INTERACTIVE GYM BUDDY

PROJECT REPORT

1. Abstract

Today we are knee deep in the technological era where even fitness has become solely connected to highly curated tech. People carried out their fitness workouts in the gym. The pandemic restricted us a lot, but the routines of life must go on. Tons of people went on about their regimes at home but with no knowledge of whether they are performing the exercises right. Our Interactive Gym Buddy is an AI-powered bot which allows you to choose exercises using hand gestures through your camera and counts the reps you do only when you perform an exercise right.

2. Introduction

During the pandemic, all gyms were closed due to a need for social distancing, therefore people did all their exercise in their homes.
As such, amateurs who were doing exercises for the first time may not know if they are doing the exercise correctly or not.
Therefore, we sought to create an automated system that was capable of assessing the state of exercise, and would tell the user if the pose was correct or not.
It doesn't restrict you to time or a place, whenever you get time, you may seek its help for your needs.
Through our model we aim at refuting the notion that the exercises are platform dependent and will be trying to make it flexible enough to fit in the schedule of everyone.
It has three exercises to choose from:

- Bicep Curl

The angle between shoulder-elbow and elbow-wrist is used to count one valid rep.

- Deadlifts

One valid rep will be considered when hands go below knees and then come back up.

- Squats

One valid rep will be counted when hip-knee range is 30^{o}.

3. Requirement Analysis (Data Collected)

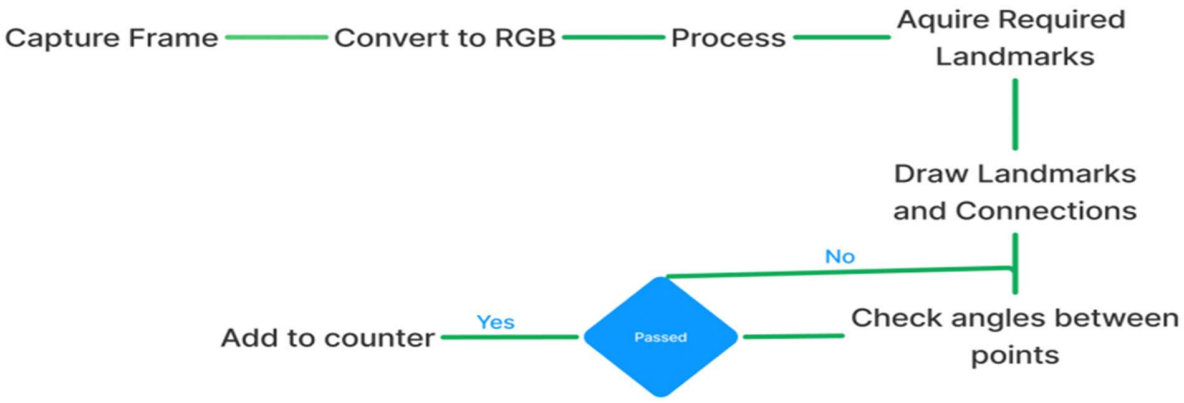
Functional requirement:

- System must have a menu for selection of exercises
- System must count reps for each of the exercise
- System must not increment the count if exercise is performed in the wrong way
- System must efficiently use sound system to notify user the count of exercise

