp(input_filename,'L_OneH_R_5.csv'))
%         label = 22;
%     elseif (strcmp(input_filename,'L_OneH_R_10.csv'))
%         label = 22;
%     elseif (strcmp(input_filename,'L_OneH_R_15.csv'))
%         label = 22;
%     elseif (strcmp(input_filename,'Push_Clockwise.csv'))
%         label = 16;
%     elseif (strcmp(input_filename,'Push_Counterclockwise.csv'))
%         label = 16;
%     elseif (strcmp(input_filename,'Pull_Clockwise.csv'))
%         label = 13;
%     elseif (strcmp(input_filename,'Pulling_OneH_L.csv'))
%         label = 14;
%     elseif (strcmp(input_filename,'Pulling_OneH_R.csv'))
%         label = 15;
%     elseif (strcmp(input_filename,'Reaching_Close.csv'))
%         label = 7;
%     elseif (strcmp(input_filename,'Reaching_Far.csv'))
%         label = 8;
%     elseif (strcmp(input_filename,'Reaching_High.csv'))
%         label = 9;
%     elseif (strcmp(input_filename,'Sitting.csv'))
%         label = 0;
%     elseif (strcmp(input_filename,'Standing.csv'))
%         label = 1;
%     elseif (strcmp(input_filename,'Static_Stoop.csv'))
%         label = 3;
%     elseif (strcmp(input_filename,'Walking.csv'))
%         label = 5;
%     elseif (strcmp(input_filename,'L_Shoulder_Waist_Twist_5.csv'))
%         label = 23;
%     elseif (strcmp(input_filename,'L_Shoulder_Waist_Twist_10.csv'))
%         label = 23;
%     elseif (strcmp(input_filename,'L_Shoulder_Waist_Twist_20.csv'))
%         label = 23;
%     elseif (strcmp(input_filename,'L_Shoulder_Waist_wo_5.csv'))
%         label = 24;
%     elseif (strcmp(input_filename,'L_Shoulder_Waist_wo_10.csv'))
%         label = 24;
%     elseif (strcmp(input_filename,'L_Shoulder_Waist_wo_20.csv'))
%         label = 24;
%     elseif (strcmp(input_filename,'Overhead_Static.csv'))
%         label = 10;
%     elseif (strcmp(input_filename,'Overhead_Dynamic.csv'))
%         label = 11;
%     else
%         % do nothing here
```

```
%     end
%
    % Update features with label as Last Column
%     flatten = [flatten ones(1,size(flatten,1))'*label];
    % Update the Super features matrix that includes features for all
    % activities per Subject
%     super_features =[super_features;flatten];
    [filepath,name,ext] = fileparts(input_filename);
    output_filename = strcat(name,'_features.csv');
    fid = fopen(output_filename, 'wt');
    dlmwrite(output_filename,flatten,'-append');
    fclose(fid);

    clear output_filename name filepath exts
    % fid=fopen(op_fname_rightshoulder,'wt');
    % dlmwrite(op_fname_rightshoulder,[data_newts3' abs(Data3)'],'-append');

    % Now clear all the variables before reading next file
    clear flatten features features_extract cal_raw_data Data0 Data7 eul_back eul_r_sh eul_l_sh eul_r_el eul_l_el eul_r_knee eul_l_knee
    clear sensor_0_quat sensor_1_quat sensor_2_quat sensor_3_quat sensor_4_quat sensor_5_quat sensor_6_quat sensor_7_quat
end % end of loop for 1 subject data
```

Appendix 3 Full code for feature extraction of the test dataset (data_preprocessTestdata.m)

```

% this code processes Lifting and other task files to remove the pauses in between
% tasks
addpath('C:\Users\Elias\Dropbox (Medizintechnik)\Berkeley\WORK\Work_Spinetrack')
addpath('E:\Programme\GoogleDrive\SpineTrack_activityData\Non Pilot\Elias_12-7-18')
cd 'E:\Programme\GoogleDrive\SpineTrack_activityData\Non Pilot\Elias_12-7-18\Elias_data'
files = dir('*.csv');
super_features = []; % matrix that will hold all training data set along with labels
figure
pause(0.00001);
frame_h = get(handle(gcf),'JavaFrame');
set(frame_h,'Maximized',1);

for ii = 1:length(files)
   input_filename = files(ii).name;
   X = ['Now processing...', input_filename]; disp(X)
   cal_raw_data = csvread(input_filename,2);

   if (strcmp(input_filename,'Carrying.csv'))
      plot(cal_raw_data(:,14:19)); title(input_filename, 'Interpreter', 'none')
      [x,y]=ginput(2);
      data_need = cal_raw_data(floor(x(1)):floor(x(2)),1:19);
      data_need(:,1) = data_need(:,1)-data_need(1,1);
      csvwrite('Processed\Carrying.csv',data_need);
      clear cal_raw_data x y data_need
   elseif (strcmp(input_filename,'Crawling.csv'))

      plot(cal_raw_data(:,14:19));title(input_filename, 'Interpreter', 'none')
      [x,y]=ginput(2);
```

