Deadlift Assistant

ECE 445 Design Document

Nosa Egiebor, Sean Luo and Johan Mufata
Group 32
TA: Jonathan Hoff
10/3/2019

1 Introduction

1.1 Problem and Solution Overview

Physical fitness is a fundamental aspect of life for a large percentage of the population. Working out is one of the ways people strive to become the best version of themselves. Even though there are several upsides to staying in shape, there are also several risks that cannot be ignored. Every year, nearly 500,000 people get injured in the gym while performing exercises. A large contributor to these injuries is poor technique and form [1]. The deadlift exercise, in particular, while being one of the most useful exercises for building functional strength, is also one of the most dangerous. This is due to the large amounts of weight and the sensitive muscles that are involved in the motion. One of the most common deadlift technique mistakes is arching your back, making your spine vulnerable to an abnormal and unsafe amount of pressure. This simple mistake can lead to severe injuries that can cost thousands of dollars in medical services to rehabilitate. A reliable technique-feedback system could save thousands of people from injury and medical costs.

Our solution to this problem is a computer-vision technique analysis system. Our system records video of a user performing a workout and provides feedback on their technique via green and red LEDs that indicate good and bad form respectively based on the angles detected between the user's joints during the lift. Our system uses a camera, accelerometer, proximity sensor, and computer vision algorithms to provide accurate feedback. This product is intended to be installed in gym equipment so that anyone can integrate it seamlessly into their workout without having to put on any physical sensors.

1.2 Background

There are two methods people typically use to learn and correct their form when lifting. One is hiring a physical trainer, and the other is looking to the internet for advice. Both have their pros and cons.

In regards to personal trainers, they can definitely be a useful asset to a member of a gym who is looking to exercise safely and use good form. However, the majority of people who go to the gym are not interested in hiring a trainer due to not being comfortable with the personal interaction or not wanting to pay more on top of a gym membership. As a result, only 15% of people with gym memberships use personal training services [3].

When it comes to looking to the internet for advice, there is a vast amount of information a person who is looking for workout advice can look at. However, the problem here is that there are a lot of varying opinions between these resources of what constitutes good form. In addition, using the internet places the responsibility of learning proper form entirely on the user. This leaves a lot of uncertainty due to the lack of active and personalized feedback for the user.

Our solution is to create a system that can give accurate analysis and accurate feedback of the user's deadlift form to ensure optimal safety to the user when they are lifting. This requires a clear cut and consistent guideline for good deadlifting form which will be extracted from professional sources and compiled into our system. We believe that such a system would provide sufficient value to the consumer to be either purchased by the individual or to be purchased by gyms to be implemented into their equipment. This system would minimize a gym member's chance of getting injured, and provide benefit to both the users and the gyms they go to via decreased injury rate and increased customer traffic.

