ARTIFICIAL INTELLIGENCE CSE3013

Winter Semester 2021-22

SCHOOL NAME

SCOPE

ITLE OF THE PROJECT

PERSONAL GYM TRAINER

TEAM MEMBER DETAILS

REG. NO.	TEAM MEMBERS (Name)
19BCE2473	Joel Shabu
20BCE2708	Padm Keshav
19BCE2474	Sulayam AR

Review

Problem Statement

Insufficient physical activity is recognized as one of the leading risk factors for death worldwide and can lead to a variety of health issues, including cardiovascular diseases, diabetes, cancer, and mental health conditions.

Over the last 1 to 2 years everyone has been confined to their homes due to the COVID pandemic. The gym is one public place which everyone uses on a regular basis as personal fitness is an important necessity in everyone's life. However, due to public places being shut, it has led to disruption in daily routine and exercise patterns for everyone. Hence, many people have struggled with various health-related problems such as obesity, irregular sleep patterns, eye strain, mental stress, decreased immunity, and hence, are at a higher risk of getting infected with various health problems. Gyms generally have a variety of equipment and personal trainers are always there who can tell you what to do. The lack of these in one's home can often be the culprit that stops them from working out.

Many people watch YouTube videos, have wearable tech and train with personal gym trainers. But all of them have their disadvantages. While watching YouTube videos, no feedback is given, and we must manually keep count of the exercises. This distracts the person performing the exercise as a lot of things need to be tracked and there is no mechanism for the same.

This is the problem our project aims to solve. We have developed a personal Gym Trainer using Artificial Intelligence. This gym trainer works on a real-time basis and keeps track of the exercises performed, repetitions in each exercise and the number of sets performed as well. However, it will also work if one uploads an already recorded video, but in order to make things easier we have decided to keep the monitoring in real time. Using our tool, users will be able to perform their daily exercises with ease from the comfort of their home without bothering too much about other things such as reps, sets, etc. We have added as many exercises as possible and have covered most of the upper body exercises in our research. Exercises such as left bicep curls, right bicep curls, right wrist rotation, left wrist rotation, etc. are some of the exercise patterns we have added into our project

Methodology

The assessment of human poses is sophisticated technology based on computer vision. It's like face recognition for the whole body. Human pose estimation systems detect and evaluate the posture of the human body using three analytical methods:

Skeleton modeling: This employs key points to depict the human body's skeletal system.

Contours modeling: This employs the body's raw breadth and extremities to display a person's figure's rectangular border boxes.

Modeling Volume: This analytical approach employs 3D body scans to capture the body using geometric meshes and forms.

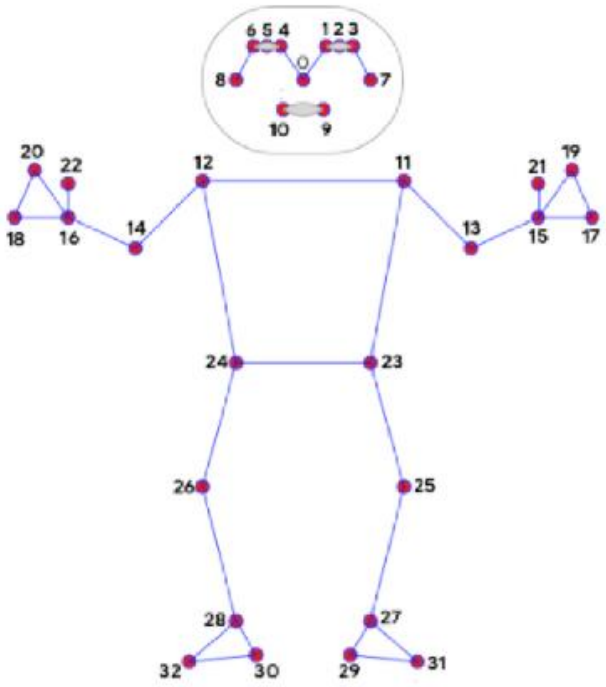
Here, we have used **mediapipe** library for pose estimation.

