

DEADLIFT ASSISTANT

By

Johan Mufuta

Nosa Egiebor

Sean Luo

Final Report for ECE 445, Senior Design, Fall 2019

TA: Jonathan Hoff

10 December 2019

Project No. 32

Abstract

Deadlift Assistant is a two-piece system designed to prevent gym-related injuries by providing accurate feedback on any user's weight-lifting technique. Using a combination of physical sensors and computer vision, the Deadlift Assistant takes multiple images of its user during a deadlift repetition and analyzes the user's body position in each photo to detect if the user had good or bad form during that phase of the lift. It then relays its analysis back to the user to tell them where they need to improve their form, if necessary.

Contents

1. Introduction	4
2 Design	5
2.1 Barbell Subsystem	6
2.1.1 Circuit Topology	7
2.1.2 Barbell Subsystem Requirements & Verifications	8
2.2. Computer Vision Subsystem	9
2.2.1 Computer Subsystem Functionality and Outputs	9
2.2.2 Computer Vision Subsystem Requirements & Verifications	11
3. Costs	12
3.1 Parts	12
3.2. Labor	12
3.3. Total Cost	12
3.4 Schedule	13
4. Conclusion	13
4.1 Accomplishments	13
4.2 Uncertainties	14
4.3 Ethical considerations	14
4.4 Future work	14
References	16
Appendix A Requirement and Verification Table	17
Appendix B Parts Costs Table	19
Appendix C Schedule Table	20

1. Introduction

Every year, over 500,000 gym-related injuries occur in the United States alone [1]. One of the primary causes of these injuries is poor weight-lifting technique. Two of the most common solutions to this problem that exist today are hiring personal trainers and looking to the internet for lifting technique advice.

The issues with the first solution are that personal trainers can be expensive and some gym-goers may feel uncomfortable with another person watching them exercise. Worse, some gym-goers may not even be aware that they have a poor lifting technique which may lead to even more unexpected injuries. In fact, we found that only 15% of gym-goers utilize the services of personal trainers [3].

The issue with looking to the internet for lifting technique advice is the lack of active feedback. A gym-goer may read as many articles and watch as many videos on the weight-lifting technique as he or she pleases, but that won't change the fact that they won't be receiving personalized feedback on their technique when are alone at the gym.

Deadlift Assistant is designed to be a solution that does not have the shortcomings of the previously mentioned technique-improvement options. By utilizing an array of sensors attached to the barbell in conjunction with a computer vision algorithm that tracks the body's joints and the angles of the joints' connecting body parts, Deadlift Assistant provides accurate and personalized feedback on lifting technique for any user during their deadlift routine. The device costs significantly less than hiring a personal trainer over time, and can be used by anyone with minimal additional learning curve beyond the standard deadlift technique. With proper use, Deadlift Assistant can teach correct deadlift technique and possibly prevent future injuries from happening.

2 Design

The design of this system consists of two subsystems: A Barbell Subsystem and a Computer Vision Subsystem. For this product to succeed, the two subsystems must be able to properly decipher the data they each collect and communicate with each other via Bluetooth. The Barbell Subsystem is attached to the user's barbell and the Computer Vision Subsystem is to be placed to the side of the user so that the angle of their back can be analyzed. Using the input and communication between these two devices will allow pictures to be taken at specific times during the lift and then the proper LEDs will light up signifying whether or not the good form was performed. The user will then be able to look to the Laptop to get more detailed feedback and analysis of their form from the Computer Vision algorithm.

