AI Virtual Fitness Assistant

Ioannis Ioannidis
Dimitrios Patiniotis Spyropoulos

University of Piraeus
NCSR Demokritos
Athens

01-07-2022

Outline

Purpose

The present project addresses the idea of introducing a virtual fitness assistant. We will train a classifier using machine learning and data available on YouTube. The ML classifier will be used in order to build a web-app, which will be keeping the records of a live feed workout.

Contents

- Introduction
- Dataset Description
- Methodology
- Results

Introduction

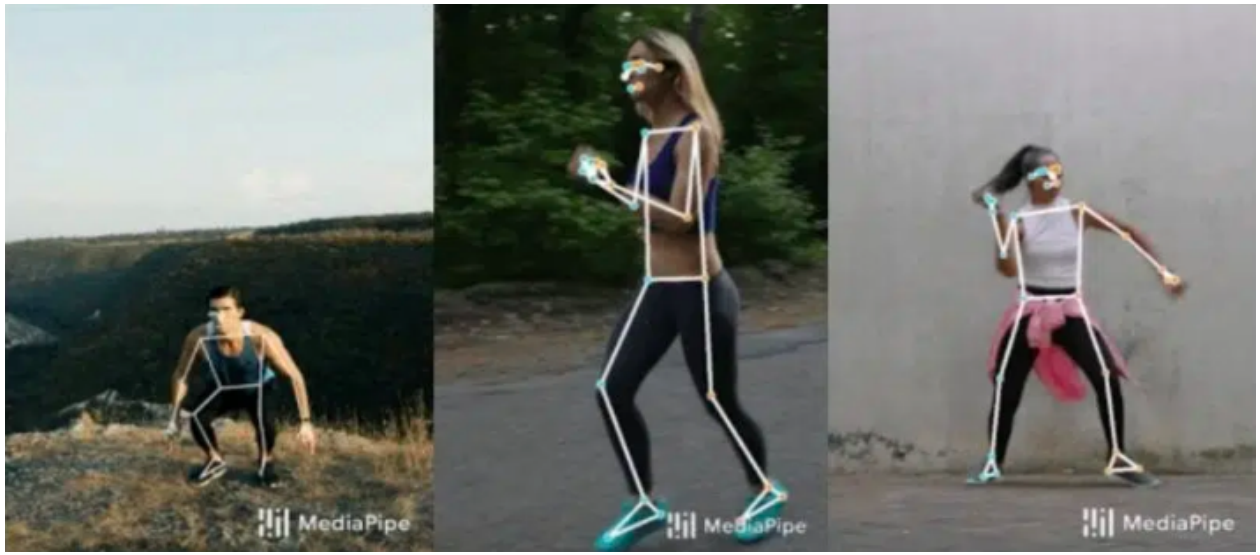
Motivation

- Pandemic and lockdown has turned staying at home to be part of our everyday lifes in the last years.
- However we need to keep on with our daily routine with respect to this limitation.
- Working out constitutes an integral and healthy part of such a routine and it is obvious that it is intensively affected by the social isolation.
- In this project we tried build a tool which would report the full workout that someone is running at home, just by using the camera of the computer.

Dataset Description

Data Source

- We gathered video data available on YouTube.
- We searched for three exercises : pull-ups, squats and plank by searching hastags (#) with relevant keywords.
- None of the videos information became public, as we just used them to extract the coordinates about specific body parts.



Dataset Description

Final Data

- We ended up having numerical features about the body parts, when a specific exercise is performed.
- It is remarkable that the created dataset is scaled, as all coordinates are measured in the same system.
- We used a sample size of 8-10 videos for each exercise and ended up with 67 features of 9290 observations.