Mediapipe - MediaPipe is a Framework for building machine learning pipelines for processing time-series data like video, audio, etc. This cross-platform Framework works in Desktop/Server, Android, iOS, and embedded devices like Raspberry Pi and Jetson Nano.

Unlike power-hungry machine learning Frameworks, MediaPipe requires minimal resources. It is so tiny and efficient that even embedded IoT devices can run it. In 2019, MediaPipe opened up a whole new world of opportunity for researchers and developers following its public release.

For further Understanding - https://learnopencv.com/introduction-to-mediapipe/

Shown below are the points which have been given to us by the mediapipe library:



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

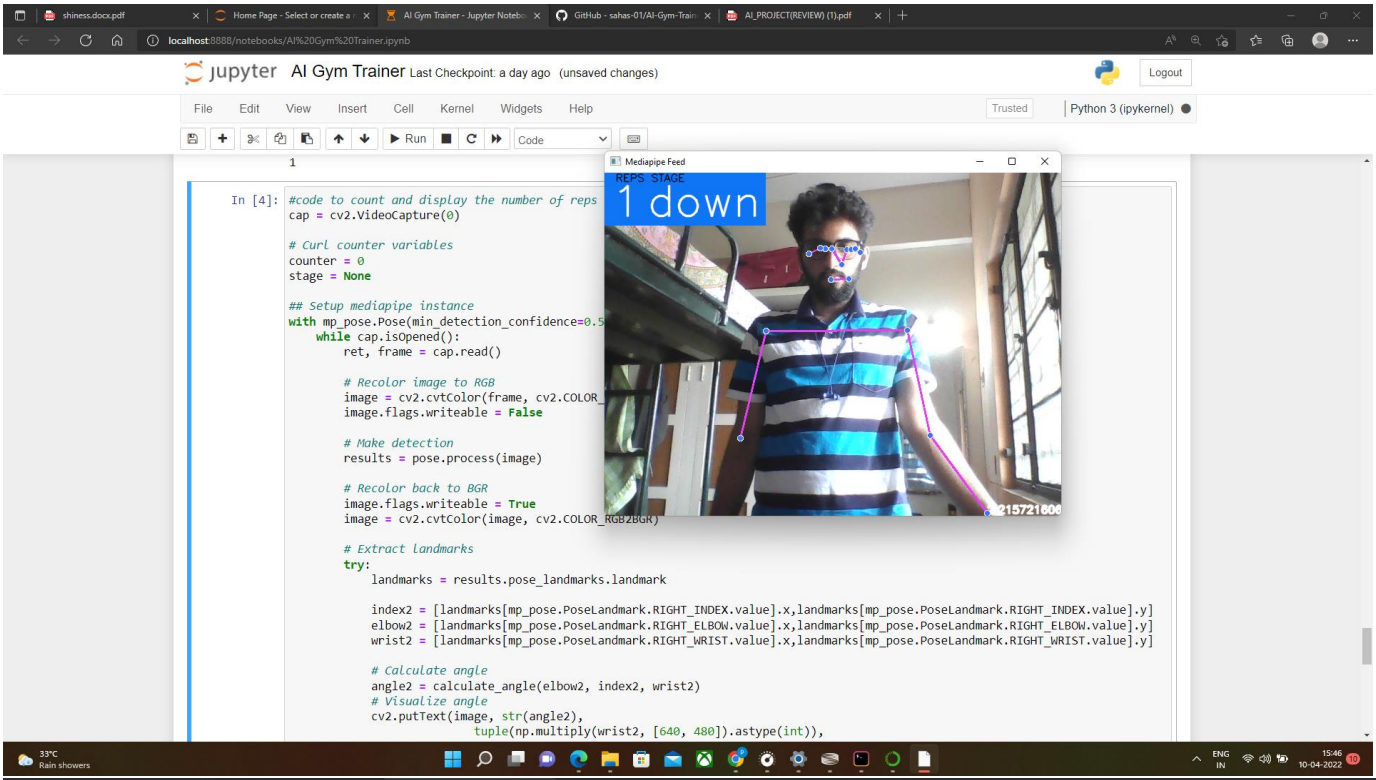
In our project, we have made use of **OpenCV** for capturing real-time feed. After capturing the real time feed, we made use of the mediapipe library to help in pose detection. Following this, we wrote a function to calculate the points in the upper half of the body again using mediapipe. Once the points were calculated, we took 3 random points (in this case – left shoulder, left wrist and left elbow were taken) and wrote a function to calculate the angle between these calculated points. Once this was done a custom function was written for the left bicep curl exercise, the methodology for all exercises remains the same, which has been explained below:

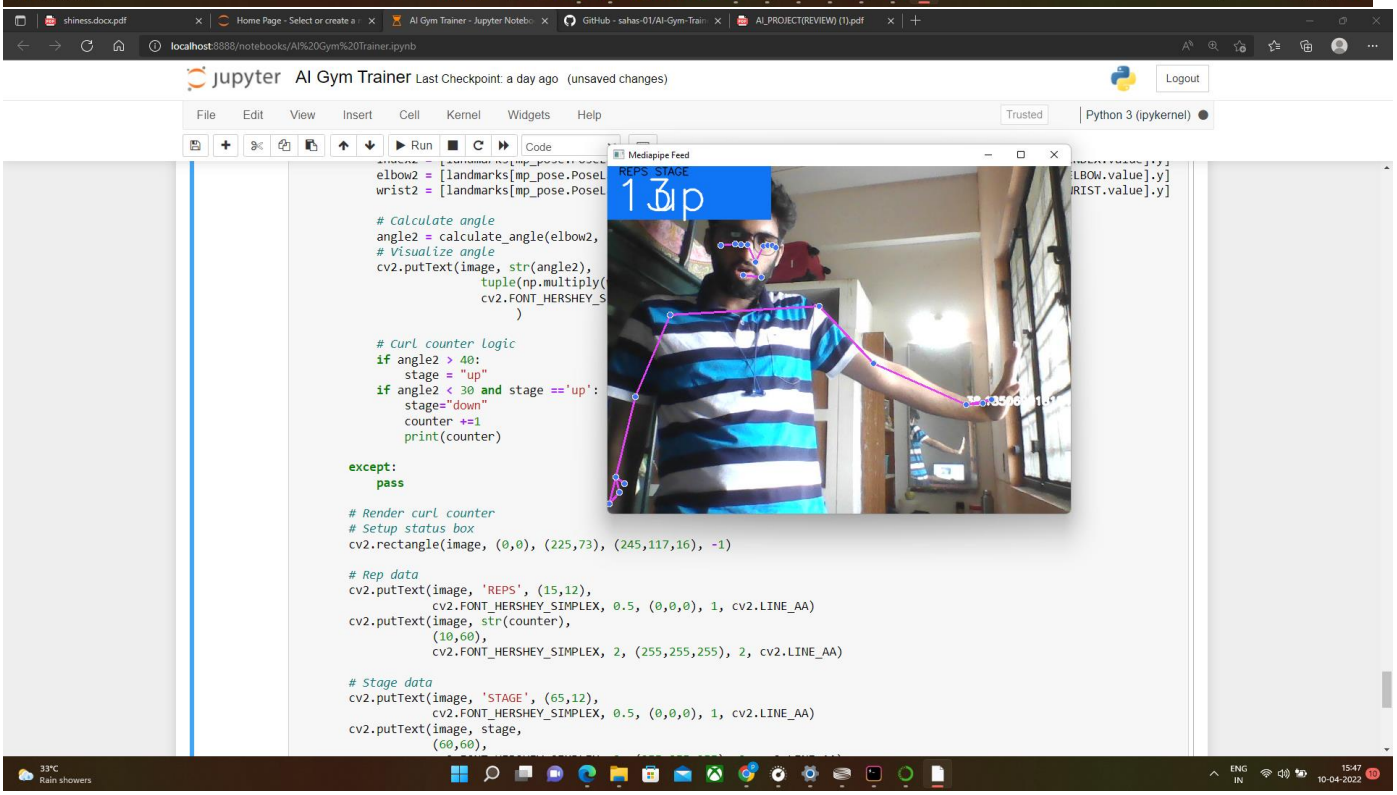
First all points are estimated and between the calculated points we find out the angle, if the angle is greater than a fixed number, we increment the rep count(up) and vice versa for rep count(down).