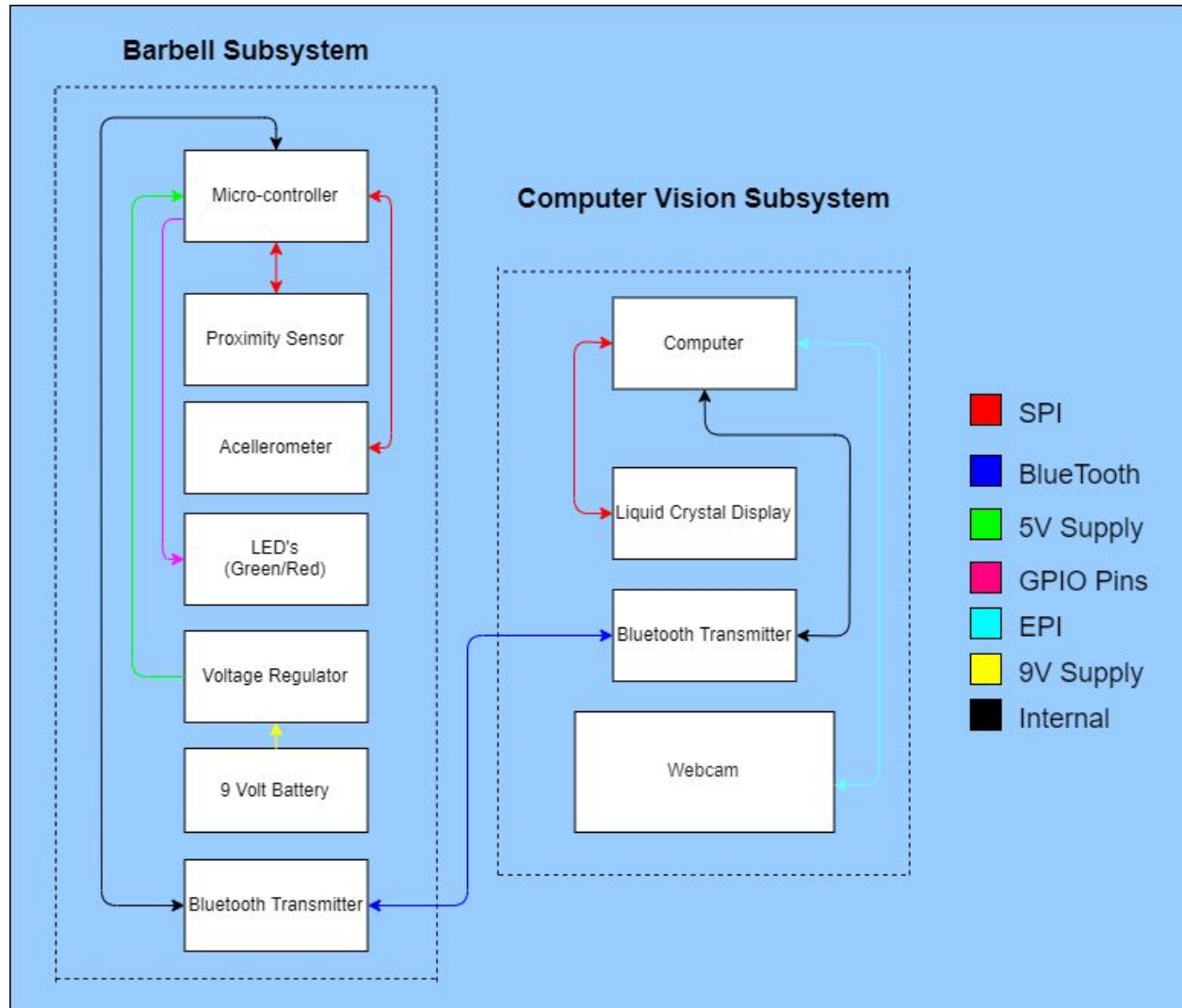


Figure 1: Deadlift Assistant Block Diagram

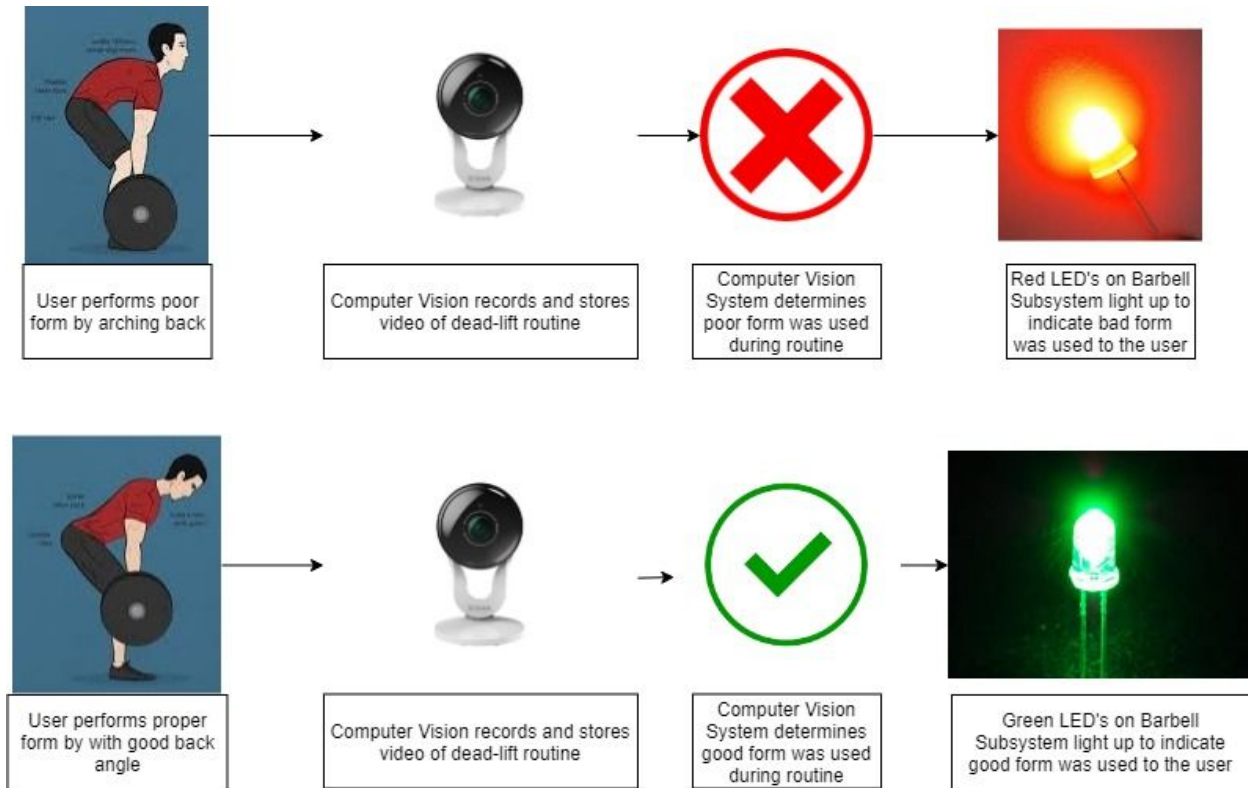


Figure 2: Visual Aid for Dead-lift Assistant Functionality

2.1 Barbell Subsystem

The Barbell Subsystem is the device that is physically attached to the barbell. Its purpose is to poll information from the barbell's movement and indicate to the Computer Vision (CV) Subsystem via Bluetooth connectivity which stage of the exercise the user is currently on. This is important to the system as a whole since it signals to the CV Subsystem when the user has the barbell on the ground, they are raising the bar, and they are at the peak of their workout. When these signals are relayed, the CV Subsystem will take a picture at each of these stages. After the routine is completed the CV Subsystem will relay back signals via Bluetooth Connectivity whether the routine was performed correctly and the appropriate LEDs will light up (Red/Green).

This subsystem needed the following components:

HiLetgo HC-05 Bluetooth Transmitter	Adafruit MMA8451 Accelerometer	Sharp IR GP2Y0A21YK0F Proximity Sensor
PoiLee 7805 Linear Regulator	16 MHz Crystal Oscillator	220 Ω Resistor (x4)
47 μ F Capacitor (x2)	22pF Capacitor (x2)	LEDs (x4)
9-volt Battery	9-volt Battery Adapter w/	1k Ω , 2k Ω , & 10k Ω Resistor