1.3 Visual Aid

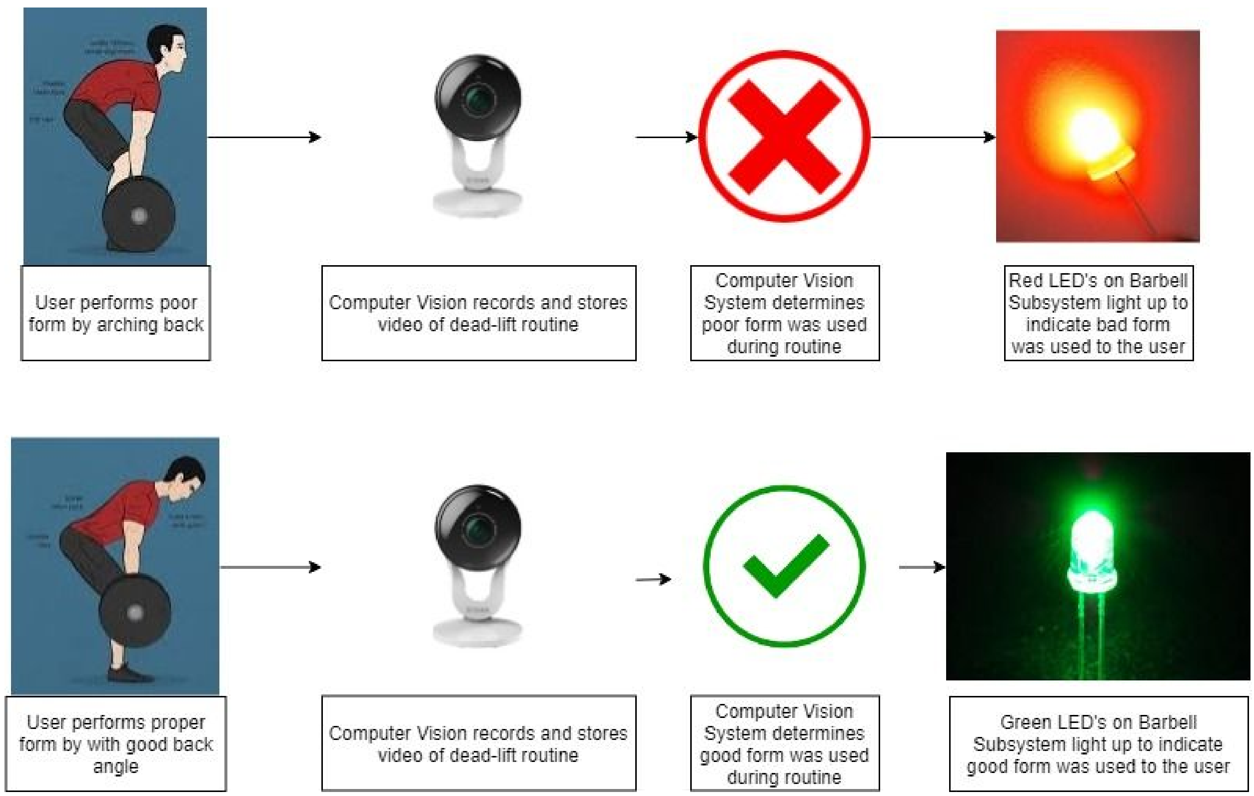


Figure 1: Visual Aid for Dead-lift Assistant Functionality

1.4 High-Level Requirements List

1. The user should be able to receive a feedback based on the angle of his/her back at every position during the lift. The feedback system will be a simple red or green LED to indicate bad or good form.
2. The accelerometer and proximity sensor should be able to convey barbell start and stop motion to the computer vision subsystem within 0.5 seconds.
3. Workout technique analysis is provided within 1 minute of exercise completion.