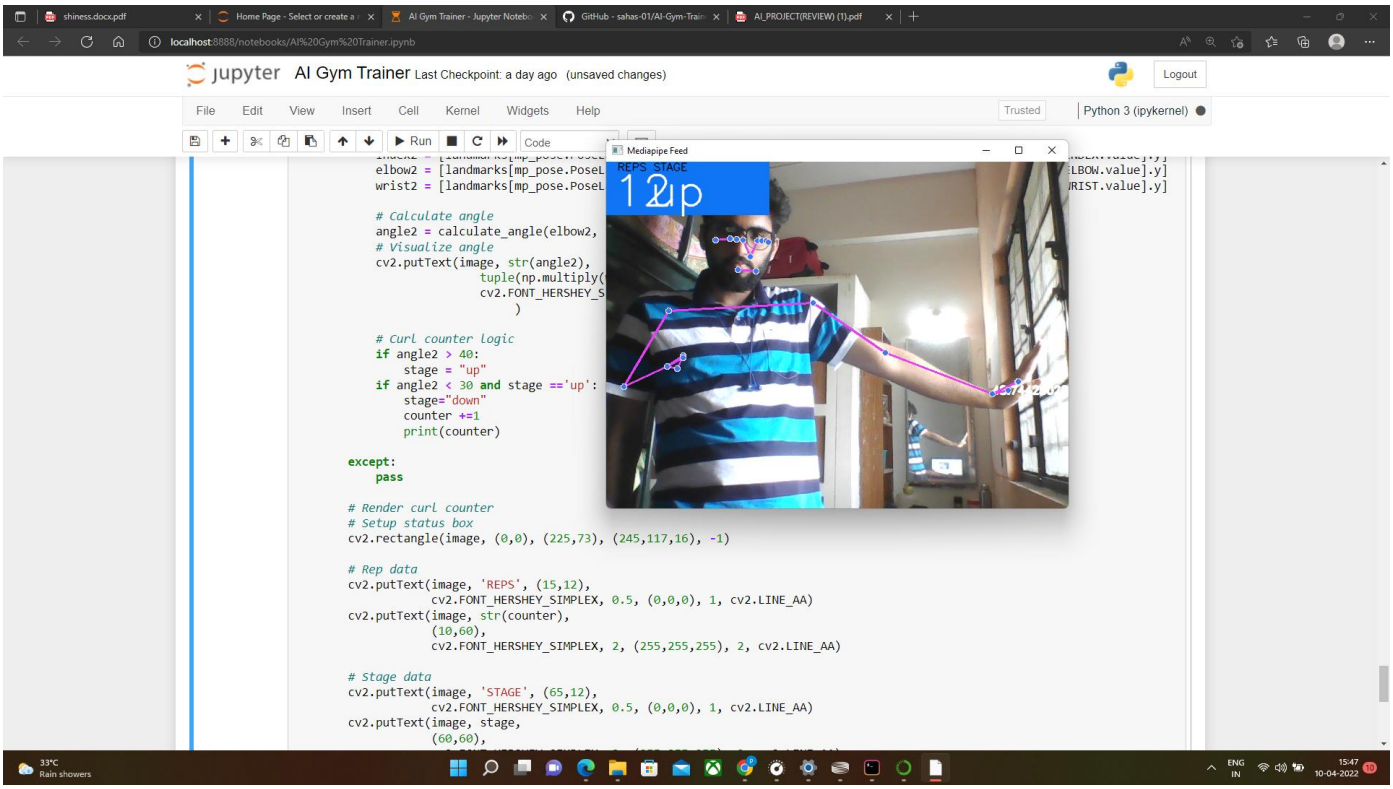
Experimentations and Results

No dataset was used by us in this project. However, we used the mediapipe library which provided us with pre trained models enabling us to calculate and estimate poses with ease.

