EEE F411

INTERNET OF THINGS

FIRST SEMESTER 2021-22

YOGA POSTURE DETECTION APPLICATION

SUBMITTED TO PROF. VINAY CHAMOLA

PREPARED BY-

MAYANK VISHNOI - 2019A7PS0098P

AMARTYA PANDEY - 2018B2A80689P

DIVYANK VYAS - 2019A7PS0092P

TABLE OF CONTENT

ABOUT

Pose estimation is a machine learning task that estimates the pose of a person from an image or a video by estimating the spatial locations of specific body parts (keypoints).

It was aimed for a raspberry pi device with a webcam. After further exploration, we decided to make it a web application which would be cross-platform. We have created a full-fledged web application so no external interfacing was required.

There is no need to attach additional sensors as the live recording of the person would be carried out by the webcam on the laptop itself.

Build flow steps:

1) Dataset collection and augmentation
2) Training and testing the ML model
3) Front-end of the application

The Dataset was simply collected from free image websites on the internet.

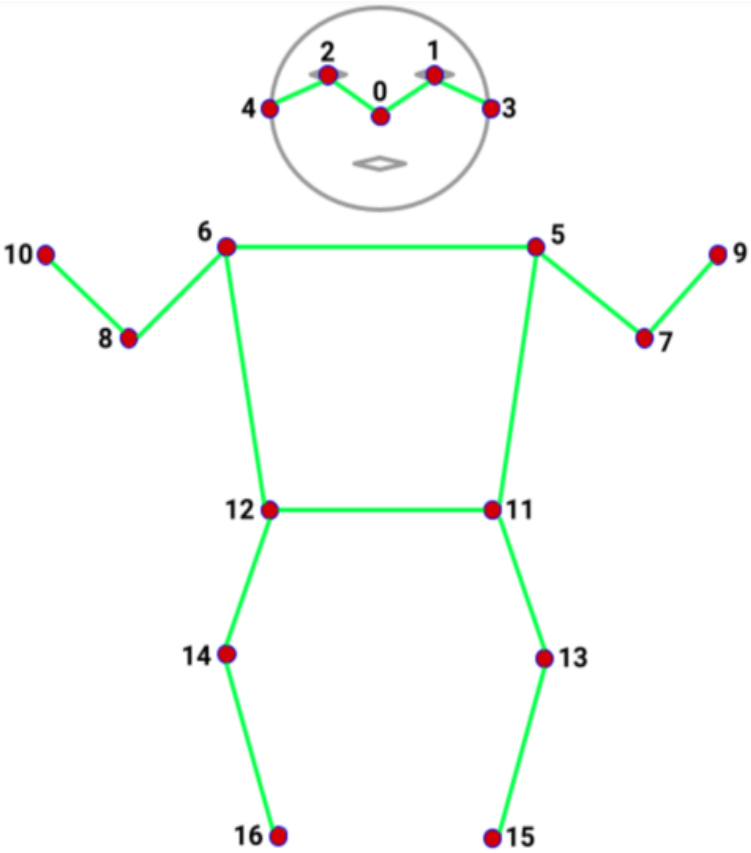
We have used MoveNet as the pose estimation model that can detect 17 key points on the body. It can run in real-time on most devices including smartphones.

The steps after this were as followed:

- ➔ Imported movenet.py- for pose estimation model. (https://blog.tensorflow.org)
- ➔ Preprocessing.py - generated .csv file from dataset.
- ➔ Training.py - trained the ML model and created model.json. This was uploaded on Cloud so to avoid running a separate backend server.
- ➔ The following dataset was used for the training of the ML model.
- ➔ Finally, the frontend of the application was fully developed using React.js.

