REAL TIME POSE TRANSFER TO 3D CHARACTERS

Sai Tarun Sathyan & Aman Raj Lnu

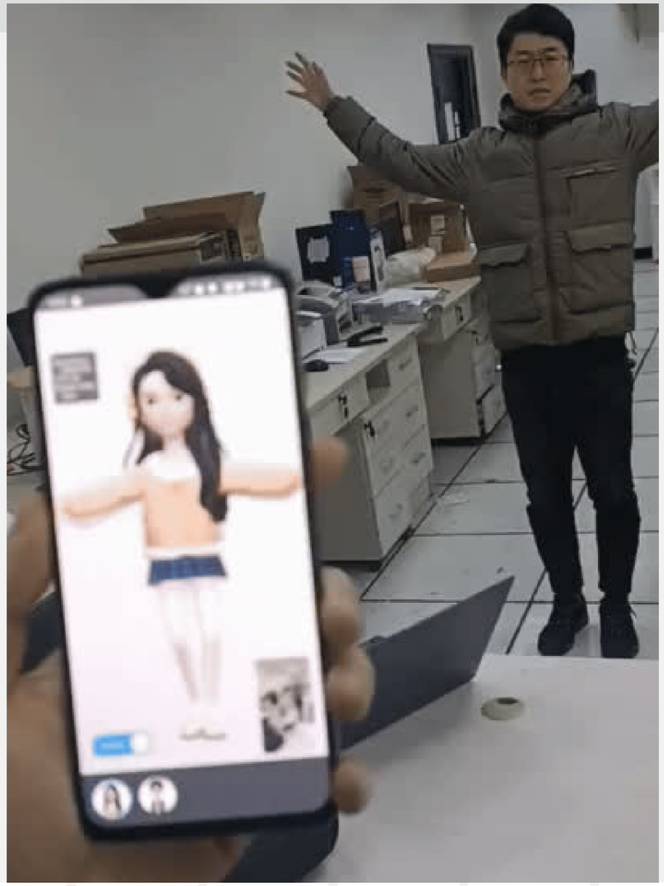
CSCI.736.01

ROCHESTER INSTITUTE OF TECNOLOGY

PROBLEM STATEMENT

The goal of this project is to develop an affordable motion capture system that uses real-time pose estimation to replace the expensive equipment required for traditional motion capture.

EXPECTED OUTCOME



INTRODUCTION

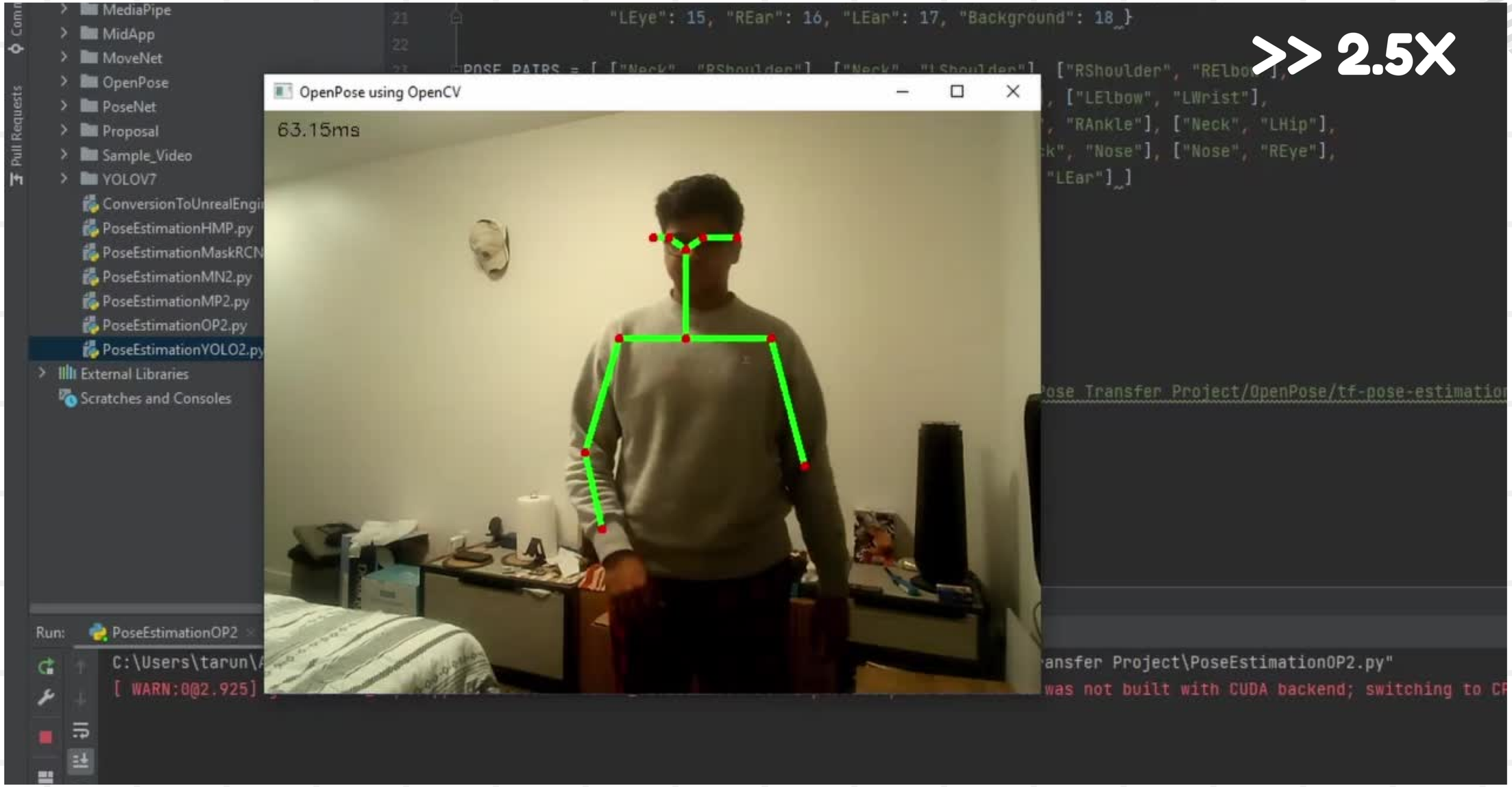
Human pose estimation is a computer vision technique that involves detecting and tracking the key points of the human body in images or videos.

We present a comprehensive evaluation of state-of-the-art pose estimation algorithms and demonstrate their practical use in real-time 3D avatar animation

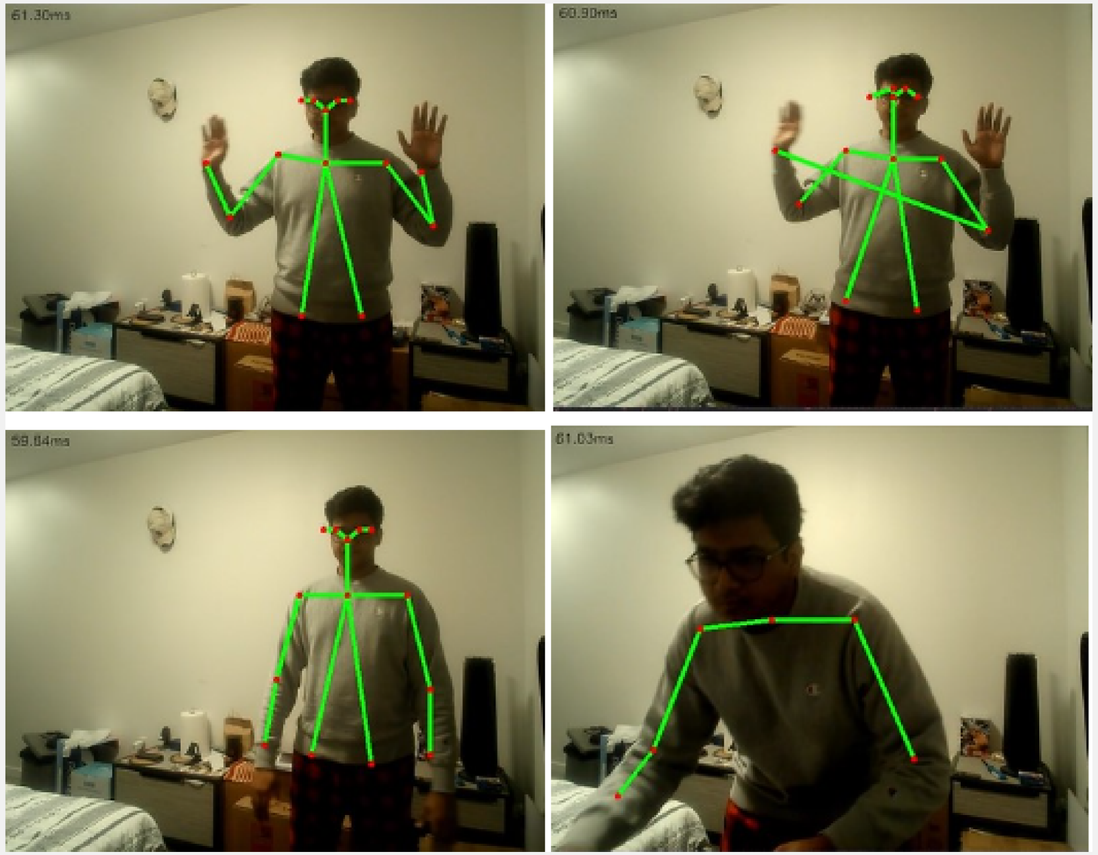
POSE ESTIMATION METHODS

- POSE NET
- MASK RCNN
- MEDIA PIPE
- OPEN POSE
- YOLOV7
- MOVE NET

OPEN POSE


>> 2.5X
OpenPose using OpenCV
63.15ms
MediaPipe
MidApp
MoveNet
OpenPose
PoseNet
Proposal
Sample_Video
YOLOV7
PoseEstimationHMP.py
PoseEstimationMN2.py
PoseEstimationMP2.py
PoseEstimationOP2.py
PoseEstimationYOLO2.py
External Libraries
Scratches and Consoles
Run:
PoseEstimationOP2
"LEye": 15, "REar": 16, "LEar": 17, "Background": 18}
["LElbow", "LWrist"],
"RAnkle"], ["Neck", "LHip"],
"Nose"], ["Nose", "REye"],
"LEar"]]