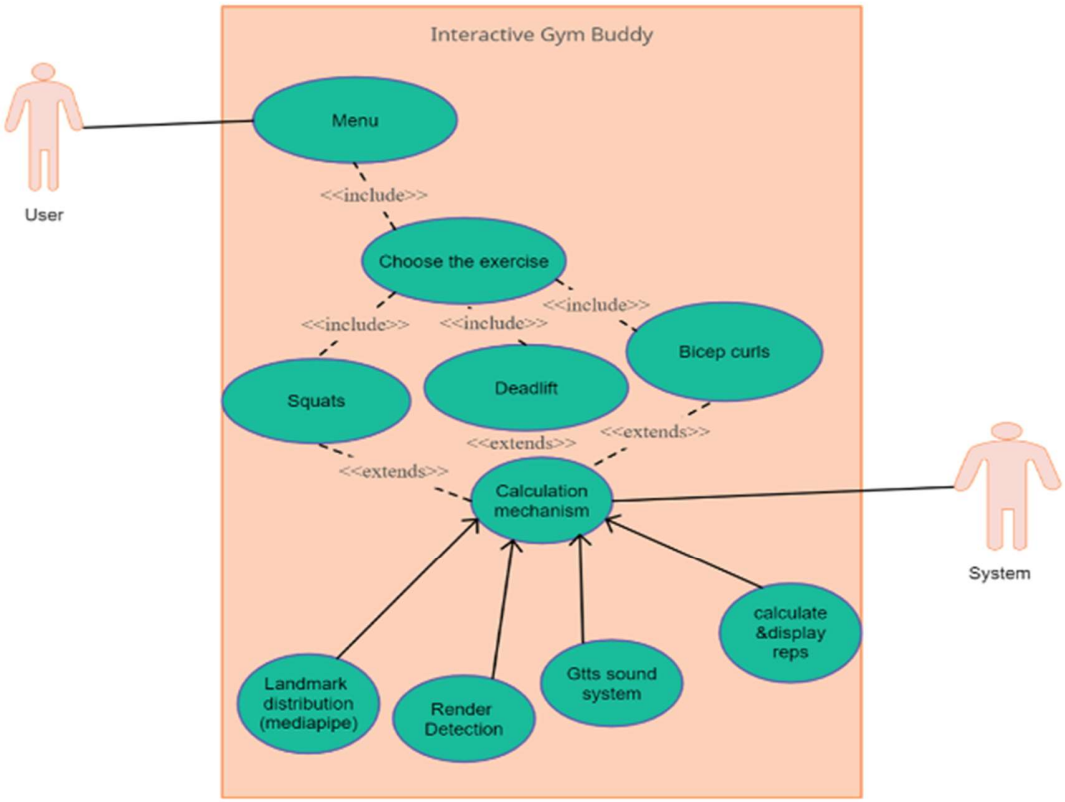
Non functional requirement:

- The tool must be responsive
- System should be able to count exercises without any lag
- System must detect every motion in top right corner to detect is user wants to switch between the exercises

4.Data Flow



Use case diagram



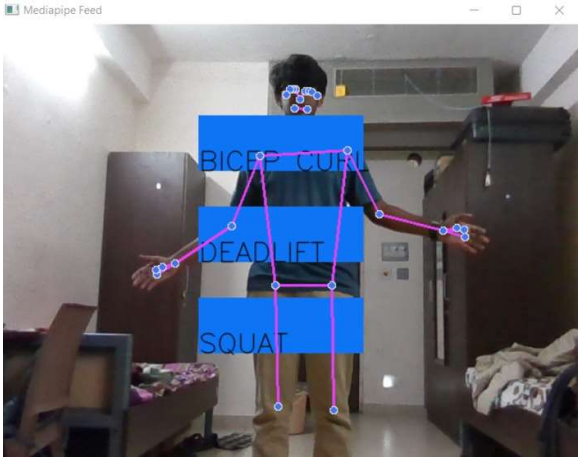
4. Nielson's 10 Heuristic evaluation & Shneiderman's 8 golden rules matching with UI

Shneiderman's 8 golden rules:

- *Strive for consistency*

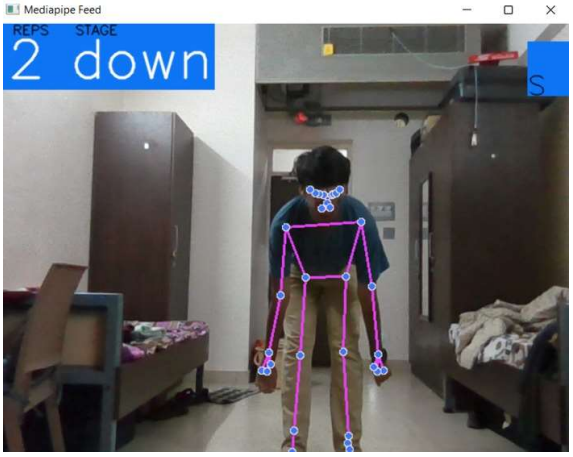
The interface is very consistent, it has a menu for exercises, you select any of the exercises using your hand gesture for the exercise you want to perform. If you want to go

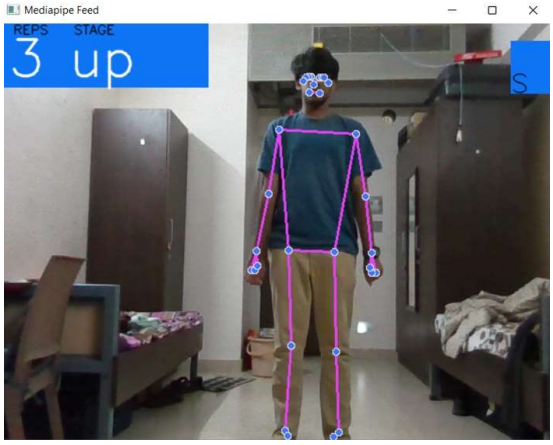
back to any other exercise, a button with an ‘S’ in the top right corner is.



