# 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 work to become the best version of themselves. Even though there are several upsides to keeping in shape, especially lifting weights, 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, is one of the most useful exercises for building functional strength, but it also is one of the most dangerous in the weight room. 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 repair. A reliable technique-feedback system would 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 a few main methods people use to learn and correct their form while lifting. One is hiring a physical trainer, and another 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, people may not be interested in having one due to not being comfortable with the personal interaction or wanting to pay more on top of the gym membership. These aspects may play a role in the fact that on average only 15% of people with gym memberships utilize personal training services[3]. Utilizing a trainer can also put the user prone to trainer negligence and still be subjected to poor exercise practices. And in regard to looking to the internet for advice, there are a vast amount of resources a person who is looking for workout advice can use. The problem here is that there are varying opinions between these resources of what good form is, and using the internet leaves the user to try to correct their own form based on the material they see. This leaves a lot of nuance due to a lack of active and personalized feedback to the user.

Our system must give be able to give accurate analysis and accurate feedback of the user's deadlift form to ensure optimal safety to the user so they know what good form. This requires and clear cut and consistent guideline for good deadlifting form. We believe that this system can provide enough value to the consumer to be either purchased by the individual or be implemented with the gym equipment. This will minimize a gym member's chance of getting injured, keeping them in the gym more as they work to become the best version of themselves.

## 1.3 Visual Aid

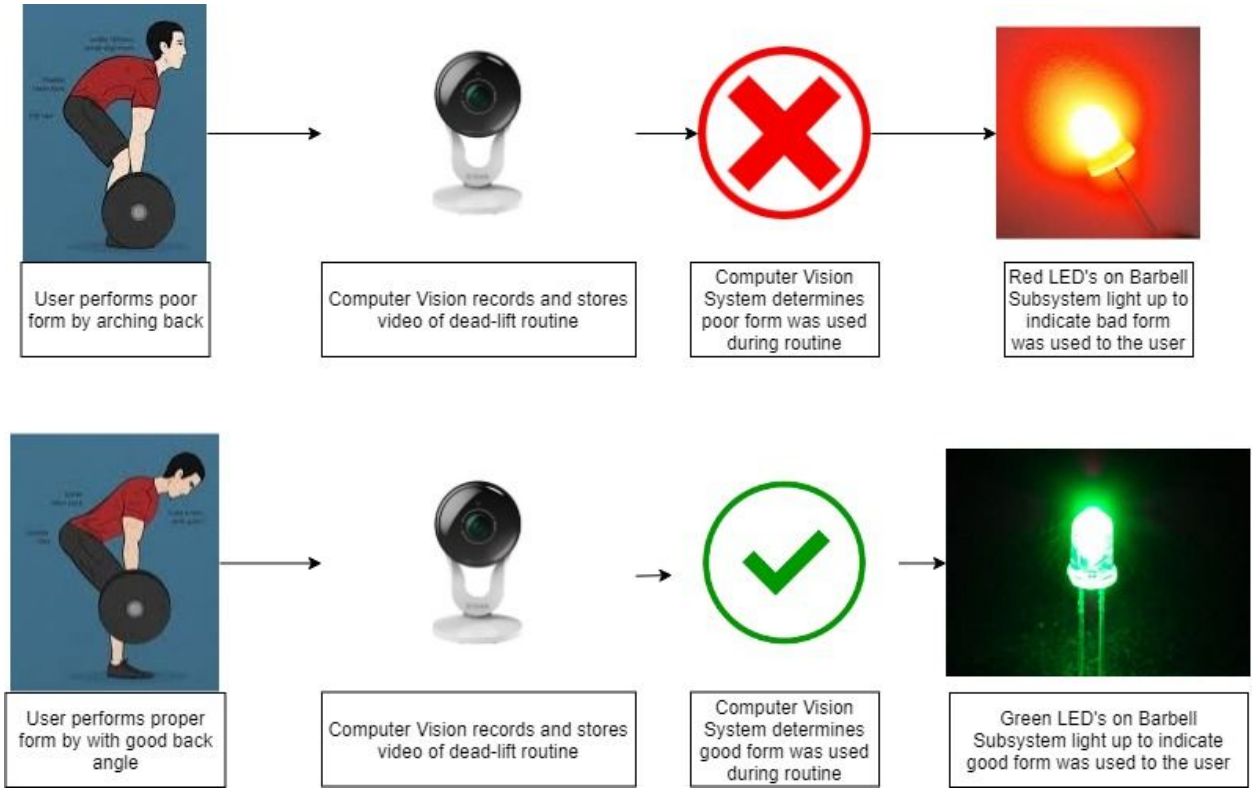



*Figure 1: Visual Aid for Dead-lift Assistant Functionality*

## 1.4 HIgh-Level Requirements List

1. Each subsystem must be able to connect and transfer data between each other automatically
2. The Barbell Subsystem must be able to compute the data relayed by the sensors to the Computer Vision Subsystem so it can track back angles properly
3. The Computer Subsystem must be able to accurate;y and consistently determine whether the proper form is being performed by the user and relay the proper signals back to the Barbel Subsystem to light up the appropriate LEDs.

