2022

Evaluating the benefits for sensorization of a prosthetic hand in the haptic bag challenge of Cybathlon 2024

DANIËLLE STEENGE, S4732561
PRE-MASTER STUDENT BIOMEDICAL ENGINEERING
BACHELOR RESEARCH PROJECT

FIRST SUPERVISOR: DR. ELISABETH WILHELM
SECOND SUPERVISER: DR. AJAY KOTTAPALLI

Abstract

The Cybathlon is a non-profit competition of ETH Zürich for assistive devices. The goal of this competition is to challenge existing technologies and develop assistive technologies suitable for everyday use with and for people with disabilities. Currently, arm prosthesis users have to look at their hand in order to adjust finger position and grip force. Amputees wish to reduce the visual attention necessary during grasping. Resulting in the research question: Is it possible by means of a proof of concept with commercial sensors to write an algorithm through which the shape and hardness of a three-dimensional object can be recognized to win the Cybathlon 2024?

Two participants were asked to recognize 12 objects while wearing a glove with three force sensors, without seeing the objects. The non-zero mean, non-zero median, maximum values, and contact counts were calculated and plotted in box plots to visually assess the differences. However, based on this visual assessment it remains unclear whether these features contain sufficient information to identify the individual objects. Due to the fact that multiple sensors broke down during the tests, and because of time restrictions of this project it was not possible to repeat the tests, the results are not reliable. However, this pilot experiment gives important information for follow up experiments.

Introduction

One of the new challenges of the Cybathlon 2024 that has not yet been solved is how to provide a sense of touch to the arm prosthesis user [1]. Currently, arm prosthesis users have to look at their hand in order to adjust finger position and grip force. As previous studies show, amputees wish to reduce the visual attention necessary during grasping [2], [3].

During this challenge of Cybathlon 2024, the arm prosthesis user (the pilot) must recognise and match objects of different shapes and compliance in the absence of visual feedback. There are three pairs of bags. For each pair of bags, the pilot is asked to remove any one object from the first bag and then to identify and re-move the identical object from the second bag. Each pair of bags contains a subset of three different objects from a pool of six objects [4]:

- Pair A: bag 1 and bag 2 contain identical objects of low compliance
- Pair B: bag 1 and bag 2 contain identical objects of high compliance
- Pair C: bag 1 contains objects of high compliance, and bag 2 contains the same objects in low compliance.

The presentation order of the pairs of bags (A, B, C) will be randomised between races [4].

For this project, Grove round force sensors (FSR402) were integrated in a flexible glove which was provided to healthy volunteers. The participants were instructed to grab into a bag and try to identify the shape and hardness of a three-dimensional object. This process was repeated 10 times for 12 objects that were presented in randomized order. The recorded sensor data was subsequentially used to develop a recognition algorithm.

The aim of this study is to find out how to get significant distinctive features from the sensors, while grasping objects from Cybathlon 2024. In addition to this, the goal is to look how to make different classes for different objects. Resulting in the research question: *Is it possible by means of a proof of concept with commercial sensors to write an algorithm through which the shape and hardness of a three-dimensional object can be recognized to win the Cybathlon 2024?*

Due to the fact that sensors broke during the tests and there was no time left to repeat the tests during this project, the results are not reliable. However, this pilot experiment gives important information for follow up experiments. An algorithm has been written that calculates different data points from the obtained sensor values. There are also points for improvement within the design that can be considered in follow-up studies.

Material and Methods

Technical setup

For this project three round force sensors (the Grove – Round Force Sensor (FSR402), Seeed Studio, made in China) are used. These sensors are force sensitive resistors. The resistance depends on the pressure applied on the resistor. The force sensitivity range is between 0.2N and 20N (Newton). The force sensitivity is optimized for use in human machine interface devices including medical systems and robotics [5]. The sensors were placed at the fingertip of the thumb, the index finger, and the middle finger. The sensors are robust and lay flat on the fingertip. Because the sensors are not flexible enough to adjust to the shape of the fingertip and are relatively large for the fingertip, it will be more difficult to feel an object than without sensors. In addition, the sensors have a smooth surface, which makes it more difficult to get a grip on the object and more force will be needed to grip the object.

The force sensors are connected to the Arduino Uno (see figure 1, here the resistors are visible but with the FSR402 this is integrated via a small connector). The Arduino is used as an analog-digital converter (ADC), it transfers the voltage of the analog input into digital values which are send to a computer via a serial connection. Further, the computer writes the data that is received at the serial port into an excel file which is saved as an CSV file. When the data is saved in an CSV file, it can be further viewed and used for the algorithm. The code of the force sensors can be found in Appendix A [5]. A disposable, Peha-soft nitrile glove (Hartmann, non-powdered, latex free, peha-soft nitrile, REF942 191, The Netherlands) was used as an imitation of an arm prosthesis. Sensors are glued to this with Pattex super glue (Pattex uni-rapide super gel, The Netherlands).

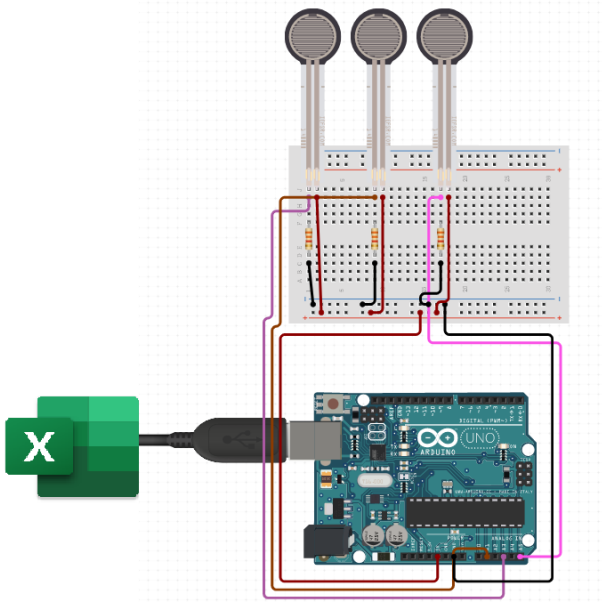

FIGURE 1 CONNECTIONS OF THE FORCE SENSORS WITH THE ARDUINO UNO [6]

During the haptic bag test of the Cybathlon 2024, 12 different objects need to be recognized (see figure 2 [4]). For the six objects with low compliance, a baseball (SSK Europe, DOL8.5 – Diamond DOL8.5, Junior size baseball, composite cork core, leather exterior, The Netherlands) was used for the sphere and a painted beech wood block from a children's play set block box was used for the cube. The other four objects with low compliance are made using a 3D printer (Creality 3D Ender 3, Creality, construction year 2018, The Netherlands). For printing, polylactic acid filament (Velleman Vertex PLA175R07, 1.75mm PLA filament, red, The Netherlands) has been used. The soft objects are cut from a soft foam (anti-static, low

density, polyurethane foam, RS PRO, The Netherlands). The settings of the 3D-printer can be found in Appendix A. Since the exact dimensions of the objects for the Cybathlon 2024 are not known, dimensions were chosen by the experimenter. The exact dimensions of the objects that has been chosen for this project can also be found in Appendix A.

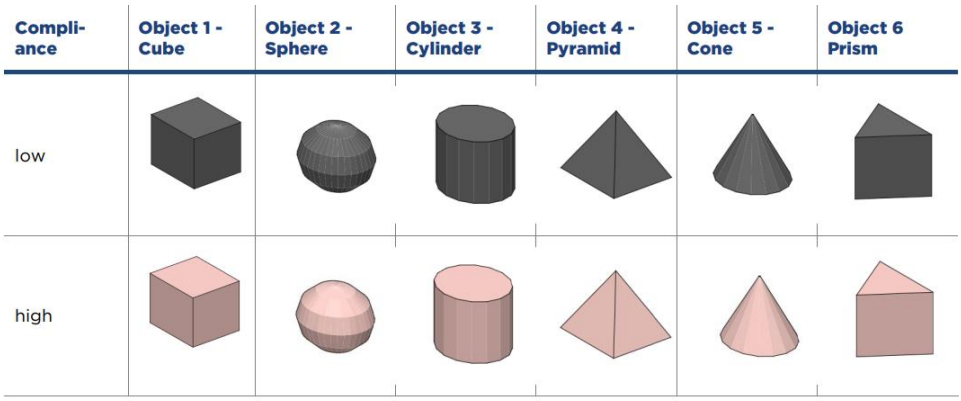

FIGURE 2 POOL OF OBJECTS THAT ARE PLACED IN THE BAGS [4]

Participants

Because this project is too short for ethical approval, no participants will be recruited. Instead, the researcher and supervisor will conduct the tests, to record pre-liminary data and verify that the study protocol developed for this project is feasible. If the results are positive a proper study with ethical approval might be conducted after the project.

