Smart Lifting: A Novel Approach to Portable Autonomous Form Detection

Zachary Farr
Computer and Information Sciences
Virginia Military Institute
Lexington, VA
farrzr22@mail.vmi.edu

Samuel Estes
Computer and Information Sciences
Virginia Military Institute
Lexington, VA
estessj22@mail.vmi.edu

Emily Hattman
Computer and Information Sciences
Virginia Military Sciences
Lexington, VA
hattmanee22@mail.vmi.edu

***Abstract*—In current society, the COVID-19 pandemic forced public gyms to close and eliminated citizens' opportunity to exercise. The creation of the Smart- Lifting device provides citizens with an inexpensive and portable solution to receive feedback on weight lifting form. The device detects movement and calculates angles based off a mathematical formula and computer vision.**

***Index Terms*—computer vision, neural networks, image processing**

I. Introduction

The Smart Lifting device will aid new and experienced weightlifters by providing insights on proper form and technique. Using pose detection and computer vision, this rechargeable device will record a person in real-time and show incorrect forms based on the chosen exercise. The main device will utilize a computer vision library and a mobile computing device to compute and send the data to the user's smartphone securely. The neural network is built upon professional lifters from the internet and real-life for correct form and other human testers that vary in size, strength, and technique. Compared to large-scale devices, Smart Lifting is a compact size meant for a tripod to be easily brought around in the gym. This device will ensure the security of the data collected from the user.

II. Initial Design

A. Concept Statement

The Smart Lifting device will aid new and experienced weightlifters by providing insights on proper form and technique. Using pose detection and computer vision, this rechargeable device will record a person in real-time and show incorrect forms based on the chosen exercise. The main device will utilize a computer vision library and a mobile computing device to compute and send the data to the user's smartphone securely. The neural network is built upon professional lifters from the internet and real-life for correct form and other human testers that vary in size, strength, and technique. Compared to large-scale devices, Smart Lifting is a compact size meant for a tripod to be easily brought around in the gym. This device will ensure the security of the data collected from the user.

B. User Models

a) Work Roles:

- Weightlifter: User utilizes system to receive feedback on weightlifting form for the specific lift they conducted.

b) User Personas:

- Weightlifter (novice): After a couple weeks at VMI, Jane Doe wants to begin weightlifting. Jane fears she will get injured because she does not know the proper form and technique for weightlifting movements. Jane utilizes smart- lifting to receive feedback on her back squat form. The software provides her feedback, such as, keeping her knees behind her toes. Jane Doe becomes more comfortable and confident in weightlifting while avoiding injury.
- Weightlifter (experienced): Zack Smith is an avid weightlifter that fears his form will be comprised as his back squat increases in weight. Zack Smith utilizes smart-lifting to watch his form and receive feedback on his form in real time.

C. Usage Model

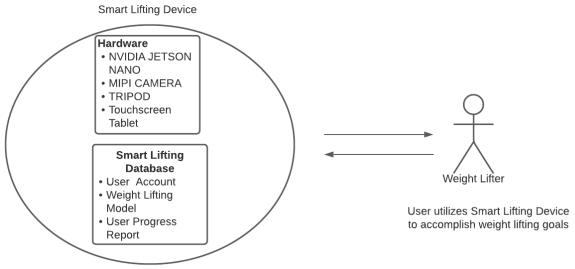

Fig. 1. Flow Model

a) Flow Model:

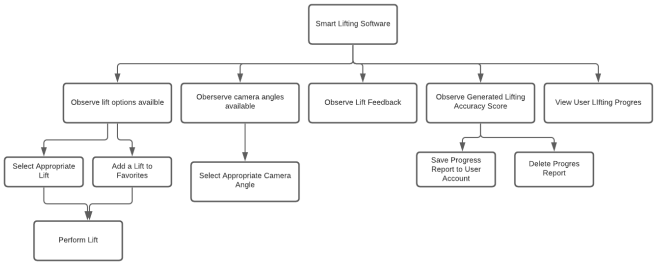

Fig. 2. Hierarchical Task Inventory

b) Hierarchical Task Inventory :

c) Task Sequence:

I. User: Presses on button, express interest to start lifting
II. System: power on and load software
III. User: select lift
IV. System: load pose model
V. System: prompt user to begin
VI. User: perform lift
VII. System: compare user to pose model
VIII. User: Perform end pose to end session
IX. System: analyze and determine user accuracy
X. System: show progress report and plays back video with pose detection
XI. User: press off button
XII. System: system power off