Key-Points: Used in PoseNet

0: nose
1: left_eye
2: right_eye
3: left_ear
4: right_ear
5: left_shoulder
6: right_shoulder
7: left_elbow
8: right_elbow
9: left_wrist
10: right_wrist
11: left_hip
12: right_hip
13: left_knee
14: right_knee
15: left_ankle
16: right_ankle

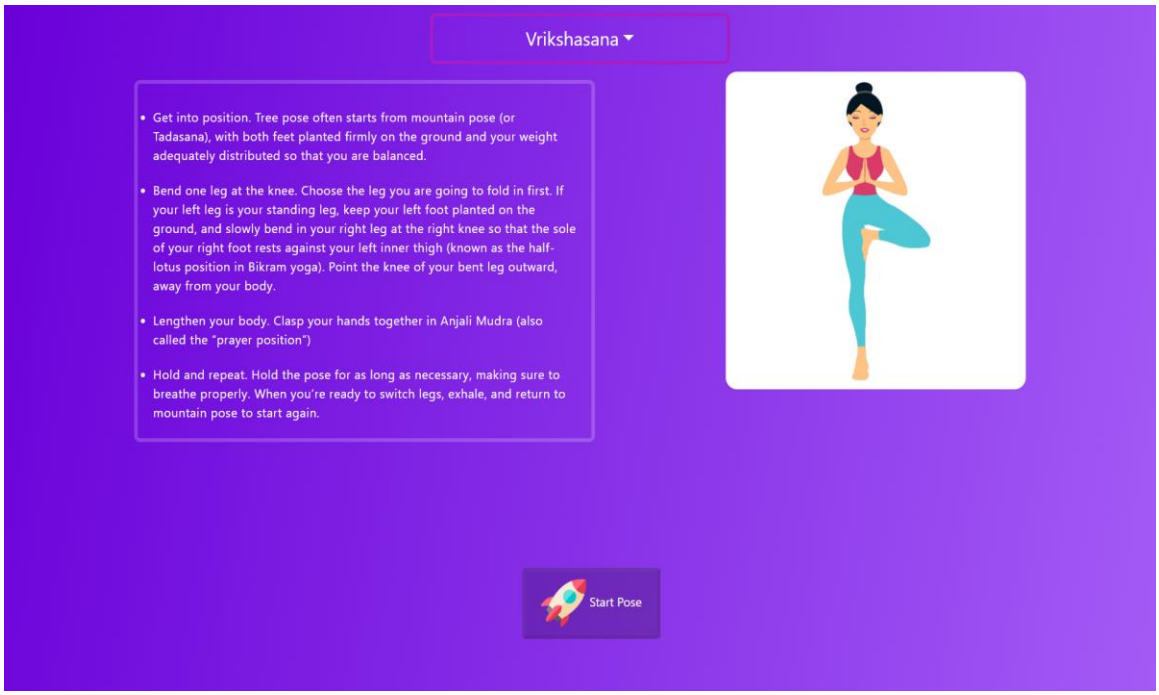


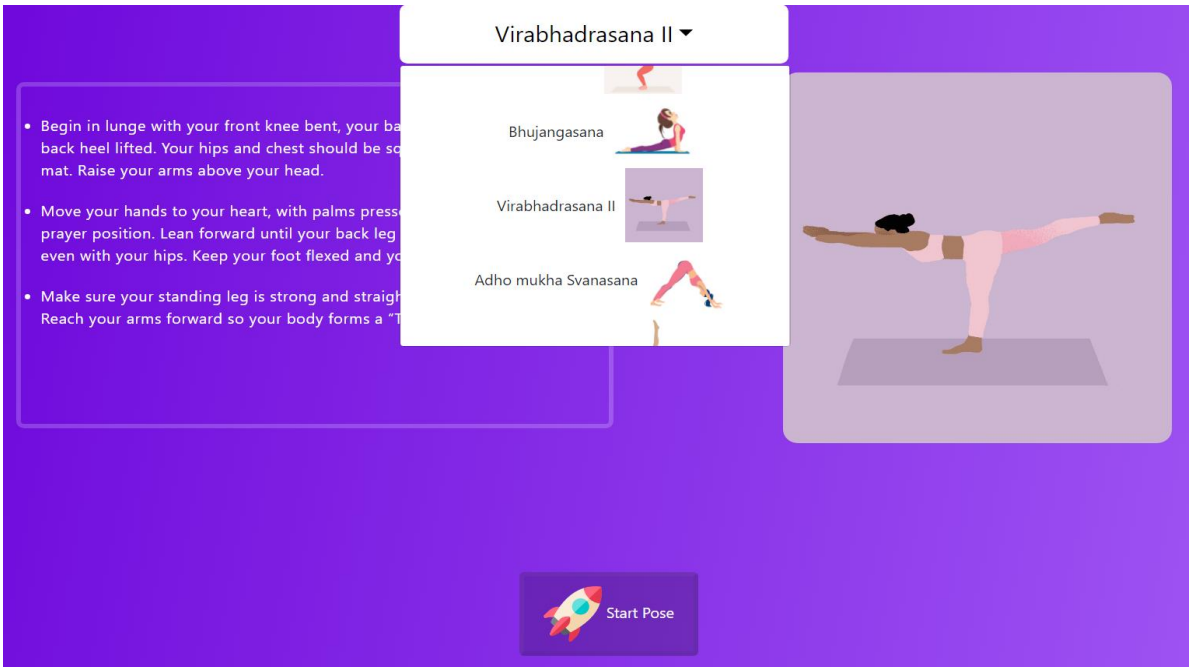
WORKING OF THE APPLICATION

Here are some snapshots of the original application:

This is the landing page of the application. Here, the user can either choose to go ahead with the pose detection or read the “About Us” section.