The researcher and supervisor will first put on the glove with sensors on the dominant hand. The dominant hand is chosen because when using both hands, everyone dominantly uses one hand, and this is called left-handedness or right-handedness [7]. Several studies show that the dominant hand has better motor skills because it is more often used for grasping and grasping objects. The dominant hand has more developed muscles due to frequent usage. This leads to more controllable and stronger muscles in this dominant hand [7], [8], [9], [10]. To determine the dominant hand, it will first be examined by the test of The Dutch Handedness Questionnaire by Jan W. Van Strien [11].

The questionnaire of Van Strien consisted of sixteen hand preference items, which can be found in Appendix A. Each question was coded from 0 to 2, with "left" receiving a score of 0 and "right" receiving a score of 2, and "both" receiving a score of 1. Therefore, the total score could range from 0 (e.g., extremely left-handed) to 32 (e.g., extremely righthanded). Subjects are considered to be strongly left-handed if their total score on the questionnaire equalled four or less, or to be strongly right-handed if this score equalled twenty-eight or more [11].

The sensors are then checked whether they are correctly positioned (on the fingertips), and whether the values are realistic (see figure 3). For example, at a higher pressure, the resistance of the sensor will be smaller, resulting in a higher output value than when a lower pressure is applied.

All objects are in a non-transparent plastic bag. The experimenter will randomly place (order is generated with Excel command Randomize), one object at a time, inside the other non-transparent bag. The participant was then asked to recognize the objects with the fingertips, since the force sensors are placed here.

After this, the test was started, from that moment the participant starts identifying the object by putting the dominant hand in the bag. The participant will not receive any feedback about whether the recognized object is correct or not. Since there is a restriction of 8 minutes for the Cybathlon 2024 to recognize 12 objects, there is a maximum of 40 seconds to recognize the object (8 minutes * 60 seconds / 12 objects = 40 seconds per object). Once the participant has recognized the object, the object will be released, and the participant takes the handout of the bag.

It is important that the participant does not see the objects during the entire test, so that no learning process can occur. The experimenter stops the recording and saves it in a CSV file for later analysis.

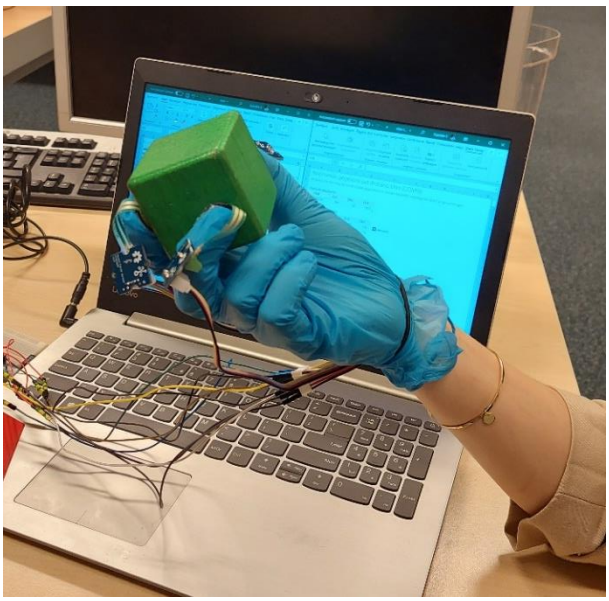

FIGURE 3 PARTICIPANT HOLDING OBJECT HARD CUBE WITH THE SENSOR GLOVE (SINCE THE OBJECT IS VISIBLE, THIS IS NOT DURING A TEST)

Subsequently, the object recognized by the participant is also written down for later analysis. This process has been repeated for all objects by the experimenter without the participant seeing it. This test will be done ten times with all twelve objects per test. This results in 120 measurements per participant. When the researcher and supervisor have both completed all the tests, there will be 240 measurements. A complete flowchart of the tests can be found in Appendix A.

Planned analysis

To start the analysis, the time between first contact moment with the object until recognition of the object will be analysed. Here it will be checked whether the participant has developed a learning curve and can therefore recognize the objects more quickly after performing the tests multiple times. All these times will be plotted, after which it can be seen whether the participant gets faster by performing the test more often.

In addition to this, the time of how long it took for the participant to recognize the object has been analysed. The purpose of this is to see if there is a relationship with the duration of recognizing the object and the object. This has been analysed by the average times it took per participant to recognize an object is looked at.

There is an algorithm written in Python (Python 3.9.10). In which the CSV files are loaded and analysed one by one. First, several calculations are made with the data points of the test. These include the maximum value of each sensor, the non-zero mean of each sensor, the non-zero median of each sensor, and the contact counts per sensor. It is important to use the nonzero values, as the sensor values are zero when no touch is detected. Subsequently, the outcomes are plotted in boxplots to get an overview of the values. The complete code can be found in Appendix A.

Results

Time

As mentioned in the chapter Materials and Methods, a total of 240 measurements were obtained from the tests. The exact order of the objects per test can be found in Appendix B. Beside this, the test results of the Dutch Handedness Questionnaire can also be found in Appendix B, which show that participant 1 had a score of 32 (extremely righthanded) and participant 2 had a score of 29 (strongly right-handed).

The data of the 240 measurements showed how long it took the participant from first touching the object to recognition of the object. In addition, the average time has been calculated as well in tables 1 and 2. Participant 1 did not recognize the soft cone twice and recognized it as a soft pyramid (during test 1 and test 6). When participant 2 performed the tests, the soft prism was recognized as a soft pyramid once (during test 3), and the soft cube was recognized as a soft prism once (during test 4). In addition, an error was also made during some of the recordings which resulted that no data was recorded. This happened during test 1 soft cube of participant 1, and test 3, hard cube of participant 2.

Sensor 1 (the sensor on the thumb) broke during test 2 at the first object from that test (soft cone). This was found out during test 3 and upon detection the sensor was exchanged with a spare one between the tests of the hard pyramid and hard cone. This resulted in a lot of noise from sensor 1 which has been resolved between the tests of the soft cylinder and soft pyramid. After the test with the soft pyramid, sensor 1 broke again and because there were no spare sensors available, it was decided to continue the tests without feedback from sensor 1. During test 7, sensor 2 (the sensor on the index finger) broke too. As a result, no data was detected during some tests (e.g., test 8 and test 9 with hard pyramid, and test 10 with soft sphere).

TABLE 1 RECOGNITION TIMES PARTICIPANT 1

Object	Time test 1	Time test 2	Time test 3	Time test 4	Time test 5	Time test 6	Time test 7	Time test 8	Time test 9	Time test 10	Average	
Hard cube	7,35	6	21,15	6,75	5,25	3,75	4,2	5,1	7,5	7,2	*7,425*	*seconds*
Hard sphere	6,15	14,4	14,7	4,2	11,7	9,6	10,95	3,6	16,5	9,6	*10,14*	*seconds*
Hard cylinder	6	3,6	4,35	10,65	5,1	10,2	3,75	6,9	7,05	3,3	*6,09*	*seconds*
Hard pyramid	3,45	3,6	6	5,25	7,05	5,85	3,6	5,85	4,65	10,95	*5,625*	*seconds*
Hard cone	4,8	7,2	6,15	13,2	12,3	6	5,55	7,65	8,1	5,7	*7,665*	*seconds*
Hard prism	4,35	3,6	5,7	6,15	1,8	2,55	7,8	4,5	4,8	6	*4,725*	*seconds*
Soft cube		4,2	5,85	5,7	10,35	6,75	7,35	8,55	6,15	4,65	*6,616666667*	*seconds*
Soft sphere	4,05	4,35	4,35	13,5	9,9	2,55	7,95	14,25	5,85	4,05	*7,08*	*seconds*
Soft cylinder	9	4,8	4,8	4,2	5,25	9,45	9,3	7,65	4,2	5,25	*6,39*	*seconds*
Soft pyramid	9,3	8,1	4,2	8,7	6	3	5,7	5,25	4,95	6,3	*6,15*	*seconds*
Soft cone	6,45	4,65	4,65	4,5	6,9	4,8	4,05	5,7	9,15	8,55	*5,94*	*seconds*
Soft prism	7,2	4,2	3,3	8,1	4,95	4,5	8,85	4,8	5,25	5,55	*5,67*	*seconds*

