Sequential Yoga Pose Assessment for Self Training: A Deep Learning Approach with Surya Namaskar

Prepared By: Anwesha Samaddar
17MF3IM02

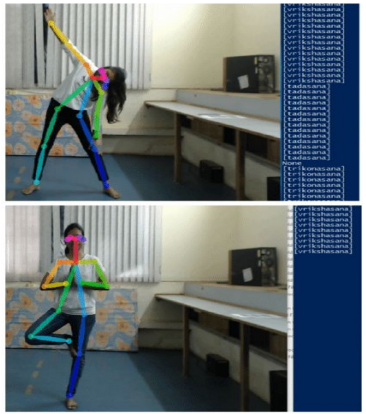
Supervisor: Prof. Mamata Jenamani

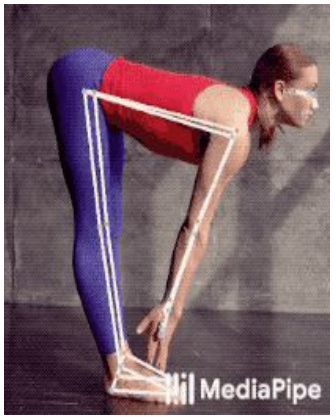
Introduction

- As per the research done in ***Occupational Health***, yoga programs at work positively affects both physiological and mental condition, most importantly, reduces stress.
- Yoga helps in ***back pain relief, improve posture, increase focus*** and ***work efficiency***.
- Nowadays people learn yoga by watching tutorials on TV, youtube videos or by teaching each other.
- ***Assessing the correctness*** *of pose or* ***tracking improvement*** with practice *may be difficult for novice learners.*

Motivation

- ***Using Pose Estimation techniques for Yoga Pose Assessment:*** a computer vision based approach for tracking human movements and detecting and analysing posture.
- Some popularly used algorithms for yoga pose tracking are ***OpenPose, PoseNet and BlazePose (MediaPipe)***.





- *Sequential yoga pose assessment for self training* would help user to ***visualize the mistakes*** *for live training*, know the ***pose sequence***, as well as *get* ***feedback*** *on pose accuracy and* ***performance improvement*** *over time.*

Literature Review

Authors	Year	Title	2D-Pose tracking	3D-Pose tracking	Pose Recognition	Pose Comparison	Sequence analysis
Yadav et al.	2019	***Real-time Yoga recognition using deep learning***	✓	✗	✓	✗	✗
Thar et al.	2019	***A Proposal of Yoga Pose Assessment Method Using Pose Detection for Self-Learning***	✓	✗	✗	✓	✗
Agrawal et al.	2020	***Implementation of Machine Learning Techniques for for Identification of Yoga Poses***	✓	✗	✓	✗	✗
Kothari et al.	2020	***Yoga Pose Classification Using Deep Learning***	✓	✗	✓	✗	✗
Chaudhari et al.	2021	***Yog-Guru: Real-Time Yoga Pose Correction System Using Deep Learning Methods***	✓	✗	✓	✓	✗

Gaps in the Literature

- ➔ Popular algorithms like Openpose, Tf-pose, PoseNet, Blazepose gives ***many incorrect keypoint detections for yoga poses with occluded parts***.
- ➔ Pose classifier based on ***2D skeleton*** (x,y) coordinates is ***unable to capture body rotations*** i.e., if body position is horizontal or vertical.
- ➔ To the best of our knowledge not much work has been done on ***pose sequence tracking, pose duration tracking or providing feedback*** based on learning history.
- ➔ We did not find any open source cloud based framework for storing and retrieving the pose analysis data.

Objectives

1. Developing a methodology for ***3D pose estimation and identification*** of individual ***Surya Namaskar*** steps

1. Developing ***indicators for feedback*** on individual pose geometry

1. Developing a methodology for ***tracking the right sequence*** of poses with time duration

1. Developing a ***cloud based framework*** for maintaining the learning history and developing indicators for tracking improvement