4. The camera module should be to identify the different joints of the body participating in the exercise.

2 Design

2.1 Block Diagram

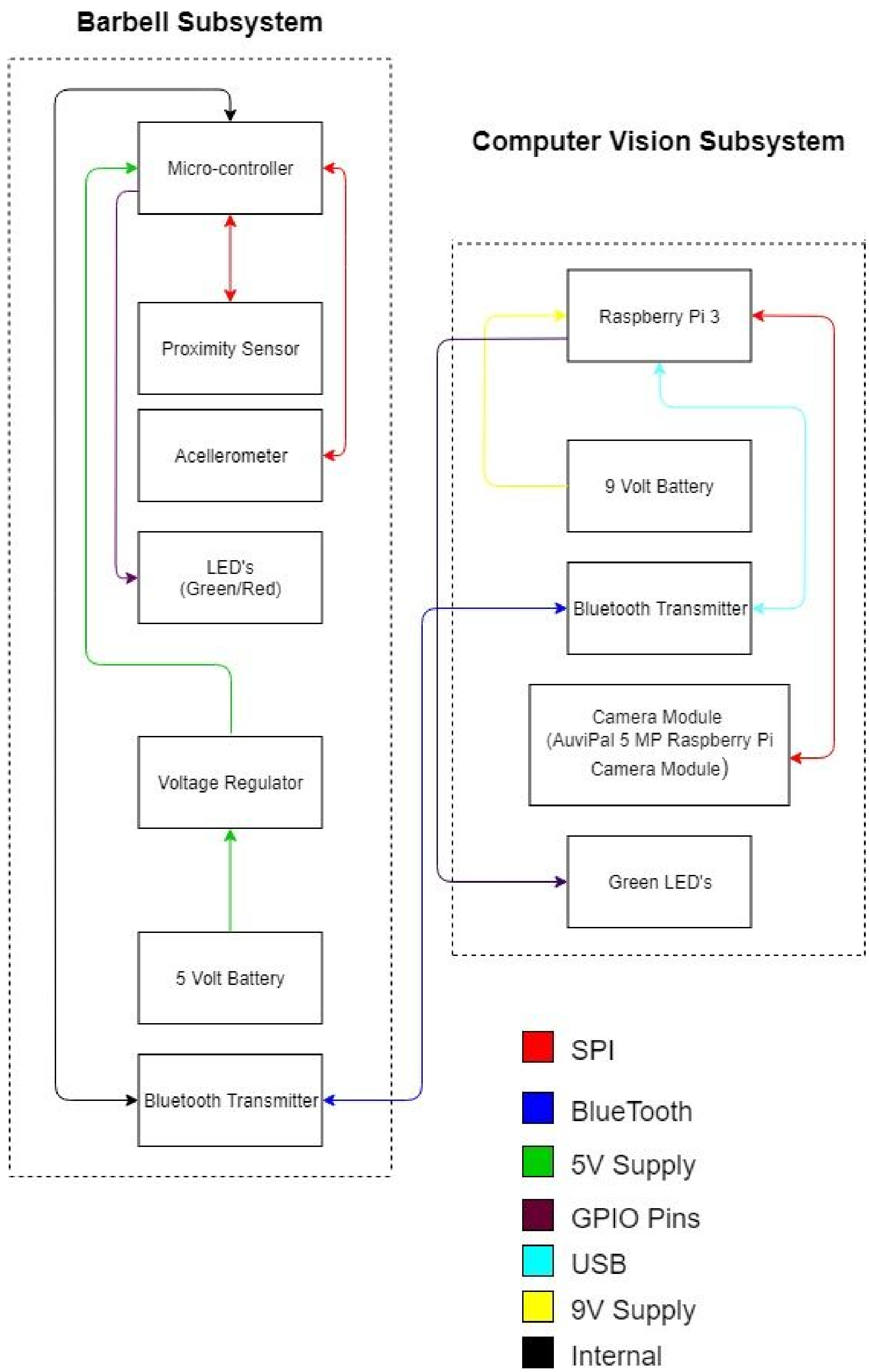


Figure 2: Deadlift Assistant Block Diagram

2.3 Physical Design

The physical design of the project will consist of two devices, one for each subsystem (see Figure 3).

The Barbell Subsystem will be held in a plastic, waterproof case to protect the module from sweat and moisture. Within the case are all the components of this subsystem. The casing contains holes on the bottom to allow the proximity sensor to detect the ground. The casing is held to the barbell via a clamp that wraps around the barbell snugly.

The other physical device is the Computer Vision Subsystem. Similar to the Barbell Subsystem, this device has all of its components contained within a see-through plastic container. It is propped up by a four-foot tall stand which is used to regulate the height at which video is recorded for all users.