III. Design Process

A. Storyboard (Figure 3 and Figure 4)

Figure 4: In the first frame, the user powers on the software through the interface on the hardware device. Before the next frame, the software is powered on, and the lifting models are loaded. Once the software is loaded, the user performs the lift they selected and the software analyzes form. In the second frame, the user will select the camera angle they wish to use. In the third frame, the user can be shown using the device and performing a squat. Following the exercise, the user is greeted with some actionable insights on things the user can improve in their form. In the fifth frame, the user is shown their score out of 100 percent, a score which is derived from their accuracy to the model. In the final frame, the device is shown powered off, showing the completed function of the device.

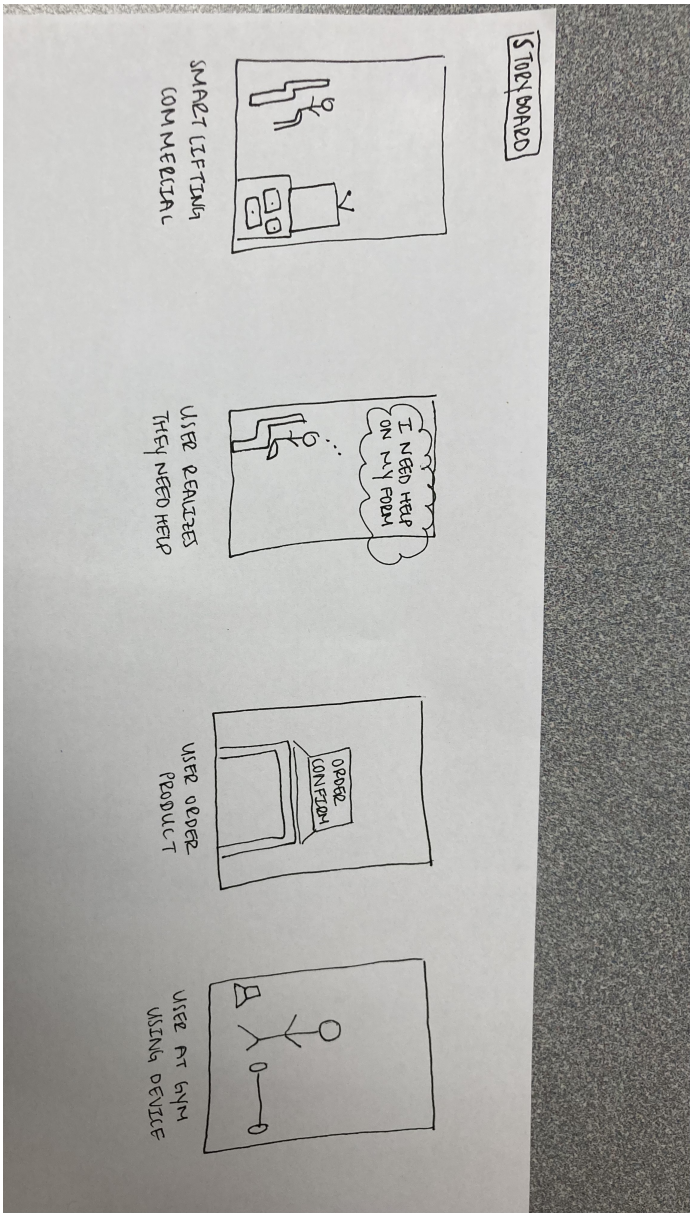

Fig. 3. Storyboard

B. Screen Layout Design

- A wireframe representation of a set of related intermediate screen designs, corresponding to the usage/design scenario you used for the storyboard above. Show screen layout and navigation.

Figure 5: The wireframe represents the screen designs for the user powering on the software via the hardware interface, then selecting their intended lift, and the software loading the model for the user. The user will then perform the lift as the software registers the key points on the user and generates a model. The user performs the T-pose to end the video. The software will generate a progress report and replay the lift video with the appropriate model.