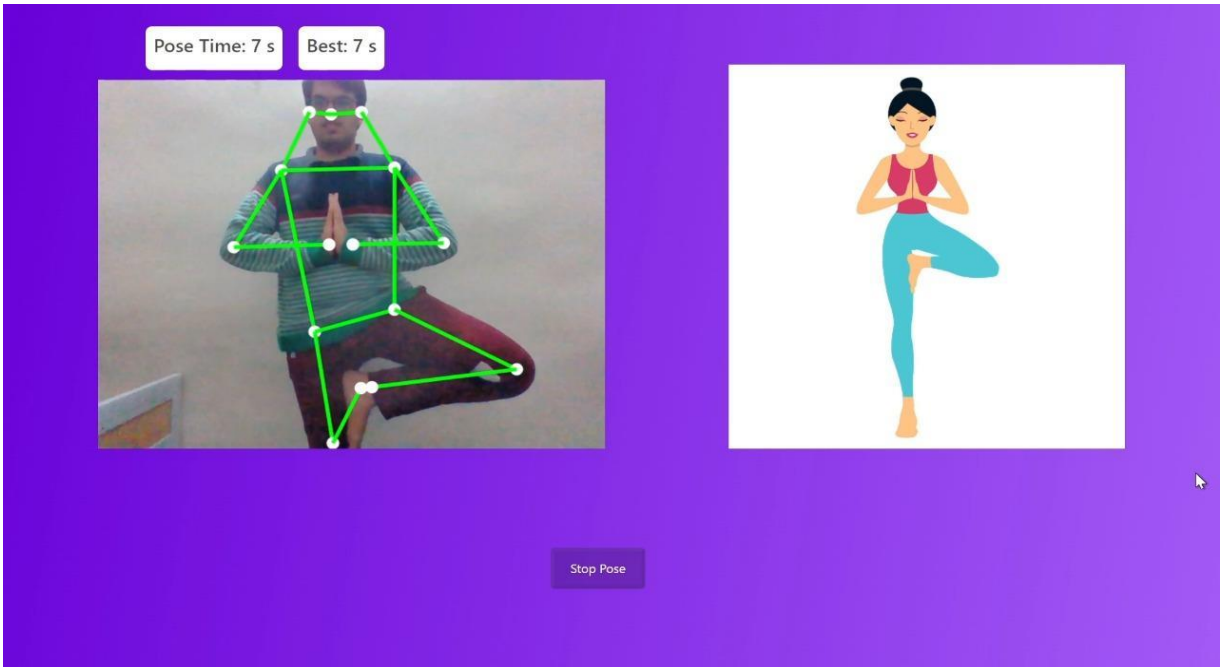
On clicking the “Let’s Start” button, the page shown below appears which shows the pose and description of the aasan which is there to help the user get accompanied with the *aasana* and there is a drop-down menu from which the user can select the available *aasana.*





The user can choose any type of *Aasana* and click on “Start Pose” to start doing the poses.

It will require the user to permit the use of a webcam on his/her web browser. After clicking the “Start Pose” button. The following screen will appear.