	LABEL	NOSE	LEFT_EYE_INNER	LEFT_EYE	LEFT_EYE_OUTER	RIGHT_EYE_INNER	RIGHT_EYE	RIGHT_EYE_OUTER	LEFT_EAR	RIGHT
0	Squats	0.504951	0.514962	0.521597	0.527880	0.490489	0.480893	0.472264	0.533444	0.4
1	Squats	0.526823	0.528318	0.531981	0.535665	0.513403	0.505798	0.497616	0.533351	0.4
2	Squats	0.531405	0.533030	0.536469	0.539779	0.518286	0.510728	0.502580	0.533579	0.4
3	Squats	0.527622	0.530922	0.535011	0.538833	0.513843	0.505464	0.497221	0.533540	0.4
4	Squats	0.529673	0.532290	0.536041	0.539568	0.516019	0.507488	0.498053	0.533145	0.4
...	...	...	...	...	...	...	...	...	...	
9285	Plank	0.221354	0.207233	0.211548	0.215775	0.195234	0.191813	0.188909	0.227866	0.1
9286	Plank	0.221485	0.207246	0.211566	0.215792	0.195373	0.191997	0.189132	0.227863	0.1
9287	Plank	0.224056	0.209296	0.213489	0.217575	0.197375	0.193857	0.190875	0.229311	0.1
9288	Plank	0.224449	0.210191	0.214352	0.218385	0.198087	0.194433	0.191319	0.229955	0.1
9289	Plank	0.224714	0.210370	0.214518	0.218552	0.198340	0.194704	0.191587	0.230111	0.1

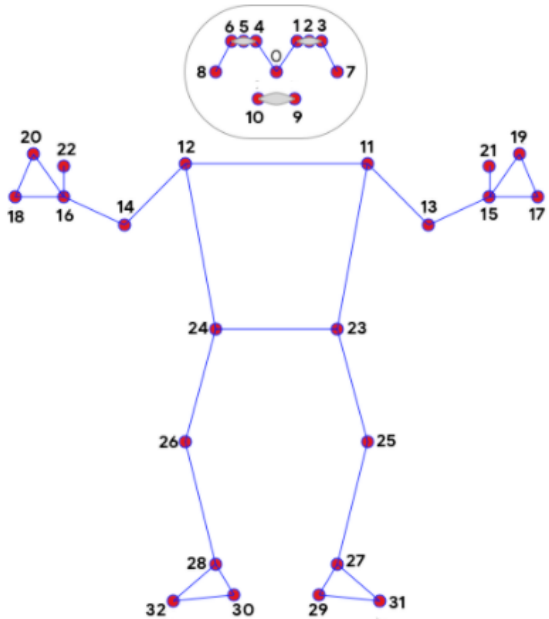
9290 rows x 67 columns

OpenCV Background

- OpenCV is an open-source library mainly used for computer vision, image processing, and machine learning, which provides great output for real-time data.
- We can process images and videos so that the implemented algorithm learns to identify objects such as cars, traffic signals, number plates, bodies, faces, or even human handwriting.
- With the help of other data analysis libraries, OpenCV is capable of processing the images and videos according to one's desire.

MediaPipe Background

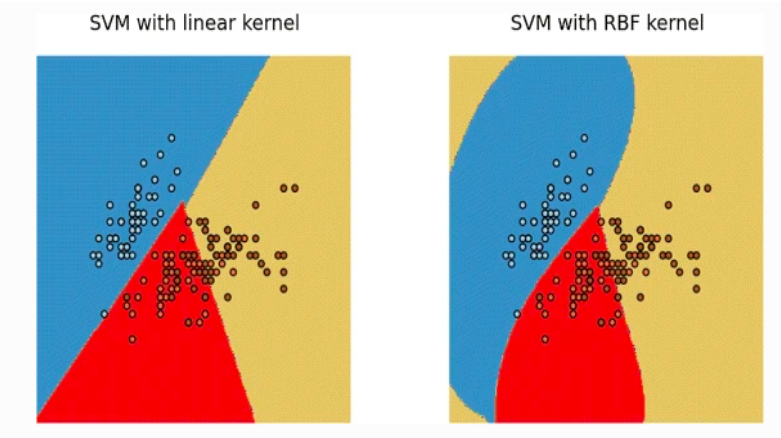
- Mediapipe is a framework mainly used for building multimodal audio, video, or any time series data.
- MediaPipe Pose is a framework for high-fidelity body pose tracking, which takes input from RGB video frames and infers 33 3D landmarks on the whole human.