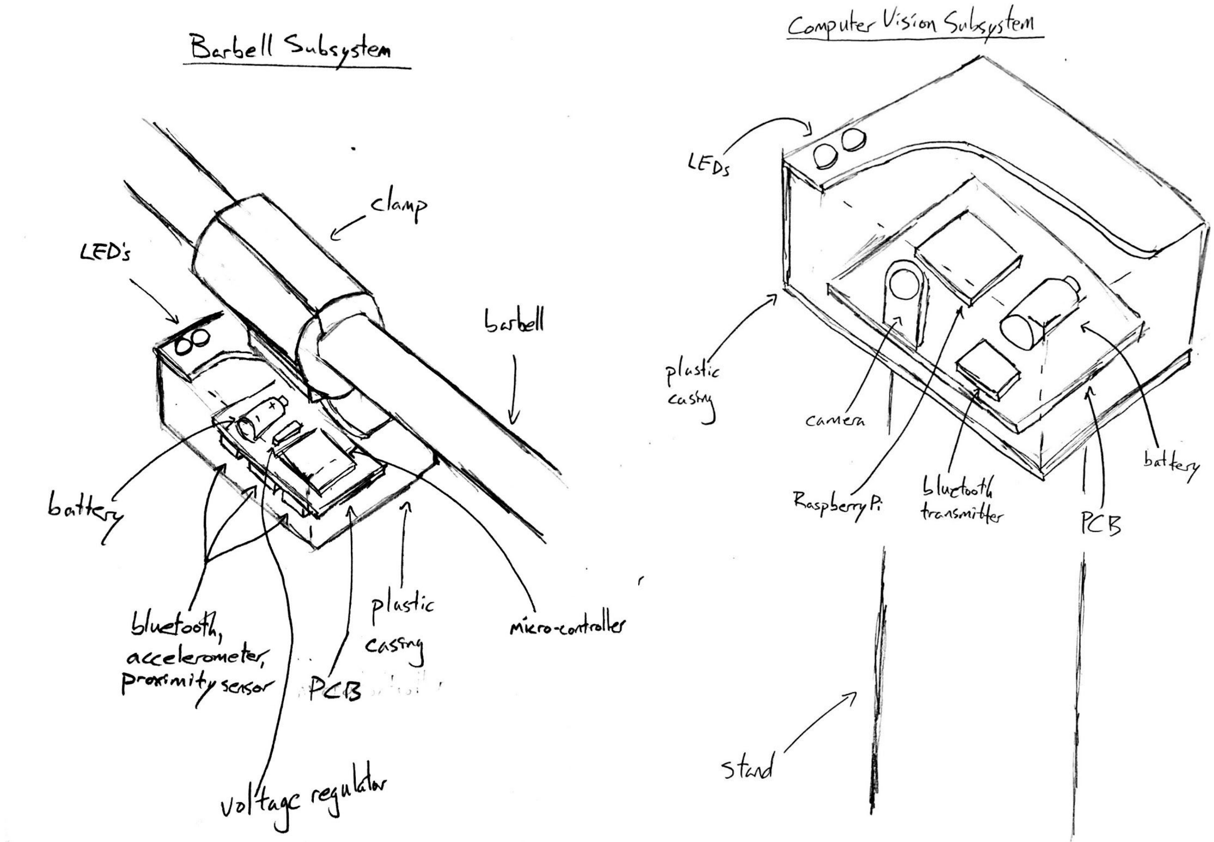


Figure 3: Physical Designs

2.4 [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 part of the exercise the user is currently on. This is important to the success of the project since it tells the Computer Vision Subsystem when the routine has begun, when the barbell is being raised, when the barbell is being lowered, and when the routine is finished. 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)

2.4.1 Proximity Sensor

The sections of the routine will be segmented by the coordination of the accelerometer and proximity sensors. The proximity sensor will be connected to the microcontroller chip through SPI and will indicate to the CV system when the routine has started and stopped. Our proximity sensor signals to our algorithm which phase of the lift the user is in. Because the deadlift motion is typically completed in under 5 seconds, we need the proximity sensor to be able to start and stop the body tracking algorithm within this time frame.

Requirements	**Verifications**
1. Relays signal to the CV Subsystem is <= .5s	1. LEDs light up <= 1s after barbell lifted and LEDs shut off <= 1s after barbell dropped and routine finishes

Table 1: Requirements and Verifications for the Proximity Sensor

2.4.2 Accelerometer

The accelerometer then tells us the direction the barbell is moving. This is important because the movement and anticipated angles of a person's back is different during these two motions. So, when the accelerometer on the barbell is being raised, the CV will do one algorithm to determine proper back angles, and when the accelerometer is being lowered The CV Subsystem then critiques the user's joints' angles depending on if the current motion of the exercise is raising and lowering the bar. The accelerometer communicates with the RaspberryPi in the Computer Vision subsystem via bluetooth transmitters.

Requirements	**Verifications**
1. Relays signal to the CV Subsystem is <= .5s	1. LEDs stay lit during routine 2. Check output signal and information to the Raspberry Pi 3
2. Can log when the motion changes so that it can be synced with the computer vision	1. Check output signal and information to the Raspberry Pi 3

Table 2: Requirements and Verifications for the Accelerometer

2.4.3 Voltage Regulator

Since we're using a microcontroller chip, we need to make sure the voltage rating of the chip is not violated, preventing the chip from being damaged while we're supplying power to it.

Requirements	**Verifications**
1. Can take a battery of 9V and can supply a constant voltage in the range of [4.9V, 5.1V] to the microcontroller	1. Use a voltmeter to ensure that the input voltage of the regulator is >5.1V. The input voltage range isn't all that important since the regulator will keep the output steady 2. Use a voltmeter to make sure the regulator doesn't surpass microcontroller rating of 5.1V or drop below 4.9V when turned on
3. Can connect to the power pin of the microcontroller	1. Use a voltmeter to check that the voltage is 5V on the input and 5V on the

	output

Table 3: Requirements and Verifications for the Voltage Regulator

2.4.4 Microcontroller chip

The ATMega328P-PU is responsible for coordinating the different signals coming from the sensors through SPI (Serial Peripheral Interface) and send those signals via Bluetooth connectivity to the Computer Vision Subsystem. It will be powered by the 5V regulator to ensure it is receiving a constant and safe voltage during operation. It must not receive or give out more than 40mA to also ensure its ratings are not violated. It will be programmed to send proximity sensor and accelerometer data and receive signals from the CV Subsystem via Bluetooth, so it must also have internal Bluetooth capabilities that are compatible with the Raspberry Pi 3 - Model B.

Requirements	**Verifications**
1. Bluetooth transmitter is compatible with Raspberry Pi 3 - Model B	1. Analyze Raspberry Pi code to see if it is receiving accelerometer and proximity sensor signals from the microcontroller chip
2. Processes start and stop signals from sensors and send them to CV subsystem in <= .5s	1. Run a timer to track CV subsystem LEDs that indicate algorithm start and stop

Table 5: Requirements and Verifications for the MicroController Chip

2.5 [Computer Vision Subsystem]

Our computer vision subsystem consists of a BlueTooth transmitter, a 9V battery, a camera module, green LEDs, and a RaspberryPi running our body tracking algorithm. It is the brain of our product that processes all of the data we are working with. While the camera module is recording the deadlift routine, the LEDs on the Computer Vision Subsystem will be lit. After this subsystem processes the Computer Vision Algorithms and if the user has performed good or bad form, it will relay back to the Barbell System to light up the appropriate LEDs on it.