OPEN POSE

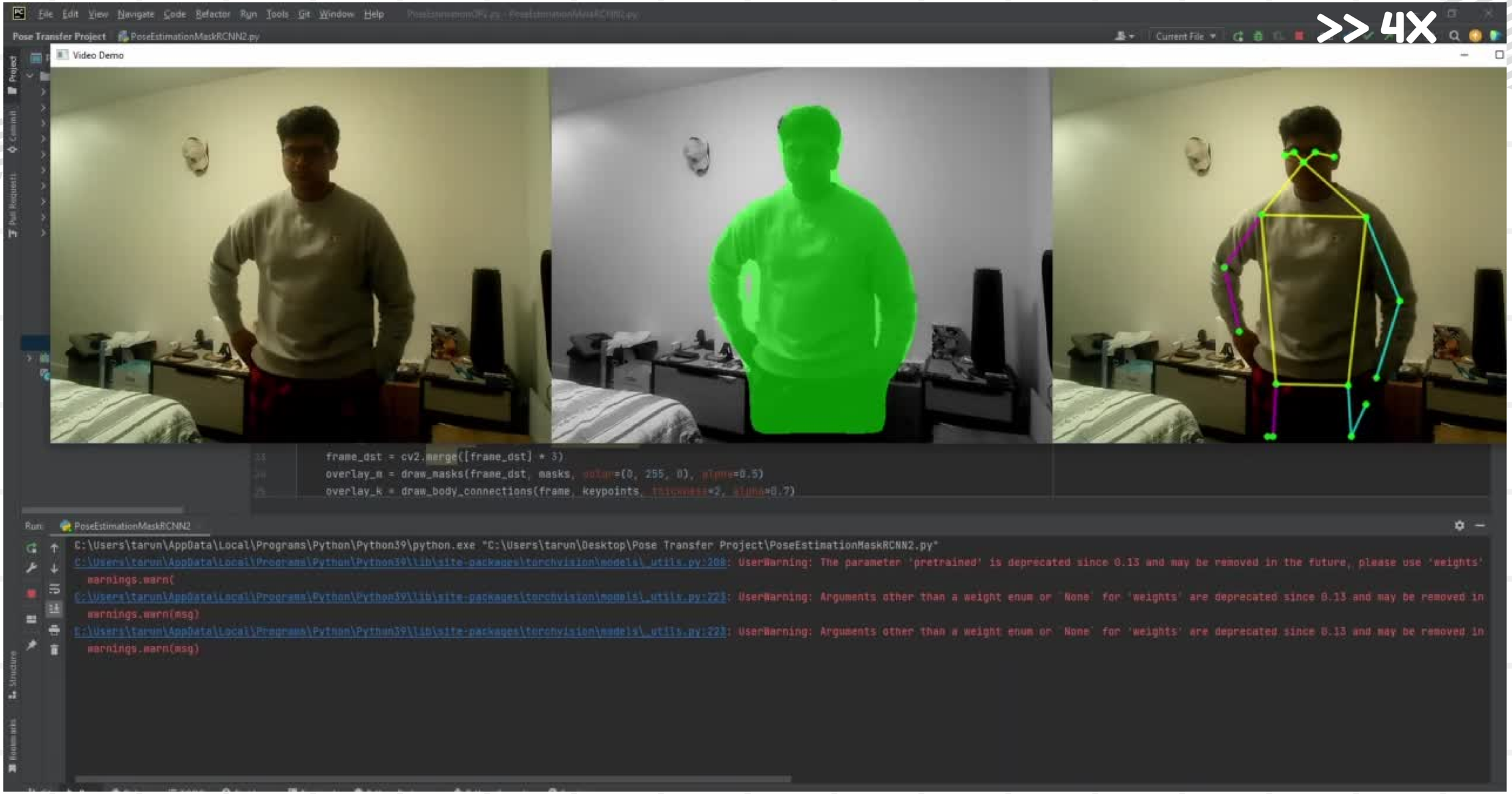


DEVICE USED : CPU AND GPU
KEY POINTS TRACKED : RETURNS 17-135 KEY POINTS

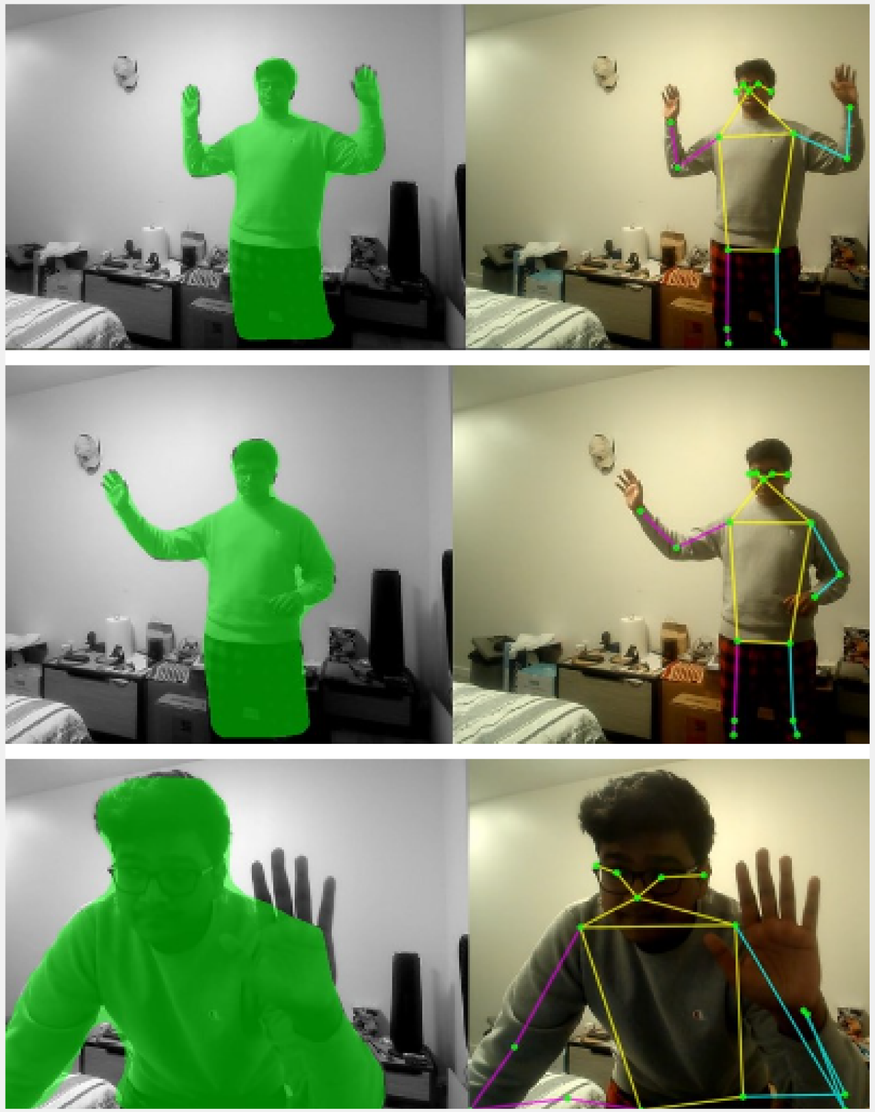
OBSERVATIONS:

- WORKS WELL ON CPU, WORKS IN REAL TIME USING CPU ALONE
- THE DETECTIONS ARE MOSTLY ACCURATE, BREAKS APART SOMETIMES DURING OVERLAPPING/COMPLEX MOVEMENTS
- FAILS TO DETECT LEGS WHILE WEARING CHECKED PANTS / DUE TO BAD LIGHTING
- FAILS TO DETECT LIMBS WHILE CLOSE TO THE CAMERA (WAIST UP)

MASK RCNN


>> 4X
Video Demo
frame_dst = cv2.merge([frame_dst] * 3)
overlay_m = draw_masks(frame_dst, masks, color=(0, 255, 0), alpha=0.5)
overlay_k = draw_body_connections(frame, keypoints, thickness=2, alpha=0.7)
Run
PoseEstimationMaskRCNN2
C:\Users\tarun\AppData\Local\Programs\Python\Python39\python.exe "C:\Users\tarun\Desktop\Pose Transfer Project\PoseEstimationMaskRCNN2.py"
C:\Users\tarun\AppData\Local\Programs\Python\Python39\lib\site-packages\torchvision\models_utils.py:208: UserWarning: The parameter 'pretrained' is deprecated since 0.13 and may be removed in the future, please use 'weights'
warnings.warn(
C:\Users\tarun\AppData\Local\Programs\Python\Python39\lib\site-packages\torchvision\models_utils.py:223: UserWarning: Arguments other than a weight enum or `None` for 'weights' are deprecated since 0.13 and may be removed in
warnings.warn(msg)
C:\Users\tarun\AppData\Local\Programs\Python\Python39\lib\site-packages\torchvision\models_utils.py:223: UserWarning: Arguments other than a weight enum or `None` for 'weights' are deprecated since 0.13 and may be removed in
warnings.warn(msg)