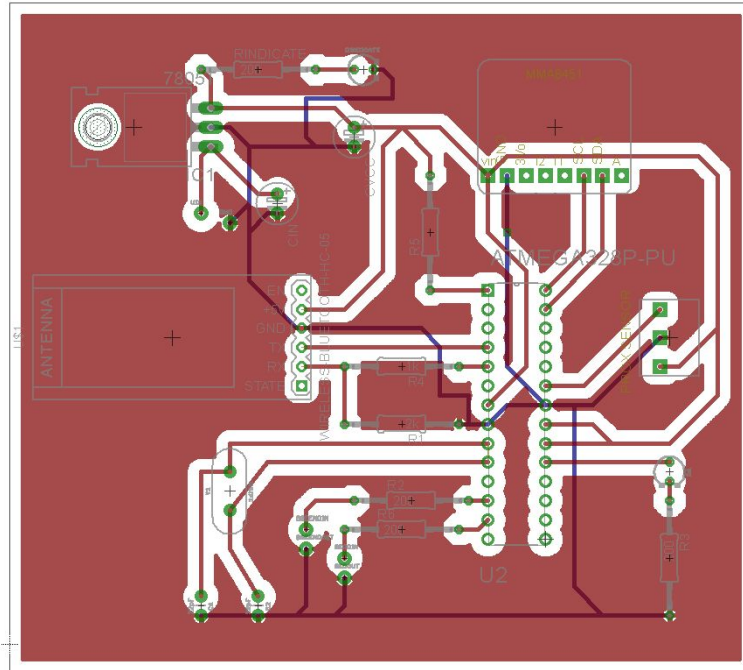


Figure 4: Barbell Subsystem PCB Layout

2.1.2 Barbell Subsystem Requirements & Verifications

In order for the barbell subsystem to work, it must be able to send and receive data via Bluetooth to and from the Computer Vision Subsystem, be able to tell its own orientation and motion, and light up the correct LEDs when the CV Algorithm determines the quality of the user's form. More specifics of this system are tabulated below in Table 2:

Requirement	Verifications
Barbell Subsystem connects to the laptop via Bluetooth	<ul style="list-style-type: none"> Port connection line in wrapper.py does not throw an error
The Barbell Module can tell what orientation the Barbell Module is currently positioned	<ul style="list-style-type: none"> The LEDs stay on when the module faces upward The LEDs turn off when the module faces downward
The Red LED lights up when the user arches their back during their lifting routine, indicating a poor form	<ul style="list-style-type: none"> Can see the Red LED turn on after photos are processed
The Green LED lights up when the user doesn't arch their back during their lifting routine, indicating the proper form	<ul style="list-style-type: none"> Can see the Green LED turn on after photos are processed

Barbell module can receive messages via Bluetooth from computer	<ul style="list-style-type: none"> • The correct LED turns on from computer vision technique analysis. Should match photo coloring
-----------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------

Table 2: Barbell Subsystem Requirements and Verifications

2.2 Computer Vision Subsystem

The Computer Vision Subsystem is a laptop computer with a webcam. The purpose of this subsystem is to capture images of the user and analyze these images using a custom computer vision algorithm.

2.2.1 Computer Subsystem Functionality and Outputs

This subsystem communicates with the Barbell Subsystem by both receiving and transmitting messages over Bluetooth. The Barbell Subsystem transmits a specific message depending on the phase of the deadlift that the user is currently in, either “gnd\n”, “up\n”, or “peak\n” which indicate that the user just started the lift, is on their way up or has reached the top of the deadlift motion respectively.

Depending on which message is received, the computer vision algorithm takes a photo and categorizes it as whatever phase the message indicates. It then sees where the joints of the user are located in the photo and computes the angle between the connected body parts. Finally, it compares the calculated angles to a professional weight-lifters joint angles at the same phase (the “ideal” model). If the user’s back joint angles are all within a range of 5 degrees of the ideal model’s respective joint angles, then the algorithm determines that the user’s technique was correct and draws green lines on the back. If not, it determines that the user’s technique was incorrect, and indicates where the mistakes were made by drawing red lines on the back. We put extra emphasis on the back joints, so we colored them differently. However, if another joint differed from the ideal model by 10 degrees, our algorithm would still indicate that an incorrect form was performed. (see Figure 5).

Taking the majority vote on the 3 images’ feedback that were captured throughout the lift, the computer vision algorithm then sends back a message of either “0” or “1” over Bluetooth to the Barbell Subsystem to indicate either bad or good form respectively.