2.5.1 Bluetooth transmitter

The BlueTooth transmitter here receives data from the barbell subsystem's equivalent transmitter. The measurements from the accelerometer and proximity sensor are received from the barbell subsystem through this bluetooth communication which will tell the computer vision algorithm which phase of the workout the user is currently in. Once the workout motion is complete, this transmitter sends information back to the barbell subsystem's transmitter to indicate whether or not the user had good form in their workout.

Requirements	**Verifications**
1. Receives information from barbell subsystem in <=.5s of the lift beginning	1. "Recording" LED lights up when barbell begins moving 2. Proper signals are relayed to Raspberry Pi 3 output
2. Relays information to the barbell subsystem after the algorithm is finished	1. Correct form indicator LED lights up on barbell subsystem after deadlift motion is complete 2. Proper signals are relayed to Raspberry Pi 3 output

Table 6: Requirements and Verifications for the BlueTooth Transmitter

2.5.2 9V Battery

The battery here provides power to the entire subsystem.

Requirements	**Verifications**
1. Battery provides power for LEDs, RaspberryPi, Bluetooth transmitter, and camera simultaneously	1. Power indicators on RaspberryPi light up, "recording" LED lights up during the workout. This indicates that bluetooth transmission works 2. Check camera by testing algorithm outputs. If good and bad form can be differentiated, this indicates that the camera was able to capture video input

Table 7: Requirements and Verifications for the 9-volt Battery

2.5.3 Camera Module

The camera module is used to record video of the user while they are performing the deadlift exercise. It starts and stops recording video according to the data received by this subsystem's BlueTooth transmitter. We are using a 5-megapixel camera. This quality of the video is high enough to allow our computer vision algorithm to clearly recognize the user's body in the video and is low enough to make sure that we don't have too much video processing overhead.

Requirements	**Verifications**
1. Begins recording at deadlift movement start	1. Recording LED lights up at the start of workout motion 2. Film starts when the proximity sensor is above the baseline height
2. Stops recording at deadlift movement end	1. Recording LED turns off at the end of workout motion 2. Film starts when the proximity sensor is back at the baseline height