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

SVM Background

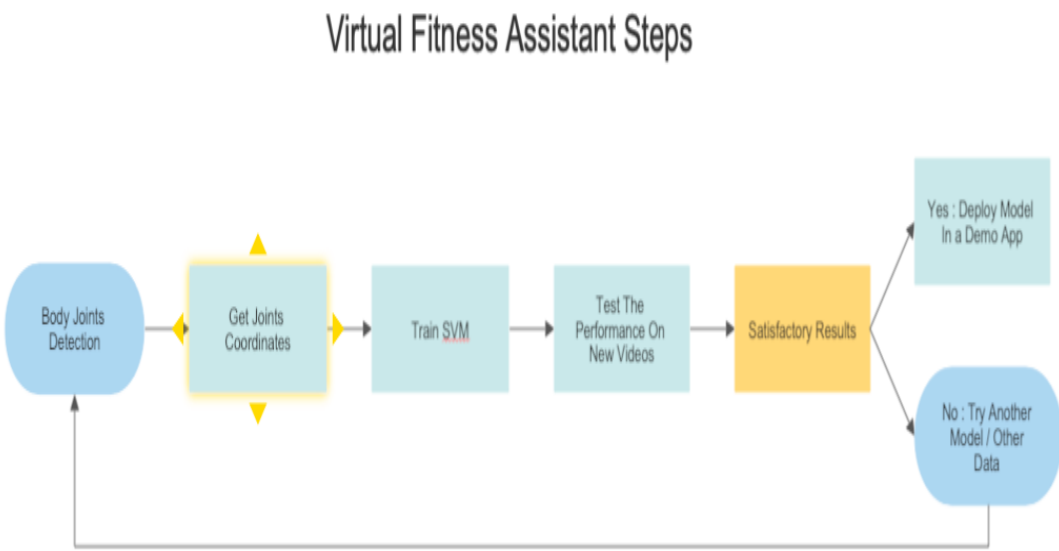
- The objective of the support vector algorithm is to find a hyperplane in an N-dimensional space that distinctly classifies the data points
- To separate the two classes of data points there exist many possible hyperpanes
- SVM finds the one that has the maximum margin
- Maximizing the margin distance provides some reinforcement so that furure data points can be classifies with more confidence