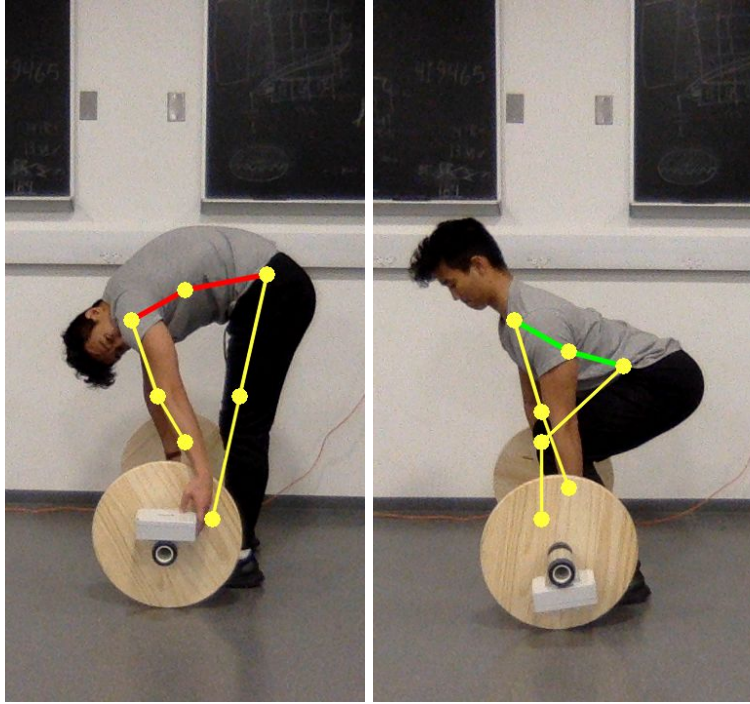


Figure 5: Computer Vision Examples

2.2.2 Computer Vision Subsystem Requirements & Verifications

Requirement	Verifications
The laptop connects to the Barbell Module via Bluetooth	<ul style="list-style-type: none">• Connection confirmed by the Bluetooth window on the displayed by the laptop• No error message from the port connection in wrapper.py
The Computer Vision Subsystem signals when it is ready to begin lift analysis	<ul style="list-style-type: none">• Unique terminal log indicating webcam and OpenCV are ready for use
The Computer Vision Subsystem signals when it is ready to capture the mid-lift technique	<ul style="list-style-type: none">• Unique terminal log indicating the initial “start” photo has already been taken
The Computer Vision Subsystem signals when it is ready to capture peak-lift technique	<ul style="list-style-type: none">• Unique terminal log indicating the “mid-lift” photo has already been taken
The Computer Vision Subsystem processes photos taken during the lift and sends feedback to the barbell subsystem	<ul style="list-style-type: none">• Can display photos from different stages of the lift• Red/green LED lights up on the barbell subsystem according to the processed feedback
The Computer Vision algorithm can detect if the user’s back is arched at any of the three recorded stages of the lift (Ground, Rising, Peak) and give back visual feedback of this detection	<ul style="list-style-type: none">• Output pictures show lines correlated to the user’s body parts• Lines are aligned with the user’s back joints and colored red in the case of bad form
The Computer Vision algorithm can detect if the user’s back is never arched at any of the three recorded stages of the lift (Ground, Rising, Peak) and give back visual feedback of this detection	<ul style="list-style-type: none">• Output pictures show lines correlated to the user’s body parts• Lines are aligned with the user’s back joints and colored green in the case of proper form

Table 3: Computer Vision Subsystem Requirements and Verifications

3. Costs and Schedule

3.1 Parts

Part	Manufacturer	Retail Cost (\$)	Bulk Purchase Cost (\$)	Actual Cost (\$)
ATMega328P-PU w/ Socket	Atmel	\$11.99	\$4/unit	\$4
MMA8451	Adafruit	\$7.95	\$7.95/unit	\$7.95
GP2Y0A21YK0F IR Proximity Sensor	Sharp	\$9.99	\$9.99	\$9.99
HC-05	HiLetgo	\$7.99	\$7.99/unit	\$7.99
7805 Voltage Regulator	PoiLee	\$5.59	\$.56/unit	\$.56
9-volt battery (x2)	Duracell	\$6.55	\$3.28/unit	\$3.28
9-volt battery holder	LAMPVPATH	\$6.38	\$3.19/unit	\$3.19
Assorted LEDs	eBoot	\$6.55	\$.07/unit	\$.28
47μF Capacitor	Hilitchi	\$15.55	\$.03/unit	\$.06
22 μF	""	""	\$.03/unit	\$.06
16 MHz Crystal Oscillator	uxcell	\$5.99	\$.60/unit	\$.60
PCB (x5)	PCBWay	\$27	\$5.40/unit	\$5.40
Resistors (x2 220Ω, x1 1kΩ)	ELEGOO	\$10.86	\$.10/unit	\$.40
Total		\$122.395	\$43.19/unit	\$43.73