Table 8: Requirements and Verifications for the Camera Module

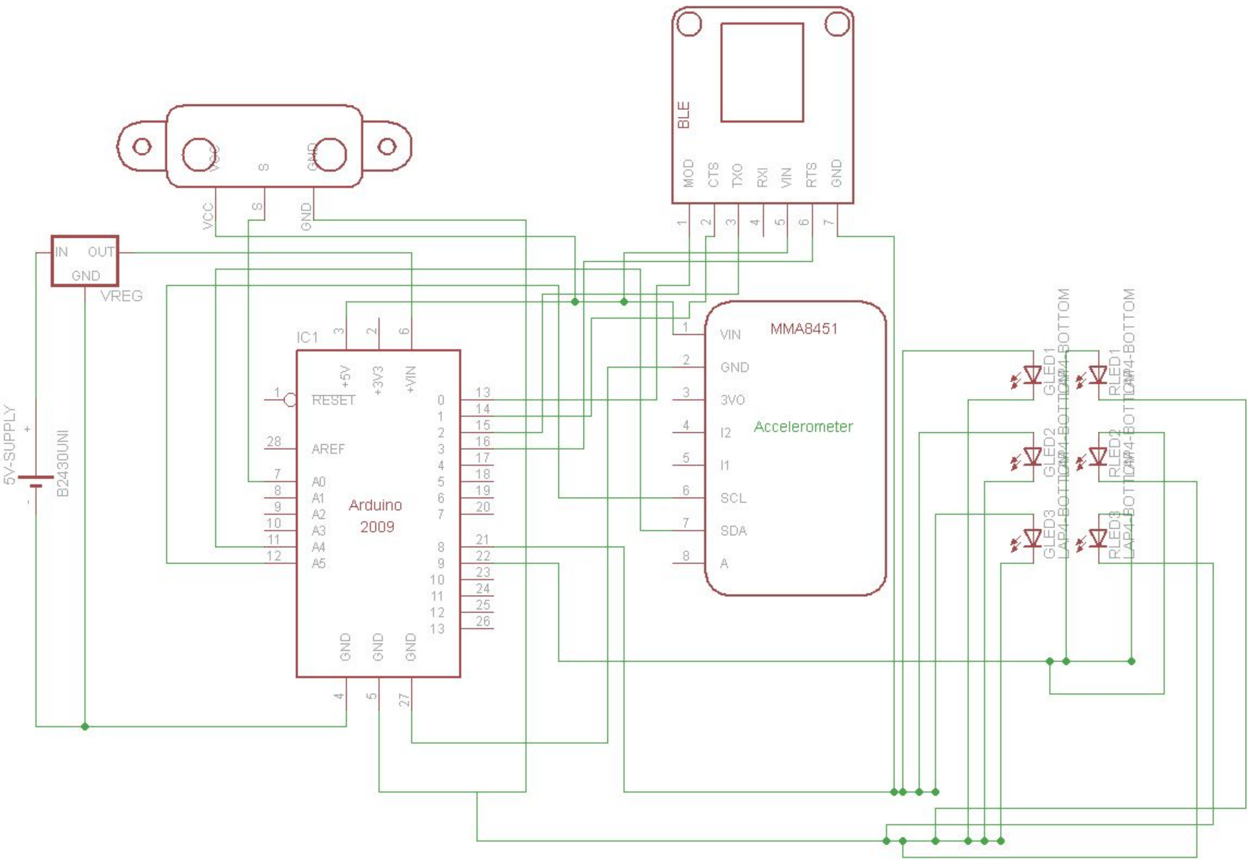
2.5.4 RaspberryPi 3

The RaspberryPi is used to process all of the data received from the barbell subsystem as well as the camera module. It runs a body-tracking algorithm, multi-person pose estimation, developed by the Perceptual Computing Lab at Carnegie Mellon University [7] to track the user while they are performing the exercise based on the data received from the barbell subsystem and the camera module. Our modified version of the algorithm also calculates the angles at every joint in the model in order to provide deadlift technique feedback. The RaspberryPi takes the output of our algorithm and sends it back to the barbell subsystem via the BlueTooth transmitters.

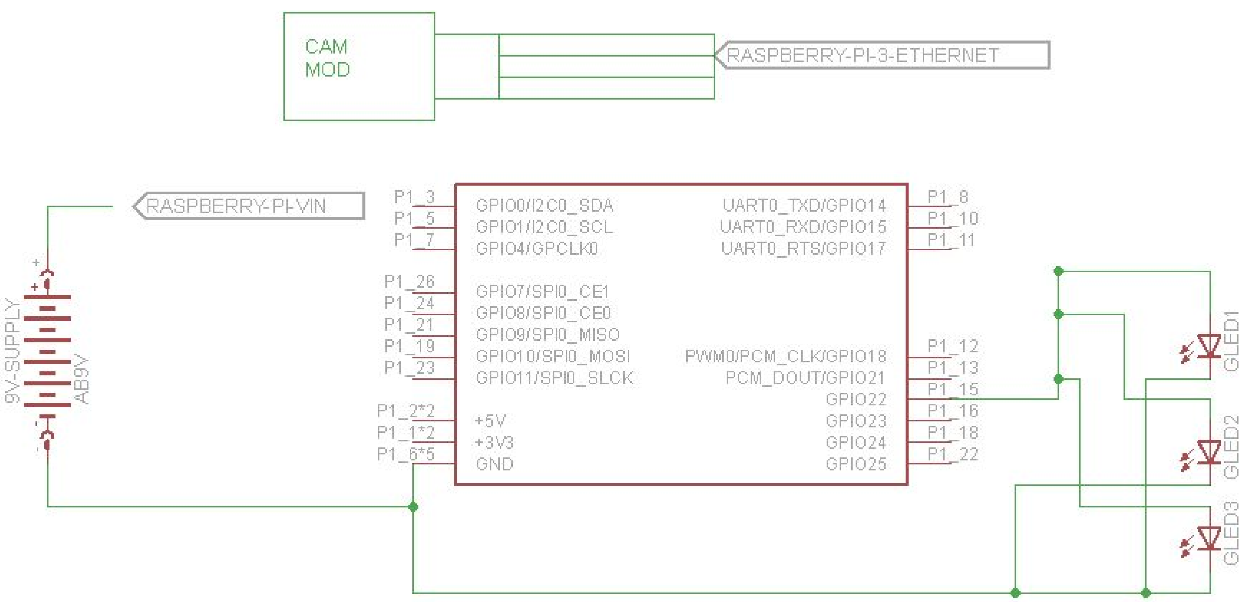
Requirements	**Verifications**
1. Receives data from the barbell subsystem and camera module	1. Feedback is correct based on user technique as shown on barbell subsystem LEDs 2. Check that input signals Raspberry Pi is receiving are consistent with the position of barbell sensors during lift
2. Returns feedback via BlueTooth transmitters	1. Check that output signals Raspberry Pi is giving out via bluetooth match correct LEDs on barbell subsystem
2. Processes video in <= 10s	1. Record time between ending workout and feedback LEDs on barbell subsystem

Table 10: Requirements and Verifications for the Raspberry Pi 3

2.6 Schematics



Barbell Subsystem Circuit Diagram



Computer Vision Subsystem Circuit Diagram

2.7 Tolerance Analysis

In the context of our project, the most critical tolerance that we must maintain is the analysis of joint angles in our computer vision algorithm. We must have very accurate calculations in order to prevent injury in users. But before proceeding with the calculation of different slopes, the RaspberryPi needs to compute our body-tracking algorithm in order to identify body joints.

