MAGIC

CV/ML ENGINEER TEST

CV/ML ENGINEER TEST

This test has been designed to assess 2 key areas of Computer Vision Machine Learning engineering, each critical for the role at Magic Tech:

1. Extracting Pose Data Patterns From Pre-Recorded Video Footage
2. Comparing Pose Data From A Video Feed To A Pre-Analysed Pattern

There are absolutely no UI requirements for these tasks. Solutions can be executed completely via console.

Understanding Our Device Functionality

The core function of our Magic Mirror devices is to:

A. Analyse a user's pose while exercising via a front-facing camera feed
    - *Pose estimation is processed via 'MediaPipe Pose', an ML solution providing 33 body pose coordinates in each frame of a video feed.*

B. Calculating how close their movements are to the correct body positioning
    - *This corresponds to point (**2**) above; the body pose coordinates provided must be analysed and compared to the correct exercise pose*

C. Counting their repetitions (reps) of the exercise
    - *Again corresponding to point (**2**) above; the body pose movements must be analysed to count exercise repetitions*

For example, a user may choose to select a workout which involves star jumps:

A. MediaPipe Pose analyses camera feed and returns user's live pose data

B. User may be asked to widen their stance on each jump

C. The number of star jumps will be counted and notified once 10 are complete

For all workouts and exercises, we have partnered with and recorded footage of famous athletes carrying out various exercises. Extracting pose data *(using MediaPipe Pose)* from that footage will provide a basis for ML algorithms to compare user's exercise poses to *(corresponding to point (**1**) above)*, which is naturally required in order to process **B** and **C**.

The following tasks will test your ability to build a drastically simplified version of the above.

TASKS

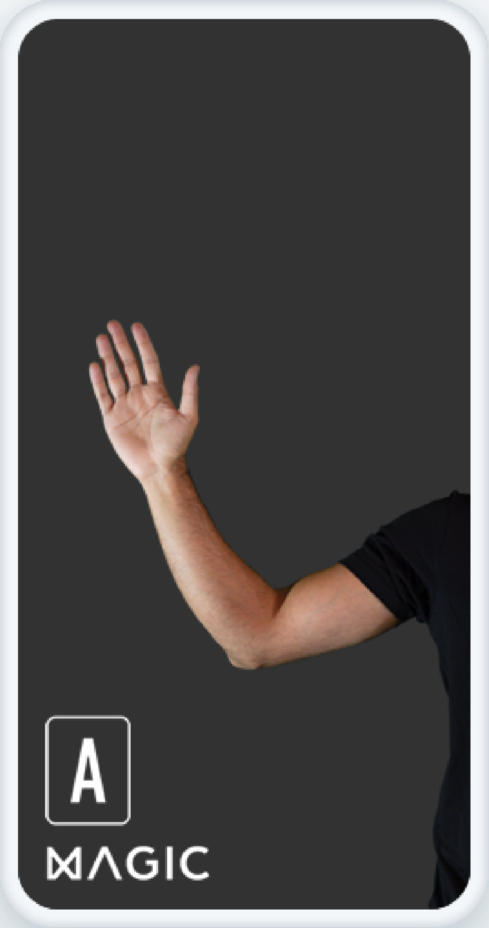
For the following tasks, you will be asked to develop an algorithm that counts the number of times an individual waves their hand in a video.

You are provided with two video files:

A. Video of a person waving 5 times at a fairly consistent speed, to be converted into Pose estimation data and used as the basis of a waving pattern;

B. Video of a person waving 10 times at inconsistent speeds, to be analysed and fed into an algorithm that counts the number of waves in real-time;

You are welcome to use existing ML solutions if you wish, or code your own maths-based algorithms to analyse patterns from data outputted by MediaPipe Pose.



1. Extract Pose Data Patterns From Video A

For the first task you are required to:

- Process **Video A** with MediaPipe Pose and extract Pose Estimation data;
- By writing your own algorithms or utilising an existing ML solution, analyse the Pose data to infer a pattern for a 'wave'

The output of this task should result with a function that can accept, over time, a parameter containing an array of changing Pose Estimation values outputted from MediaPipe Pose that should return an array of two values:

1. The % completion of the current 'wave' action
2. The number of times a 'wave' has been completed

2. Analyse Pose Estimation Data From Video B

For the second task you are required to:

- Feed **Video B** into MediaPipe Pose and pass the output into your function from Task 1
- Print to console the output data of your function as the video is being processed frame by frame

Once this is successfully working, please produce a screen recording of the data being printed in the console.

REQUIREMENTS

Programming Language

There is no specific coding language required for these tasks. The design of your solution is more important here.

Keep It Lightweight

It's extremely important your solution is a lightweight algorithm that can run smoothly on a mobile device. As you'll discover or already know about MediaPipe Pose, their lightweight models are designed to run on mobile devices, so you must ensure your solution will not hinder that advantage as Magic Mirror Devices are also low specification similar to mobile devices.

SUBMISSION

The following will be required in your submission::

A. Link to a public git repo containing your code;
B. Video of your working solution as outlined in Task 2;
C. Description & reasoning behind the algorithms designed & utilised in your solution

Your Git repository should contain all your files, including the video file, with your ReadMe outlining the locations of key files in your directories alongside your answer to **C**.

Your submission is to be emailed to: **sunil@magic.fit**