TABLE 2 RECOGNITION TIMES PARTICIPANT 2

Object	Time test 1	Time test 2	Time test 3	Time test 4	Time test 5	Time test 6	Time test 7	Time test 8	Time test 9	Time test 10	Average	
Hard cube	6	5,1		8,4	5,1	8,25	7,65	8,25	9,3	7,05	*7,23333*	*seconds*
Hard sphere	5,55	4,5	5,55	5,25	8,85	8,85	10,35	6,6	3,75	3	*6,225*	*seconds*
Hard cylinder	4,5	4,5	6,6	8,25	9,3	3	7,95	2,55	4,05	3,3	*5,4*	*seconds*
Hard pyramid	6,9	10,2	13,8	8,4	6,3	8,85	4,5			7,05	*8,25*	*seconds*
Hard cone	14,1	6,3	6,45	2,25	7,35	6,9	4,05			5,1	*6,5625*	*seconds*
Hard prism	13,95	5,7	4,8	6,3	6,45	7,95	8,55			8,85	*7,81875*	*seconds*
Soft cube	13,05	7,95	4,5	5,5	6,75	3,15	6,3	3,3	10,2	6,75	*6,745*	*seconds*
Soft sphere	14,25	6,6	8,7	6,9	4,95	6,75	6,45	4,05	7,8		*7,38333*	*seconds*
Soft cylinder	8,85	6,3	7,2	7,8	5,55	5,1	5,1	5,25	1,65	3	*5,58*	*seconds*
Soft pyramid	7,2	6	5,1	3,3	5,4	4,05	4,2	3,15	4,35		*4,75*	*seconds*
Soft cone	15,45	9,9	5,4	4,5	7,5	7,95	4,2	5,25	3,9	3	*6,705*	*seconds*
Soft prism	4,8	5,4	5,25	6,75	2,85	7,05	7,5	4,95	4,65	1,95	*5,115*	*seconds*

All times are plotted in a graph to see if there is a learning curve in the participant (see figures 4, 5, 6, and 7). If this were the case, the time per object to recognize it would have to decrease. The time it took to recognize objects did not decline from the first attempts average number to the 10th attempt average time. Therefore it seems that the participants did not learn during the experiment

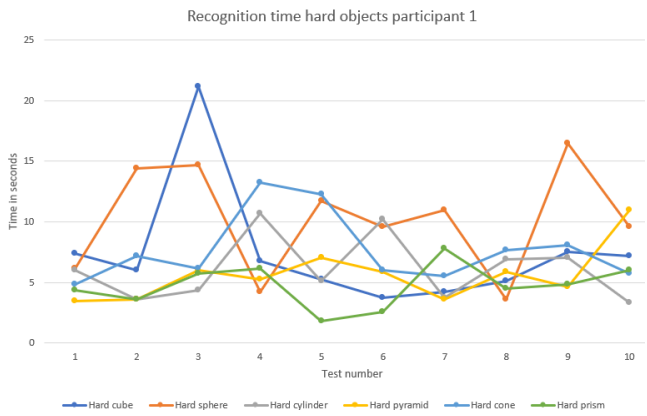

FIGURE 4 RECOGNITION TIME PARTICIPANT 1 HARD OBJECTS PLOTTED

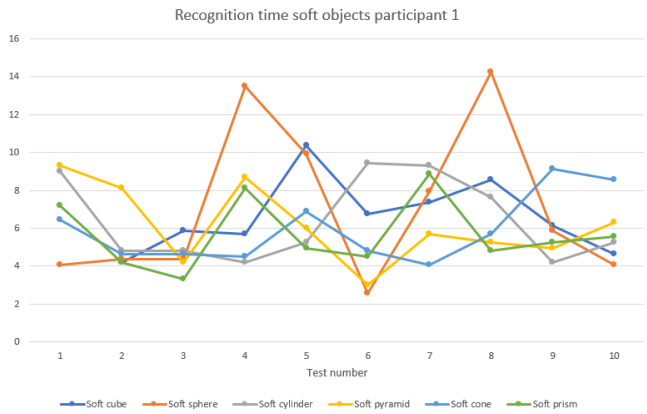

FIGURE 5 RECOGNITION TIME PARTICIPANT 1 SOFT OBJECTS PLOTTED

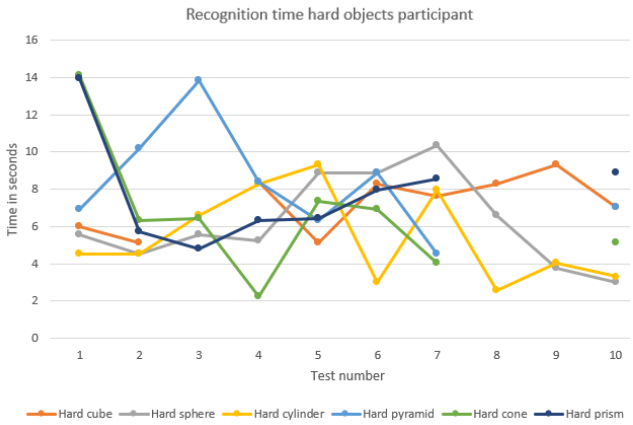

FIGURE 6 RECOGNITION TIME PARTICIPANT 2 HARD OBJECTS PLOTTED

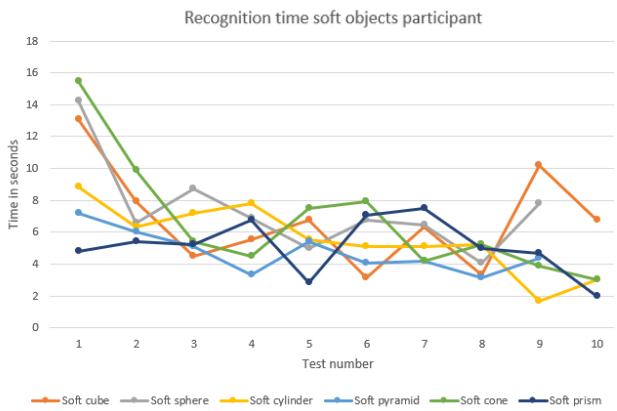

FIGURE 7 RECOGNITION TIME PARTICIPANT 3 SOFT OBJECTS PLOTTED

To identify whether the hardness of an object has an influence on the time of identifying the objects, the times to recognition per object of both participant 1 and participant 2 are plotted in boxplots in figure 8. The data with the standard deviation are shown in table 3. There are objects where the difference in time is relatively large (e.g., for the hard sphere) and for other objects the difference in time is relatively small (e.g., for the soft pyramid).

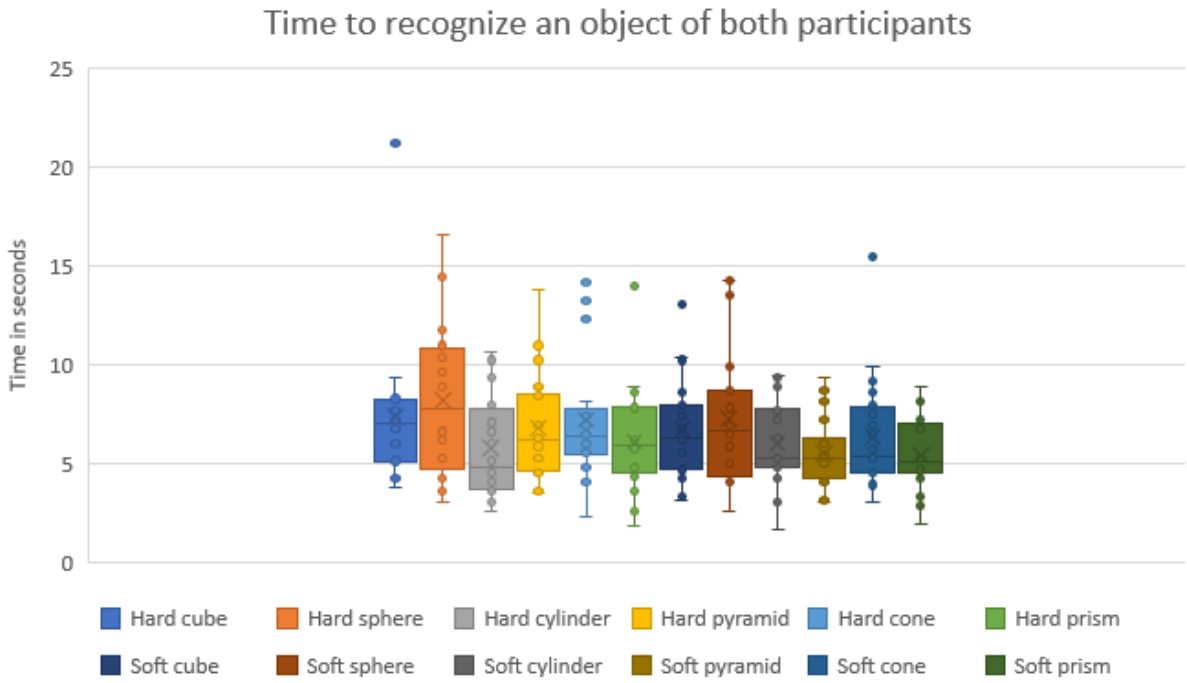

FIGURE 8 BOXPLOTS OF THE TIME TO RECOGNIZE AN OBJECT

TABLE 3 AVERAGE TIME TO RECOGNIZE AN OBJECT WITH STANDARD DEVIATION IN SECONDS

Object	Average time Participant 1 (in seconds)	Average time Participant 2 (in seconds)	Standard deviation (in seconds)
Hard cube	7,4	7,2	3,7
Hard sphere	10,1	6,2	4,0
Hard cylinder	6,1	5,4	2,5
Hard pyramid	5,6	8,3	2,8
Hard cone	7,7	6,6	3,1
Hard prism	4,7	7,8	2,7
Soft cube	6,6	6,7	2,5
Soft sphere	7,1	7,4	3,5
Soft cylinder	6,4	5,6	2,2
Soft pyramid	6,2	4,8	1,8
Soft cone	5,9	6,7	2,9
Soft prism	5,7	5,1	1,8

Recognition

To detect if it is possible to recognize the objects from the data, an algorithm has been written in Python. The maximum values, the nonzero median and the nonzero mean of the three sensors are plotted in the boxplots in figures 9 and 10.