As explained in the RaspberryPi module, we are running a multi-person pose estimation algorithm which was developed by the Perceptual Computing Lab at Carnegie Mellon University. The important tolerance that we must meet is the one associated with the OpenPose model derived from said algorithm [8]. This is because the accuracy of the slopes depends up on the accuracy of the joint positions.

OpenPose uses a metric called Object Keypoint Similarity(OKS) which tells how close the predicted keypoint is with the ground truth; the ground truth being the real-life position of the keypoint. A higher OKS means that the predicted keypoints are closer to the ground truth.

The accuracy is determined by the COCO Dataset which will compare the OpenPose and wrnchAI model (we decided to use OpenPose instead of wrnchAI). COCO uses Precision and Recall as metrics to do so. Before jumping into the results, we have to go over some basic parameters used during the experiment.

The COCO evaluation used Precision and Recall at 50% and 75% (AP_50, AP_75, AR_50, AR_75) and Precision and Recall for medium (AP_medium, AR_medium) and also large objects (AR_large, AP_large).

The COCO experiment provides the following results:

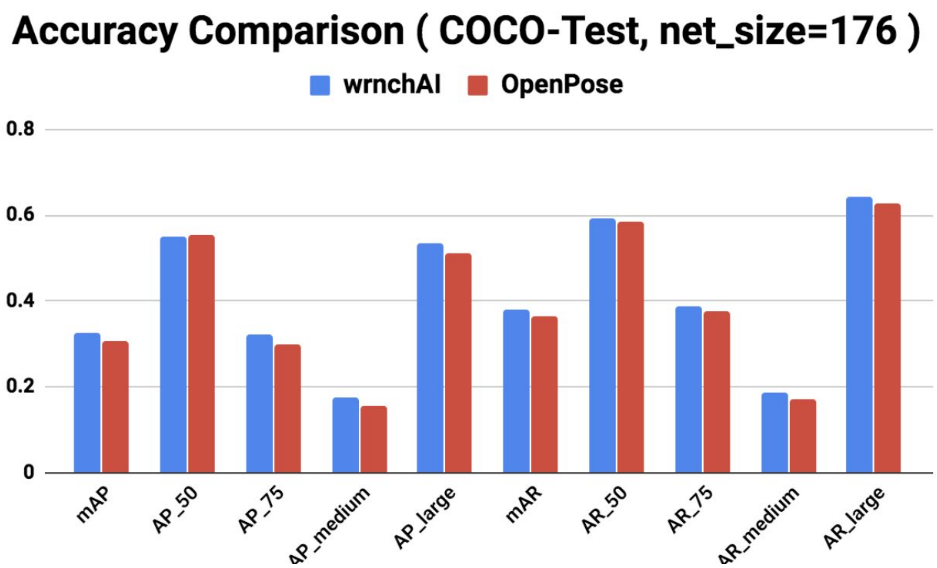


Figure 4: Results on Validation Data

While comparing wrnchAI and OpenPose is important, we will only look focus on the OpenPose OKS tolerance since it is the one that we are using.

Whether it is a small, medium or large image, the COCO experiment shows a 4% tolerance in identifying joints. This is a rather average tolerance value for most situations.

We conducted our own experiment with our prototype code. Holding a 90° angle in our elbow joint, we found that our algorithm gave us angle calculations between 87° and 93°. Therefore, we will be allowing a 3° deviation from “proper” deadlift form to constitute “good” form for our users.