MASK RCNN

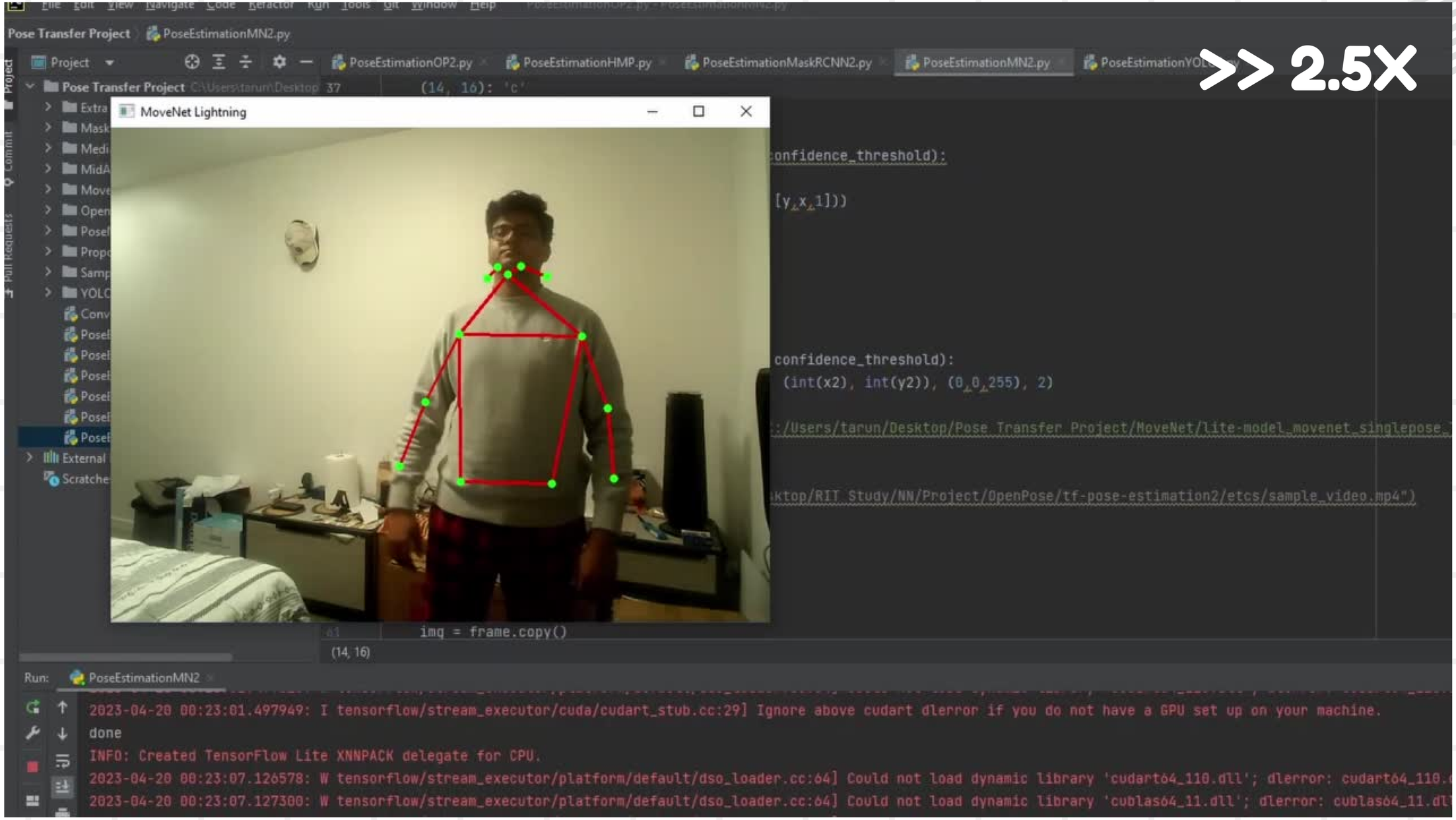


DEVICE USED : CPU AND GPU
KEY POINTS TRACKED : RETURNS 17

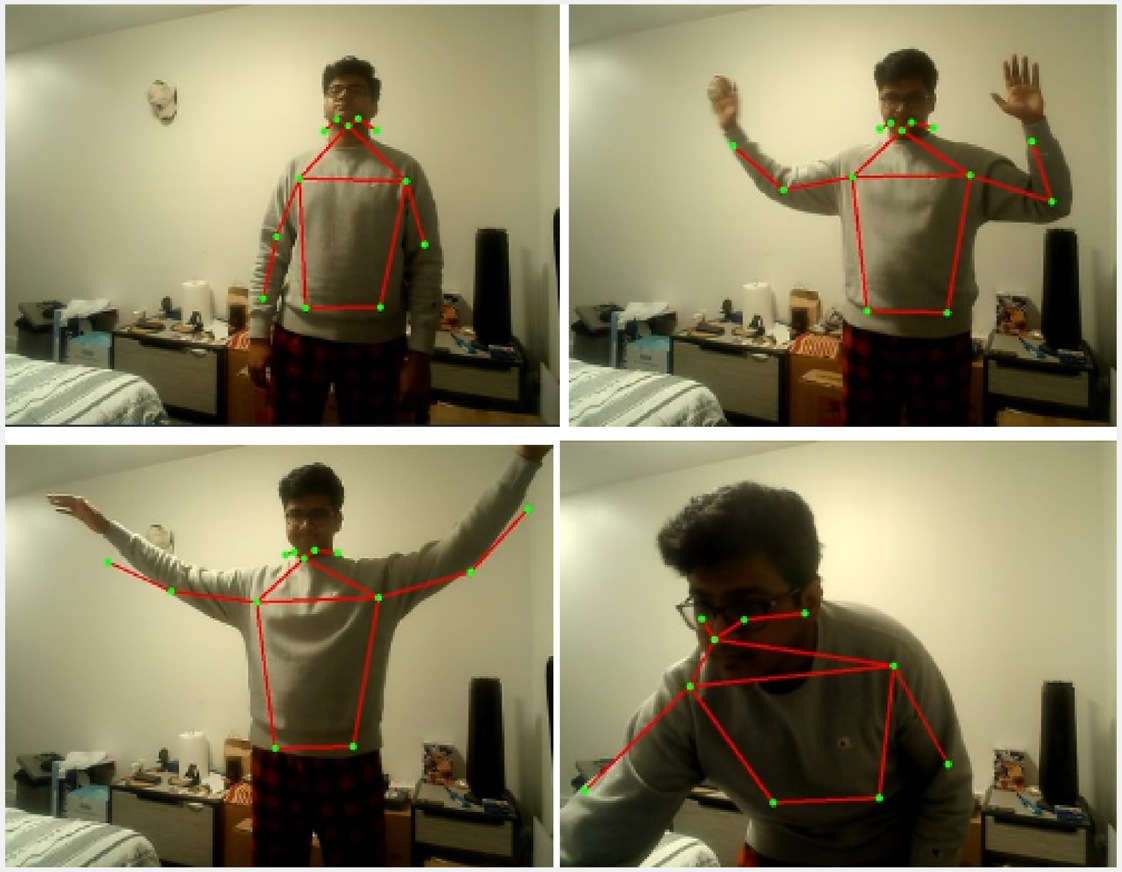
OBSERVATIONS:

- EXTREMELY ACCURATE
- WORKS WITH MULTIPLE PEOPLE
- ALSO WORKS AS A SEGMENTATION ALGORITHM
- DETECTS LEGS PROPERLY AND ALSO DETECTS LIMBS WHEN CLOSE TO THE CAMERA
- SOMETIMES DETECTS PHANTOM LIMBS OR JOINTS THAT DO NOT EXIST
- IT IS EXTREMELY SLOW ON CPU AND FAILS TO WORK IN REAL TIME
- NEEDS GPU TO WORK FAST REAL TIME DETECTIONS

MOVE NET


>> 2.5X
MoveNet Lightning
Pose Transfer Project
PoseEstimationMN2.py
PoseEstimationOP2.py
PoseEstimationHMP.py
PoseEstimationMaskRCNN2.py
PoseEstimationMN2.py
img = frame.copy()
(14, 16)
Run: PoseEstimationMN2
2023-04-20 00:23:01.497949: I tensorflow/stream_executor/cuda/cudart_stub.cc:29] Ignore above cudart dlerror if you do not have a GPU set up on your machine.
done
INFO: Created TensorFlow Lite XNNPACK delegate for CPU.
2023-04-20 00:23:07.126578: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'cudart64_110.dll'; dlerror: cudart64_110.
2023-04-20 00:23:07.127300: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'cublas64_11.dll'; dlerror: cublas64_11.dll

MOVENET

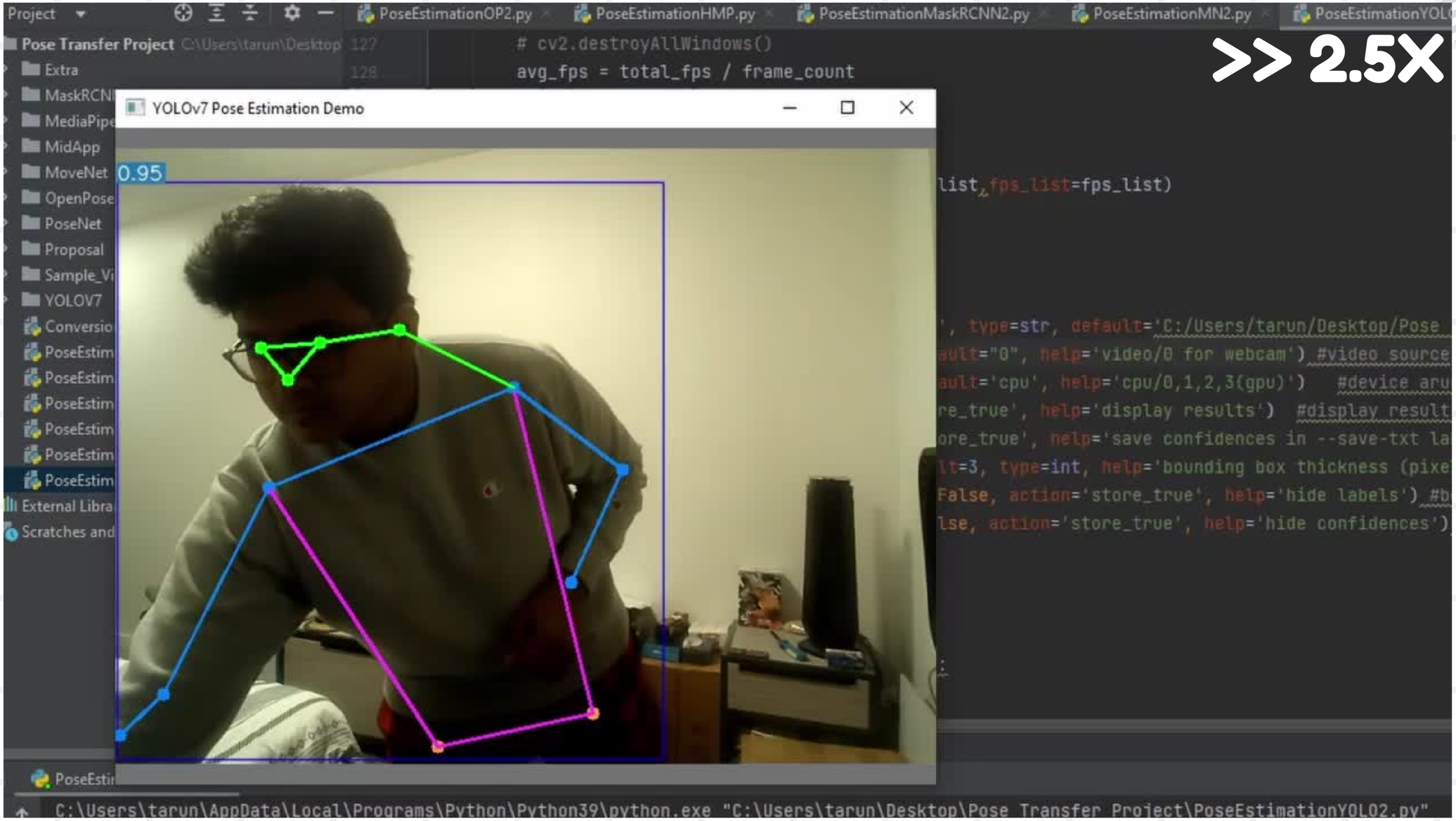


DEVICE USED : CPU AND GPU
KEY POINTS TRACKED : RETURNS 17

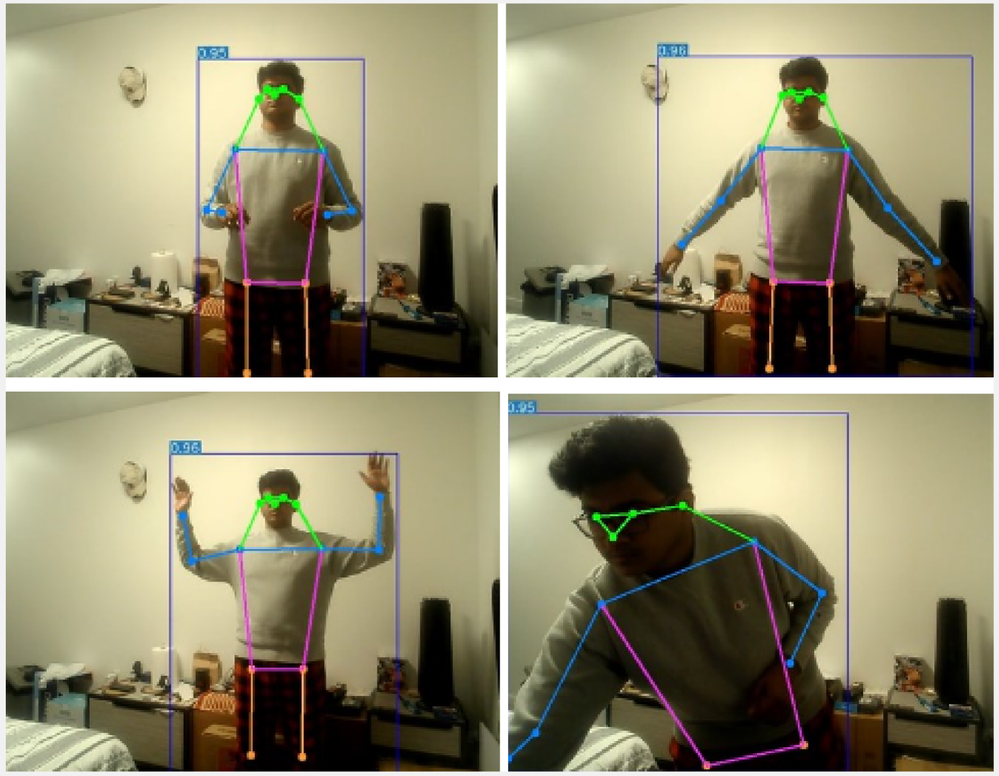
OBSERVATIONS:

- MOSTLY ACCURATE , EVEN DURING LOWLIGHT CONDITIONS.
- WORKS IN REAL TIME WITH CPU ALONE
- IT'S GOOD AT IDENTIFYING LIMBS WHEN CLOSE TO THE CAMERA
- FAILS AT PROPERLY TRACKING THE EYES,NOSE,EARS IF THE SUBJECT IS WEARING GLASSES AND IN LOW LIGHT CONDITIONS
- FAILS TO DETECT LEGS WHILE WEARING CHECKED PANTS / DUE TO BAD LIGHTING

YOLOV7


>> 2.5X
YOLOv7 Pose Estimation Demo
0.95
Project
Pose Transfer Project
Extra
MaskRCN
MediaPipe
MidApp
MoveNet
OpenPose
PoseNet
Proposal
Sample_Vi
YOLOV7
Conversio
PoseEstim
External Libra
Scratches and
PoseEstimationOP2.py
PoseEstimationHMP.py
PoseEstimationMaskRCNN2.py
PoseEstimationMN2.py
# cv2.destroyAllWindows()
avg_fps = total_fps / frame_count
list, fps_list=fps_list)
', type=str, default='C:/Users/tarun/Desktop/Pose
ault="0", help='video/0 for webcam') #video source
ault='cpu', help='cpu/0,1,2,3(gpu)') #device aru
re_true', help='display results') #display result
ore_true', help='save confidences in --save-txt la
lt=3, type=int, help='bounding box thickness (pixe
False, action='store_true', help='hide labels') #b
lse, action='store_true', help='hide confidences')
C:\Users\tarun\AppData\Local\Programs\Python\Python39\python.exe "C:\Users\tarun\Desktop\Pose Transfer Project\PoseEstimationYOLO2.py"

YOLOV7

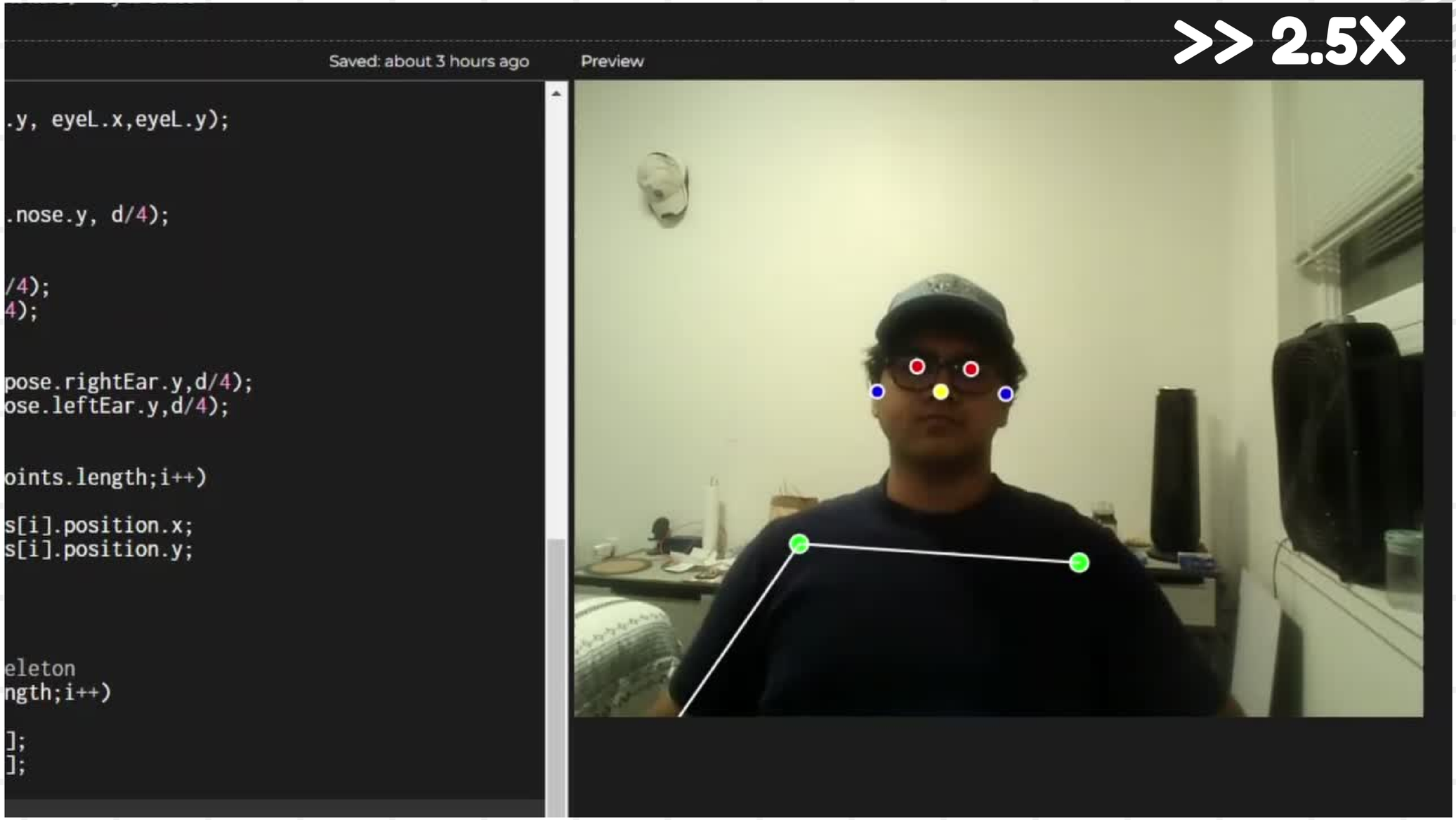


DEVICE USED : CPU AND GPU
KEY POINTS TRACKED : RETURNS 17

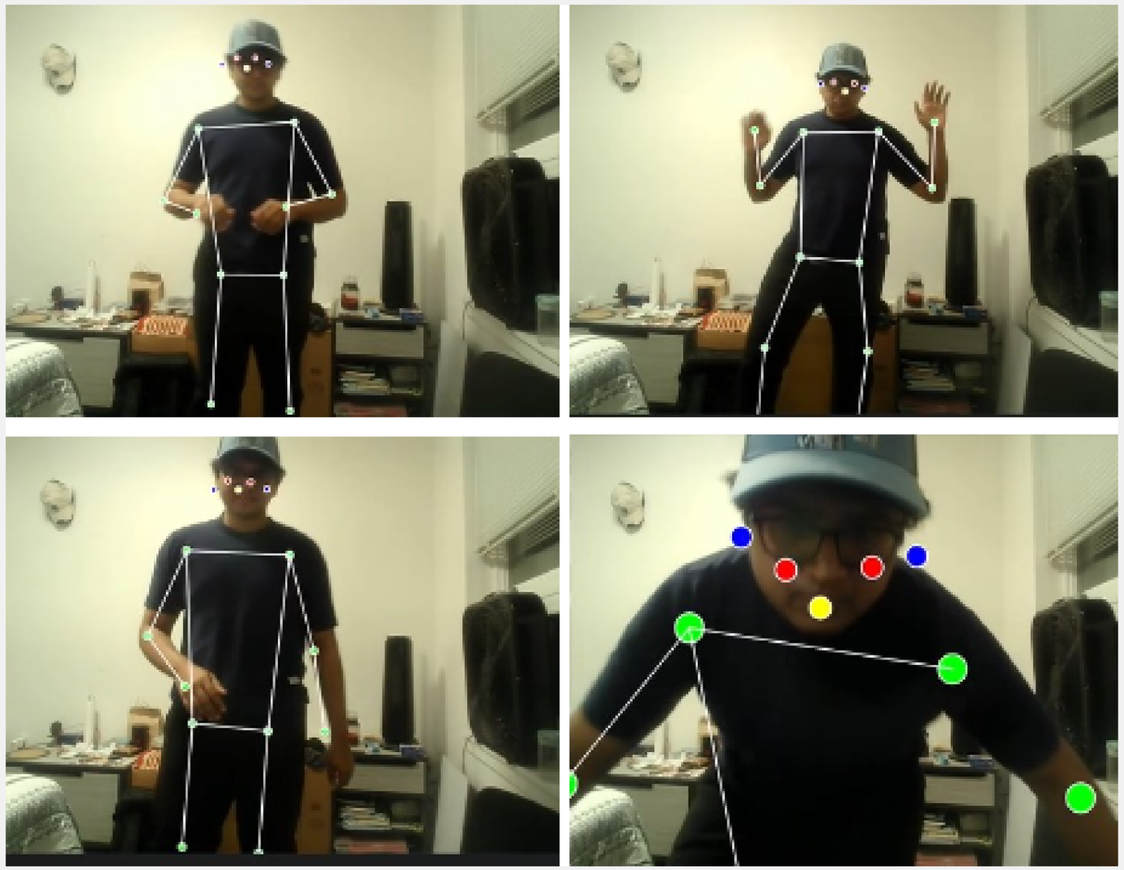
OBSERVATIONS:

- EXTREMELY ACCURATE , EVEN DURING LOWLIGHT CONDITIONS.
- WORKS MODERATELY WELL ON CPU (<=12FPS)
- WORKS WITH MULTIPLE PEOPLE
- WORKS REALLY WELL ON GPU
- IT’S GOOD AT IDENTIFYING LIMBS WHEN CLOSE TO THE CAMERA
- YOU NEED A GPU FOR IT TO WORK SMOOTHLY IN REAL TIME

POSENET


Saved: about 3 hours ago
Preview
>> 2.5X
.y, eyeL.x,eyeL.y);
.nose.y, d/4);
/4);
4);
pose.rightEar.y,d/4);
ose.leftEar.y,d/4);
oints.length;i++)
s[i].position.x;
s[i].position.y;
eleton
ngth;i++)
];
];