In figure 9 the boxplots sorted by the hard and soft objects are plotted. There is a lot of overlap with the different box plots. To assess the data, the mean, the median, the maximum values are compared for hard and soft. The minimum values were not included since at every boxplot the minimum is zero (see table 4 and plot of table 4 in Appendix B). In these boxplots, the differences in values for a hard object compared to a soft object, are shown.

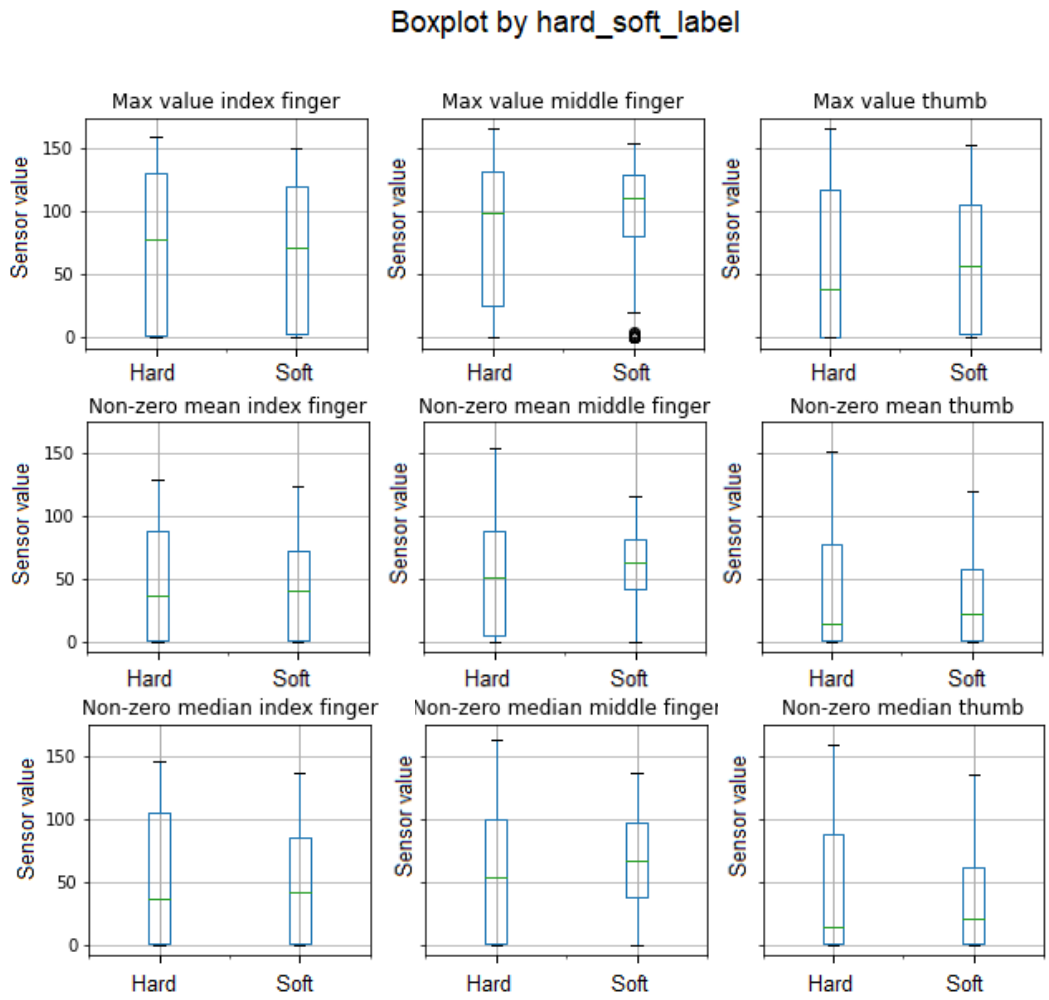

FIGURE 9 BOXPLOTS SORTED BY HARD OR SOFT OBJECTS AND SENSOR VALUES

In figure 10 the boxplots that are sorted by the shape of the objects are plotted. There also in these boxplots lots of overlap. That is why also the mean, the median, the maximum values are also calculated by the algorithm (the table and plot can be found in Appendix B).

TABLE 4 DATAPOINTS FROM THE BOXPLOTS OF FIGURE 9

		Mean	Median	Maximum
Max value index vinger	*Hard*	72	78	159
	Soft	68	72	150
Max value middle vinger	*Hard*	82	99	166
	Soft	100	112	154
Max value thumb	*Hard*	58	38	166
	Soft	56	58	153
Non-zero mean index finger	*Hard*	47	37	129
	Soft	41	40	123
Non-zero mean middle finger	*Hard*	53	51	154
	Soft	60	63	115
Non-zero mean thumb	*Hard*	38	15	151
	Soft	33	23	120
Non-zero median index finger	*Hard*	52	36	146
	Soft	46	41	136
Non-zero median middle finger	*Hard*	58	53	163
	Soft	65	67	136
Non-zero median thumb	*Hard*	43	14	158
	Soft	36	20	135

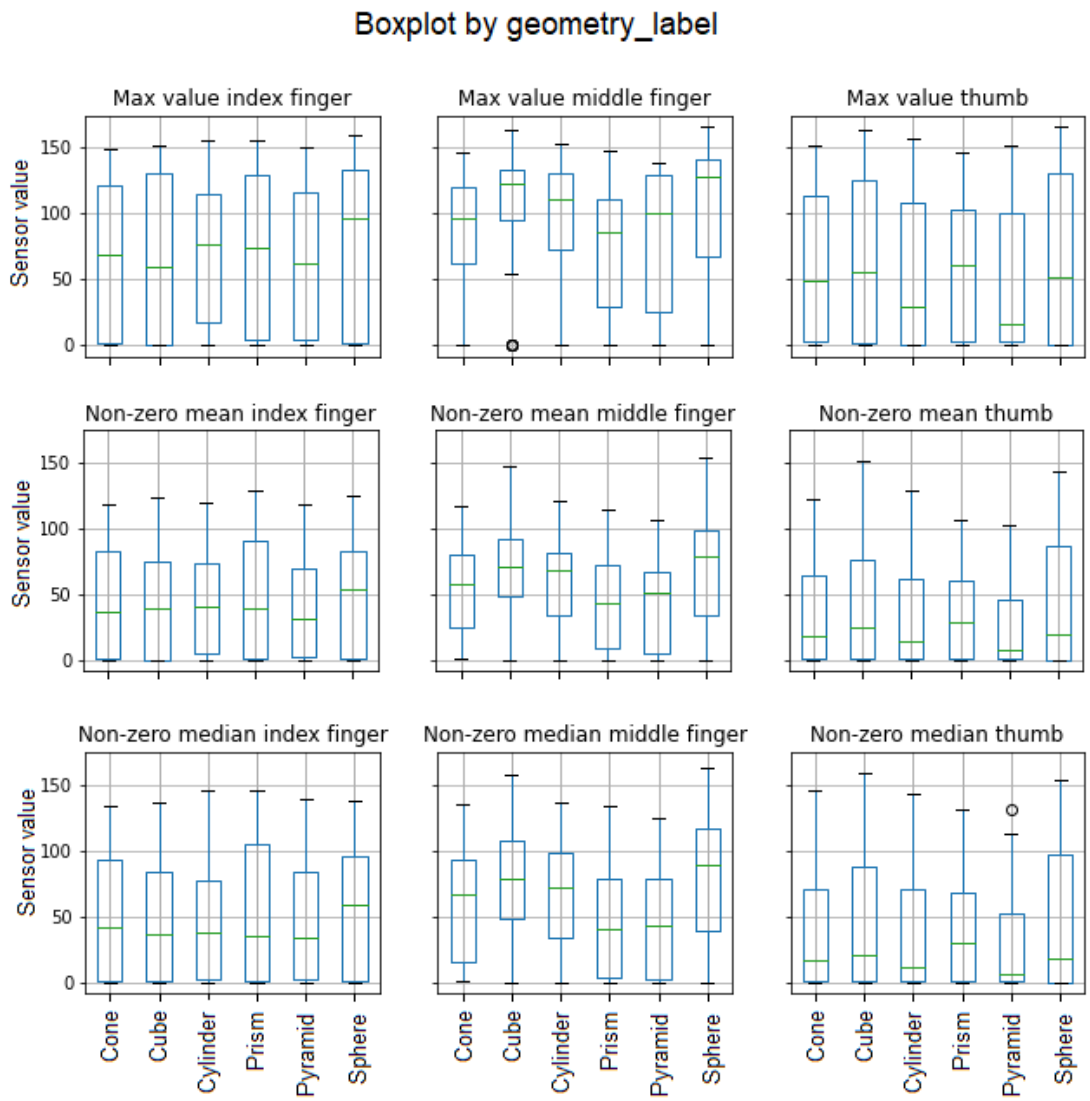

FIGURE 10 BOXPLOTS SORTED BY SHAPE OF THE OBJECT

Discussion

The technical setup of this project was not optimal, resulting in sensors breaking due the freely hanging wires. This can be fixed by taping the wires on the flexible glove which could also give the participant more room to move the hand without being worried about breaking the wires.