On holding the proper position, the timer will start, and the grid lines will become green suggesting that the optimal position has been achieved and the user should hold that position.

FUTURE SCOPE

This device was meant to be solely used for Raspberry Pi devices. But on exploring the practical aspects, we made it as a web-based application which could run on any platform. Due to time constraints, we could not provide some features to it and we would continue developing this application on the same lines.

- ➔ Number of Yoga poses can be increased.
- ➔ Dataset used for training, validation and testing of the ML model can be increased to get better results.
- ➔ On the backend a simple database could be added to register the date and time of the user on a daily, weekly and monthly basis.
- ➔ Furthermore, some more features like Calorie counter, weight and BMI calculator could be added.
- ➔ When hosted on the internet a feature of login could be added, which will enable many users to use this application.
- ➔ Some video tutorials (long term goal) or YouTube links (short-term goal) could be added to assist the user.
- ➔ Full Yoga Workout with Virtual trainer using AI.

REFERENCES

- ➔ POSE ESTIMATION - TENSORFLOW.ORG
- ➔ PROJECT POSENET GITHUB REPOSITORY
- ➔ ALIGNED YOGA APP GITHUB REPOSITORY
- ➔ MSCOCO KEYPOINT EVALUATION METRIC

RESEARCH PAPERS:

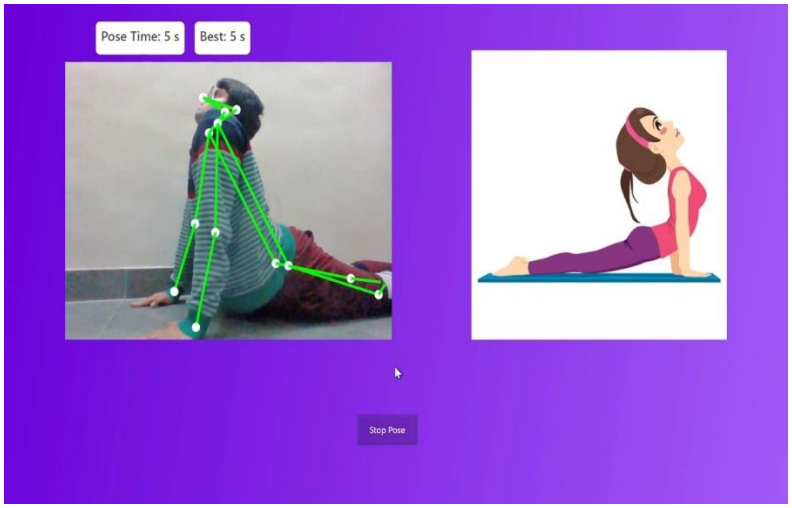
- ➔ Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields

  Zhe Cao, Tomas Simon et. al.

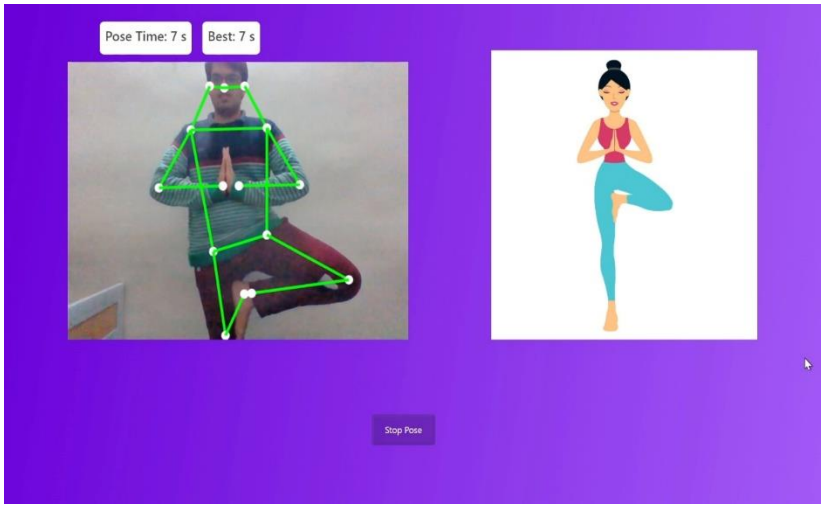
- ➔ 2D human pose estimation: new benchmark and state of the art analysis

