PRDv1

Team Name: Pretty Lil Leetcoders
Project Title: LogMyMotion
Team members: Sung (Sam) Kim, Nikhil Patil, Etgar (Eric) Freilafert, Yuyang (Ethan) Su, Aryaman (Arvan) Das

Intro

Problem

Working out and fitness is essential to both physical and mental health; however, not everyone is aware of how to go about lifting weights the correct way. Progress is often difficult for beginners who have little to no knowledge about safe and efficient workout technique. As a result, many people end up getting injured or becoming stagnant in their strength growth due to poor form. In order to progressively become stronger, one must ensure that proper form is established early on in the training process. The solution is to provide an application that can give feedback on form to ensure that workout exercises are done in a safe and effective manner.

Innovation

We will utilize a joint detection machine learning model that allows us to keep track of key cues that will indicate a proper training form for our exercises. This requires us to develop a system that can analyze the performance of the repetitions that our trainees perform.

This will be accomplished by examining the squat form of professional trainers and weightlifters using joint detection and creating measurable guidelines that can be utilized to analyze the form of our trainees. By applying these guidelines to a trainee’s program, we can ensure that a trainee would never strain their body and injure themselves by forming bad habits early on in the program. In addition to joint detection, we will leverage audio recognition to track breathing patterns to ensure every component of the exercise is done a proper way. In our app, we innovate joint detection and audio recognition to accomplish a holistic evaluation about a trainee’s execution of weight-lifting exercises. We understand the importance and safety of performing these exercises correctly. Another important aspect to successful workout repetitions is proper breathing. Irregular breathing can hinder good form and cause greater stress on muscles during exertion.

Team Goals/Objectives

Our goal is to develop a web based platform where users can keep track of their training progression and receive feedback that maximizes the efficiency of their workouts while addressing issues that lead to injury. Our application should help guide users to be able to live fit lives and prevent users from being discouraged by the tricky learning curve of lifting weights.

Stretch goals include creating a system that analyzes and tracks users performance and has the ability to leverage data from video analysis to provide feedback that encourages progression on the user's end. Some long term features we would like to implement are challenges and a social media aspect to our application.

Soft goals

Server should be running 24/7 and powerful enough to not lag. AI speech should be realistic and smooth

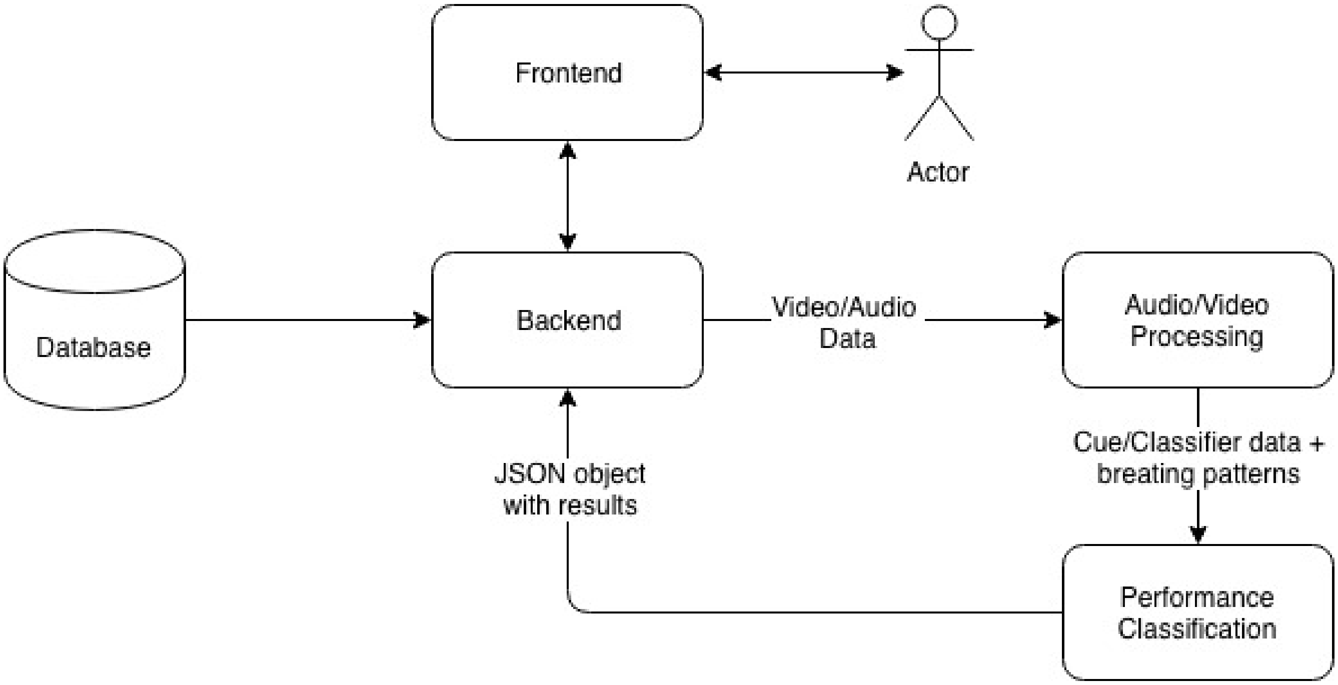
Background

Modern solutions to ensure proper form while exercising include hiring a personal trainer or watching YouTube videos created by professional bodybuilders. However, the former costs money and the latter doesn't ensure that your form will be as good as the professional bodybuilders in the video. Our application hopes to bridge this gap by using machine learning to analyze videos of specific exercises and provide feedback on main classifiers used to determine proper form.