Pose detection using Mediapipe


jupyter AI Gym Trainer Last Checkpoint: a day ago (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code
Mediapipe Feed
REPS STAGE
1 down
In [4]:
#code to count and display the number of reps
cap = cv2.VideoCapture(0)
# Curl counter variables
counter = 0
stage = None
Setup mediapipe instance
with mp_pose.Pose(min_detection_confidence=0.5
while cap.isOpened():
ret, frame = cap.read()
# Recolor image to RGB
image = cv2.cvtColor(frame, cv2.COLOR_
image.flags.writeable = False
# Make detection
results = pose.process(image)
# Recolor back to BGR
image.flags.writeable = True
image = cv2.cvtColor(image, cv2.COLOR_RGB2BGR)
# Extract landmarks
try:
landmarks = results.pose_landmarks.landmark
index2 = [landmarks[mp_pose.PoseLandmark.RIGHT_INDEX.value].x,landmarks[mp_pose.PoseLandmark.RIGHT_INDEX.value].y]
elbow2 = [landmarks[mp_pose.PoseLandmark.RIGHT_ELBOW.value].x,landmarks[mp_pose.PoseLandmark.RIGHT_ELBOW.value].y]
wrist2 = [landmarks[mp_pose.PoseLandmark.RIGHT_WRIST.value].x,landmarks[mp_pose.PoseLandmark.RIGHT_WRIST.value].y]
# Calculate angle
angle2 = calculate_angle(elbow2, index2, wrist2)
# Visualize angle
cv2.putText(image, str(angle2),
tuple(np.multiply(wrist2, [640, 480]).astype(int)),
33°C
Rain showers
ENG
IN
15:46
10-04-2022


jupyter AI Gym Trainer Last Checkpoint: a day ago (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code
Mediapipe Feed
REPS STAGE
1 up
elbow2 = [landmarks[mp_pose.PoseL
wrist2 = [landmarks[mp_pose.PoseL
# Calculate angle
angle2 = calculate_angle(elbow2,
# Visualize angle
cv2.putText(image, str(angle2),
tuple(np.multiply(
cv2.FONT_HERSHEY_S
)
# Curl counter logic
if angle2 > 40:
stage = "up"
if angle2 < 30 and stage =='up':
stage="down"
counter +=1
print(counter)
except:
pass
# Render curl counter
# Setup status box
cv2.rectangle(image, (0,0), (225,73), (245,117,16), -1)
# Rep data
cv2.putText(image, 'REPS', (15,12),
cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0,0,0), 1, cv2.LINE_AA)
cv2.putText(image, str(counter),
(10,60),
cv2.FONT_HERSHEY_SIMPLEX, 2, (255,255,255), 2, cv2.LINE_AA)
Stage data
cv2.putText(image, 'STAGE', (65,12),
cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0,0,0), 1, cv2.LINE_AA)
cv2.putText(image, stage,
(60,60),
33°C
Rain showers
ENG
IN
15:47
10-04-2022



Challenges we ran into

Our understanding of the mediapipe library was not well versed enough which made us face difficulties in using mediapipe to estimate poses.

Writing our own function to estimate the angles, it was simple trigonometry, but we initially were doing it without the use of **numPy** library.

Counting the reps and sets for each exercise performed.

Future Scope

- Add in a mechanism to detect other exercises for the lower body parts

- Deploy this as a model to Heroku or AWS
- A static website where users will be able to register and keep track of their daily exercises