  M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele

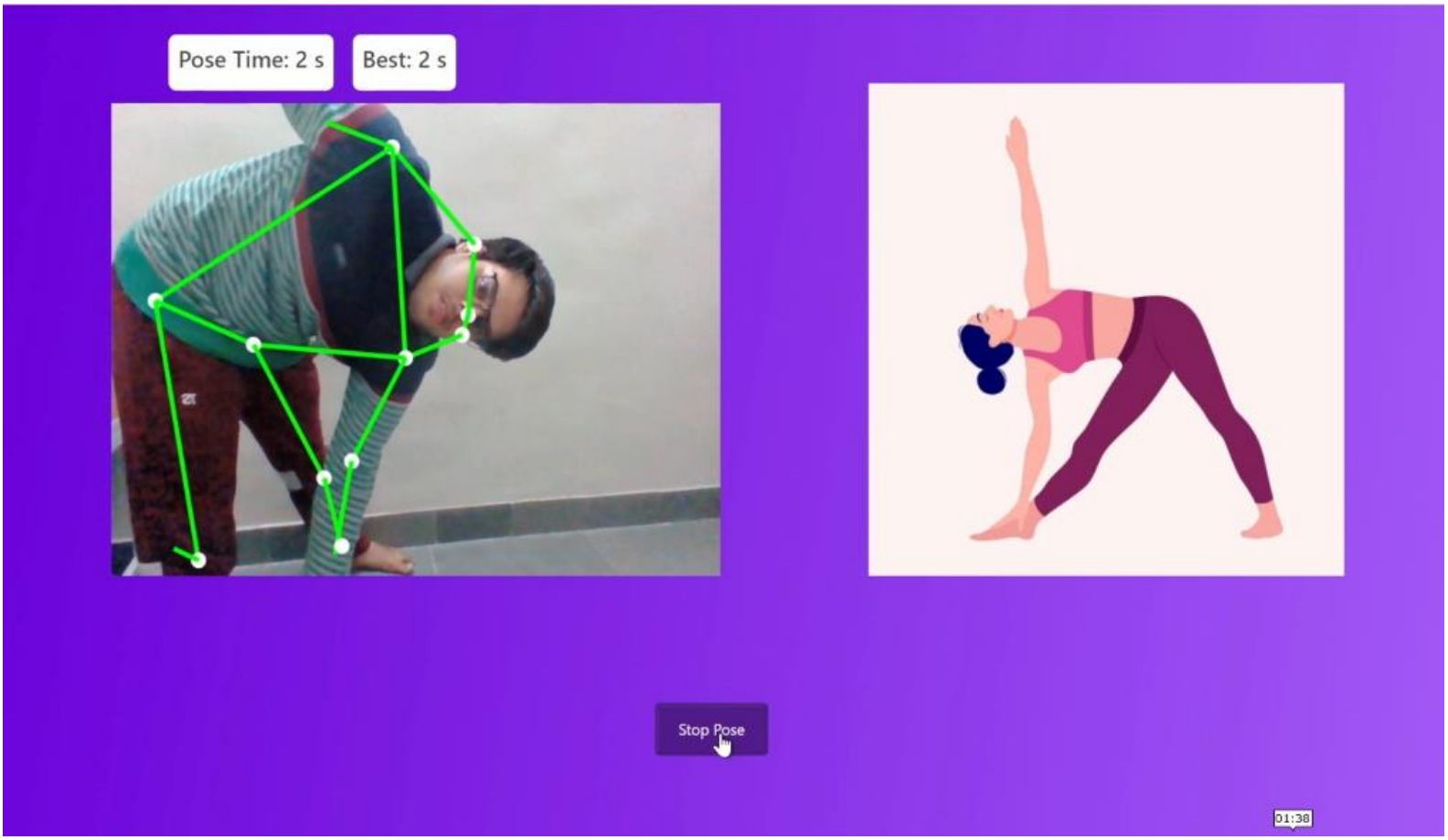
ANNEXURE- A: SOME DEMONTRATIONS OF DIFFERENT POSES



BHUJANGASANA



VRIKHSHASANA



TRIKONASANA