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

Report submitted in fulfillment
of the requirements for the degree
of

Master of Technology
in Industrial and Systems Engineering

by

Anwesha Samaddar
17MF3IM02

Under the guidance of

Prof. Mamata Jenamani

INDIAN INSTITUTE OF TECHNOLOGY, KHARAGPUR

NOVEMBER 2021

DECLARATION

I thus declare that this thesis represents my views in my own words, and that I have appropriately cited and acknowledged the primary sources where others' ideas or words were included. I further certify that I have followed all academic honesty and integrity rules in my research and have not distorted, misrepresented, or falsified any idea, information, fact, or reference. I understand that any breach of the foregoing will result in disciplinary action by the Institute, as well as legal action from the sources who were not properly referenced or from whose sufficient permission was not obtained when required.

Anwesha Samaddar
17MF3IM02

CERTIFICATE OF APPROVAL

This is to certify that the Dissertation Report entitled, "Sequential Yoga Pose Assessment for Self Training: A Deep Learning Approach with Surya Namaskar" submitted by Anwesha Samaddar to Indian Institute of Technology, Kharagpur, India, is a record of bona fide Project work carried out by her under my supervision and guidance and is worthy of consideration for the award of the degree of Master of Technology in Industrial Engineering and Management of the Institute. The Dissertation Report has fulfilled all the requirements as per the regulations of the institute and in my opinion, reached the standard for submission.

_____

Prof. Mamata Jenamani
Department of Industrial and Systems Engineering
Indian Institute of Technology, Kharagpur
Date: November 2021

ACKNOWLEDGEMENTS

First and foremost, I would like to thank Prof. Mamata Jenamani, of the Department of Industrial and Systems Engineering, IIT Kharagpur, for giving me the opportunity to implement this relevant project titled "Sequential Yoga Pose Assessment for Self Training: A Deep Learning Approach with Surya Namaskar". Her guidance and support were instrumental in completing my dissertation on this fascinating topic.I would also like the thank all of the authors, journalists, columnists, and scientists whose thoughts and contributions were used in my project. My heartfelt gratitude goes out to the entire University personnel and administration for providing the infrastructure in the form of our online library and virtual classes, which they made so simple. Finally I would like to thank my parents, seniors, mentors and friends at the institution for their unwavering support during this effort.

Anwesha Samaddar
17MF3IM02

Contents

Abstract

In recent times, the COVID-19 pandemic has brought our fast moving world to a standstill. The imposed extended periods of lockdown in order to suppress the rapid spread of the disease also has limited our daily physical activities and social interaction resulting in serious physiological as well as psychological issues. Working from home comes with several drawbacks such as decreased productivity, lack of focus, risk of overworking and even health problems due to improper workspace conditions. According to the research done in Occupational Health, Yoga is considered as the most efficient way of achieving a calm nervous system, improving bone and muscle strength, easing body pain and relaxing our system. Self training for yoga is integral, however for beginners, proper assessment of the pose may be difficult and an incorrect pose might lead to serious muscle or ligament injuries. This study presents a deep learning approach for self-training and assessment of Surya Namaskar yoga postures. The proposed method uses a convolutional neural networks (CNN) based pose estimator Movenet, for live yoga pose tracking along with a 3D pose estimation network - VideoPose3D, for yoga pose recognition. This study also presents an approach for developing performance indicators, tracking pose duration and sequence and constructing and displaying feedback based on pose analysis.

Chapter 1: Introduction

Incorporating a disciplined yoga regime into employees' lives helps them boost their energy levels, improve work efficiency, release stress and create a positive impact on their physical and mental health. Practice of yoga and meditation helps calm down the nervous system, motivates employees to work, and reduces absenteeism at work. According to Occupational Research, Benefits of Yoga in workplace include:

- improves posture, balance and flexibility.
- helps with back pain relief
- increases focus and work efficiency
- improves immunity and strength
- helps us relax, develops patience
- increased mental and physical energy, lowers stress

Performing Yoga: Nowadays people learn yoga by watching tutorials on TV, youtube videos or by teaching each other. However, assessing the correctness of pose may be difficult for novice learners. Incorrect yoga practices may even lead to injuries. Further, correct yoga pose assessment can give us an indication about the health status of an individual.