10 Surya Namaskar poses

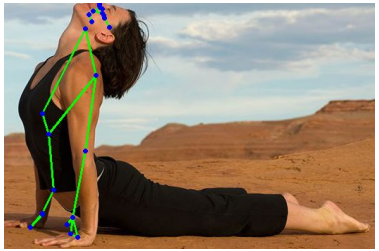
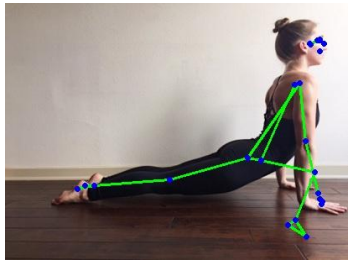
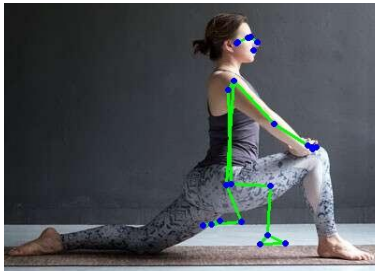
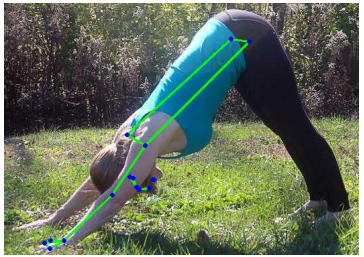
Progress towards Objective 1

Developing a methodology for ***3D pose estimation and identification*** of individual ***Surya Namaskar*** steps

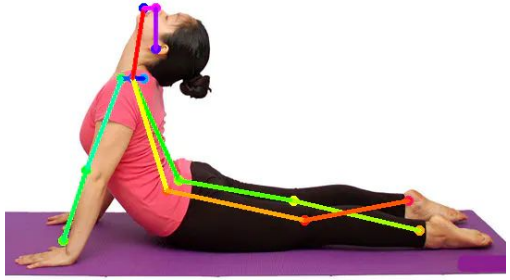
2D Yoga Pose Tracking Networks used so Far

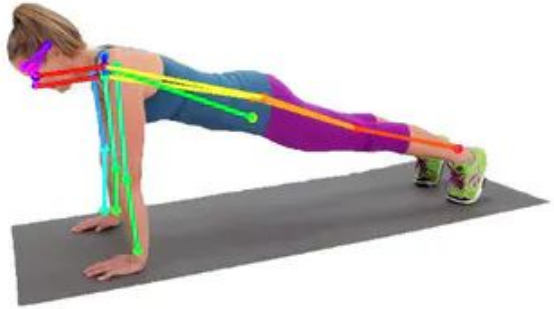
Average Precision measures how accurate our predictions are. i.e. the percentage of keypoint predictions that are correct as compared to the original annotations.

2D Network	Average Precision
Blazepose	62.6
Openpose	68.2
Posenet	48.8
AlphaPose	**78.6**

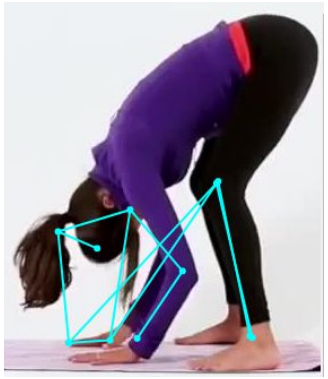





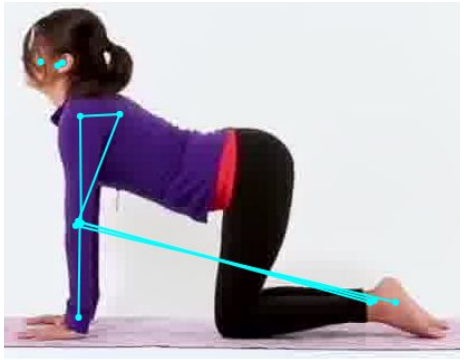
Incomplete/Wrong predictions by Blazepose