```
        data_need = cal_raw_data(floor(x(1)):floor(x(2)),1:19);
        data_need(:,1) = data_need(:,1)-data_need(1,1);
        csvwrite('Processed\Crawling.csv',data_need);
        clear cal_raw_data x y data_need

    elseif (strcmp(input_filename,'Crouching.csv'))

        plot(cal_raw_data(:,14:19));title(input_filename, 'Interpreter', 'none')
        [x,y]=ginput(2);
        data_need = cal_raw_data(floor(x(1)):floor(x(2)),1:19);
        data_need(:,1) = data_need(:,1)-data_need(1,1);
        csvwrite('Processed\Crouching.csv',data_need);
        clear cal_raw_data x y data_need

    elseif (strcmp(input_filename,'Kneeling.csv'))
        plot(cal_raw_data(:,14:19));title(input_filename, 'Interpreter', 'none')
        [x,y]=ginput(2);
        data_need = cal_raw_data(floor(x(1)):floor(x(2)),1:19);
        data_need(:,1) = data_need(:,1)-data_need(1,1);
        csvwrite('Processed\Kneeling.csv',data_need);
        clear cal_raw_data x y data_need

    elseif (strcmp(input_filename,'L_Floor_60in_1.csv'))

        plot(cal_raw_data(:,14:19));title(input_filename, 'Interpreter', 'none')
        [x,y]=ginput(60);
        data_need = append_lift_data(cal_raw_data,x);
        csvwrite('Processed\L_Floor_60in_1.csv',data_need);
        clear cal_raw_data x y data_need

    elseif (strcmp(input_filename,'L_Floor_60in_2.csv'))

        plot(cal_raw_data(:,14:19));title(input_filename, 'Interpreter', 'none')
        [x,y]=ginput(60);
        data_need = append_lift_data(cal_raw_data,x);
        csvwrite('Processed\L_Floor_60in_2.csv',data_need);
        clear cal_raw_data x y data_need

    elseif (strcmp(input_filename,'L_Floor_Shoulder_1.csv'))

        plot(cal_raw_data(:,14:19));title(input_filename, 'Interpreter', 'none')
        [x,y]=ginput(60);
        data_need = append_lift_data(cal_raw_data,x);
        csvwrite('Processed\L_Floor_Shoulder_1.csv',data_need);
        clear cal_raw_data x y data_need

    elseif (strcmp(input_filename,'L_Floor_Shoulder_2.csv'))

        plot(cal_raw_data(:,14:19));title(input_filename, 'Interpreter', 'none')
        [x,y]=ginput(60);
        data_need = append_lift_data(cal_raw_data,x);
        csvwrite('Processed\L_Floor_Shoulder_2.csv',data_need);
        clear cal_raw_data x y data_need

    elseif (strcmp(input_filename,'L_Floor_Waist_Squat_1.csv'))

        plot(cal_raw_data(:,14:19));title(input_filename, 'Interpreter', 'none')
        [x,y]=ginput(60);
        data_need = append_lift_data(cal_raw_data,x);
```

```
        csvwrite('Processed\L_Floor_Waist_Squat_1.csv',data_need);
        clear cal_raw_data x y data_need

    elseif (strcmp(input_filename,'L_Floor_Waist_Squat_2.csv'))

        plot(cal_raw_data(:,14:19));title(input_filename, 'Interpreter', 'none')
        [x,y]=ginput(60);
        data_need = append_lift_data(cal_raw_data,x);
        csvwrite('Processed\L_Floor_Waist_Squat_2.csv',data_need);
        clear cal_raw_data x y data_need

    elseif (strcmp(input_filename,'L_Floor_Waist_Stoop_1.csv'))