Table 4: Parts Costs

3.2 Labor

Taking the average salaries of UIUC ECE undergraduates in 2016-17, \$71,166 and \$96,518 for electrical engineers and computer engineers respectively [10], we can estimate the development costs it would take to develop this product. With a 40-hour workweek, 52 weeks of the year worked, every hour of labor would cost roughly \$34.21 and \$46.40 for electrical and computer engineering new grads. Our team consisted of 1 electrical engineering undergrad and 2 computer engineering undergrads. With an average of 15 hours of work per week over the 16-week semester for each of us, the total labor cost estimate to develop this product is $(\$34.21 \times 15 \times 16) + (\$46.40 \times 15 \times 16 \times 2) = \$30,482.40$.

3.3 Total Cost

With the parts and labor combined, the total cost to develop this product is \$30,526.13.

3.4 Schedule

Week	Sean	Johan	Nosa
10/07	Clean up existing code	Formulate math for angle calculations	Order parts
10/14	Optimize code	Integrate joint-angle calculations in the algorithm	Determine how to build plastic casings
10/21	Learn how to work with RaspberryPi	Update code for I/O with hardware	Design and order PCB
10/28	Integrate hardware feedback into the algorithm	Integrate hardware feedback into the algorithm	Verify that hardware works with each other
11/04	Define specific values for joint angles in the algorithm	Define specific values for joint angles in the algorithm	solder parts at a time for modularity
11/11	Initial testing in a controlled setting	Initial testing in a controlled setting	Initial testing in a controlled setting
11/18	Testing in the lab	Testing in the lab	Testing in the lab
11/25	Algorithm optimization	Algorithm optimization	Testing in the lab

Table 5: Schedule

4. Conclusion

4.1 Accomplishments

We have successfully developed a low cost (< \$150) deadlift assistant that can provide real-time feedback. Based on the user's deadlift form, the deadlift assistant will provide a green or red light with pictures as evidence.

4.2 Uncertainties

Even though completing this project was a tremendous success, there are few things that could be improved:

- The current flow of our algorithm does not automatically switch from analyzing left joints (left shoulder, left knee, etc) to the right joints. We have separate blocks of codes for the two scenarios and we have to manually load them.
- The video feedback is not as fast as the picture one. This is the main reason why we had to scrap it.

4.3 Ethical considerations

Our project was conducted in compliance with the IEEE Code of Ethics[5]. Due to the fact that our project provides feedback on the user's form, honesty in stating claims and making calculations was a goal of the utmost importance. We are glad that we successfully followed the third article of the IEEE Code of Ethics.

Article 3 of the IEEE Code of ethics was not the only concern we had to cover: We do not intend on having an individual hurt by using our project as that is a direct violation of #9 of the IEEE Code of ethics. This is the reason why positive feedback only comes when the user's back angle is within a range of 5° compared to the professional trainer. This allows the user's back to always be on the safe side.

4.4 Future work

There are a lot of improvements and additions that we can make to this product moving forward. First and foremost, the Barbell Subsystem ended up weighing roughly 2 lbs because it was attached to the barbell via a large metal clamp. It was contained in a plastic box with 6in. x 8 in. x 3.5 in. dimensions. We would like to reduce the size and weight of this device as much as possible by designing a more compact PCB, casing, and attachment to reduce its impact on the deadlift workout as much as possible.

Additionally, we would like to expand our product to other workouts such as the bench press. Theoretically, all this would require is for us to run our algorithm again on professionals performing these workouts and store the "proper" angles at each phase of the motions again, just like we did for the deadlift. After this, users would be able to perform whatever workouts we have joint-angles stored for, and compare their angles to that of the professionals.