During this project, it was noted that three sensors at the fingertips alone is limiting the recognition of the objects. It was naturally noticed by the participants that the objects also included feedback from the palm to recognize the objects. The three force sensors placed at the fingertips is relatively low compared to competitors. For example, CyberLimb uses eight force sensors, and the Technische Universität München (TUM) uses twelve force sensors [12], [13]. The results from the CyberLimb were that the pilot was able to accomplish the Haptic Box task during the training sessions but was not able to accomplish the task during the competition due to time constraints and a sensor failure of the haptic sensors [12].

Subsequently, because only human hands were used during this project, the results of an arm prosthesis will most likely differ at the contact forces, since the arm prothesis has less degrees of freedom than a human hand. That is why in follow-up studies it should be considered to perform the tests with the arm prosthesis instead of human hands.

To get a more accurate picture of whether time has an influence on the recognition of the different objects, the arm prothesis should also perform the tests for measurement. This is to get a better view on how the arm prosthesis makes contact with the object to recognize it.

In addition, it is not certain for participant 2 how long some tests lasted, which also influenced the data. This is because the participant has not used sensor 3 to recognize the objects during some of the tests where sensors 1 and 2 were broken down. The times of recognizing the object of the tests after sensor 1 has broken are therefore not reliable since it is not exactly clear when the first and last contact with the object took place.

Beside this, the Grove round force sensors (FSR402) were used during this project. It is a good point to also look at other sensors during follow-up studies. Unfortunately, this was outside the scope of this project, but it is certainly important for achieving the more accurate results.

Moreover, a foundation has been made for follow-up studies to work further on machine learning. In the algorithm are several calculations made that can be useful for machine learning. With machine learning the computer can learn to identify patterns in the recorded sensor values that reveal which specific object was touched. The CyberLimb and the TUM have also done this during Cybathlon 2020 [12], [13]. The makers of the CyberLimb made a machine learning algorithm for detecting different objects based on force signals from force resistor sensors on the gripper. Several features that could be used for training a feature-based machine learning algorithm are calculated during this project, like the mean, median and maximum values.

The tests that were performed by the participants during this project differ from the challenges during Cybathlon 2024. It was decided that this pilot experiment gives important information for follow up experiments and to see whether it is possible to recognize the objects on the basis of the sensor data.

During follow-up studies, when it is clear which sensors and which algorithms are possible to recognize the objects, the tests should also be performed according to the Cybathlon Races and Rules 2024 [4], where the exact test during the Cybathlon is described.

To answer the research question: '*Is it possible by means of a proof of concept with commercial sensors to write an algorithm through which the shape and hardness of a three-dimensional object can be recognized to win the Cybathlon 2024?',* previous studies show that with force

resistor sensors and a machine learning algorithm, it is possible to recognize different compliance and shapes of objects without visual feedback. Due to limited time during this project, it has not been accomplished to fully answer the research question by means of a proof of concept. A foundation has been made for the machine learning algorithm and it was found how to improve the sensor attachment on the hand.

References

[1] "This is Cybathlon," [Online]. Accessed 19 April 2022. Available: https://cybathlon.ethz.ch/en/cybathlon.

[2] Pylatiuk C, Schulz S, Döderlein L. Results of an Internet survey of myoelectric prosthetic hand users. Prosthet Orthot Int. 2007 Dec;31(4):362-70. doi: 10.1080/03093640601061265. PMID: 18050007.

[3] E. Comission, “Accidents at work by NACE Rev. 2 activity and type of injury,” 2017. [Online]. Available: http://appsso.eurostat.ec.europa.eu/nui/show.do?dataset=hsw n2 lang=en.

[4] Cybathlon, "Races and Rules 2024 are published," [Online]. Accessed 7 April 2022. Available: https://cybathlon.ethz.ch/en/news/5710/races-and-rules-2024-are-published.

[5] "Grove Round Force Sensor FSR402," Seeed Studios, [Online]. Accessed 24 May 2022. Available: https://wiki.seeedstudio.com/Grove-Round_Force_Sensor_FSR402/.

[6] “Design your circuit with circuit.io” [Online]. Accessed 2 June 2022. Available: https://www.circuito.io/.

[7] Kim, J. S., Lee, S. G., Park, S. K., Lee, S. M., Kim, B. K., Choi, J. H., & Kim, S. H. (2011). Comparison of grip and pinch strength between dominant and non-dominant hand according to type of handedness of female college students. *Journal of International Academy of Physical Therapy Research*, *2*(1), 201-206.

[8] Koley, S., & Singh, A. P. (2010). Effect of hand dominance in grip strength in collegiate population of Amritsar, Punjab, India. *The Anthropologist*, *12*(1), 13-16.

[9] Mcsp ICB, Dipcot JA. A Comparison of Dominant and Non-dominant Hand Function in both Right- and Left-Handed Individuals using the Southampton Hand Assessment Procedure (SHAP). The British Journal of Hand Therapy. 2003;8(1):4-10. doi:10.1177/175899830300800101

[10] Gedela, R. K., Kirby, A., & Huhtala, H. (2008). Does the dominant hand possess 10% greater grip strength than the non-dominant hand. *Indian Journal of physiotherapy*, *45*(2), 116.

[11] J. W. Van Strien, "The Dutch Handedness Questionnaire," FSW, Department of Psychology, Erasmus University Rotterdam, Rotterdam, December 2002.

[12] Seppich, N., Tacca, N., Chao, K. Y., Akim, M., Hidalgo-Carvajal, D., Pozo Fortunić, E., ... & Haddadin, S. (2022). CyberLimb: a novel robotic prosthesis concept with shared and intuitive control. *Journal of NeuroEngineering and Rehabilitation*, *19*(1), 1-20.

[13] Trimborn, N. (2021). *Design, implementation and evaluation of an embedded electronic system for the control of a trans-radial prosthesis* (Master's thesis).

[14] "Pandas," [Online]. Accessed May 2022. Available: https://pandas.pydata.org/.

[15] "Numpy," [Online]. Accessed May 2022. Available: https://numpy.org/. [Accessed May 2022].

Appendix A

```
//Constants:
const int sensorPin0 = A0; //pin A0 to read analog input
const int sensorPin1 = A1; //pin A1 to read analog input
const int sensorPin2 = A2; //pin A2 to read analog input


//Variables:
int value0; //save analog value
int value1;
int value2;


void setup(){

  Serial.begin(9600);       //9600 bits per second
}

void loop(){

  int value0 = analogRead(sensorPin0);
  int value1 = analogRead(sensorPin1);
  int value2 = analogRead(sensorPin2);

  value0 = map(value0, 0, 1023, 0, 255); //Map value 0-1023 to 0-255 (PWM)
  value1 = map(value1, 0, 1023, 0, 255); //Map value 0-1023 to 0-255 (PWM)
  value2 = map(value2, 0, 1023, 0, 255); //Map value 0-1023 to 0-255 (PWM)



  Serial.print(value0);               //Print value
  Serial.print(",");

  Serial.print(value1);               //Print value
  Serial.print(",");



  Serial.print(value2);               //Print value
  Serial.println(",");


}
```

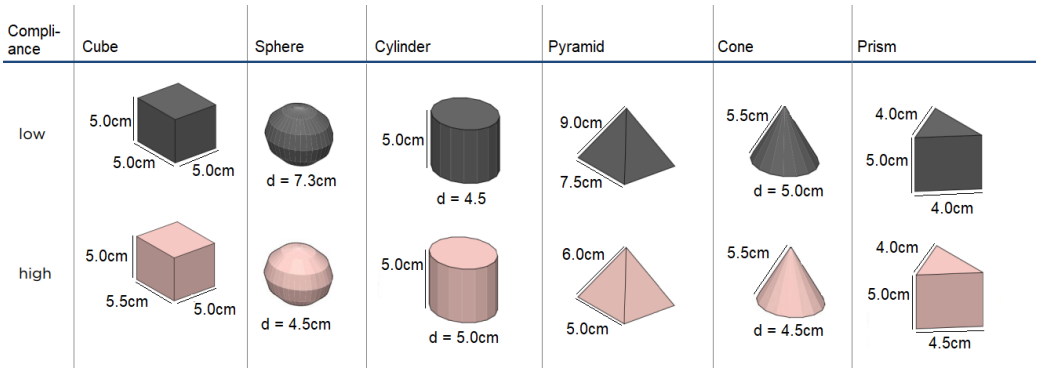

FIGURE 11 THE EXACT DIMENSIONS OF THE OBJECTS [4]