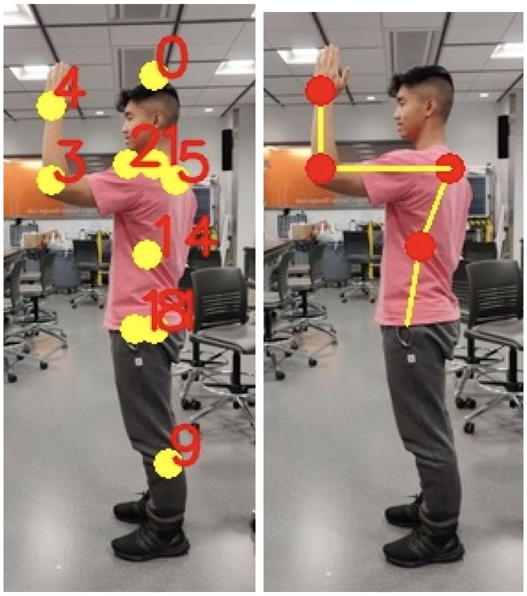


Figure 5: Personalized test for tolerance

3 Cost and Schedule

3.1 Labor Costs

We are working as a team of three engineers to develop this project. Using the average UIUC ECE new-graduate annual salary as of 2016-17 of $96,518, we can get a rough estimate on the cost it would require to develop this project in industry. Assuming that this salary is the outcome of 40 hour work weeks, 52 weeks a year, the hourly pay for ECE new-grads is $46.40. This course is 16 weeks long, and each of us put in, on average, 15 hours a week for this course. With all these assumptions, the development cost for this product would be ($46.40/hour)(16 weeks)(15 hours/week)(3 students) = $33,408.

3.2 Parts Cost

Part Name	Part Description	Cost
Raspberry Pi 3 Model B Motherboard	Used for Computer Vision processing	$35.88
Bluetooth Transmitter: ideapro USB Bluetooth Adapter	Relays signals back and forth between Raspberry Pi and Barbell Subsystem Microcontroller	$7.99
Proximity Sensor: Adafruit IR distance sensor includes cable (10cm-80cm) - GP2Y0A21YK0F [ADA164]	Senses when the barbell is on the ground or not	$20.19
Accelerometer: Gy-521 MPU-6050 MPU6050 Module 3 Axis Analog	Tracks motion of the barbell	$5.86

Gyro Sensors+ 3 Axis Accelerometer Module		
5-Volt Regulator: 5v Regulator, DROK 5pcs Mini Voltage Reducer DC	Ensures the power supply the appropriate voltage to the Barbell Microcontroller	$8.99
9-Volt Battery: Energizer E522 Max 9V Alkaline battery	Powers the CV Subsystem	$8.04
Red and Green LED's:	Indicates when the camera is recording, good form, and bad form	$9.88
Camera Module: Dorhea Raspberry Pi 4 B 3 B+ Camera Module	Records deadlift routine	$21.59
Micro-Controller: KeeYees ESP32 ESP-32S Development Board	Coordinates and relays back and forth between the subsystems	$12.99

Table 11: Cost Analysis Table

3.3 Total Cost

Development cost + Parts cost = $33,539.41

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 algorithm	Define specific values for joint angles in algorithm	solder parts at a time for modularity

11/11	Initial testing in controlled setting	Initial testing in controlled setting	Initial testing in controlled setting
11/18	Testing in public gym setting	Testing in public gym setting	Testing in public gym setting
11/25	Algorithm optimization	Algorithm optimization	Testing in public gym setting

Table 12: Work Schedule

4 Discussion of Ethics and Safety

There are a handful of safety hazards that we must consider with our project. Because our project is meant to be commercialized within the fitness community, all of the intrinsic safety hazards that come with working out are attached to our product. In accordance to the IEEE Code of Ethics #1, we need to inform the public about any dangers that can come with the use of our equipment [5]. Injuries due to poor form as well as miscellaneous accidents that can occur in the gym are things we must consider while designing our product. In addition, we will be keeping our user information confidential. Video recordings of our users will not be saved anywhere post-workout and will not be sent anywhere outside of our device's modules. Thus, our users' privacy is not at risk.

As our product does not require any peripherals to be placed on the user, there are very few added physical risks when using our product beyond the regular risks of working out.

The most common injury that occurs when performing a deadlift is straining your back due to too much curvature in the back while moving the barbell up during the concentric phase of the exercise. To address this issue, we need to make sure that our algorithm places an emphasis on analyzing joint angles correctly. In order to do this, we will be very strict with the allowed angles in the back joints in our model. We will be restricting the back curve range to ± 3 ° from the ideal angle which will be determined by a professional weight lifter. This extra emphasis on back angles will make sure that our algorithm only tells the user they have good form when their back matches the ideal model more closely.

Another risk that our product faces is short-circuiting due to being in a gym. Spilled water and sweat are things that we need to be wary of when designing our product. While the computer vision subsystem will not be in direct contact with our users, the barbell subsystem will be because it will be attached to the barbell that the user is moving. We need to make sure that the barbell subsystem is water-proof so that it cannot short-circuit and hurt the users. In order to accomplish this, we will be placing the barbell subsystem in a plastic casing that will protect it from liquids. Protection from liquids should be sufficient in preventing short-circuiting, but in the case that our device still short-circuits, we will be attaching the barbell-subsystem to the actual barbell via a clamp that separates the electronics from physically touching the barbell. This way, users can use our device without any risk of being electrocuted.

Citations

[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?NGBgKPVglSz.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.ri.cmu.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/
[10] 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