Lastly, moving forward, we could even integrate our entire product in software. Writing a more advanced algorithm that can detect when a workout motion begins and ends without the use of the

sensors we had in our barbell subsystem would open endless possibilities for our product. We could move to exercises beyond the weight room such as shooting a basketball. If we are able to recreate our product using only software, we would be able to create mobile apps with the same idea and make our product even more accessible to a potential audience.

These steps would entice us to do industry analysis and customer research to discover any competitors within this market. That way, we can tailor the functionality of this system to best fit our target market's needs as well as help create a pricing model for our product to optimize sales and profit.

References

- [1] Consumer Product Safety Commission, "NEISS Data Highlights," *National Electronic Injury Surveillance System*, July 2018. [Online]. Available: https://www.cpsc.gov/s3fs-public/2018%20Neiss%20data%20highlights.pdf?NGBgKPVgISz.ShB_W6Y0ENRjhr6qAtWr. [Accessed Oct. 3, 2019].
- [2] J. McCarron, "Back Injuries From Deadlift," *Livestrong*, September 2017. [Online]. Available: <https://www.livestrong.com/article/497414-back-injuries-from-deadlifts/>. [Accessed Oct. 3, 2019].
- [3] M. Aiello, "Personal Training Usage Continues to Grow Among Health Club Members," *International Health, Racquet and Sportsclub Association*, June 2016. [Online]. Available: <https://www.ihrsa.org/improve-your-club/personal-training-usage-continues-to-grow-among-health-club-members/>. [Accessed Oct. 3, 2019]
- [4] CMU Perceptual Computing Lab, "OpenPose: Multi-Person Pose Estimation," *People Image Analysis Consortium*, May 2019. [Online]. Available: <http://www.consortium.ricmu.edu/projOpenPose.php>. [Accessed Oct. 3, 2019].
- [5] IEEE, "IEEE Code of Ethics," *IEEE Policies*, Section 7, no. 8, June 2019. [Online]. Available: <https://www.ieee.org/about/corporate/governance/p7-8.html>. [Accessed Oct. 3, 2019].
- [6] American Chiropractic Association, "Back Pain Facts and Statistics," *American Chiropractic Association*, May 2016. [Online]. Available: <https://www.acatoday.org/Patients/Health-Wellness-Information/Back-Pain-Facts-and-statistics>. [Accessed Oct. 3, 2019].
- [7] Z. Cao, T. Simon, S. Wei, and Y. Sheikh, "Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields," *Cornell University*, November 2016. [Online]. Available: <https://arxiv.org/abs/1611.08050>. [Accessed Oct. 3, 2019].
- [8] Vikas Gupta, "Pose Detection Comparison: wrnchAI vs OpenPose", Learn OpenCV, February 2019. [Online]. Available: <https://www.learnopencv.com/pose-detection-comparison-wrnchai-vs-openpose/>
- [9] Zhe Cao, "OpenPose: Realtime Multi-Person 2D Pose Estimation Using Part Affinity Field", IEEE Transactions on Pattern Analysis, May 2019.[Online]. Available: <https://arxiv.org/pdf/1812.08008.pdf>
- [10] Engineering Career Services, "Salary Averages:: ECE ILLINOIS", December 1, 2019. [Online]. Available: <https://ece.illinois.edu/admissions/why-ece/salary-averages.asp>. [Accessed Dec. 10, 2019].

Appendix A Requirement and Verification Tables

Requirement	Verifications
Barbell Subsystem connects to the laptop via Bluetooth	<ul style="list-style-type: none"> Port connection line in wrapper.py does not throw an error
The Barbell Module can tell what orientation the Barbell Module is currently positioned	<ul style="list-style-type: none"> The LEDs stay on when the module faces upward The LEDs turn off when the module faces downward
The Red LED lights up when the user arches their back during their lifting routine, indicating a poor form	<ul style="list-style-type: none"> Can see the Red LED turn on after photos are processed
The Green LED lights up when the user doesn't arch their back during their lifting routine, indicating the proper form	<ul style="list-style-type: none"> Can see the Green LED turn on after photos are processed
Barbell module can receive messages via Bluetooth from computer	<ul style="list-style-type: none"> The correct LED turns on from computer vision technique analysis. Should match photo coloring