C. UX Evaluation Plan

Our plan to evaluate the UX from this point is to devise theoretical use cases and question individuals on what they would expect to see at each step of the process. Also, we would ask users how they would like to view the output of the algorithm. A cognitive walk-through is the most effective UX inspection process we can use at this stage of our development. We decided upon the RITE UX evaluation method, as this is the best evaluation method for our team to use at this stage of development. One of the key tasks we used to drive the rapid

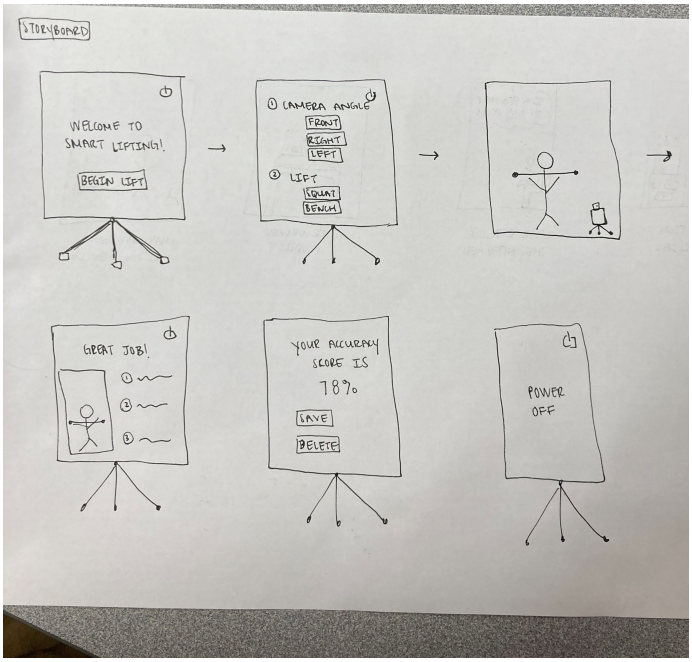

Fig. 4. Storyboard 2

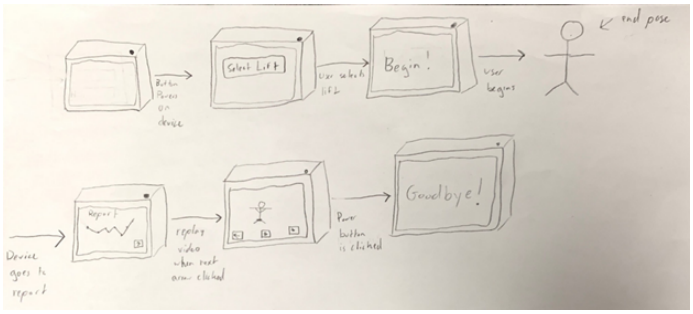

Fig. 5. Screen Design Layout

empirical evaluation process was to have a theoretical user perform tasks using their imagination then report what they would expect at each stage. The user must also determine an easy way to view the data.

D. UX problems discovered with the UX inspection process

I. Will I be able to understand the changes I need to make?
II. Do I need the device to see how well I did on the lift?
III. Can my phone tell me what I can do better?
IV. What if I don't even know where to start on a lift?
V. How far back can I see in terms of the progress I've made?
VI. Can I see last month's lift score and recording?

E. UX problems discovered with the rapid empirical process

I. What will the output of the system look like?
II. How will the device know which lift you are about to perform?
III. How do I stop the program once I am finished?
IV. How will I know how to make a change in my form?
V. Where do I place the camera system?
VI. Can I see my progress over time?
VII. How do I know how well I did on the lift?

F. Merged union list of UX problems discovered from both approaches

I. What will the output of the system look like? Is it easy to understand?
II. How will the device know which lift you are about to perform?
III. How do I stop the program once I am finished?
IV. How will I know how to make a change in my form?
V. Where do I place the camera system?
VI. Can I see my progress over time? How far back can I see?
VII. How do I know how well I did on the lift?

G. UX problems selected for cost-importance analysis

I. What will the output of the system look like?
II. How will the device know which lift you are about to perform?
III. How do I stop the program once I am finished?
IV. How will I know how to make a change in my form?
V. Where do I place the camera system?
VI. Can I see my progress over time?
VII. How do I know how well I did on the lift?
VIII. Will I be able to understand the changes I need to make?
IX. Do I need the device to see how well I did on the lift?
X. Can my phone tell me what I can do better?
XI. What if I don't even know where to start on a lift?
XII. How far back can I see in terms of the progress I've made?
XIII. Can I see last month's lift score and recording?

Problem	Importance	Solution	Cost	Ratio	Priority	Cum Cost	Resolution
1. User clicks on wrong exercise	2	Allow user to revert back and select correct lift	1	2000	1	1	Fixing this version
2. User is not being registered in camera	5	User re-collaborates and moves camera angle	3	2500	2	3	Fixing this version
3. User is unsure where to position camera	3	User read comments from software where to position	2	1500	3	5	Fixing this version

Fig. 6. Cost Importance Diagram

IV. Implementation

For our proof-of-concept demonstration, we established four steps for our December goal.