The exact settings of the 3D-printer, Creality Ender 3, in Ultimaker Cura 4.2.1.:

Infill		
Infill Density	5	%
Infill Line Distance	17.6	mm
Infill Pattern	Grid	
Connect Infill Lines		
Infill Line Directions	[]	
Infill X Offset	0	mm
Infill Y Offset	0	mm
Infill Line Multiplier	1	
Infill Overlap Percentage	30.0	%
Infill Overlap	0.132	mm
Skin Overlap Percentage	10.0	%
Skin Overlap	0.044	mm
Infill Wipe Distance	0.0	mm
Infill Layer Thickness	0.16	mm
Gradual Infill Steps	0	
Infill Before Walls		
Minimum Infill Area	0	mm²

Quality		
Layer Height	0.16	mm
Initial Layer Height	0.12	mm
Line Width	0.44	mm
Wall Line Width	0.44	mm
Outer Wall Line Width	0.44	mm
Inner Wall(s) Line Width	0.44	mm
Top/Bottom Line Width	0.44	mm
Infill Line Width	0.44	mm
Skirt/Brim Line Width	0.44	mm
Initial Layer Line Width	100.0	%

Shell		
Wall Thickness	1	mm
Wall Line Count	3	
Outer Wall Wipe Distance	0.0	mm
Top Surface Skin Layers	0	
Top/Bottom Thickness	1.08	mm
Top Thickness	1.08	mm
Top Layers	3	

Material		
Default Printing Temperature	185	°C
Printing Temperature	185	°C
Printing Temperature Initial Layer	185	°C
Initial Printing Temperature	185	°C
Final Printing Temperature	185	°C
Default Build Plate Temperature	60	°C
Build Plate Temperature	60	°C
Build Plate Temperature Initial Layer	60	°C
Flow	100	%
Wall Flow	100	%
Outer Wall Flow	100	%
Inner Wall(s) Flow	100	%
Top/Bottom Flow	100	%
Infill Flow	100	%
Skirt/Brim Flow	100	%
Support Flow	100	%
Prime Tower Flow	100	%
Initial Layer Flow	100	%

Speed		
Print Speed	50	mm/s
Infill Speed	50	mm/s
Wall Speed	25.0	mm/s
Outer Wall Speed	25.0	mm/s
Inner Wall Speed	25.0	mm/s
Top/Bottom Speed	25.0	mm/s
Travel Speed	150.0	mm/s
Initial Layer Speed	20.0	mm/s
Initial Layer Print Speed	20.0	mm/s
Initial Layer Travel Speed	100.0	mm/s
Skirt/Brim Speed	20.0	mm/s
Number of Slower Layers	2	

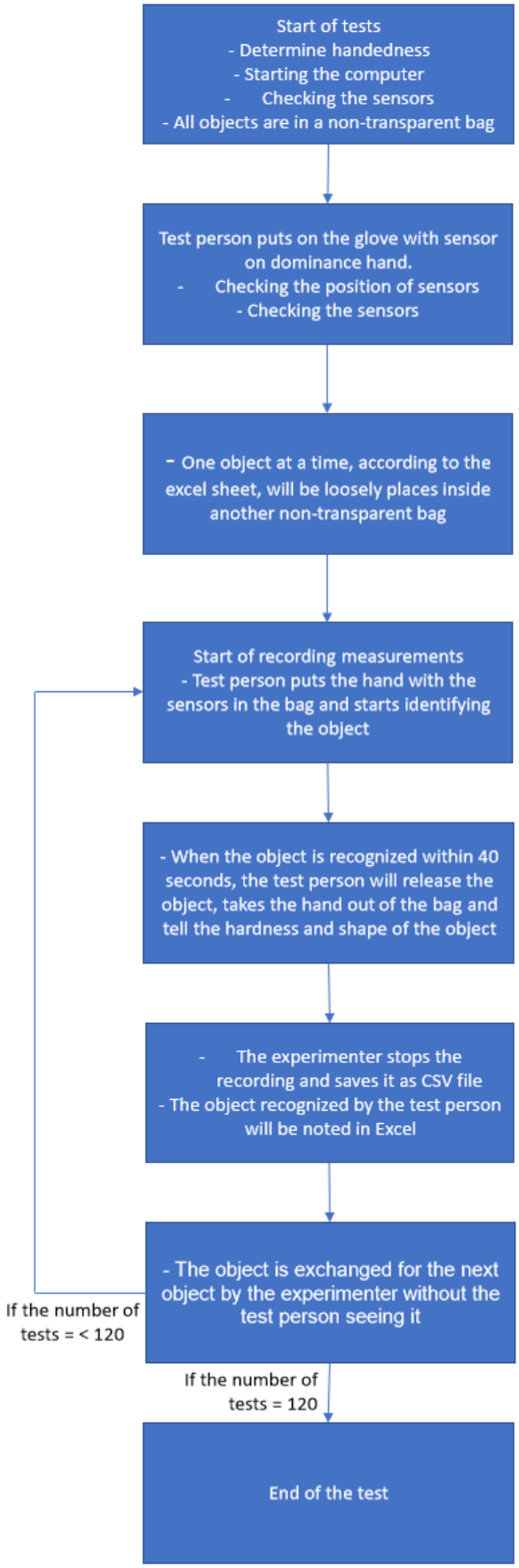

FIGURE 12 FLOWCHART OF TESTS

Python code for recognition

Sources: [14] [15]

```
import numpy as np
import pandas as pd
import os
from matplotlib import pyplot as plt

#######
filelocation=['C:/Users/Steenge/Documents/Master Universiteit Groningen/Pre-master/2b/Bachelor Research project BME/Test results Danielle', 'C:/Users/Steenge/Documents/Master Universiteit Groningen/Pre-master/2b/Bachelor Research project BME/Test results Elisabeth'] # TODO: Rename those two like your folders are named


# in the beginning we need to initiate lists for our features
max_sen1=[]
max_sen2=[]
max_sen3=[]
min_sen1=[]
mean_sen1=[]
contact_counts_sen1=[]
non_zero_mean_sen1=[]
non_zero_median_sen1=[]
mean_sen2=[]
contact_counts_sen2=[]
non_zero_mean_sen2=[]
non_zero_median_sen2=[]
mean_sen3=[]
contact_counts_sen3=[]
non_zero_mean_sen3=[]
non_zero_median_sen3=[]
filenames=[]
hard_soft_label=[]
geometry_label=[]
participant_label=[]

for count3 in range(len(filelocation)):

    datafile_names=os.listdir(filelocation[count3])

    #print(datafile_names)
# to get all data in we loop through all the file names in the folder
    for count in range(len(datafile_names)):
  # as there might be some system files we first ensure that the filename starts with t in case there are other relevant files you need ot adjust the if statement
        if datafile_names[count][0]=='t':
        # the following lines are from your code
            data=pd.read_csv(os.path.join(filelocation[count3], datafile_names[count]))
            filenames.append(datafile_names[count])

            data.rename(columns={'#!': 'Sensor 1', 'Workbook:': 'Sensor 2', 'Custom': 'Sensor 3'}, inplace = True)  #rename the colums

            data=data.drop([1,])

            data=data.drop([0,])


      #    plt.plot(data['Sensor 1'])
```

```
#     plt.plot(data['Sensor 2'])

#     plt.plot(data['Sensor 3'])

#     plt.show()

 # here we calculate our features and fill the lists that we initiated above

 #sensor 1

        raw_data_sen1= data['Sensor 1'].to_numpy().astype(float)
        max_sen1.append(np.max(raw_data_sen1))
        mean_sen1.append(np.mean(raw_data_sen1))
        contact_counts_sen1.append(len(np.where(raw_data_sen1!=0)))
        non_zero_mean_calc=np.mean(raw_data_sen1[np.where(raw_data_sen1!=0)])
        if np.isnan(non_zero_mean_calc):
            non_zero_mean_sen1.append(0)
        else:
            non_zero_mean_sen1.append(non_zero_mean_calc)

        non_zero_median_calc=np.median(raw_data_sen1[np.where(raw_data_sen1!=0)])
        if np.isnan(non_zero_median_calc):
            non_zero_median_sen1.append(0)
        else:
            non_zero_median_sen1.append(non_zero_median_calc)

 #sensor 2

        raw_data_sen2= data['Sensor 2'].to_numpy().astype(float)
        max_sen2.append(np.max(raw_data_sen2))
        mean_sen2.append(np.mean(raw_data_sen2))
        contact_counts_sen2.append(len(np.where(raw_data_sen2!=0)))
        non_zero_mean_calc=np.mean(raw_data_sen2[np.where(raw_data_sen2!=0)])
        if np.isnan(non_zero_mean_calc):
            non_zero_mean_sen2.append(0)
        else:
            non_zero_mean_sen2.append(non_zero_mean_calc)

        non_zero_median_calc=np.median(raw_data_sen2[np.where(raw_data_sen2!=0)])
        if np.isnan(non_zero_median_calc):
            non_zero_median_sen2.append(0)
        else:
            non_zero_median_sen2.append(non_zero_median_calc)


#sensor 3
        raw_data_sen3= data['Sensor 3'].to_numpy().astype(float)
        max_sen3.append(np.max(raw_data_sen3))
        mean_sen3.append(np.mean(raw_data_sen3))
        contact_counts_sen3.append(len(np.where(raw_data_sen3!=0)))
        non_zero_mean_calc=np.mean(raw_data_sen3[np.where(raw_data_sen3!=0)])
        if np.isnan(non_zero_mean_calc):
            non_zero_mean_sen3.append(0)
        else:
            non_zero_mean_sen3.append(non_zero_mean_calc)

        non_zero_median_calc=np.median(raw_data_sen3[np.where(raw_data_sen3!=0)])
        if np.isnan(non_zero_median_calc):
            non_zero_median_sen3.append(0)
        else:
```

```
                non_zero_median_sen3.append(non_zero_median_calc)


        #reading filenames
                if datafile_names[count].find('hard')>0:
                    hard_soft_label.append(1)
                else:
                    hard_soft_label.append(2)

                if datafile_names[count].find('cone')>0:
                    geometry_label.append(3)
                elif datafile_names[count].find('cube')>0:
                    geometry_label.append(4)
                elif datafile_names[count].find('cylinder')>0:
                    geometry_label.append(5)
                elif datafile_names[count].find('prism')>0:
                    geometry_label.append(6)
                elif datafile_names[count].find('pyramid')>0:
                    geometry_label.append(7)
                else:
                     geometry_label.append(8)
                participant_label.append(count3)



print('max sen1=', max_sen1)
#print('contact counts sen1= ', contact_counts_sen1)
print('nonzeromean sen1=', non_zero_mean_sen1)
print('nonzeromedian sen1=', non_zero_median_sen1)
print('max sen2=', max_sen2)
#print('contact counts sen2=', contact_counts_sen2)
print('nonzeromean sen2=', non_zero_mean_sen2)
print('nonzeromedian sen2=', non_zero_median_sen2)
print('max sen3=', max_sen3)
#print('contact counts sen3=', contact_counts_sen3)
print('nonzeromean sen3=', non_zero_mean_sen3)
print('nonzeromedian sen3=', non_zero_median_sen3)


feature_frame=pd.DataFrame(list(zip( max_sen1, max_sen2, max_sen3, non_zero_mean_sen1,
non_zero_mean_sen2, non_zero_mean_sen3, non_zero_median_sen1, non_zero_median_sen2,
non_zero_median_sen3, geometry_label)),
                                columns=['Max value thumb', 'Max value index finger', 'Max value middle finger',
'Non-zero mean thumb', 'Non-zero mean index finger', 'Non-zero mean middle finger', 'Non-zero median thumb',
'Non-zero median index finger', 'Non-zero median middle finger', 'geometry_label'])

bp= feature_frame.boxplot(by='geometry_label', grid=True, fontsize=10, layout=(6,3),figsize=(10,18))


print(feature_frame.groupby(['geometry_label']).mean())


# for changing arangement of plot see documentaion:
https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.boxplot.html
```

