

# Yoga Posture Detection Application

Instructor: Dr. Vinay Chamola  
Group No: 11





# Introduction

• 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**.

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.



# Objective

The aim of this device is to help user perform perfect yoga poses without any help from yoga instructors. It also tracks the time for which the user holds the perfect pose. 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.

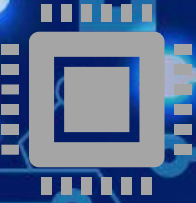


# MoveNet Demo





# Work Flow



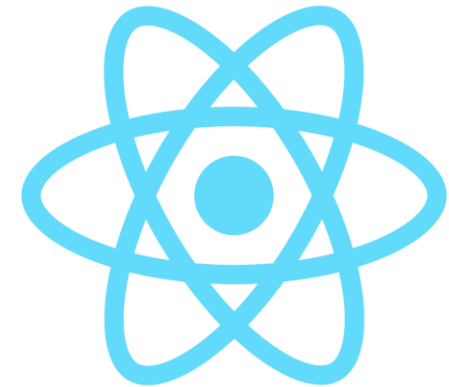
## Python

Training and testing of the model



## React

Developed the web Application using React Application



# Build Flow Steps

01

Dataset collection and augmentation

02

Training and testing the ML model

03

Front-end of the application

# Code

```
object to mirror  
mirror_mod.mirror_object  
operation == "MIRROR_X":  
mirror_mod.use_x = True  
mirror_mod.use_y = False  
mirror_mod.use_z = False  
operation == "MIRROR_Y":  
mirror_mod.use_x = False  
mirror_mod.use_y = True  
mirror_mod.use_z = False  
operation == "MIRROR_Z":  
mirror_mod.use_x = False  
mirror_mod.use_y = False  
mirror_mod.use_z = True
```

```
selection at the end -add  
obj.select= 1  
obj.select=1  
context.scene.objects.active  
Selected = str(modifier)  
obj.select = 0  
= bpy.context.selected_objects  
data.objects[one.name].select  
print("please select exactly
```

--- OPERATOR CLASSES ---

```
bpy.types.Operator):  
X mirror to the selected  
object.mirror_mirror_x"  
mirror X"
```

is not

# Implementation

- ❖ Movenet.py - for pose estimation
- ❖ Preprocessing.py - generated csv file from dataset.
- ❖ Training.py - trained the ML model and created json file.
- ❖ Developing the frontend of the application.