- User will power on the device and the software will load with an OpenGL display box.
- User will perform a squat.
- Software will detect form accuracy and provide a percentage of correctness (out of 100) in the terminal log of the software.
- User will power off the device.

Using the open library, jetson-inference, we developed a program that would take the camera input of the Nvidia

Jetson and make calculations based on the results of the pose detection [1].

For each person detected, a ”pose” object is created containing 18 points from the resnet model. Those points include

1) nose
2) left eye
3) right eye
4) left ear
5) right ear
6) left shoulder
7) right shoulder
8) left elbow
9) right elbow
10) left wrist
11) right wrist
12) left hip
13) right hip
14) left knee
15) right knee
16) left ankle
17) right ankle
18) neck

To start understanding the way posenet processes the images, Figure 8 and 9 show the before and after results of a given image.

Fig. 7. Picture before using Pose Detection

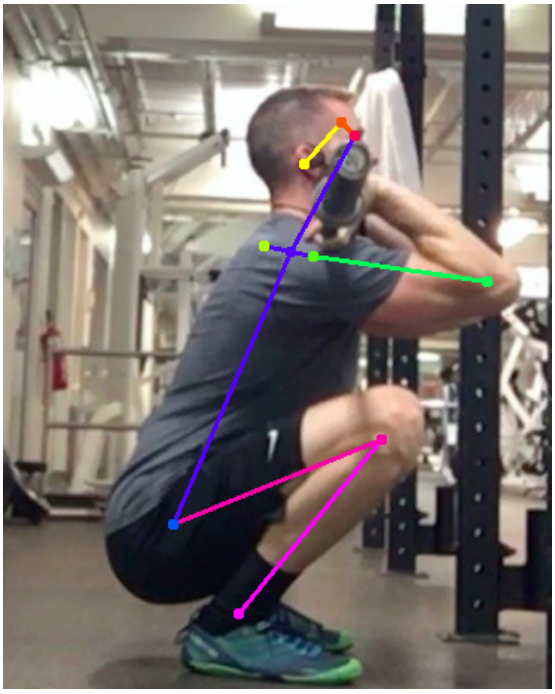

Fig. 8. Results of Pose Detection Model

As seen, the model draws as many of the 18 points as possible and the lines connecting them. The next step was to show the capabilities of interactions between the 18 points. Preliminary methods were made that showed where the user was pointing to on the x, y coordinate plane and whether or not the user’s hip was below the knee.

To determine the quality of squat, we focused on the down position on a right-side view of the user. By using the formula to calculate the angle based on the slope of two intersecting lines, the angles for the knee and the hip were found. The below figure shows the intended angles and points focused on for the proof-of-concept prototype.

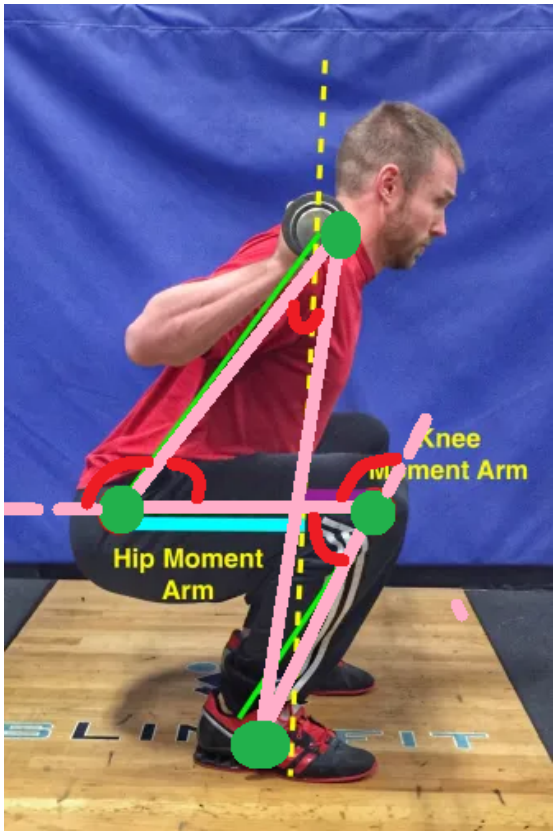

Fig. 9. Example Diagram for angles and points used

The slope of the lines were calculated by using the coordinates of which of the 18 points were needed. For example, to calculate the angle of the right knee, two lines were needed: one connecting the right hip and right knee and another connecting the right knee and the right ankle. By using the slope formula:

$$m = \frac{rise}{run} = \frac{y_2 - y_1}{x_2 - x_1} \tag{1}$$

The slope was found for the right upper leg. The same was done for the right lower leg using the points for the right knee and right ankle. The angle given the slope of two intersecting lines was found by using

$$\theta = arctan(\frac{m_2 - m_1}{1 + m_2 * m_1}) \tag{2}$$

[2] By then comparing the produced knee angle to the desired angle of 55°, the difference is outputted and scored [3]. Below is the output of the previous pose detection model based on these calculations using the below grading rubric.

```
detected 1 objects in image
--------------------
Right Upper Leg Slope: 0.40262138038706113
Right Lower Leg Slope: 1.223151240247287
Right knee angle = 28.801157700581943
     Off by -26.198842299418057 degrees
--------------------
--------------------
Right Back Slope: 2.292390647567188
Right Upper Leg Slope: 0.40262138038706113
Hip angle = 44.50115721391102
     Off by -10.498842786088979 degrees
--------------------
####################
Squat is too deep. Raise hips to have the hip joint parallel to knee. Score = 29.337%
Keep chest upward and look straight ahead. Score = 81.671%
Current Score: 55.504%
################################
BEST SCORE = 55.504%
################################
```

Fig. 10. Log Output of Pose Detection Test Image

1. 125 degree Knee angle (upper leg-lower leg knee inner angle)
2. 55 degree Hip angle (torso-upper leg inner angle)

+-5 deg = 100

+-8 = 90

+-11 = 80

+-14 = 70

+-17 = 60

+- 20 or more = 50

Take the average of score reached for points 1 and 2, this will be the score received. There is a margin of error of 3 degrees in between each next score. Example below

1. 125 degree Knee angle (upper leg-lower leg knee inner angle), found angle: 138, score = 80
2. 55-degree Hip angle (torso-upper leg inner angle), found angle: 61, score 100

Fig. 11. Grading Rubric and Example for the Squat

The grading rubric shows an example for scoring the two different angles. Any angles with a difference below a 5°difference receives a 100 while a difference above 35°receives a 0.

By then adding these methods to the program intended for live camera feed as an input. The below figures show the results of both the OpenGL box and terminal log of the program.

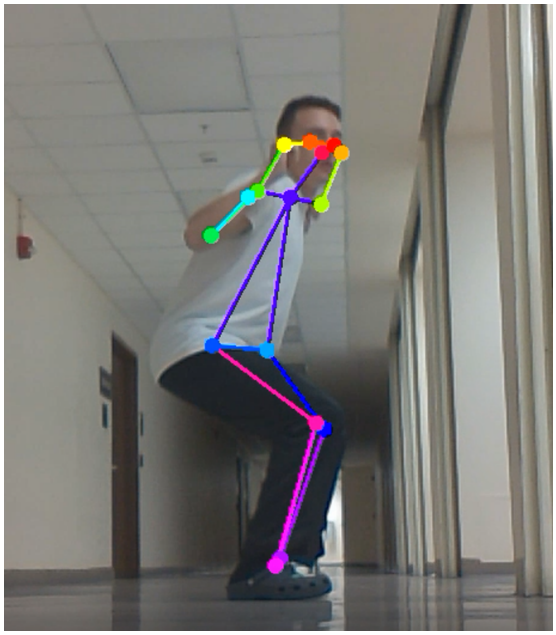

Fig. 12. Bad Form Live Demonstration OpenGL Screenshot

```
--------------------
####################
Squat is too deep. Raise hips to have the hip joint parallel to knee
Keep chest upward and look straight ahead
Current Score: 463.639047355362
detected 1 objects in image
--------------------
Right Upper Leg Slope: 6.9190705154611996
Right Lower Leg Slope: 6.075690106300957
Right knee angle = 1.1226313960921308
     Off by -53.87736860390787 degrees
--------------------
--------------------
Right Back Slope: 8.053901999120933
Right Upper Leg Slope: 6.9190705154611996
Hip angle = 1.1460871036599871
     Off by -123.85391289634 degrees
--------------------
####################
```