Appendix B

Order test 1	Order test 2	Order test 3	Order test 4	Order test 5	Order test 6	Order test 7	Order test 8	Order test 9	Order test 10
Hard cube	Soft cone	Hard cube	Hard prism	Hard cylinder	Hard prism	Hard cone	Hard pyramid	Hard cube	Hard cube
Soft cube	Hard sphere	Hard sphere	Soft prism	Hard sphere	Soft cube	Hard prism	Hard cone	Hard cone	Hard sphere
Hard cylinder	Soft cylinder	Hard cylinder	Soft cylinder	Soft cube	Hard cone	Soft pyramid	Soft prism	Hard cylinder	Soft prism
Hard sphere	Hard pyramid	Hard pyramid	Hard pyramid	Hard pyramid	Soft sphere	Hard pyramid	Soft sphere	Hard pyramid	Hard pyramid
Hard pyramid	Soft cube	Hard cone	Hard cone	Soft pyramid	Hard pyramid	Hard cube	Soft cube	Hard sphere	Soft cylinder
Hard cone	Hard prism	Hard prism	Hard cube	Hard cube	Soft cylinder	Hard sphere	Hard prism	Hard prism	Soft cube
Soft cylinder	Hard cone	Soft cube	Soft cube	Soft sphere	Hard cylinder	Hard cylinder	Hard cube	Soft pyramid	Hard prism
Soft sphere	Soft sphere	Soft sphere	Soft sphere	Hard cone	Soft pyramid	Soft sphere	Soft pyramid	Soft sphere	Soft sphere
Soft prism	Hard cylinder	Soft cylinder	Hard sphere	Soft cylinder	Hard sphere	Soft cylinder	Soft cylinder	Soft prism	Hard cone
Hard prism	Soft pyramid	Soft pyramid	Soft pyramid	Hard prism	Soft prism	Soft prism	Hard sphere	Soft cube	Soft pyramid
Soft cone	Hard cube	Soft cone	Soft cone	Soft cone	Soft cone	Soft cone	Soft cone	Soft cone	Soft cone
Soft pyramid	Soft prism	Soft prism	Hard cylinder	Soft prism	Hard cube	Soft cube	Hard cylinder	Soft cylinder	Hard cylinder

TABLE 5 ORDERS OF THE OBJECTS FROM THE TESTS

Each question is coded from 0 to 2, with "left" receiving a score of 0 and "right" receiving a score of 2, and "both" receiving a score of 1.

The participants also received the following written instruction [11]:

"*A number of activities, in which you can use either your left or your right hand, are specified below. Indicate which hand you usually use for these activities. Visualize the activity in question if you are not immediately sure of an answer. If you don't have a clear preference, indicate that you use both hands*."

TABLE 6 HANDEDNESS TEST PARTICIPANT 1 [8]

1	Which hand do you use to hold scissors?	
2	With which hand do you draw?	
3	With which hand do you screw the top off a bottle?	
4	With which hand do you deal cards?	
5	Which hand do you use to hold a toothbrush when cleaning teeth?	
6	With which hand do you use a bottle opener?	
7	With which hand do you throw a ball away?	
8	Which hand do you use to hold a hammer?	
9	With which hand do you thread a needle?	
10	With which hand do you hold a racket when playing tennis?	
11	With which hand do you open the lid of a smart box?	
12	With which hand do you turn a key?	
13	With which hand do you cut a cord with a knife?	
14	With which hand do you stir with a spoon?	
15	With which hand do you use an eraser on paper?	
16	With which hand do you strike a match?	
	Total score	

Test participant 1 had a score of 32 and test participant 2 had a score of 29

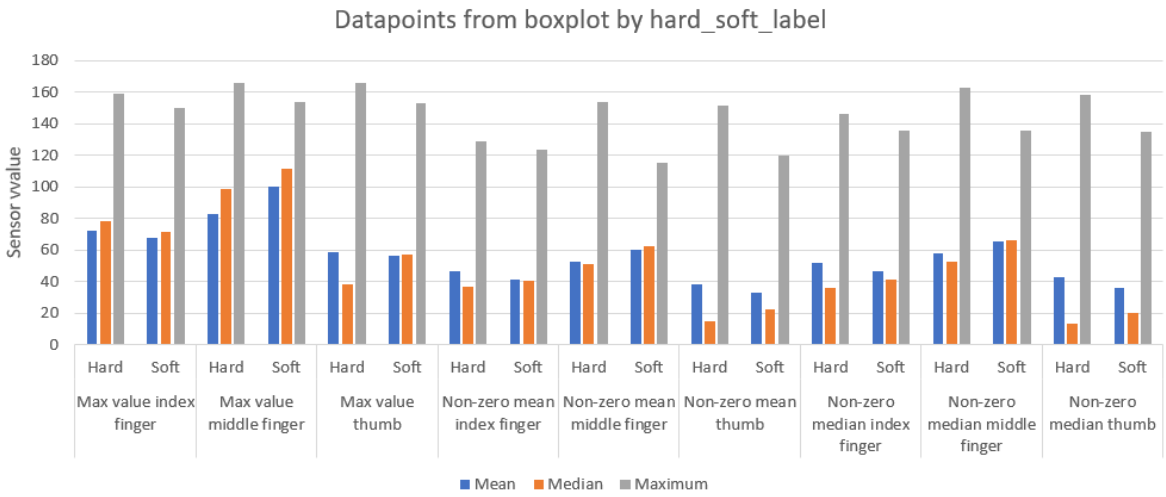

FIGURE 13 DATAPOINTS FROM BOXPLOT BY HARD_SOFT_LABEL

TABLE 7 CALCULATED DATAPOINTS FROM BOXPLOT BY GEOMETRY_LABEL

		Mean	Median	Maximum
Max value index finger	*Cone*	65,8	69,0	149,0
	Cube	69,2	60,0	151,0
	Cylinder	71,6	77,0	155,0
	Prism	70,7	73,5	156,0
	Pyramid	65,8	62,5	150,0
	Sphere	77,2	96,0	159,0
Max value middle finger	*Cone*	83,2	96,0	146,0
	Cube	106,6	123,0	164,0
	Cylinder	101,5	111,0	153,0
	Prism	72,0	86,0	148,0
	Pyramid	80,4	100,5	139,0
	Sphere	105,9	128,5	166,0
Max value thumb	*Cone*	57,3	49,0	152,0
	Cube	61,4	56,0	164,0
	Cylinder	54,1	29,5	157,0