        plot(cal_raw_data(:,14:19));title(input_filename, 'Interpreter', 'none')
        [x,y]=ginput(60);
        data_need = append_lift_data(cal_raw_data,x);
        csvwrite('Processed\L_Floor_Waist_Stoop_1.csv',data_need);
        clear cal_raw_data x y data_need

    elseif (strcmp(input_filename,'L_Floor_Waist_Stoop_2.csv'))

        plot(cal_raw_data(:,14:19));title(input_filename, 'Interpreter', 'none')
        [x,y]=ginput(60);
        data_need = append_lift_data(cal_raw_data,x);
        csvwrite('Processed\L_Floor_Waist_Stoop_2.csv',data_need);
        clear cal_raw_data x y data_need

    elseif (strcmp(input_filename,'L_OneH_L_1.csv'))

        plot(cal_raw_data(:,14:19));title(input_filename, 'Interpreter', 'none')
        [x,y]=ginput(60);
        data_need = append_lift_data(cal_raw_data,x);
        csvwrite('Processed\L_OneH_L_1.csv',data_need);
        clear cal_raw_data x y data_need

    elseif (strcmp(input_filename,'L_OneH_L_2.csv'))

        plot(cal_raw_data(:,14:19));title(input_filename, 'Interpreter', 'none')
        [x,y]=ginput(60);
        data_need = append_lift_data(cal_raw_data,x);
        csvwrite('Processed\L_OneH_L_2.csv',data_need);
        clear cal_raw_data x y data_need

    elseif (strcmp(input_filename,'L_OneH_R_1.csv'))
        plot(cal_raw_data(:,14:19));title(input_filename, 'Interpreter', 'none')
        [x,y]=ginput(60);
        data_need = append_lift_data(cal_raw_data,x);
        csvwrite('Processed\L_OneH_R_1.csv',data_need);
        clear cal_raw_data x y data_need

    elseif (strcmp(input_filename,'L_OneH_R_2.csv'))
        plot(cal_raw_data(:,14:19));title(input_filename, 'Interpreter', 'none')
        [x,y]=ginput(60);
        data_need = append_lift_data(cal_raw_data,x);
        csvwrite('Processed\L_OneH_R_2.csv',data_need);
        clear cal_raw_data x y data_need

    elseif (strcmp(input_filename,'Pushing_Clockwise.csv'))
        plot(cal_raw_data(:,14:19));title(input_filename, 'Interpreter', 'none')
```

```
    [x,y]=ginput(2);
    data_need = cal_raw_data(floor(x(1)):floor(x(2)),1:19);
    data_need(:,1) = data_need(:,1)-data_need(1,1);
    csvwrite('Processed\Pushing_Clockwise.csv',data_need);
    clear cal_raw_data x y data_need

elseif (strcmp(input_filename,'Pushing_Counterclockwise.csv'))
    plot(cal_raw_data(:,14:19));title(input_filename, 'Interpreter', 'none')
    [x,y]=ginput(2);
    data_need = cal_raw_data(floor(x(1)):floor(x(2)),1:19);
    data_need(:,1) = data_need(:,1)-data_need(1,1);
    csvwrite('Processed\Pushing_Counterclockwise.csv',data_need);
    clear cal_raw_data x y data_need

elseif (strcmp(input_filename,'Pulling_Clockwise.csv'))
    plot(cal_raw_data(:,14:19));title(input_filename, 'Interpreter', 'none')
    [x,y]=ginput(2);
    data_need = cal_raw_data(floor(x(1)):floor(x(2)),1:19);
    data_need(:,1) = data_need(:,1)-data_need(1,1);
    csvwrite('Processed\Pulling_Clockwise.csv',data_need);
    clear cal_raw_data x y data_need

elseif (strcmp(input_filename,'Pulling_OneH_L.csv'))

    plot(cal_raw_data(:,14:19));title(input_filename, 'Interpreter', 'none')
    [x,y]=ginput(2);
    data_need = cal_raw_data(floor(x(1)):floor(x(2)),1:19);
    data_need(:,1) = data_need(:,1)-data_need(1,1);
    csvwrite('Processed\Pulling_OneH_L.csv',data_need);
    clear cal_raw_data x y data_need

elseif (strcmp(input_filename,'Pulling_OneH_R.csv'))

    plot(cal_raw_data(:,14:19));title(input_filename, 'Interpreter', 'none')
    [x,y]=ginput(2);
    data_need = cal_raw_data(floor(x(1)):floor(x(2)),1:19);
    data_need(:,1) = data_need(:,1)-data_need(1,1);
    csvwrite('Processed\Pulling_OneH_R.csv',data_need);
    clear cal_raw_data x y data_need