POSENET

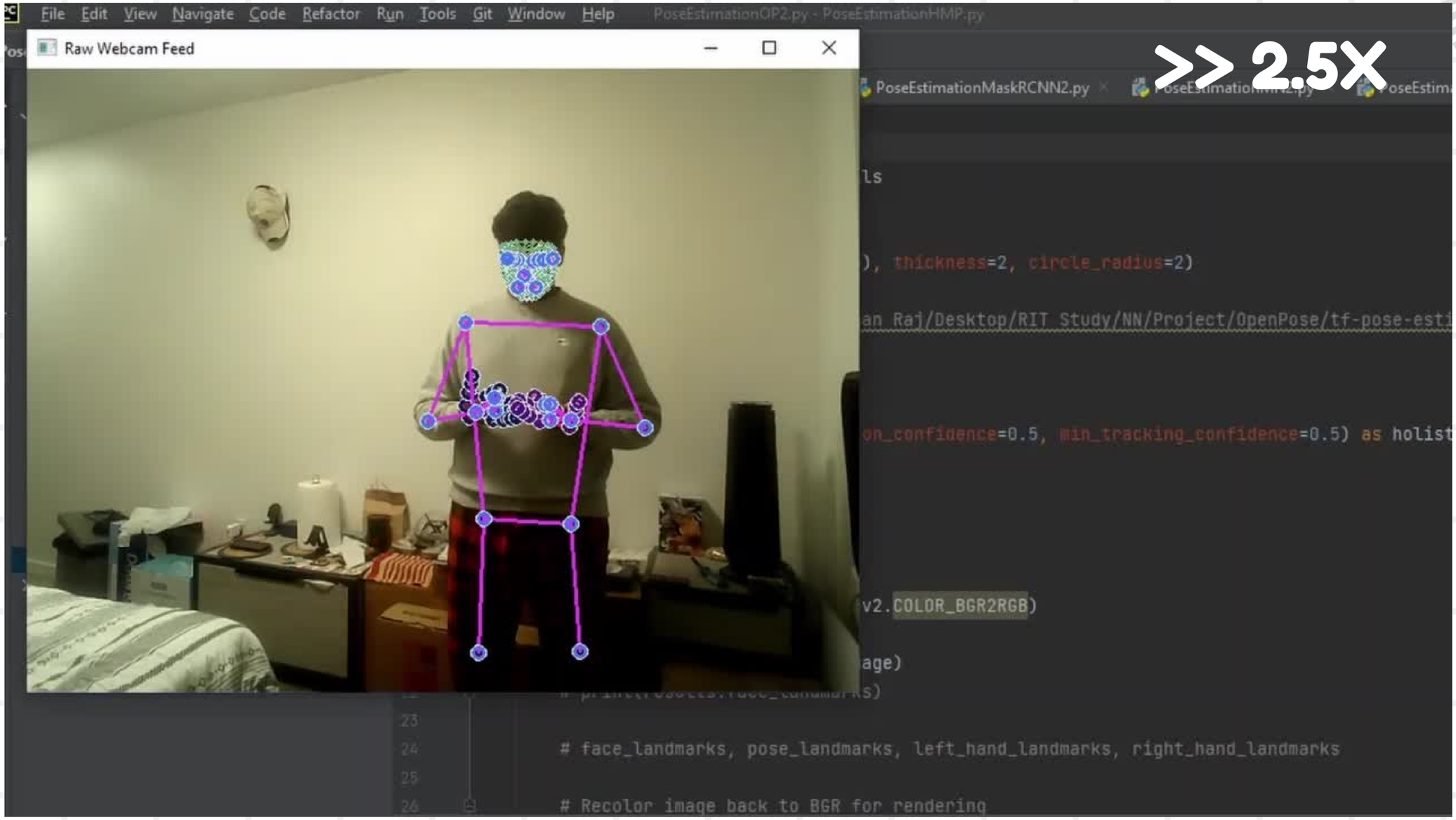


DEVICE USED : WORKS ON CPU ONLY
KEY POINTS TRACKED : RETURNS 17

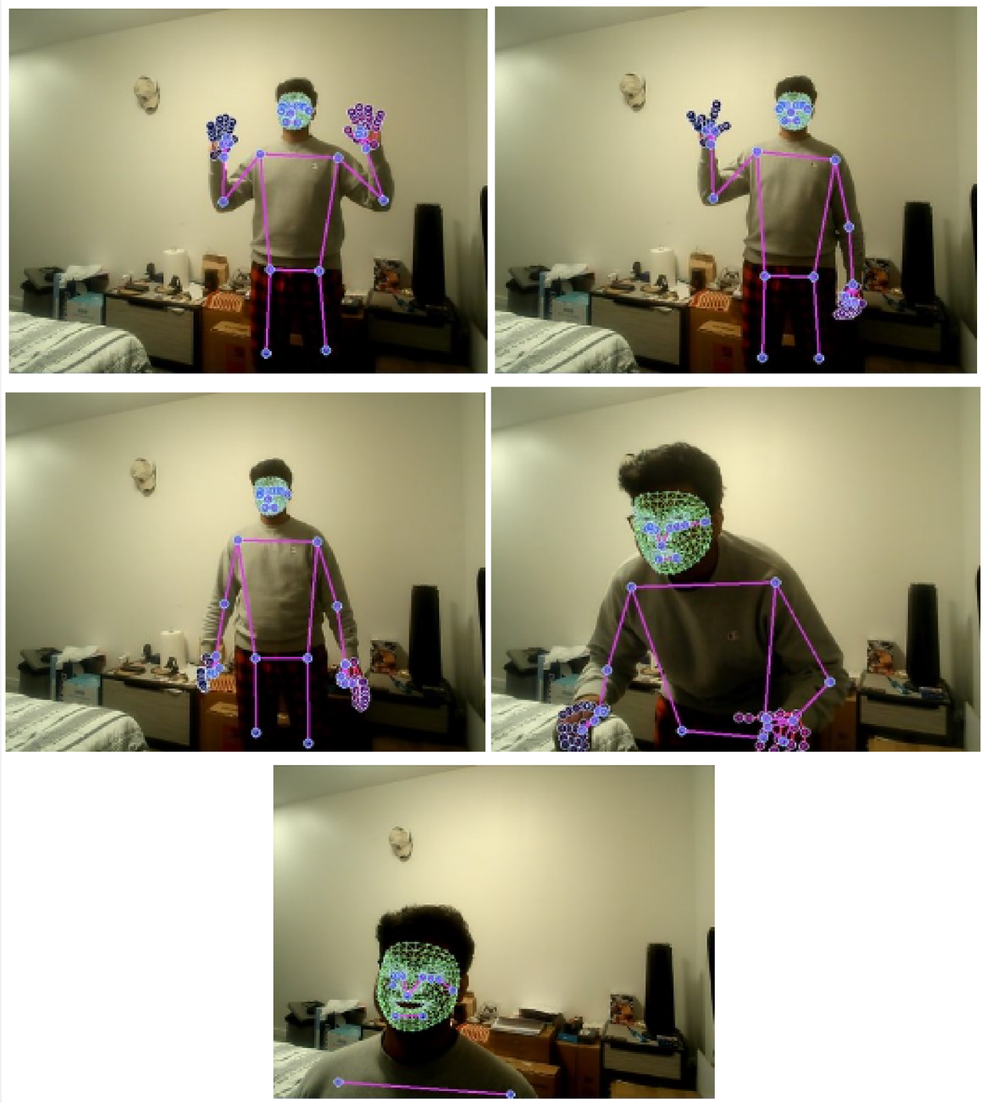
OBSERVATIONS:

- LIGHTWEIGHT POSE ESTIMATOR, WORKS WITH VERY LITTLE COMPUTING RESOURCES
- EXTREMELY ACCURATE , EVEN DURING LOWLIGHT CONDITIONS.
- WORKS MODERATELY VERY WELL ON CPU
- DOES NOT WORK ON/USE GPU
- IT'S GOOD AT IDENTIFYING LIMBS WHEN CLOSE TO THE CAMERA
- IT IS NOT THE OPTIMAL CHOICE IF YOU HAVE A DECENT PC BUT THE BEST WHEN USING IT ON A LOW END COMPUTER

MEDIA PIPE


File Edit View Navigate Code Refactor Run Tools Git Window Help
PoseEstimationOP2.py - PoseEstimationHMP.py
Raw Webcam Feed
>> 2.5X
PoseEstimationMaskRCNN2.py
ls
), thickness=2, circle_radius=2)
an Raj/Desktop/RIT Study/NN/Project/OpenPose/tf-pose-esti
on_confidence=0.5, min_tracking_confidence=0.5) as holist
v2.COLOR_BGR2RGB)
age)
s)
face_landmarks, pose_landmarks, left_hand_landmarks, right_hand_landmarks
Recolor image back to BGR for rendering

MEDIA PIPE

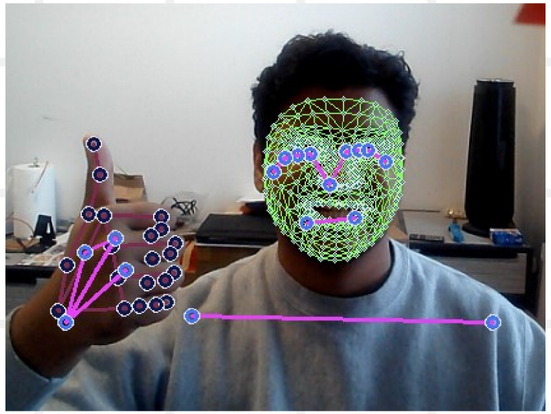


DEVICE USED : WORKS ON CPU ONLY
KEY POINTS TRACKED : RETURNS 135

OBSERVATIONS:

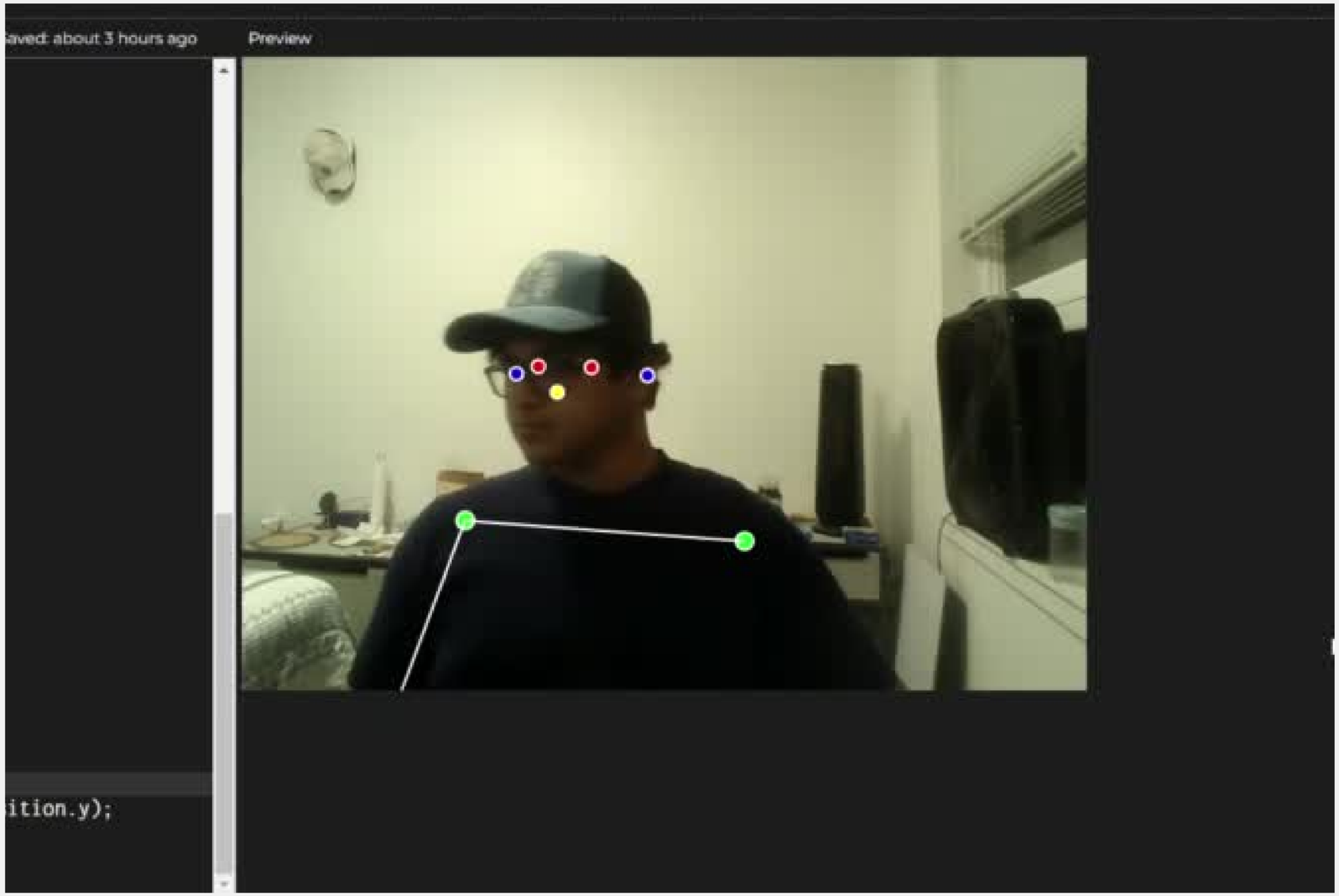
- WORKS EXTREMELY WELL WITH CPU RESOURCES ALONE
- PROVIDES TRACKING FOR FACIAL EXPRESSION AND FINGERS AS WELL
- DOES NOT WORK REQUIRE GPU
- THE MOST ACCURATE POSE ESTIMATOR SO FAR
- WORKS WELL CLOSE UP AND IN LOW LIGHT CONDITIONS