	Prism	57,6	60,5	147,0
	Pyramid	49,7	16,5	151,0
	Sphere	63,8	51,5	166,0
Non-zero mean index finger	*Cone*	51,4	57,8	116,9
	Cube	68,2	71,2	146,7
	Cylinder	61,5	68,3	121,2
	Prism	42,5	43,8	113,7
	Pyramid	44,3	51,2	106,8
	Sphere	71,0	78,4	154,1
Non-zero mean middle finger	*Cone*	42,0	37,4	117,9
	Cube	41,4	39,4	123,3
	Cylinder	42,5	40,3	119,5
	Prism	47,1	39,0	128,7
	Pyramid	40,0	3,1	118,1
	Sphere	49,9	54,4	125,4
Non-zero mean thumb	*Cone*	36,3	17,7	121,8
	Cube	40,2	25,5	151,2
	Cylinder	32,6	15,0	129,0
	Prism	33,1	28,4	106,4
	Pyramid	29,2	7,6	102,8
	Sphere	43,0	20,2	143,9
Non-zero median index finger	*Cone*	46,6	41,8	134,0
	Cube	46,0	36,0	136,0
	Cylinder	46,5	38,0	146,0
	Prism	54,1	34,5	145,5
	Pyramid	45,1	33,0	139,0
	Sphere	56,5	59,3	138,0

Non-zero median middle finger	*Cone*	57,2	66,0	135,0
	Cube	75,9	78,0	157,0
	Cylinder	68,0	71,5	136,0
	Prism	46,6	40,0	134,0
	Pyramid	45,3	43,3	125,0
	Sphere	78,3	89,5	163,0
Non-zero median thumb	*Cone*	40,4	16,5	146,0
	Cube	44,6	21,0	158,0
	Cylinder	35,0	11,3	143,0
	Prism	36,8	29,8	130,5
	Pyramid	31,3	6,5	131,0
	Sphere	48,0	18,0	153,5

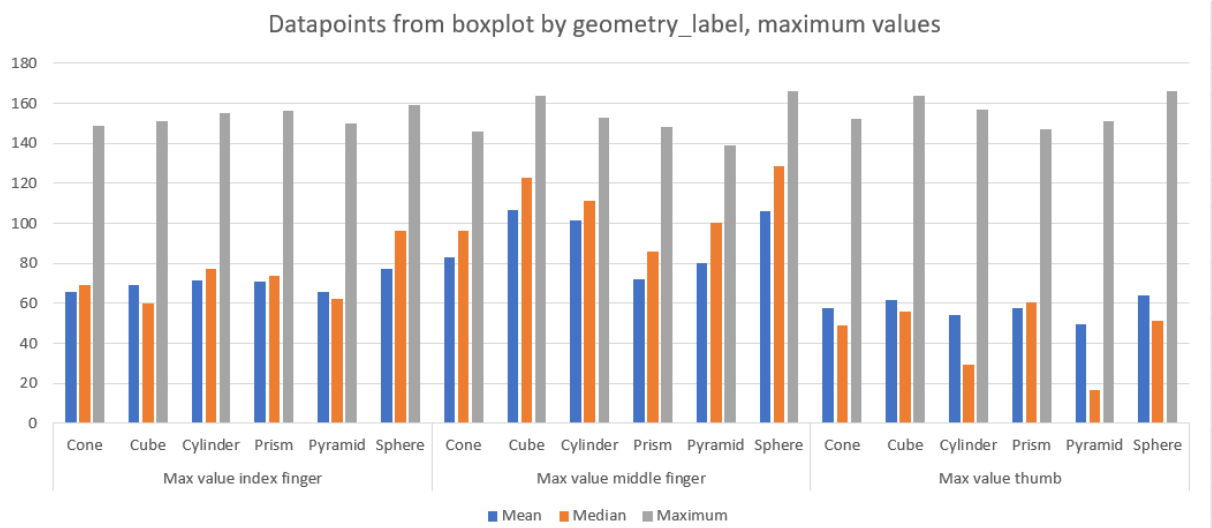

FIGURE 14 DATAPOINTS FROM BOXPLOT BY GEOMETRY_LABEL, MAXIMUM VALUES

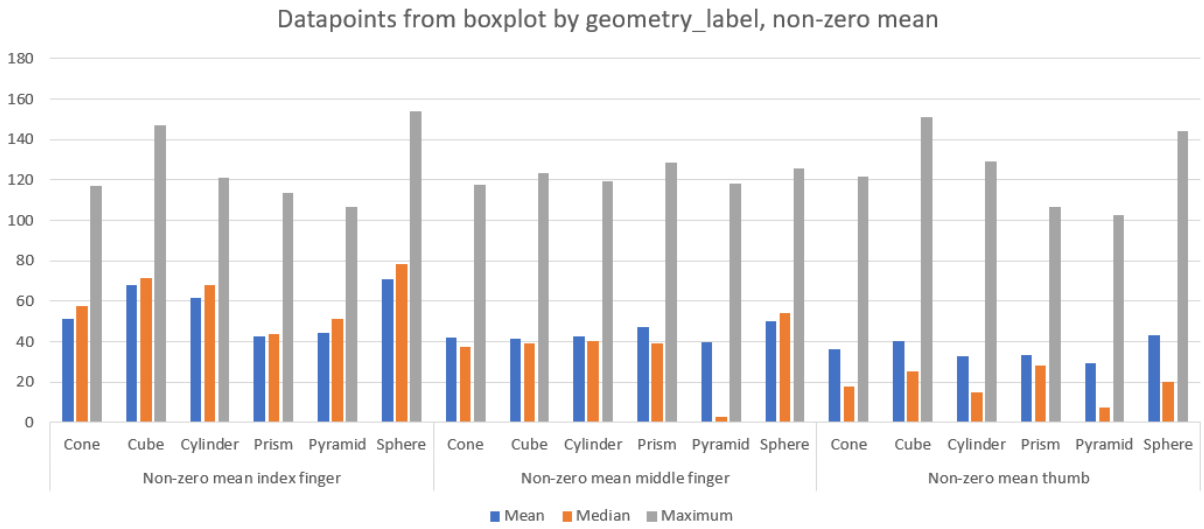

FIGURE 15 DATAPOINTS FROM BOXPLOT BY GEOMETRY_LABEL, NON-ZERO MEAN

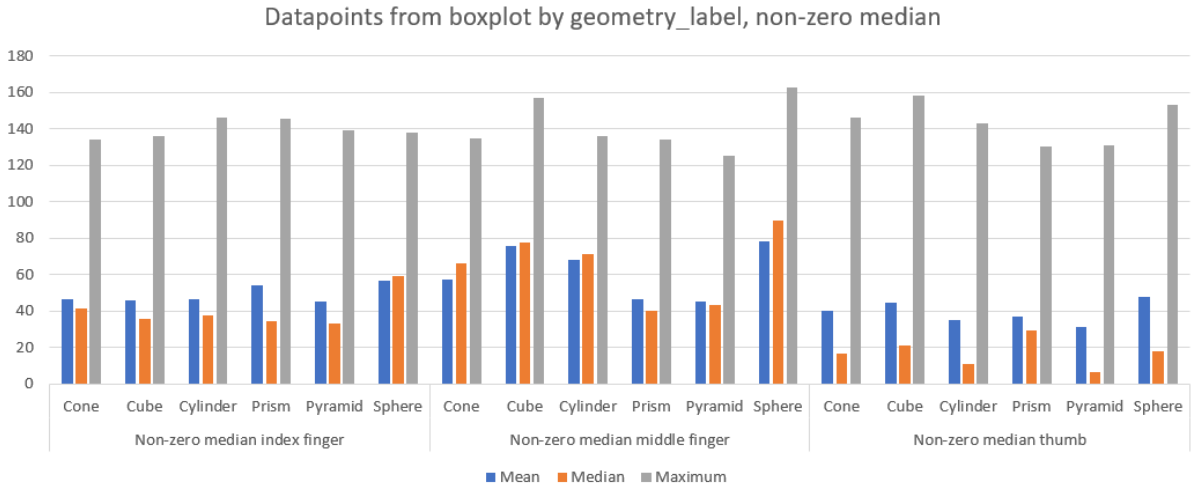

FIGURE 16 DATAPOINTS FROM BOXPLOT BY GEOMETRY_LABEL, NON-ZERO MEDIAN