elseif (strcmp(input_filename,'Reaching_Close.csv'))
    plot(cal_raw_data(:,14:19));title(input_filename, 'Interpreter', 'none')
    [x,y]=ginput(2);
    data_need = cal_raw_data(floor(x(1)):floor(x(2)),1:19);
    data_need(:,1) = data_need(:,1)-data_need(1,1);
    csvwrite('Processed\Reaching_Close.csv',data_need);
    clear cal_raw_data x y data_need

elseif (strcmp(input_filename,'Reaching_Far.csv'))

    plot(cal_raw_data(:,14:19));title(input_filename, 'Interpreter', 'none')
    [x,y]=ginput(2);
    data_need = cal_raw_data(floor(x(1)):floor(x(2)),1:19);
    data_need(:,1) = data_need(:,1)-data_need(1,1);
    csvwrite('Processed\Reaching_Far.csv',data_need);
    clear cal_raw_data x y data_need

elseif (strcmp(input_filename,'Reaching_High.csv'))

```

```
    plot(cal_raw_data(:,14:19));title(input_filename, 'Interpreter', 'none')
    [x,y]=ginput(2);
    data_need = cal_raw_data(floor(x(1)):floor(x(2)),1:19);
    data_need(:,1) = data_need(:,1)-data_need(1,1);
    csvwrite('Processed\Reaching_High.csv',data_need);
    clear cal_raw_data x y data_need

elseif (strcmp(input_filename,'Sitting.csv'))

    plot(cal_raw_data(:,14:19));title(input_filename, 'Interpreter', 'none')
    [x,y]=ginput(2);
    data_need = cal_raw_data(floor(x(1)):floor(x(2)),1:19);
    data_need(:,1) = data_need(:,1)-data_need(1,1);
    csvwrite('Processed\Sitting.csv',data_need);
    clear cal_raw_data x y data_need

elseif (strcmp(input_filename,'Standing.csv'))

    plot(cal_raw_data(:,14:19));title(input_filename, 'Interpreter', 'none')
    [x,y]=ginput(2);
    data_need = cal_raw_data(floor(x(1)):floor(x(2)),1:19);
    data_need(:,1) = data_need(:,1)-data_need(1,1);
    csvwrite('Processed\Standing.csv',data_need);
    clear cal_raw_data x y data_need

elseif (strcmp(input_filename,'Static_Stoop.csv'))

    plot(cal_raw_data(:,14:19));title(input_filename, 'Interpreter', 'none')
    [x,y]=ginput(2);
    data_need = cal_raw_data(floor(x(1)):floor(x(2)),1:19);
    data_need(:,1) = data_need(:,1)-data_need(1,1);
    csvwrite('Processed\Static_Stoop.csv',data_need);
    clear cal_raw_data x y data_need

elseif (strcmp(input_filename,'Walking.csv'))

    plot(cal_raw_data(:,14:19));title(input_filename, 'Interpreter', 'none')
    [x,y]=ginput(2);
    data_need = cal_raw_data(floor(x(1)):floor(x(2)),1:19);
    data_need(:,1) = data_need(:,1)-data_need(1,1);
    csvwrite('Processed\Walking.csv',data_need);
    clear cal_raw_data x y data_need

elseif (strcmp(input_filename,'L_Shoulder_Waist_Twist_1.csv'))

    plot(cal_raw_data(:,14:19));title(input_filename, 'Interpreter', 'none')
    [x,y]=ginput(60);
    data_need = append_lift_data(cal_raw_data,x);
    csvwrite('Processed\L_Shoulder_Waist_Twist_1.csv',data_need);
    clear cal_raw_data x y data_need

elseif (strcmp(input_filename,'L_Shoulder_Waist_Twist_2.csv'))

    plot(cal_raw_data(:,14:19));title(input_filename, 'Interpreter', 'none')
    [x,y]=ginput(60);
    data_need = append_lift_data(cal_raw_data,x);
    csvwrite('Processed\L_Shoulder_Waist_Twist_2.csv',data_need);
    clear cal_raw_data x y data_need

```