Methodology

Workflow

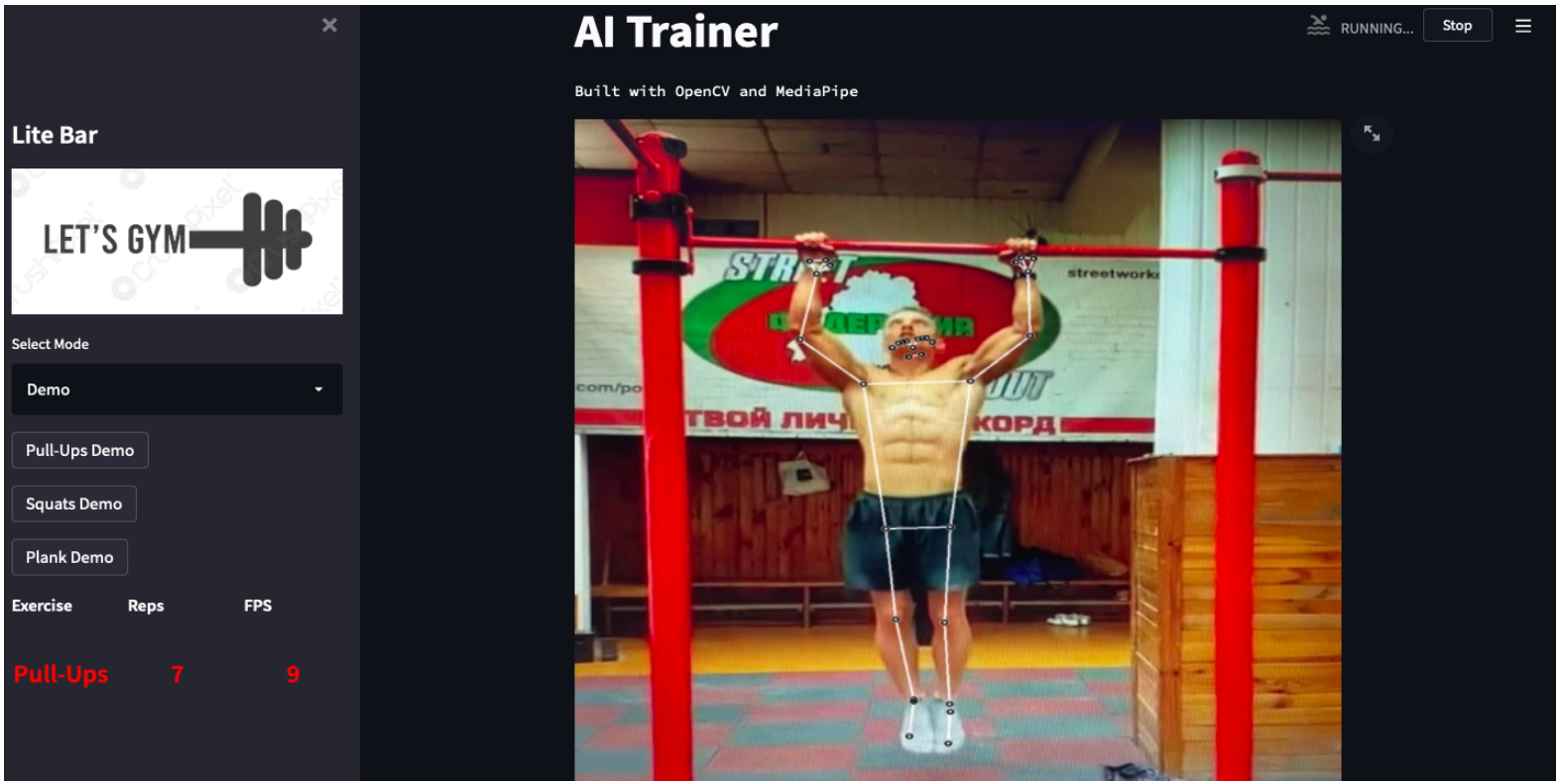
1. Step 1: Detect a person from a front side view and identify each one of the 33 joints that MediaPipe provides.
2. Step 2: Using the position of those joints, we train and test a Support Vector Machine model to classify which exercise is performed.
3. Step 3: Count repetitions using angle thresholds and states "UP" and "DOWN".