- *Cater to wide range & type of Users*

System can be understood by even a novice user. It doesn’t limit the users with regards to their age and other factors.





- *Offer Informative feedback*

The system only counts the reps that the user does correctly & tells you which state you are at-up or down during the rep.

- *Design Dialogs to Yield Closure*

Counts the reps that the user does properly & announces each of them, does not count if the posture is wrong during exercise.

- *Prevent Errors*

The system is very easy to use, it has only two main pages, one is with the exercise menu, the other one is when the exercise starts where number of reps are counted, so it is quite easy to understand and there is almost no chance for an error to happen.

- *Permit easy reversal of actions*

If a wrong exercise is selected by the user, by just gesturing towards the ‘S’ in the top right corner, users can access the exercise menu again to select any of the exercises they want.

- *Support internal locus of control*

The user has full control over what exercise to do, change in between exercises and does not feel lost while using this system.

- *Reduce short-term memory load*

System is very self-explanatory, options can easily be recognized and the user does not have to remember anything.

Nielsen's ten heuristics

- *Visibility of system status*

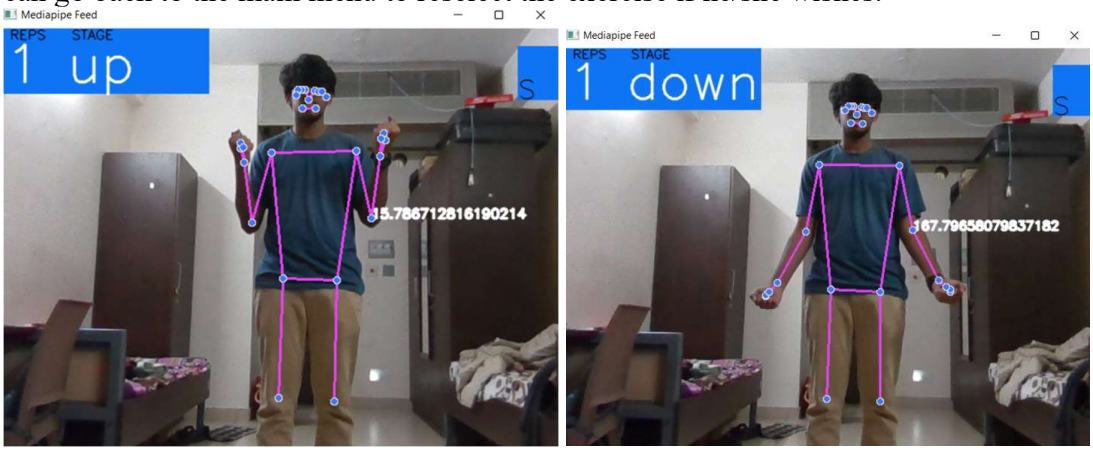
Since this is a real time system, the status of the system is displayed along with the actions of the user.
As a user completes a set of a specific exercise, he/she will be shown whether he/she is doing it correctly at the same time. If the user does the exercise correctly, the count of the specific exercise will increase, else it will stay constant.

- *Match between system and real world*

The system is easily understandable to even the novice users, as it involves zero to no difficult terms and is mostly entirely visual-oriented and UI based.

- *User control and freedom*

Once the program has started, full control is passed to the user. The user must decide what exercise to do, and if he does the exercise correctly, the count of the exercise increases. As this is a real time system, the user cannot undo his/her exercises. The user can go back to the main menu to reselect the exercise if he/she wishes.



- *Consistency of standards*

The system is easily understandable to even the novice users, as it involves zero to no difficult terms and is mostly entirely visual-oriented and UI based.

- *Error prevention*

Since this is a UI based system, and the user only needs to perform an exercise to obtain results, there is a minimal chance of an error occurring.

- *Recognition rather than recall*

Again, this is an entirely UI based system, and does not need the user to do a lot and to navigate much, as all the work is done in a minimal fashion. There is one menu which the user can choose his/her exercise from, and do the exercise, and then go back to the menu by clicking on the S icon in the top right.