# Screenshots of Web App



*Internet of Things Project*

About Us

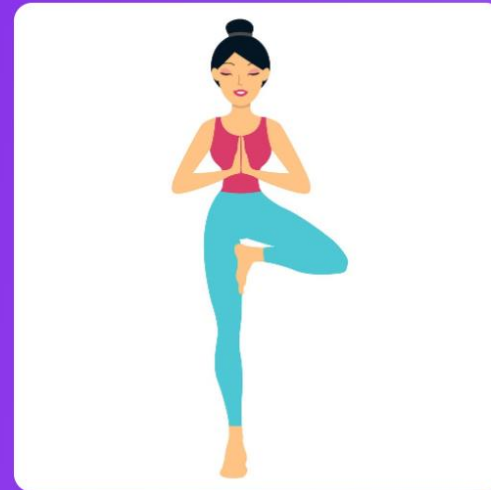
## Yoga Posture Detection



# Screenshots of Web App

## Vrikshasana ▾

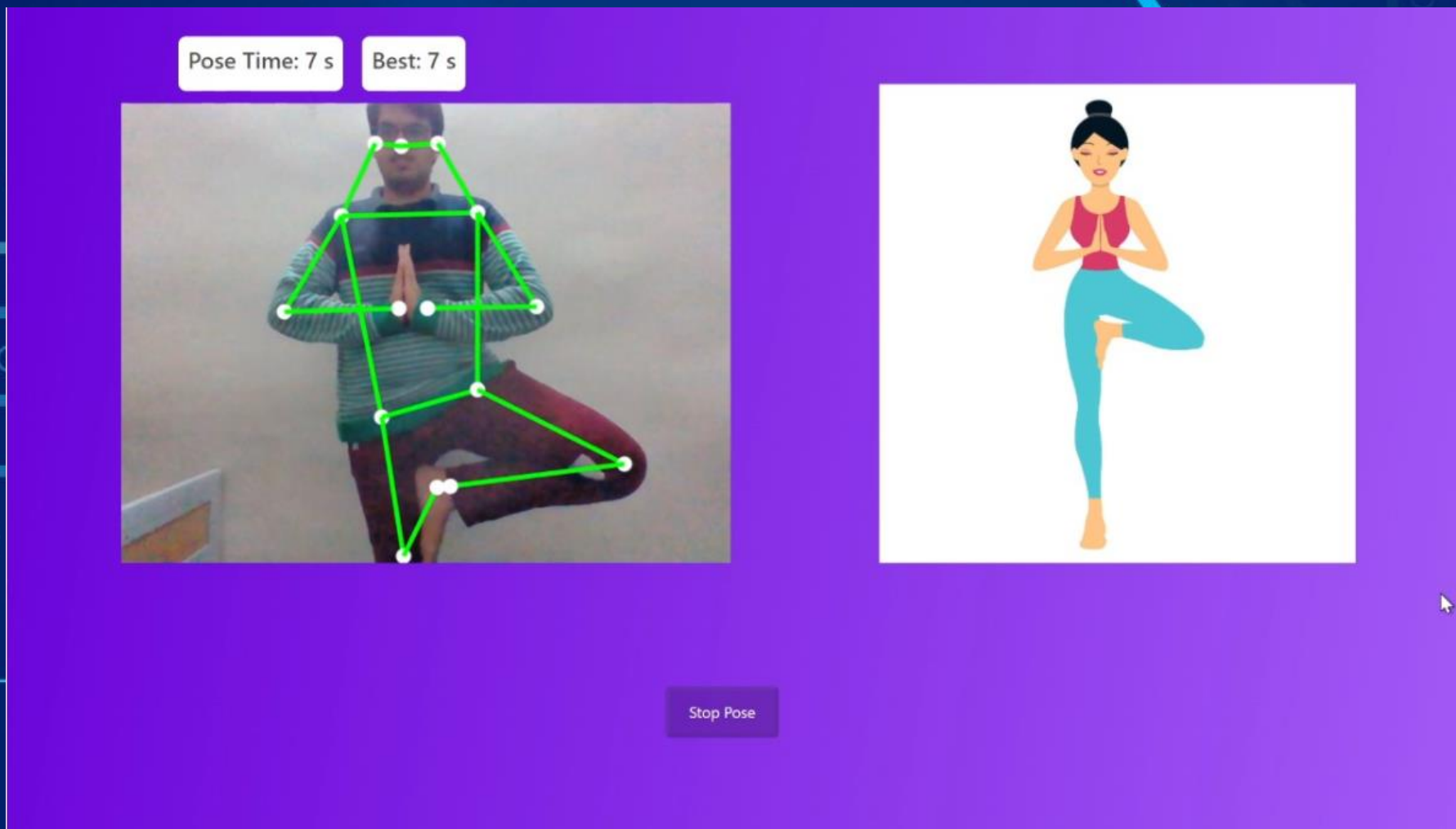
- Get into position. Tree pose often starts from mountain pose (or Tadasana), with both feet planted firmly on the ground and your weight adequately distributed so that you are balanced.
- Bend one leg at the knee. Choose the leg you are going to fold in first. If your left leg is your standing leg, keep your left foot planted on the ground, and slowly bend in your right leg at the right knee so that the sole of your right foot rests against your left inner thigh (known as the half-lotus position in Bikram yoga). Point the knee of your bent leg outward, away from your body.
- Lengthen your body. Clasp your hands together in Anjali Mudra (also called the "prayer position")
- Hold and repeat. Hold the pose for as long as necessary, making sure to breathe properly. When you're ready to switch legs, exhale, and return to mountain pose to start again.



Start Pose



# Screenshots of Web App



# Future Scope

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.



# Future Scope Cont'd..

- ❖ 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.

# Our Team

Divyank Vyas  
(2019A7PS0092P)

Amartya Pandey  
(2018B2A80689P)

Mayank Vishnoi  
(2019A7PS0098P)



The background is a detailed, high-angle view of a computer circuit board. A large, square, 3D-rendered chip is positioned in the upper left quadrant. The chip has a dark, flat top surface and a metallic, reflective border. The circuit board itself is covered in a complex network of fine, golden-brown lines representing circuit traces. Various other components, including smaller chips and capacitors, are visible in the background, though they are slightly out of focus. The overall color palette is dominated by the metallic gold of the traces, the dark grey/black of the chip's surface, and a cool blue/purple tint that gives the image a technological and digital feel. A white rectangular border is superimposed over the center of the image, framing the text.

**Thank You**