Wrong predictions by Openpose

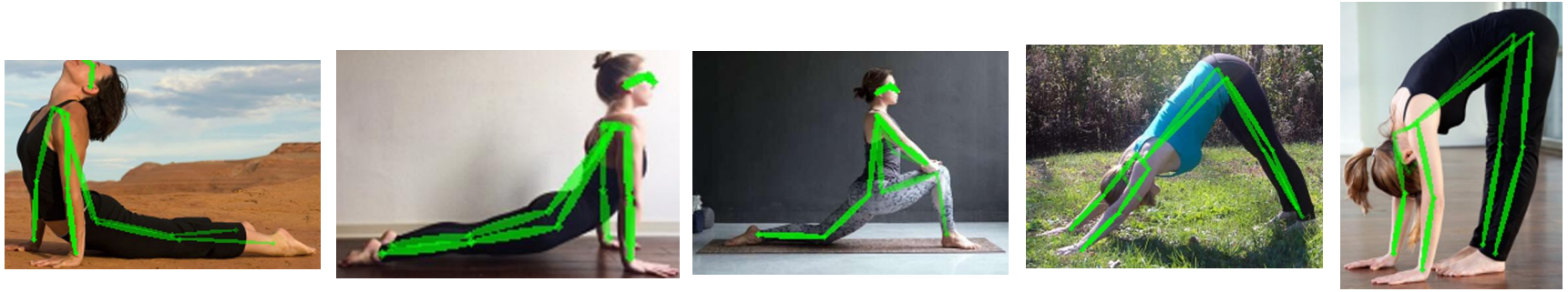




Wrong predictions by Posenet

Proposed 2D Pose Tracking Network used: **AlphaPose**

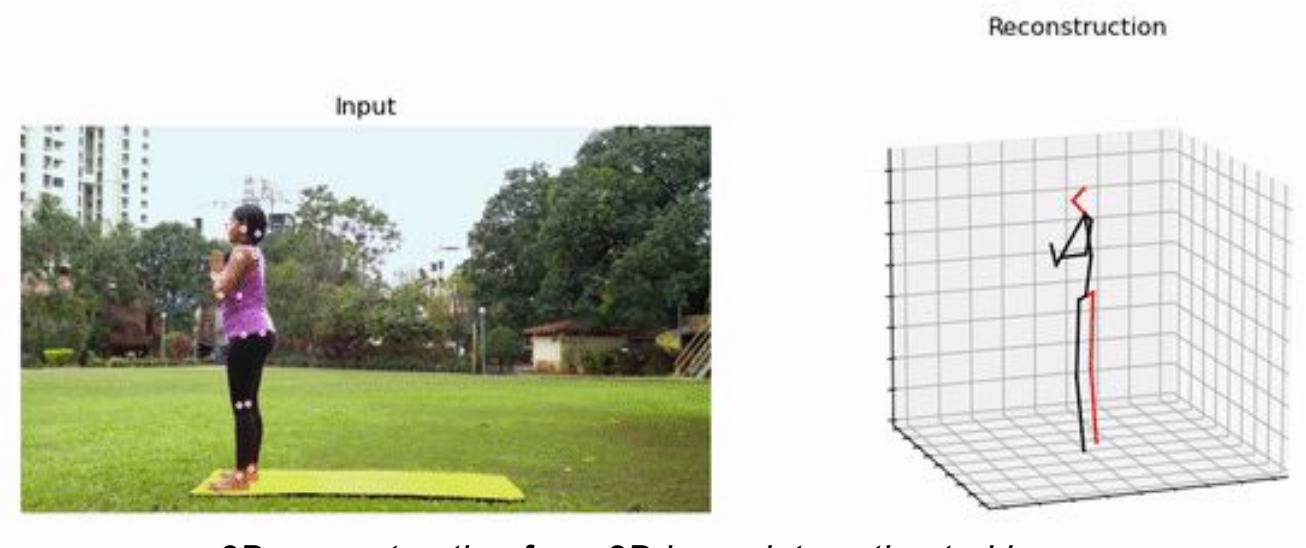
- ***AlphaPose:*** an accurate Regional Multi-person Pose Estimation method, which much higher accuracy for yoga pose tracking than Blazepose, Openpose or PoseNet.
- It is trained on the **COCO-Keypoints dataset** and follows a ***top-down approach*** using a ***person detector*** first in a frame and then estimating 18 keypoints to draw the skeleton.



Improved posture tracking with AlphaPose

3D Pose Tracking from 2D Keypoints

- **VideoPose3D:** using the predicted 2D keypoints by AlphaPose, the 3D pose (x,y,z keypoints) will be estimated and finally back-projected into the 2D space.