BEST ONE



Pose Estimator	Devices used	No.Key points	Works Real time	Subjects Tracked
Open Pose	CPU/GPU	17 - 135	Yes/ CPU	1 person
MaskRCNN	CPU/GPU	17	No	N people
Move Net	CPU/GPU	17	Yes/CPU	1 person
Yolov7	CPU/GPU	17	Yes/GPU	N people
Pose Net	CPU	17	Yes/CPU	1 person
Media Pipe	CPU	135	Yes/CPU	1 person

PROBLEM WITH 2D



PROJECT DESIGN

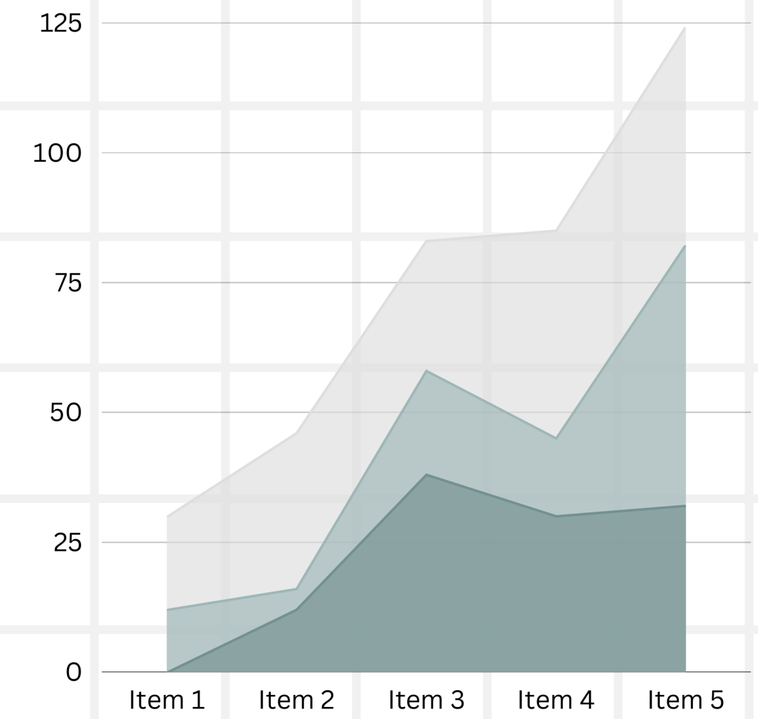
BLENDER

MODEL, ARMATURE AND RIG

MEDIAPIPE

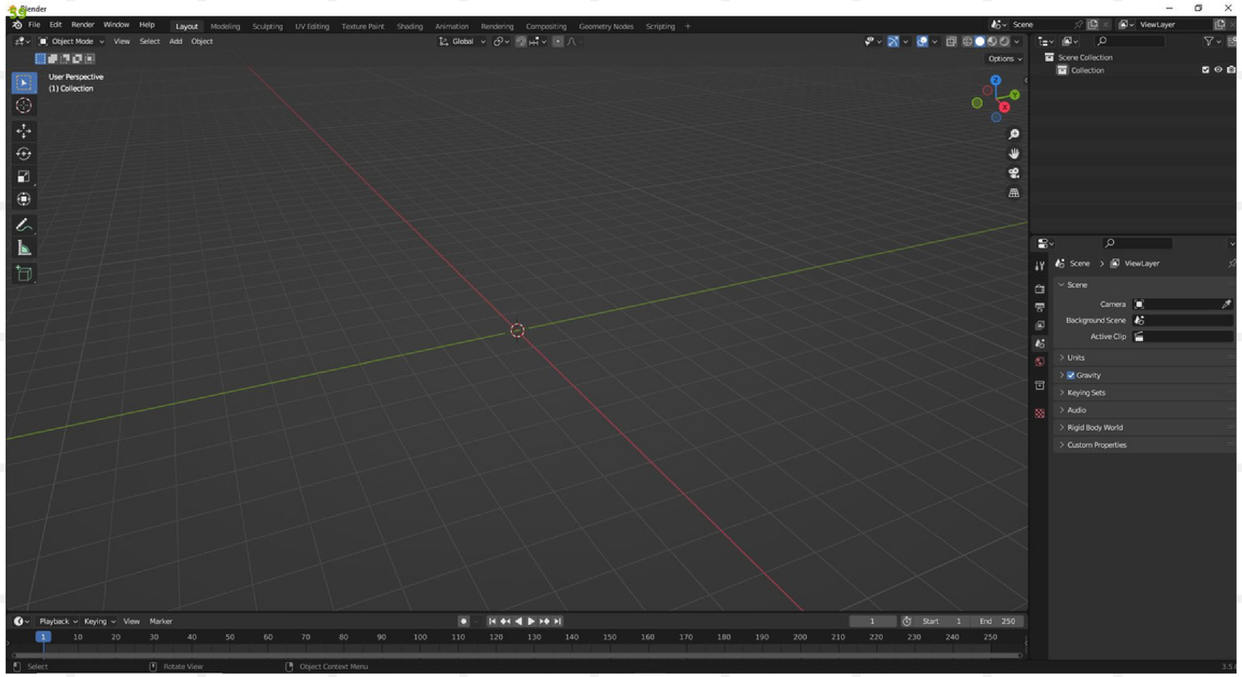
2D TO 3D CONVERSION

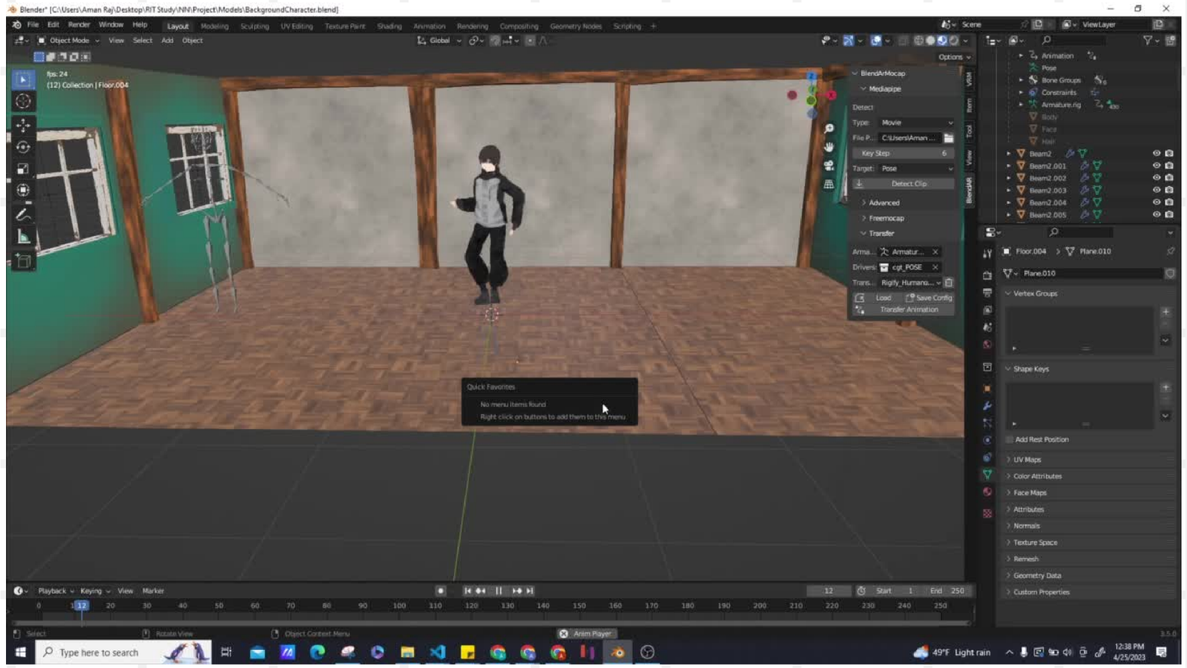
APPLYING TO RIG TO ANIMATE



BLENDER DESIGN

What is blender and how it Functions

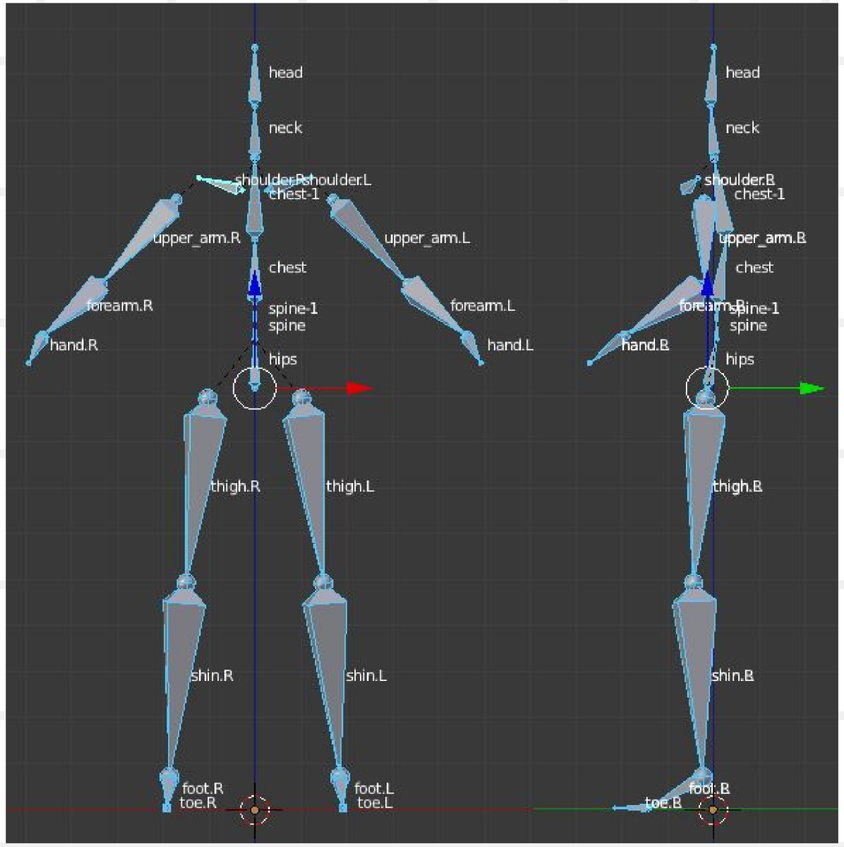




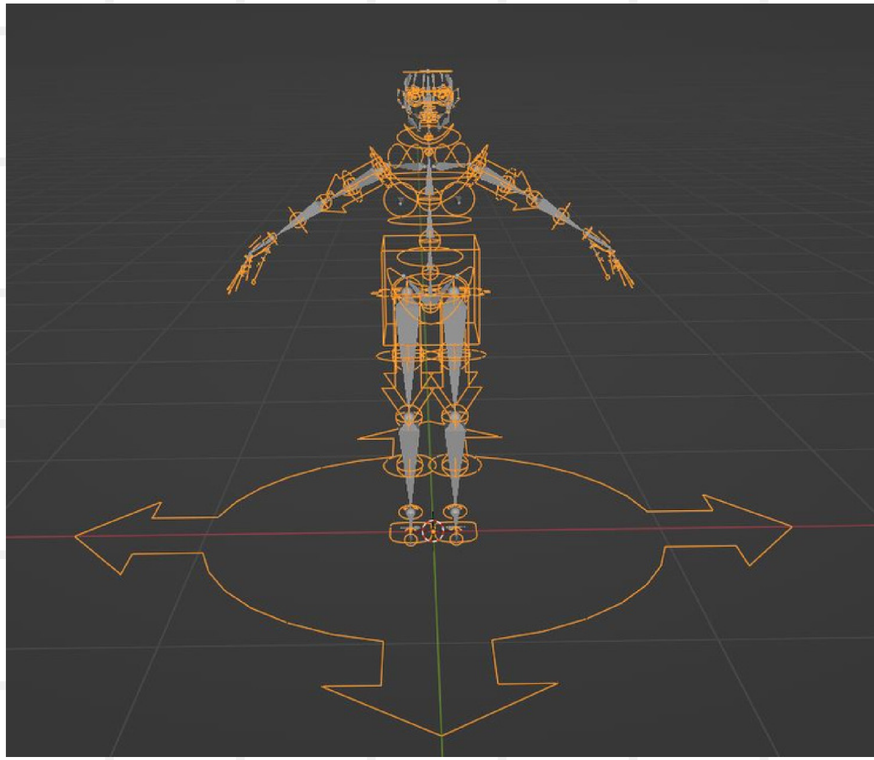
MODELS AND RIGS

MODEL/MESH

ARMATURE

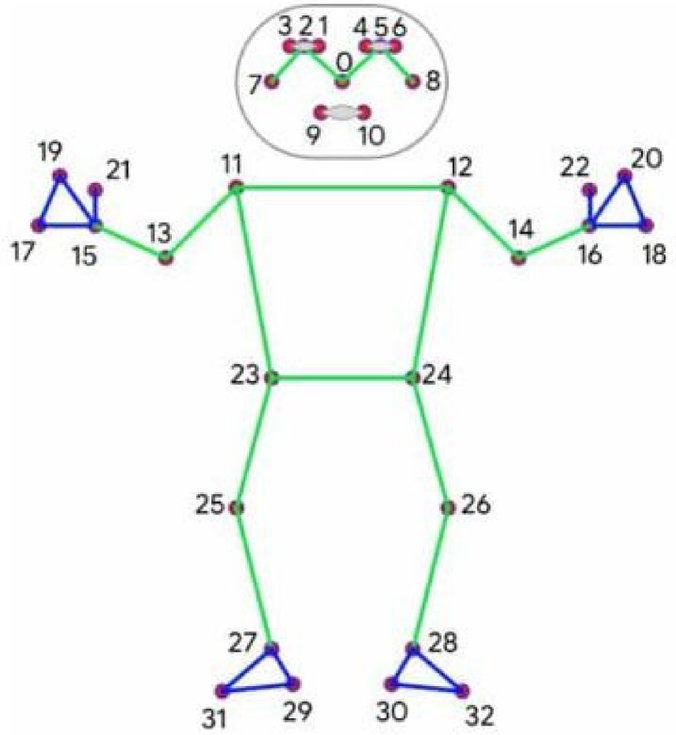


RIG



MEDIAPIPE

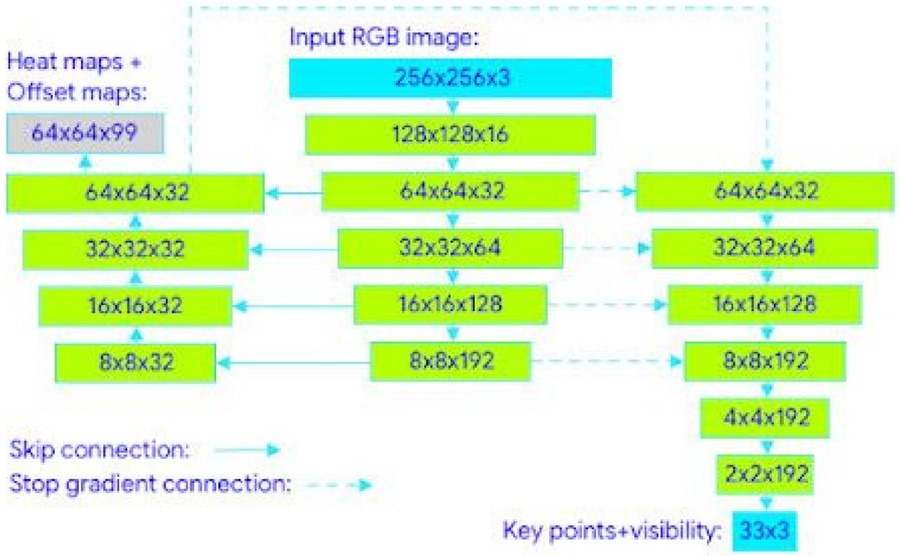
MediaPipe Pose is a single-person pose estimation framework. It uses BlazePose 33 landmark topology



0. nose
1. right eye inner
2. right eye
3. right eye outer
4. left eye inner
5. left eye
6. left eye outer
7. right ear
8. left ear
9. mouth right