# 2 Design

## 2.1 Block Diagram

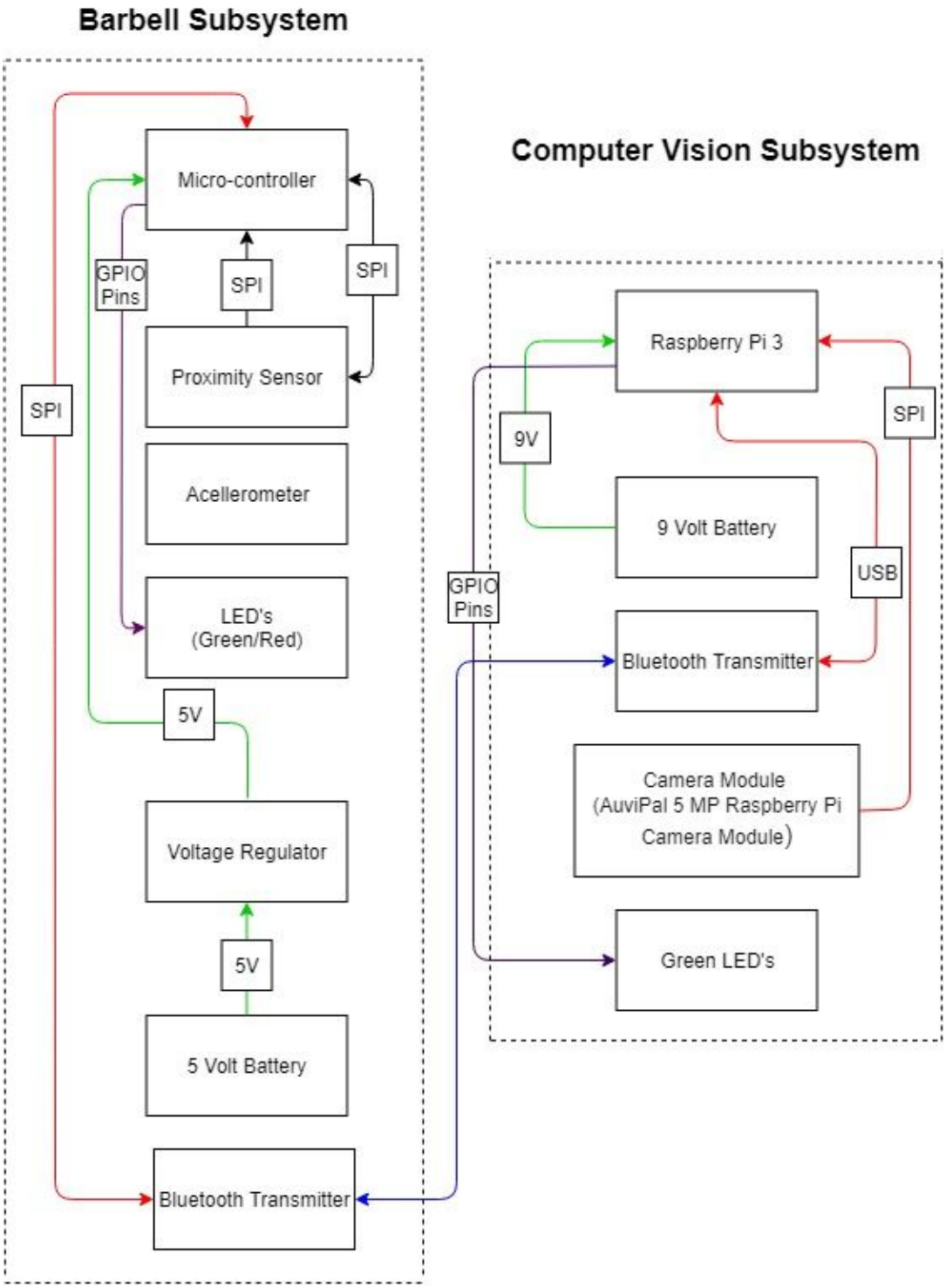



*Figure 2: Deadlift Assistant Block Diagram*

## 2.3 Physical Design

The physical design (Figure 3 below) of the project will consist of two devices, one for each subsystem. The barbell subsystem will be held in a plastic, waterproof case to protect the module from sweat and moisture. It will also use a clamp that can fit on the barbell. It must be tight enough to stay on the barbell while it is in motion, but loose enough to ensure that the proximity from the floor can be measured accurately. The other deice will be Computer Vision Subsystem. This will also have a plastic casing and possibly a stand depending on the range of the frame of the camera. This device will be placed at the side of the user to accurately analyze the angle of their spine.