This study presents an efficient approach for allowing the practitioner to track his/her own performance while practicing Surya Namaskar, get an indication of incorrect pose angles as compared to the target pose of the instructor and recieve a feedback based on performance.

Rest of the paper has been arranged in the following way:

- Chapter 2 discusses some relevant studies from the literature.
- Chapter 3 presents the objectives for this project.
- Chapter 4 contains the proposed pose estimation models with illustrations.
- Chapter 5 describes the proposed solution methodology and workflow
- Chapter 6 shows the experimental details and discussions
- Chapter 7 concludes our study and gives future research directions.

Prayer Raised arms Forward bend Equestrian (left side) High Plank

Salute with eight parts Cobra Downward Dog Equestrian (right side)

Fig 1.1 Surya Namaskar Poses

Chapter 2: Literature Review

Section 2.1: Using Pose Estimation techniques for Yoga Pose Assessment

Human pose estimation and activity recognition is a computer-vision based approach for tracking human movements and detecting and analyzing posture for several applications like video surveillance, training robots, motion capture for interactive gaming and augmented reality and also in AI powered sports or workout coaches. Yoga Pose assessment is a relatively newer application . Some popularly used algorithms for 2D pose tracking are TfPose, OpenPose, PoseNet, Movenet, AlphaPose and BlazePose MediaPipe. The precision of keypoints tracking is inversely proportional to the frame rate per second for each algorithm. The Movenet models developed by Tensorflow come in two variants: Lightning (for latency-critical applications) and Thunder (for higher tracking accuracy). Both Movenet models are suitable for building posture tracking applications in edge devices like Raspberry Pi, are super fast and highly accurate. Fig 2.1 shows the 17 keypoints tracked by Movenet, Openpose and Alphapose.

Fig 2.1 17 keypoints tracked by Movenet, Openpose and Alphapose.

Section 2.2: Real-time Yoga pose recognition

The works related to yoga posture estimation and classification/recognition that have been found can be broadly divided into two categories: image based and keypoint based classification. Shailesh et al. (2021) created a deep learning algorithm that can detect a yoga pose from an image or a video frame. They have used convolutional neural networks (VGG16 network architecture) and transfer learning based approach to identify images of 10 different asanas. Their classification model has 82% prediction accuracy. Yadav et al. (2019) proposed a hybrid deep learning model for Yoga recognition on real-time videos that uses a convolutional neural network (CNN) and a long short-term memory (LSTM), with the CNN layer extracting features from keypoints of each frame obtained from OpenPose and the LSTM layer providing temporal predictions.

Agrawal et al. (2020) has proposed a real-time pose recognition system for ten asanas which detects the skeleton using tf-pose estimation algorithm. Angles of the joints in the human body are extracted and asana classification is done using Random Forests, KNN, SVM and logistic regressionSruti Kothari demonstrated a CNN-based computational technique for identifying Yoga poses from images. They used a dataset of 1000 images spread across six classes to create the classification model, which was approximately 85 percent accurate.

Section 2.3: Yoga pose assessment and correction

To aid in the self-learning of Yoga, Thar et al. (2019) suggested a Performance Evaluation System for Yoga Pose Training. The system evaluated a user's yoga posture by computing the difference in body angles between the displayed instructor's pose and that of the user, marking the erroneous part between the learner and the instructor, and classifying the pose into four categories based on the average angle difference. The system's usefulness was demonstrated by applications of three Yoga positions to three different persons of various ages, genders, and body types.

Chaudhari et al. (2021) developed a system which consists of a pipeline for pose identification and then point localization on the human body and then followed by an error identification process. The system uses deep learning techniques can assess the user's pose from the front perspective and provide feedback to help them improve their yoga pose. In their project, a user-friendly dashboard was also established, on which the generated model is deployed.

