Project plan

1. Project name

Real-time AI Fitness Monitoring System

2. Project Description

Based on human posture, we can develop various applications, such as fitness or yoga trackers. As an example, we showcase an AI Squat Counter, which can automatically track user data or verify the quality of the performed exercise. Such use cases can be achieved by using additional classifier networks or simple pairwise distance lookup algorithms that match the closest pose in a normalized pose space. This project uses the MediaPipe toolkit, where MediaPipe Pose is an ML solution for high-fidelity body pose tracking, leveraging our BlazePose research to infer 33 3D landmarks and background segmentation masks for the entire body from RGB video frames. One of the applications that BlazePose can achieve is fitness, specifically posture classification and repetition counting. Push-ups and squats, as the most common exercises, will be used for demonstration.

1. Project Plan
2. Phase 1 (4-6 weeks)

Learn MediaPipe Pose Detection Toolkit

MediaPipe is a multimedia machine learning application framework developed and open-sourced by Google Research. At Google, a number of important products such as YouTube, Google Lens, ARCore, Google Home, and Nest have deeply integrated MediaPipe. MediaPipe has a wide range of applications, including object detection, selfie segmentation, hair segmentation, face detection, hand tracking, motion tracking, and more. Based on this, more advanced functionality can be achieved.

1. Phase 2 (6-8 weeks)

Create a dataset and collect image samples of the target exercises.

In the preparatory stage, we will discuss which exercise counts need to be performed, which technology to use for recognizing and estimating exercise postures, and perform data collection. To build a good classifier, appropriate samples should be collected for the training set: in theory, there should be several hundred samples for each final state of each exercise (e.g., the "up" and "down" positions of push-ups and squats). The collected samples should cover different camera angles, environmental conditions, body shapes, and motion variations. In practice, we will take about 15-25 shots of each state, and when shooting, we should pay attention to diversifying the angles, taking one shot every 15 degrees.

1. Phase 3 (8-9 weeks).

Annotate the dataset by labeling the images and generating a label file (CSV file).

1. Phase 4 (9-10 weeks)

Write the code for the human pose classification module.

For the classifier, we decided to choose the simple and easy-to-use k-nearest neighbors algorithm (k-NN) as the classifier. This algorithm determines the category of an object based on the closest samples in the training set, rather than based on the angular characteristics between the limbs of each exercise. Therefore, this method has good generalization and versatility, and can be widely used to classify given postures in exercises such as squats and push-ups.

1. Phase 5 (10-11 weeks)

Writing code for the counter module.

To calculate the repetition count, the algorithm monitors the probability of the target pose category. For example, in the case of the "up" and "down" end states of a squat, when the probability of the "down" pose class first exceeds a certain threshold, the algorithm marks the entry into the "down" pose class. Once the probability drops below the threshold (i.e., the person rises above a certain height), the algorithm marks the "up" pose class, exits, and increments the counter.

1. Phase6 (11-13 weeks)

Write code for detecting and counting movements in videos, and for real-time detection and counting of movements using a camera.

1. Phase7 (13-15 weeks)

During the testing phase, the code for each component is integrated and tested. Bugs are identified based on error messages and fixed one by one. Finally, a final summary is made.