3D reconstruction from 2D keypoints estimated by AlphaPose

Existing Yoga Pose Recognition Techniques

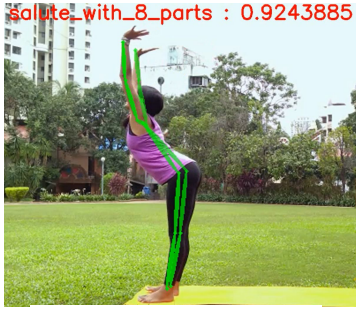
- Classifiers based on 2D Skeleton tracking
    - Deep learning based - LSTM , ANN
    - Machine learning based - SVM, Logistic Regression
- CNN based Classifier (transfer-learning based Image Classification)

Drawbacks of above methods:

➢ ***unable to capture body rotations*** (+90 /-90 degree)

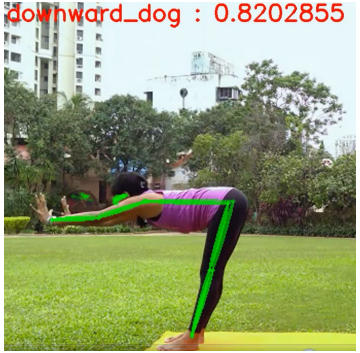
➢ cannot distinguish between ***left and right directions***

'Salute_with_8_parts' position



'Raised_arms' position

'Downward_dog' position



'Forward_bend' position

Wrong pose classifications with 2D tracking method

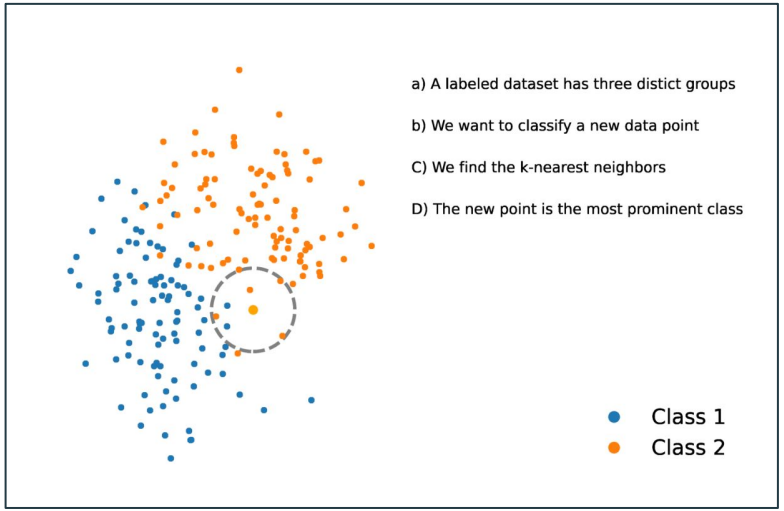
Proposed Yoga Pose Recognition Method :

Dataset Preparation for 3D Pose Recognition

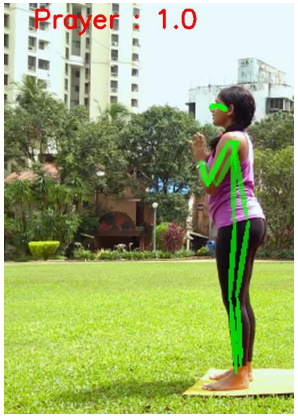
- Collected images of ***Surya Namaskar poses*** from open source Yoga pose images datasets
- ***Extracted 3D (x,y,z) pose landmarks*** from the 2D joints ***(extracted by Alphapose)*** and stored them in a .csv format along with the labelled pose as class.

Classification Using KNN:

- ***K-nearest neighbors algorithm*** is used as the classifier.
- ***Frame wise pose classification*** is done based on the top 10 nearest neighbours by calculating the ***euclidean distance*** *(most frequent class is the predicted pose).*

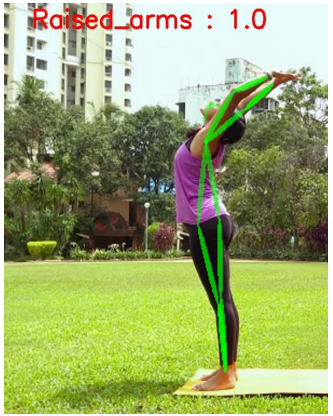


Correct Pose Recognition Outputs



‘Prayer’ Pose

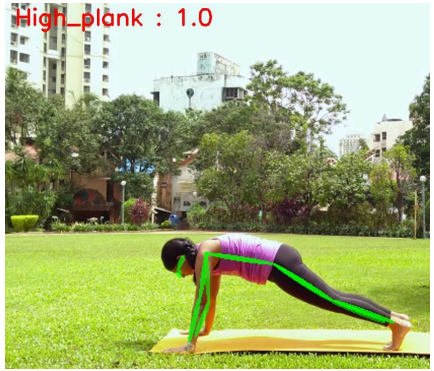
‘Forward bend’ Pose



‘Raised arms’ Pose

‘Equestrian’ pose (left leg)