10. mouth left
11. right shoulder
12. left shoulder
13. right elbow
14. left elbow
15. right wrist
16. left wrist
17. right pinky knuckle #1
18. left pinky knuckle #1
19. right index knuclke #1
20. left index knuckle #1
21. right thumb knuckle #2
22. left thumb knuckle #2
23. right hip
24. left hip
25. right knee
26. left knee
27. right ankle
28. left ankle
29. right heel
30. left heel
31. right foot index
32. left foot index

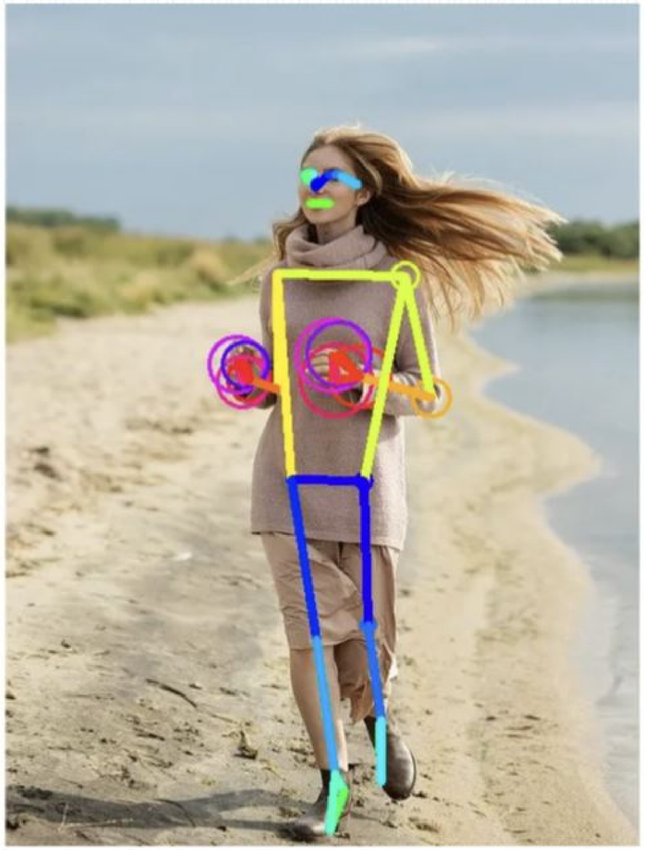
BLAZE POSE

MediaPipe Pose is a single-person pose estimation framework. It uses BlazePose 33 landmark topology

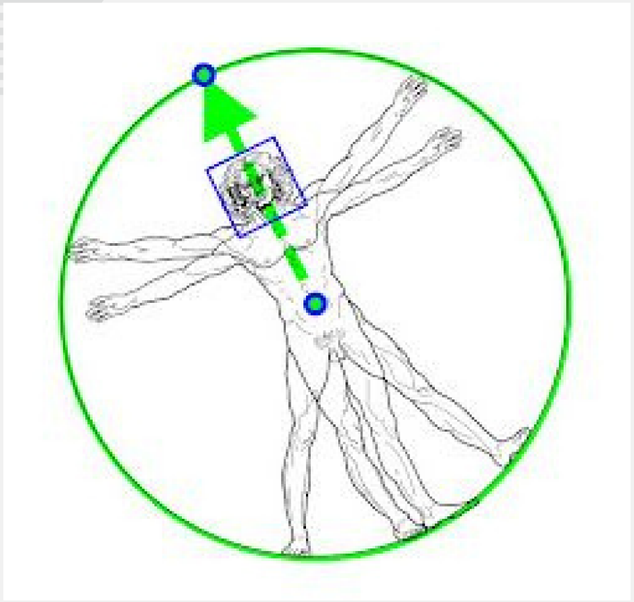


Tracking network architecture: regression with heatmap supervision

Pose With Z value Shown



DEPTH ESTIMATION AND PLUGIN CALACULATION



Z coordinate measures distances in "image pixels" and is based on the subject's hips, which serve as the origin of the Z axis.

Input Data:

Landmark: `[idx: int, [x: float, y: float, z: float]]`

Output Data:

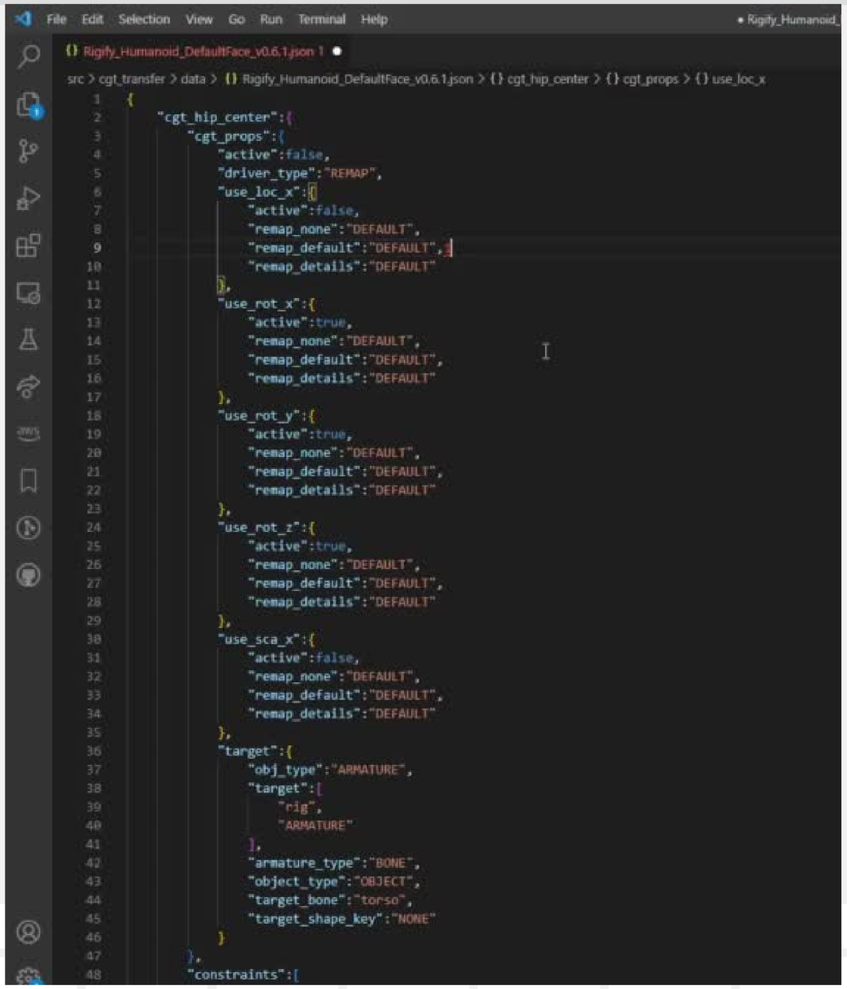
Location: `[idx: int, List[x: float, y: float, z: float]]'