Assumptions

Our product targets weightlifters that want a way to track progress and receive feedback on videos that they will upload to our application themselves. Due to PoseNet's limitations, we assume that users will record exercises in specific angles and that users will not be wearing clothes that blend in with the gym's background. For the purpose of audio recognition, we will assume that users will be wearing mic-in headphones.

High level diagram of System Architecture



Use case: Login
Actors: User, User Account Database.
System Precondition: User must be in the login page.
Flow of Events: Basic Path: 1. User types in username/password. 2. System checks the database to confirm if user is valid. If the user credentials are valid, then the user information will be loaded.
Alternative Paths: If the user credentials are not valid, an error will be thrown and the user will be asked to enter a valid username/password.
Postcondition: User should now be in the user's home page of our web application.

Use case: Upload video
Actors: User, User Account Database
System Precondition: User has logged in.
Flow of Events: Basic Path: 1. User clicks upload video. 2. User chooses video from computer (or cloud drives). 3. Upon confirming, video will now be stored in system database to be analyzed.
Alternative Paths: If the file size is too big than our system accounts for, an error message will be thrown and the user will be asked to upload a file within the size limit.
If the upload fails, an error will be thrown and the user will be asked to try again.
Postcondition: User should be able to see the video he/she uploaded in her uploaded videos either in an analyzed or to be analyzed state.

Use case: Analyze video
Actors: System Workout Analyzer
System Precondition: There exists a video in the to be analyzed state in the system's database.
Flow of Events:
1. System retrieves a video to be analyzed in the database.
2. System applies our algorithm to determine the grade of the workout.
3. System will update the state of the video and upload analysis of the workout to the respective user's database.

Alternative Paths: If the algorithm fails for some reason, an error will be thrown and the user will be informed that the analysis was not performed.
If the video is not found in the database, an error will be thrown that the video was not found in the database.
Postcondition: An analysis of the workout video will be shown in the respective user's page.

Use case: Delete video
Actors: User, User account database.

System Precondition: A user has uploaded a video before and is currently showing in the uploaded videos page.
Flow of Events: Basic Path: 1. User selects video from uploaded videos. 2. Upon confirmation of video chosen to be deleted, system will remove the video from the database.
Alternative Paths: Video is not found in database
Postcondition: Selected video will no longer appear in user's list of uploaded videos (as well as the database).

Use case: Logout
Actors: User, User account database.
System Precondition: A user has to be logged into their account.
Flow of Events: Basic Path: 1. User clicks the logout button. 2. The user is brought back onto the home page.
Alternative Paths: If the user does not log out, a cookie would be used to set a timer for how long the user could be long on for.
Postcondition: The cookie should be deleted and the user should be logged out and be on the homepage.

Use case: Progress
Actors: User, User account database
System Precondition: User has uploaded sufficient number of videos to be analyzed
Flow of Events: User opens progress page. System analyzes data from user videos in relation to each other. System informs user of his/her progress, and whether the user is recommended to increase weight.
Alternative Paths: If the user has not uploaded sufficient number of videos for progress to be checked, the user will be notified that they need to upload more videos.
Postcondition: User is redirected to homepage. New progress bar is set.

Use case: Create an Account
Actors: User, User account database
System Precondition: User does not have an account already for a specific email.
Flow of Events: User clicks on button to create an account. User enters information. Account is created and added to backend storage system.
Alternative Paths: If the email already has an account associated with it, the user is informed that an account already exists.
If the user's password is invalid, the user is informed to create a password within the required constraints.
If the system fails to create an account for the user, the user is informed that account creation failed.
Postcondition: User is brought into the application signed in with their account.

Use case: Create/Edit Profile
Actors: User, User account database.

System Precondition: User just created their account and has not set up the profile. User wants to make changes to their profile.
Flow of Events: User goes to their profile. User changes their info on their profile. User saves the profile.
Alternative Paths: User's profile fails to update, so user is notified. User cancels out from editing a profile.
Postcondition: User has updated their profile information

Use case: Review Feedback File
Actors: User, User Account Database
System Precondition: User has created
Flow of Events: User has logged into their account
Alternative Paths: Feedback file is not found, so error is thrown and user is notified.
Postcondition: Analysis of feedback file is cleaned and displayed to user of the application.

Use case: Analyze Audio
Actors: System
System Precondition: The video has been uploaded to the server
Flow of Events: 1. System retrieves audio file to be analyzed in the database.
2. System applies our algorithm to determine the grade of the breathing.
3. System will upload analysis of the workout to the respective user's database.
Alternative Paths: Audio file is not found, so error is thrown and user is notified that their audio was not found.
Postcondition: User is able to view analysis of their breathing on their homepage.

USER STORIES/GIT COMMITS:

1. **Be able to get a stream of coordinates of joints from PoseNet.**
2. **Be able to authenticate users.**
3. **Get frontend running.**
4. **Get database running.**
5. **Be able to detect the knees over toes classifier.**