Correct Pose Recognition Outputs

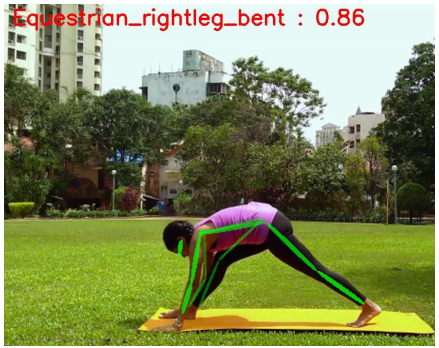


'High plank' pose

'Salute with 8 parts' pose

'Cobra' pose

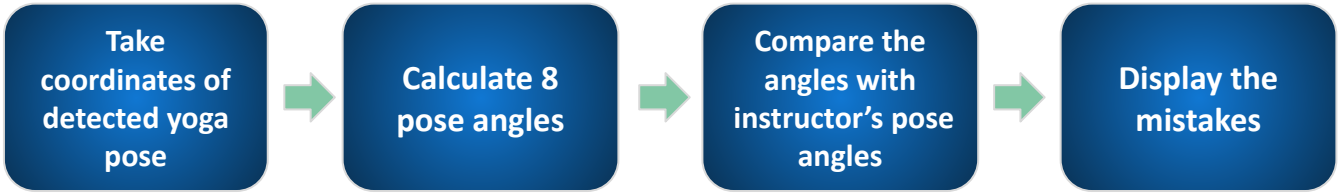
'Downward dog" pose



'Equestrian' pose (right leg)

Progress towards Objective 2

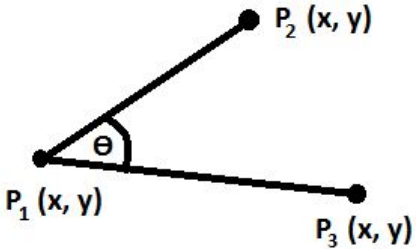
Developing ***indicators for feedback*** on individual pose geometry



Displaying mistakes by comparing the target pose angles with the user's recognized pose

To calculate angle between any 3 keypoints/joints:

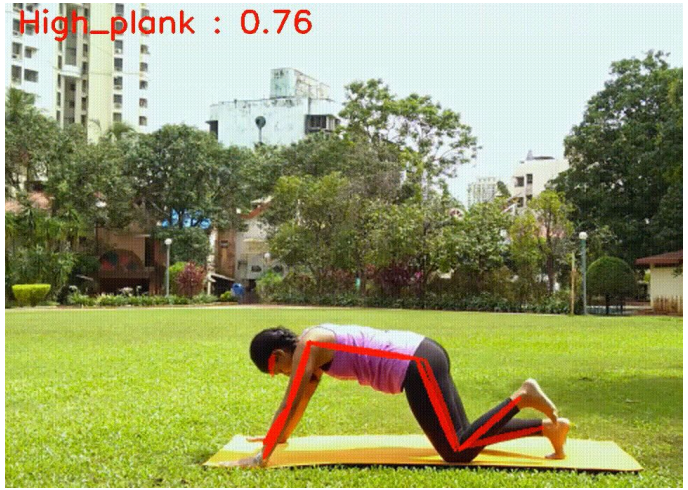
$$\theta = \arccos(x \cdot y / |x|\,|y|)$$



- **8 angles** have been compared between the target pose and the practitioner's pose: **hip** (left and right), **knee** (left and right), **elbow** (left and right), **shoulder** (left and right)
- Based on a threshold range for each angle, the display is done in **red**/**green** for **wrong**/**right** angles respectively

Display of indicators for right/wrong angles

Target Poses



Display of intermediate stage/wrong angles

Conclusions and Future Work

1. ***Chronological Pose Sequence*** and ***Pose Duration tracking*** for a complete cycle of Surya Namaskar.
2. Construction of a ***feedback*** based on analysis each cycle of poses.
3. Building a ***Cloud based*** framework for ***storing the Pose tracking analysis*** data and recording improvement.

Thank You!