Rotation: `[idx: int, Euler(x: float, y: float, z: float)]'

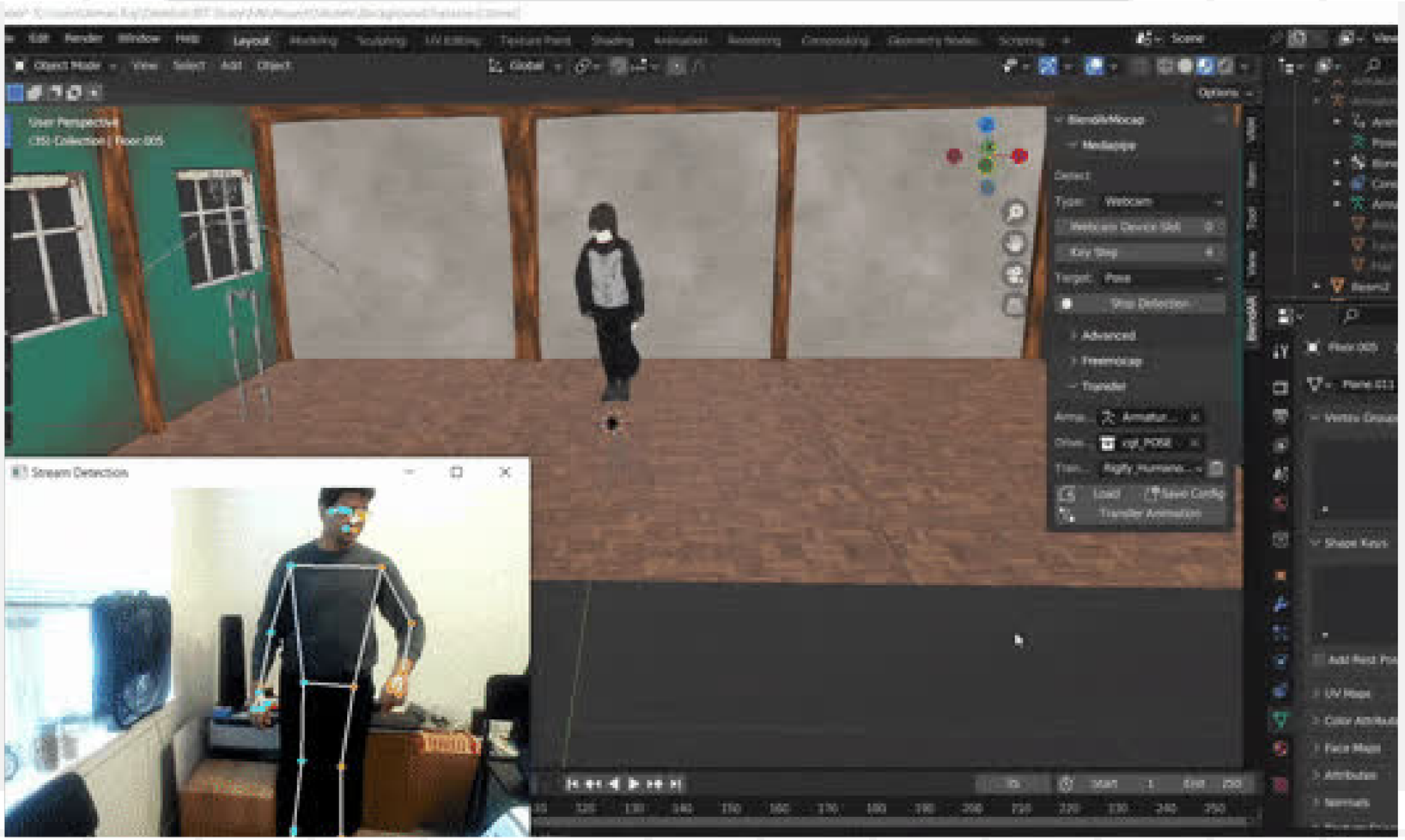
Scale: `[idx: int, List[x: float, y: float, z: float]]'

BONE REMMAPPING
FROM MEDIAPIPE TO BLENDER RIG

```
"pose":{
    "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",
    "33":"hip_center",
    "34":"shoulder_center",
    "35":"pose_location",
    "36":"pose_none"
},
```



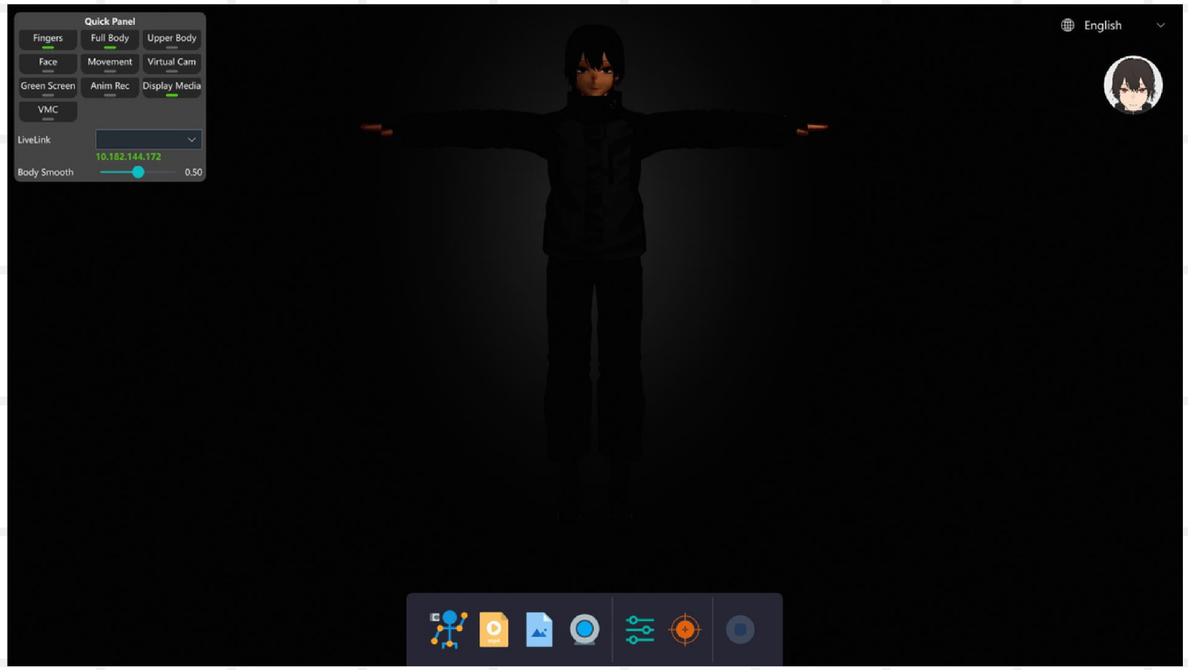
DEMONSTRATION(ROUGH)



LESSONS LEARNED

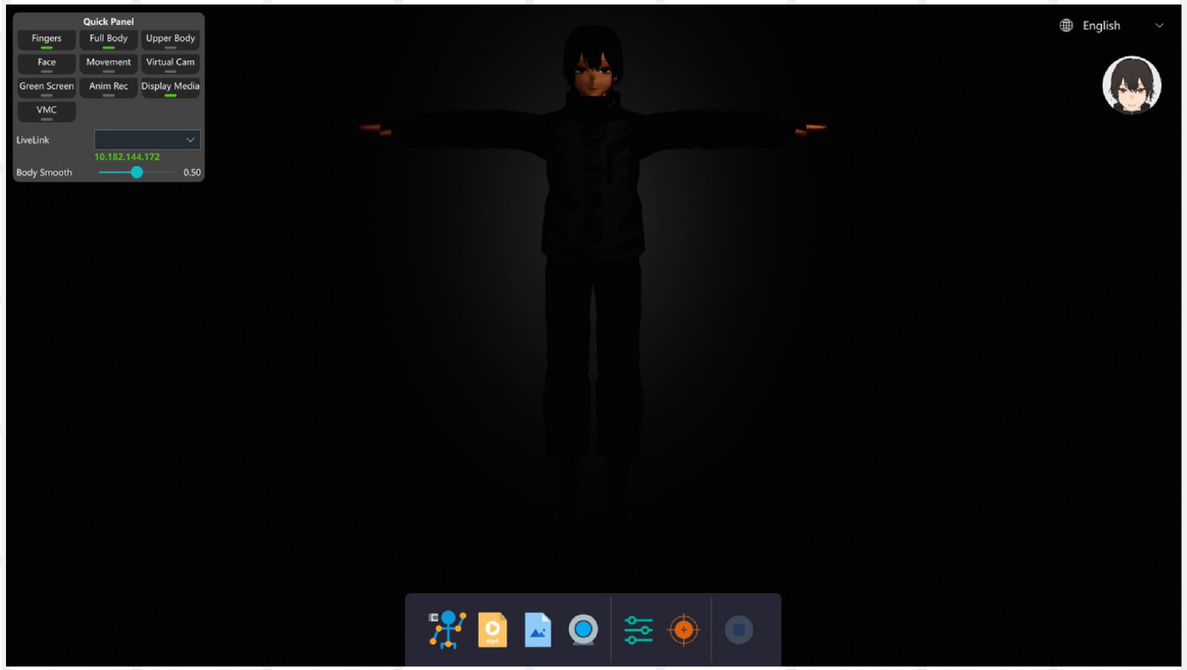
- Unreal Engine too difficult to work with right now for us because of new features being added right now.

- Overestimating our capabalities and also the hardware we had



FUTURE SCOPE

- Make it more accurate by using two cameras
- Deploy on more rigs/models
- Try it on Unreal Engine
- Try it for video of animals



REFERENCES

- Bazarevsky, V., Grishchenko, I., Raveendran, K., Zhu, T., Zhang, F. and Grundmann, M., 2020. Blazepose: On-device real-time body pose tracking. arXiv preprint arXiv:2006.10204.
- Sánchez Riera, J. and Moreno-Noguer, F., 2020. Integrating human body mocaps into Blender using RGB images. In 2020 13th International Conference on Advances in Computer-Human Interactions (ACHI) (pp. 285-290). International Academy, Research, and Industry Association (IARIA).
- Borodulina, A., 2019. Application of 3D Human Pose Estimation of Motion Capture and Character Animation (Doctoral dissertation, Master's thesis, University of Oulu, 6 2019).

THANK YOU

ANY QUESTIONS?

Sai Tarun Sathyan & Aman Raj Lnu