```
    elseif (strcmp(input_filename,'L_Shoulder_Waist_wo_1.csv'))

        plot(cal_raw_data(:,14:19));title(input_filename, 'Interpreter', 'none')
        [x,y]=ginput(60);
        data_need = append_lift_data(cal_raw_data,x);
        csvwrite('Processed\L_Shoulder_Waist_wo_1.csv',data_need);
        clear cal_raw_data x y data_need

    elseif (strcmp(input_filename,'L_Shoulder_Waist_wo_2.csv'))

        plot(cal_raw_data(:,14:19));title(input_filename, 'Interpreter', 'none')
        [x,y]=ginput(60);
        data_need = append_lift_data(cal_raw_data,x);
        csvwrite('Processed\L_Shoulder_Waist_wo_2.csv',data_need);
        clear cal_raw_data x y data_need

    elseif (strcmp(input_filename,'Overhead_Static.csv'))
        plot(cal_raw_data(:,14:19));title(input_filename, 'Interpreter', 'none')
        [x,y]=ginput(2);
        data_need = cal_raw_data(floor(x(1)):floor(x(2)),1:19);
        data_need(:,1) = data_need(:,1)-data_need(1,1);
        csvwrite('Processed\Overhead_Static.csv',data_need);
        clear cal_raw_data x y data_need

    elseif (strcmp(input_filename,'Overhead_Dynamic.csv'))

        plot(cal_raw_data(:,14:19));title(input_filename, 'Interpreter', 'none')
        [x,y]=ginput(2);
        data_need = cal_raw_data(floor(x(1)):floor(x(2)),1:19);
        data_need(:,1) = data_need(:,1)-data_need(1,1);
        csvwrite('Processed\Overhead_Dynamic.csv',data_need);
        clear cal_raw_data x y data_need

    else
        % do nothing here
    end

end
```

Appendix 4 Full code for data_process_liftingv3.m, final code for preprocessing (cutting) for the whole training dataset

```
clear all;
%instrfindall

data = csvread('F:\CLOUD\GoogleDrive\SpineTrack_activityData\Non Pilot\Elias_12-7-18\Elias_data\Kneeling.csv');
rows_0 = find(data(:,2)==0);
rows_1 = find(data(:,2)==1);
rows_2 = find(data(:,2)==2);
rows_3 = find(data(:,2)==3);
rows_4 = find(data(:,2)==4);
rows_5 = find(data(:,2)==5);
rows_6 = find(data(:,2)==6);
rows_7 = find(data(:,2)==7);

t_0 = data(rows_0,:);
t_1 = data(rows_1,:);
t_2 = data(rows_2,:);
```

```
t_3 = data(rows_3,:);
t_4 = data(rows_4,:);
t_5 = data(rows_5,:);
t_6 = data(rows_6,:);
t_7 = data(rows_7,:);

figure()
subplot(1,2,1)
hold on
plot(diff(t_0(:,1)),'b-o')
plot(diff(t_1(:,1)),'k-o')
plot(diff(t_2(:,1)),'r-o')
plot(diff(t_3(:,1)),'g-o')
plot(diff(t_4(:,1)),'y-o')
plot(diff(t_5(:,1)),'c-o')
plot(diff(t_6(:,1)),'m-o')
plot(diff(t_7(:,1)),'b--o')
xlabel('data points')
ylabel('delta t [ms]')

subplot(1,2,2)
hold on
grid on
S_0 = diff(t_0(:,1));
S_1 = diff(t_1(:,1));
S_2 = diff(t_2(:,1));
S_3 = diff(t_3(:,1));
S_4 = diff(t_4(:,1));
S_5 = diff(t_5(:,1));
S_6 = diff(t_6(:,1));
S_7 = diff(t_7(:,1));
group = [ones(size(S_0));...
        2*ones(size(S_1));...
        3*ones(size(S_2));...
        4*ones(size(S_3));...
        5*ones(size(S_4));...
        6*ones(size(S_5));...
        7*ones(size(S_6));...
        8*ones(size(S_7))];
boxplot([S_0; S_1; S_2; S_3; S_4; S_5; S_6; S_7], group)
set(gca,'XTickLabel',{'0','1','2','3','4','5','6','7'})
xlabel('sensor number')
ylabel('delta t [ms]')

% plot(t_0(:,13),'b-o')
% plot(t_1(:,13),'k-o')
% plot(t_2(:,13),'r-o')
% plot(t_3(:,13),'g-o')
% plot(t_4(:,13),'y-o')
% plot(t_5(:,13),'c-o')
% plot(t_6(:,13),'m-o')
% plot(t_7(:,13),'b-o')
```

Appendix 5 Full code for Test_eight.m, final code for analyzing data drops