### 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. Proximity sensors have a smaller threshold for movement, so we are aiming for the sensor to detect height differential up to a centimeter.

| **Requirements** | **Verifications** |
|---|---|
| 1. Relays signal to the CV Subsystem is <= ___s<br>2. Can detect a proximity difference at a threshold of 1 cm | 1. LEDs light up <= ___s after barbell lifted<br>2. LEDs shut off <= ___s after barbell dropped and routine finishes<br>3. Use an ammeter to make sure the proximity sensor doesn't surpass microcontroller rating of ___[mA] |

### 2.4.2 Accelerometer

The accelerometer then differentiates the direction of the barbell. 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.

| **Requirements** | **Verifications** |
|---|---|
| 1. Relays signal to the CV Subsystem is <= ___s<br>2. Can also log when the motion changes so that it can be synced with the computer vision | 1. LEDs stay lit during routine<br>2. Use an ammeter to make sure the accelerometer doesn't surpass microcontroller rating of ___[mA] |

### 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 ___V and can supply a fairly constant voltage of __V to the microcontroller<br>2. Can connect to the power pin of the microcontroller | 1. Use a voltmeter to make sure the regulator doesn't surpass microcontroller rating of ___[V], and to ensure that its own rating isn't being violated by the battery |

### 2.4.4 LEDs

The Computer Vision Subsystem will relay back signals to the Barbell Subsystem to turn on the appropriate LEDs (Green for Good form detected, Red for bad form detected). This is how the user gets feedback on their form.

| **Requirements** | **Verifications** |
|---|---|
| 1. LEDs can handle __ [A] rating<br>2. Light up after Computer Vision processes results | 1. Ammeter detects <= __[A]<br>2. Visual detection of lights turning on |

### 2.4.5 Microcontroller chip

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

| **Requirements** | **Verifications** |
|---|---|
| 1. BlueTooth compatibility with Raspberry Pi 3 - Model B<br>2. Must not exceed __[mA] ratings<br>3. Pins for the voltage regulator, the proximity sensor, the accelerometer, and general-purpose input/output (GPIO) pins for the LEDs | 1. Analyze raspberry pi code to see if it is receiving the proper signals from the microcontroller chip<br>2. Use an ammeter to ensure it doesn't surpass __[mA] rating during operation<br>3. Look at the datasheet to ensure a sufficient amount of pins |

## 2.5 [Computer Vision Subsystem]

Our computer vision subsystem consists of a BlueTooth transmitter, a 9V battery, a camera module, green LED's, 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 LED's on the Computer Vision Subsystem while be lit. After this subsystem processes the Computer Vision Algotrith and if the user has performed good or bad form, it will relay a  back to the Barbell System to light up the appropriate LED's on it.

### 2.5.1 Bluetooth transmitter

The BlueTooth transmitter here receives data from the barbell subsystem's equivalent transmitter. The information it is receiving tells 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 correctly and <= __[s]<br>2. Relays information to the barbell subsystem after the algorithm is finished | 1. "Recording" LED lights up when barbell begins moving<br>2. Correct form indicator LED lights up on barbell subsystem after deadlift motion is complete |

### 2.5.2 9V Battery

The battery here provides power to the entire subsystem.

| **Requirements** | **Verifications** |
|---|---|
| Battery powers entire subsystem | Power indicators on RaspberryPi light up, "recording" LED lights up during the workout |

### 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<br>2. Stops recording at deadlift movement end | 1. Recording LED lights up at the start of workout motion<br>2. Recording LED turns off at the end of workout motion |

### 2.5.5 Recording LED

This LED is used to indicate if the camera is recording.

| **Requirements** | **Verifications** |
|---|---|
| 1. The LED's lights up during deadlift motion<br>2. LED turns off when deadlift motion is over<br>3. The LED's can handle a voltage supply of __[V] | 1. LED only turns on when motion is in progress<br>2. LED is off otherwise<br>3. Voltmeter to ensure the LED potential is <= __[V] |