- *Flexibility and Efficiency of use*

The system is very efficient, as it has a simple menu to choose exercises from, and a simple count value that increases if the user does the exercise correctly, and a S icon at the top right to take the user back to the menu to choose another exercise.

- *Aesthetic and minimalist design*

The application is a UI based system that does not need much navigation, and the user remains on the same window throughout his/her use. It has a simple design, and does require a user to have complicated knowledge to use the system.

- *Help user recognize, diagnose and recover from an error*

As this is a real time system, error correction can unfortunately not be possible. However, the user may know if he/she has done the exercise incorrectly if the count does not increase.

- *Help and Documentation*

As such, no separate documentation has been made for this project, but ample information about the project has been provided in the github repository of the project.

5. Testing

Test Cases:

- Smooth flow of events through hand gestures.
- Proper count of valid reps of an exercise.

6. Implementation

Tech Stack:

Software specification

● Python 3.5.0 or above

Libraries

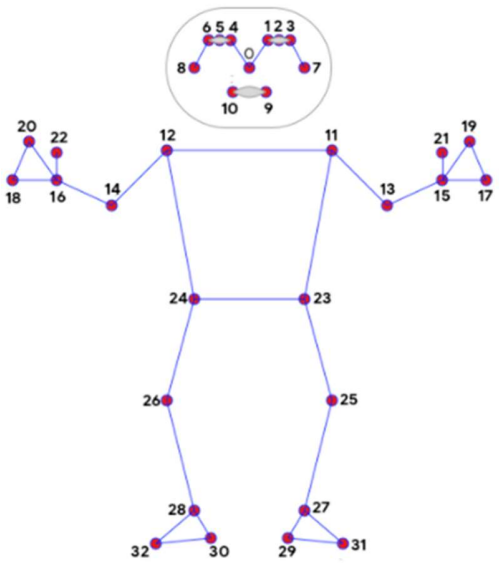
- OpenCV
- MediaPipe
- Numpy
- Playsound
- Gtts

Hardware specification

- Webcam
- 4GB RAM or above

MediaPipe and OpenCV Implementation:

We will be using Mediapipe for body pose tracking and body segmentaion. It segments the subject to 33 3D landmarks. These landmarks are tracked and it also uses z-index analysis to measure depth as in to indicate whether user's hands are behind their hips or not.
This utilizes two step detector-tracker proven to be effective in Media Pipe hands and face Mesh solutions. Using the detector, it locates the person's region-of-interest(ROI). Then proceeds to predict the pose landmarks and segmentation mask within the ROI using the ROI-cropped frame as input.



0. nose
1. left_eye_inner
2. left_eye
3. left_eye_outer
4. right_eye_inner
5. right_eye
6. right_eye_outer
7. left_ear
8. right_ear
9. mouth_left
10. mouth_right
11. left_shoulder
12. right_shoulder
13. left_elbow
14. right_elbow
15. left_wrist
16. right_wrist
17. left_pinky
18. right_pinky
19. left_index
20. right_index
21. left_thumb
22. right_thumb
23. left_hip
24. right_hip
25. left_knee
26. right_knee
27. left_ankle
28. right_ankle
29. left_heel
30. right_heel
31. left_foot_index
32. right_foot_index

We have also added the gtts sound system, to incorporate added functionality where, the system announces the reps done by the user for ease of use.

Repetition Counting

To count the repetitions, the algorithm monitors the probability of a target pose class.
Let's take push-ups with its "up" and "down" terminal states:
When the probability of the "down" pose class passes a certain threshold for the first time, the algorithm marks that the "down" pose class is entered.
Once the probability drops below the threshold, the algorithm marks that the "down" pose class has been exited and increases the counter.

7. Conclusion

This project proves very useful for the people who don't have much time on their hand due to their busy lives but want to take care of their health, the system can be accessed whenever its convenient for them to help them. The future scope of this project is that it can be upgraded to support more exercises, a mobile application can be made for convenient access, personalized workouts according to the users' weight and height can be added to the application, the interface can be added in outdoor gyms and places to increase access and save money.

8. References

Code: https://github.com/zubairatha/AI-Gym-Trainer

https://google.github.io/mediapipe/solutions/pose.html

https://scholarworks.calstate.edu/downloads/n009w777f