Fig. 13. Bad Form Live Demonstration Terminal Screenshot

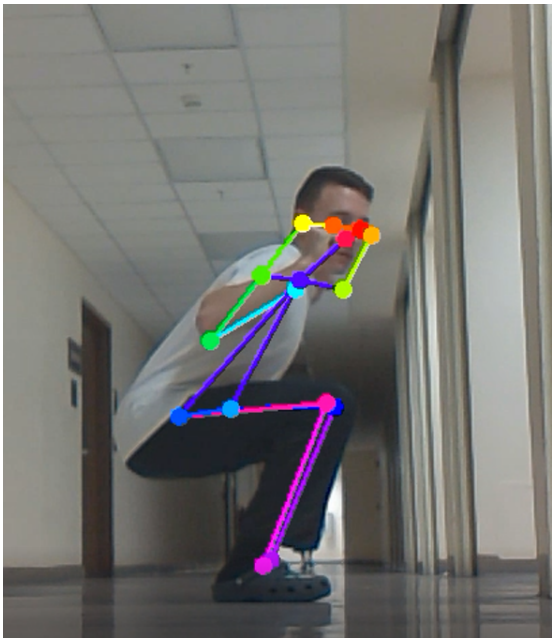

Fig. 14. Good Form Live Demonstration OpenGL Screenshot

```
--------------------
####################
Squat is too deep. Raise hips to have the hip joint parallel to knee
Lean forward to prevent excessive stress on the lower back
Current Score: 493.9829621314
detected 1 objects in image
--------------------
Right Upper Leg Slope: 0.18096105153925976
Right Lower Leg Slope: 2.605796717182179
Right knee angle = 58.747905050276586
     Off by 3.747905050276586 degrees
--------------------
--------------------
Right Back Slope: 1.15000233213617
Right Upper Leg Slope: 0.18096105153925976
Hip angle = 38.733669753830675
     Off by -86.26633024616933 degrees
--------------------
####################
```

Fig. 15. Good Form Live Demonstration Terminal Screenshot

V. Future Works

While the proposed proof-of-concept prototype was achieved, there are still many plans for improving upon this system. While running static images through the program, the proper percentages were generated based on the grading scale and implemented formulas. However, with the live feed, the angles were correct but the percentages were not. This will require adjusting the code so that out-of-bounds numbers do not throw off the system.

More calculations will are in need of being determined based on different positions of the squat since only the ”down” position is being tested. Other angles need to be implemented to add to more potential for insights and scoring. Finally, to present this as a user-friendly software, a GUI will have to implemented to show the recorded camera feed, scoring, insights, and a progress report based on lifts completed over time.

Additionally, the Smart Lifting Device could be implemented as a physical therapy device in order to ensure patients

are performing their rehabilitation exercises correctly. The device is able to detect movement of the user and calculate any specific angle. In theory, the Smart Device can be implemented accordingly to a specific physical therapy exercises and provide the user with feedback if their form is incorrect.

VI. Lessons Learned and Conclusion

This document helped codify our thoughts that were free floating and allowed us to better organize ourselves and see where we stand in terms of progress and future work. By strategically planning every use case and user scenario for the design we were able to consider factors such as the presentation of the outcome in a way that allows the user to easily understand the results. However, we believe that if we were able to work on this document over the course of this class as opposed to trying to remember all of the work and ideas generated at a later date, it would prove more beneficial. Another highly beneficial lesson we learned early on was to work side by side, not over the internet. In working side by side, we can more adequately communicate and review other team member's decisions. This rapid feedback was integral to the completion of this project. Another lesson we learned through this project is that there is a lot of work that has been done, and by researching what others have done, we can both save time by utilizing others' work, as well as helping the creator by paying their work respect and giving them recognition for their accomplishment. Finally, the most important lesson we learned was to create the system design early on to ensure everyone is on the same page when it comes to the function of the product. This road map helps everyone understand what we have done and what still remains.

References

[1] Dusty-Nv, "Dusty-NV/Jetson-inference: Hello AI World Guide to deploying deep-learning inference networks and deep vision primitives with TENSORRT and Nvidia Jetson.," GitHub. [Online]. Available: https://github.com/dusty-nv/jetson-inference. [Accessed: 14-Dec-2021].

[2] "Angle between two lines - formula, examples," Cuemath. [Online]. Available: https://www.cuemath.com/geometry/angle-between-two-lines/. [Accessed: 14-Dec-2021].

[3] Ahorschig, "The real science of the squat," Squat University, 06-Mar-2021. [Online]. Available: https://squatuniversity.com/2016/04/20/the-real-science-of-the-squat/. [Accessed: 14-Dec-2021].

List of Figures