### 2.5.6 RaspberryPi 3

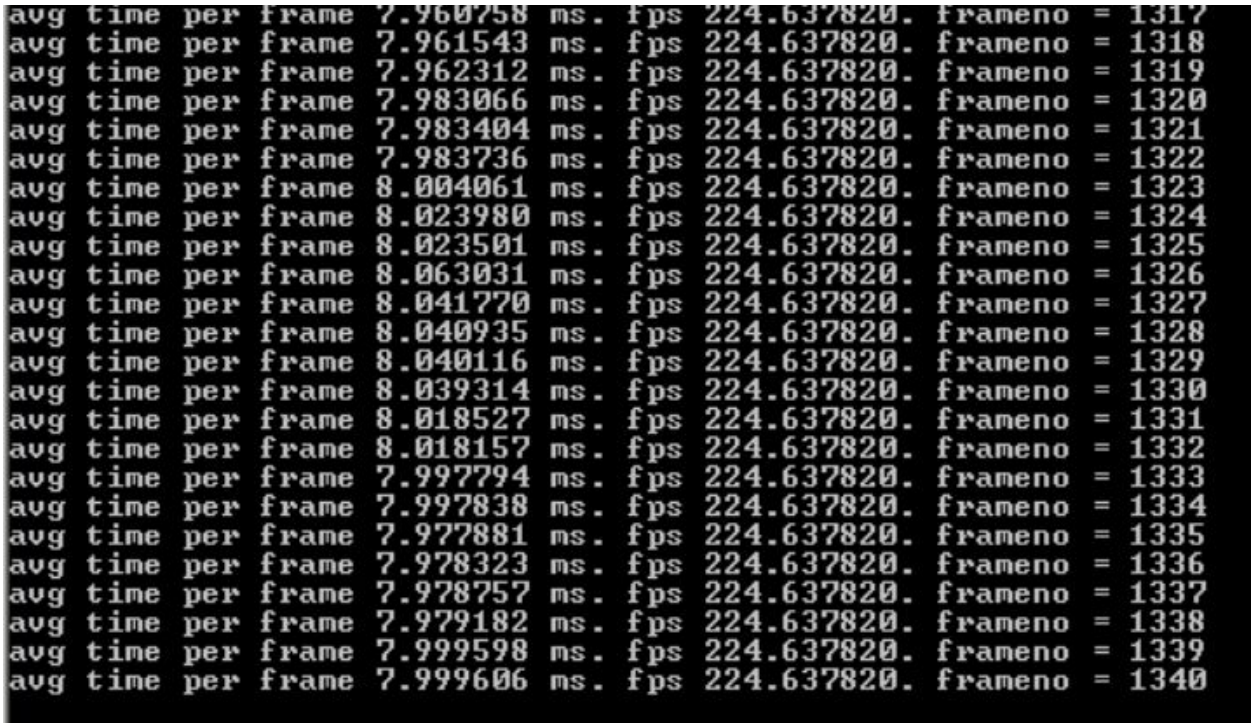
The RaspberryPi is used to process all of the data received from the barbell subsystem as well as the camera module. It runs a body-tracking algorithm, multi-person pose estimation, developed by the Perceptual Computing Lab at Carnegie Mellon University [7] to track the user while they are performing 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<br>2. Returns feedback via BlueTooth transmitters | Feedback is correct based on user technique as shown on barbell subsystem LEDs |

## 2.6 Tolerance Analysis

In the context of our project, the most critical part of it is the combination of the computer vision and the algorithm used to detect joins within the body of the user. The components involved at these critical stages are the camera module within the Computer Vision subsystem and the Micro-controller within the Barbell subsystem. Based on our current estimation, the feedback coming from the processed video by the Micro-controller will be running with a low frame rate.

Here is a display of the current frame rate coming fro  OpenCv:




In order to obtain these values, we used the cvGetTickCount() method offered by OpenCV and we put a Sleep inside process() in order  to measure the correct fps. In order to increase the frame rate , we are going to use  a lower quality video.

```
#include "opencv2/highgui/highgui.hpp"

#include "opencv2/imgproc/imgproc.hpp"

#include <opencv/cv.h> #include <sys/timeb.h> using namespace cv;

#if defined(_MSC_VER) || defined(WIN32) || defined(_WIN32) || defined(__WIN32__) \ ||
defined(WIN64) || defined(_WIN64) || defined(__WIN64__)

#include <windows.h> bool _qpcInited=false; double PCFreq = 0.0; __int64 CounterStart
= 0; void InitCounter() { LARGE_INTEGER li; if(!QueryPerformanceFrequency(&li)) {
```

```
std::cout << "QueryPerformanceFrequency failed!\n"; } PCFreq =
double(li.QuadPart)/1000.0f; _qpcInited=true; } double CLOCK() { if(!_qpcInited)
InitCounter(); LARGE_INTEGER li; QueryPerformanceCounter(&li); return
double(li.QuadPart)/PCFreq; } #endif #if defined(unix) || defined(__unix) ||
defined(__unix__) \ || defined(linux) || defined(__linux) || defined(__linux__) \ ||
defined(sun) || defined(__sun) \ || defined(BSD) || defined(__OpenBSD__) ||
defined(__NetBSD__) \ || defined(__FreeBSD__) || defined __DragonFly__ \ ||
defined(sgi) || defined(__sgi) \ || defined(__MACOSX__) || defined(__APPLE__) \ ||
defined(__CYGWIN__) double CLOCK() { struct timespec t;
clock_gettime(CLOCK_MONOTONIC, &t); return (t.tv_sec * 1000)+(t.tv_nsec*1e-6); }
#endif double _avgdur=0; double _fpsstart=0; double _avgfps=0; double _fps1sec=0;
double avgdur(double newdur) { _avgdur=0.98*_avgdur+0.02*newdur; return _avgdur; }
double avgfps() { if(CLOCK()-_fpsstart>1000) { _fpsstart=CLOCK();
_avgfps=0.7*_avgfps+0.3*_fps1sec; _fps1sec=0; } _fps1sec++; return _avgfps; } void
process(Mat& frame) { Sleep(3); } int main(int argc, char** argv) { int frameno=0;
cv::Mat frame; cv::VideoCapture cap(0); for(;;) { //cap>>frame; double start=CLOCK();
process(frame); double dur = CLOCK()-start; printf("avg time per frame %f ms. fps %f.
frameno = %d\n",avgdur(dur),avgfps(),frameno++ ); if(waitKey(1)==27) exit(0); }
return 0; }
```

# 3 Cost and Schedule

## 3.1 Cost Analysis

| 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 Subsystemm 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 Gyro Sensors+ 3 Axis Accelerometer Module | Tracks motion of the barbell | $5.86 |
| **5-Volt Regulator:** 5v Regulator, DROK 5pcs Mini | Ensures the power supply the appropriate voltage to the | $8.99 |

| | | |
|---|---|---|
| Voltage Reducer DC | Barbellmicrocontrollerr | |
| **5-Volt Battery:** | Powers Barbell Subsystem | |
| **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 | |
| **Camera Module:** | Records deadlift routine | |
| **Micro-Controller:** KeeYees ESP32 ESP-32S Development Board | Coordinates and relays back and forth between the subsystemss | $12.99 |

## 3.2 Schedule

| Week | Milestone |
|---|---|
| 9/30 | - Design document rough draft |
| 10/07 | - Order parts |
| 10/14 | - Integrate joint-angle calculations in the algorithm |
| 10/21 | - Build hardware |
| 10/28 | - Integrate hardware feedback into the algorithm |
| 11/04 | - Teach algorithm what "good" form is |
| 11/11 | - Field testing |
| 11/18 | - Algorithm optimization |
| 11/25 | - Final testing |

# 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.

Luckily, as our product does not require any peripherals to be placed on the user, there are not many 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 extra emphasis on analyzing back angles correctly. In order to do this, we will be more strict with the allowed angles in the back joints in our model. For example, while we may allow a range of plus or minus 5 degrees from the ideal angle for knee bend to constitute good form, we will restrict the back curve range to 2 degrees from the ideal angle. 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 casing that will protect it from liquids. This casing will be able to be attached to a barbell without any risk to the user being electrocuted.

# Citations


[1]
https://www.cpsc.gov/s3fs-public/2018%20Neiss%20data%20highlights.pdf?NGBgKPVglSz.ShB_W6Y0ENRjhr6qAtWr

[2]
https://www.livestrong.com/article/497414-back-injuries-from-deadlifts/

[3]
https://www.ihrsa.org/improve-your-club/personal-training-usage-continues-to-grow-among-health-club-members/

[4]
https://github.com/CMU-Perceptual-Computing-Lab/openpose/blob/master/python/openpose/openpose_python.cpp

[5]
"IEEE IEEE Code of Ethics." IEEE - IEEE Code of Ethics, https://www.ieee.org/about/corporate/governance/p7-8.html Accessed 19 Sept. 2019.

[6]
"Back Pain Facts and Statistics." Accessed 19 Sept. 2019
https://www.acatoday.org/Patients/Health-Wellness-Information/Back-Pain-Facts-and-statistics

[7] https://arxiv.org/pdf/1611.08050.pdf

https://courses.engr.illinois.edu/ece445/guidelines/design-document.asp
https://courses.engr.illinois.edu/ece445/documents/examples/ExampleDD.pdf