Results

Web App Environment

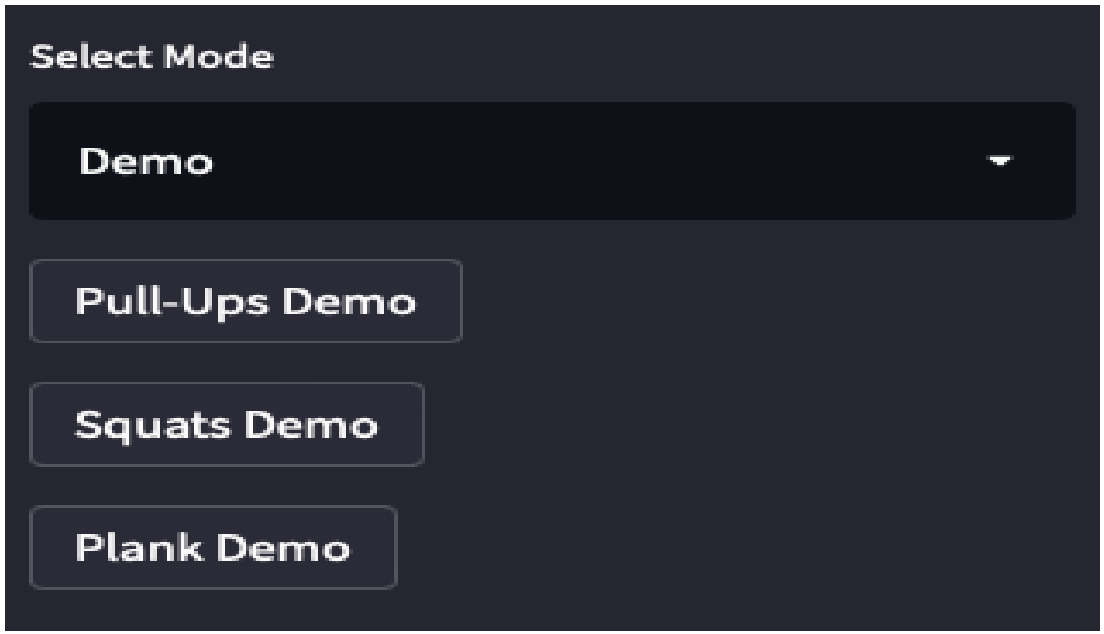
- We used this model to deploy a web-app, which counts the repetitions of any video displaying pull-ups, squats or plank, achieving good performance.
- Our app has 3 modes : *Demo*, *Your Video*, *About*



Results

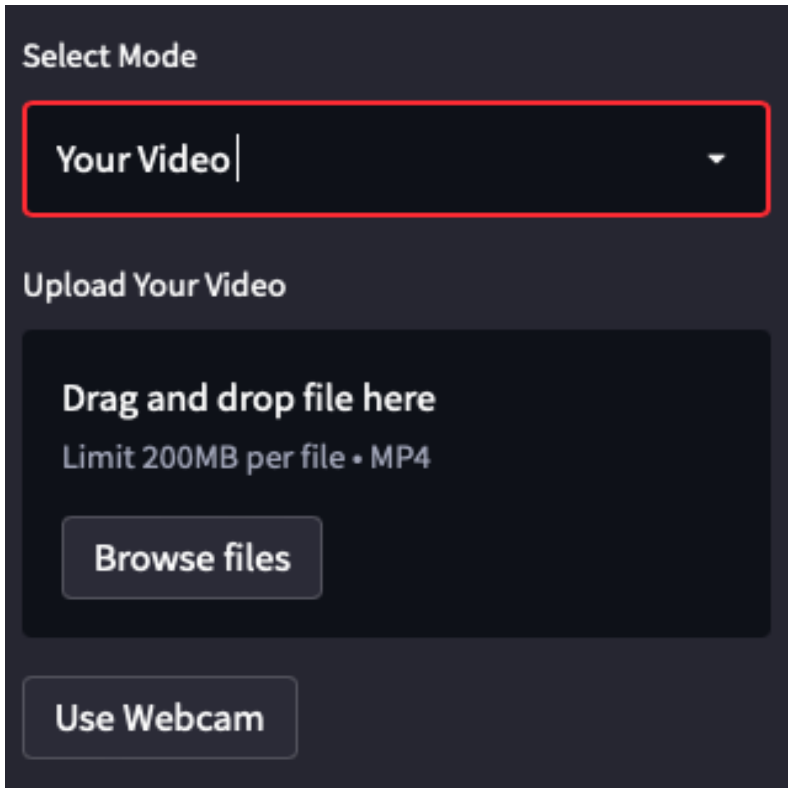
Demo Mode

- We created *Demo* feature so that a user can see how the app works, without having to upload any videos or use webcam.
- In this mode user has the opportunity to select between displaying a pull-up, squat or plank video.



Your Video Mode

- *Your Video* mode is the main feature, where the user can test the app himself/herself, either by uploading his/her video or by using webcam.



Results

Your Video Mode

- Results are previewed in real time, while video is running.

About Mode

- Finally, in the *About* mode user can see the purpose we ran this project and some proposed future work.