Table 2: Barbell Subsystem Requirements and Verifications

Requirement	Verifications
The laptop connects to the Barbell Module via Bluetooth	<ul style="list-style-type: none"> Connection confirmed by the Bluetooth window on the displayed by the laptop No error message from the port connection in wrapper.py
The Computer Vision Subsystem signals when it is ready to begin lift analysis	<ul style="list-style-type: none"> Unique terminal log indicating webcam and OpenCV are ready for use
The Computer Vision Subsystem signals when it is ready to capture the mid-lift technique	<ul style="list-style-type: none"> Unique terminal log indicating the initial "start" photo has already been taken
The Computer Vision Subsystem signals when it is ready to capture peak-lift technique	<ul style="list-style-type: none"> Unique terminal log indicating the "mid-lift" photo has already been taken
The Computer Vision Subsystem processes photos taken during the lift and sends feedback to the barbell subsystem	<ul style="list-style-type: none"> Can display photos from different stages of the lift Red/green LED lights up on the barbell subsystem according to the processed feedback

<p>The Computer Vision algorithm can detect if the user's back is arched at any of the three recorded stages of the lift (Ground, Rising, Peak) and give back visual feedback of this detection</p>	<ul style="list-style-type: none"> • Output pictures show lines correlated to the user's body parts • Lines are aligned with the user's back joints and colored red in the case of bad form
<p>The Computer Vision algorithm can detect if the user's back is never arched at any of the three recorded stages of the lift (Ground, Rising, Peak) and give back visual feedback of this detection</p>	<ul style="list-style-type: none"> • Output pictures show lines correlated to the user's body parts • Lines are aligned with the user's back joints and colored green in the case of proper form

Table 3: Computer Vision Subsystem Requirements and Verifications

Appendix B Parts Costs Table

Part	Manufacturer	Retail Cost (\$)	Bulk Purchase Cost (\$)	Actual Cost (\$)
ATMega328P-PU w/ Socket	Atmel	\$11.99	\$4/unit	\$4
MMA8451	Adafruit	\$7.95	\$7.95/unit	\$7.95
GP2Y0A21YK0F IR Proximity Sensor	Sharp	\$9.99	\$9.99	\$9.99
HC-05	HiLetgo	\$7.99	\$7.99/unit	\$7.99
7805 Voltage Regulator	PoiLee	\$5.59	\$.56/unit	\$.56
9-volt battery (x2)	Duracell	\$6.55	\$3.28/unit	\$3.28
9-volt battery holder	LAMPVPATH	\$6.38	\$3.19/unit	\$3.19
Assorted LEDs	eBoot	\$6.55	\$.07/unit	\$.28
47 μ F Capacitor	Hilitchi	\$15.55	\$.03/unit	\$.06
22 μ F	""	""	\$.03/unit	\$.06
16 MHz Crystal Oscillator	uxcell	\$5.99	\$.60/unit	\$.60
PCB (x5)	PCBWay	\$27	\$5.40/unit	\$5.40
Resistors (x2 220 Ω , x1 1k Ω)	ELEGOO	\$10.86	\$.10/unit	\$.40
Total		\$122.395	\$43.19/unit	\$43.73

Table 4: Parts Costs

Appendix C Schedule Table

Week	Sean	Johan	Nosa
10/07	Clean up existing code	Formulate math for angle calculations	Order parts
10/14	Optimize code	Integrate joint-angle calculations in the algorithm	Determine how to build plastic casings
10/21	Learn how to work with RaspberryPi	Update code for I/O with hardware	Design and order PCB
10/28	Integrate hardware feedback into the algorithm	Integrate hardware feedback into the algorithm	Verify that hardware works with each other
11/04	Define specific values for joint angles in the algorithm	Define specific values for joint angles in the algorithm	solder parts at a time for modularity
11/11	Initial testing in a controlled setting	Initial testing in a controlled setting	Initial testing in a controlled setting
11/18	Testing in the lab	Testing in the lab	Testing in the lab
11/25	Algorithm optimization	Algorithm optimization	Testing in the lab

Table 5: Schedule