Chiddarwar et al. (2020) proposed a system where 17 keypoints are predicted using PoseNet network and body angles are compared with the ideal angles of target pose. Voice-controlled Yoga activity is enabled by KontinousSpeechRecognizer module that allows us to get user sound input and process it continuously.

Section 2.4: Gaps in the literature

A few important conclusions which could be obtained from previous literature and which can be mentioned as research gaps are:

- ➔ Popular pose tracking algorithms like Openpose, Tf-pose, PoseNet, Blazepose give many *incorrect keypoint detections for yoga poses* with occluded parts.
- ➔ Pose classifiers based on 2D skeleton (x,y) coordinates cannot distinguish between the poses when the body positions are different. It is unable to *capture the body rotations* (+90 /-90 degree) i.e., if the body position is *horizontally lying down* on the ground or if the person is in an upright

vertical position. To overcome this, we need to incorporate depth (z-coordinate) in the classifier model.

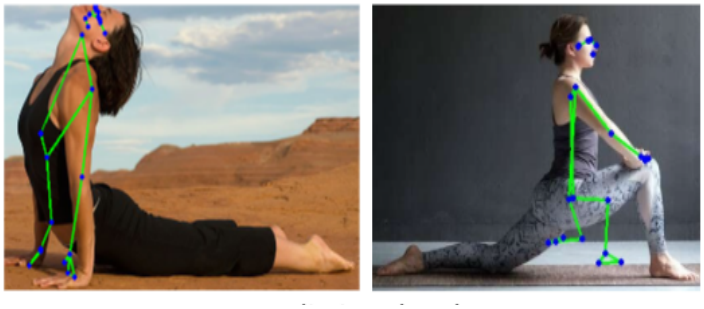


Fig 2.4.1 Wrong tracking by Blazepose

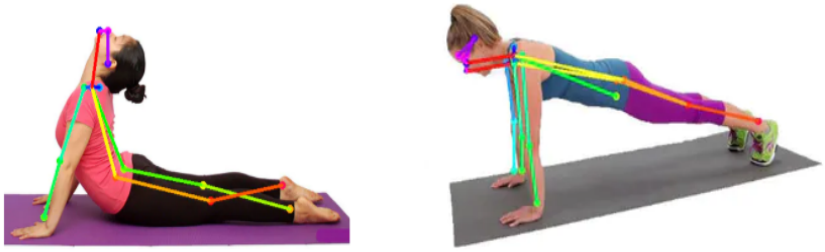


Fig 2.4.2 Wrong tracking by Openpose

➔ Mediapipe Blazepose (GHUM 3D) provides a 3D pose tracking network. A KNN classification model built on Blazepose may provide good classification accuracy, however the keypoints tracking is not very accurate due to their face detection model. Occluded or invisible facial features leads to improper tracking which makes it difficult to display the mistakes to the user or develop performance indicators effectively.

➔ To the best of our knowledge not much work has been done on yoga pose sequence tracking, pose duration tracking or providing feedback based on learning history.

Chapter 3: Objectives

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

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

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

4. Developing a *Graphical User Interface* for user-friendly demonstration of pose tracking, pose comparison and feedback on sequence analysis and accuracy of poses.

Fig 3: Sequence of the 10 Suryanamaskar poses

Chapter 4: Pose tracking in 2D and 3D

Section 4.1: Open Source datasets

MoveNet algorithm is used for 2D pose tracking in real-time. MoveNet was trained on two datasets: the COCO Keypoints Dataset and an internal Google dataset called Active. COCO consists of over 100K person instances labeled with over 1 million total keypoints. It is the standard benchmark dataset used for pose estimation in 2D, however it does not contain enough image annotations for yoga or fitness applications. To expand the training data and boost the model performance, the dataset - Active was created following the same 17 COCO keypoints format. It contains keypoints labellimg on yoga, fitness and dance image frames extracted from YouTube videos.

The pose analysis for displaying the errors/mistakes will be performed by tracking with AlphaPose network as it gives higher tracking accuracy but is low on latency (unable to give realtime output). AlphaPose is trained on the COCO keyponts dataset and can be used only on pre-recorded videos in CPU.

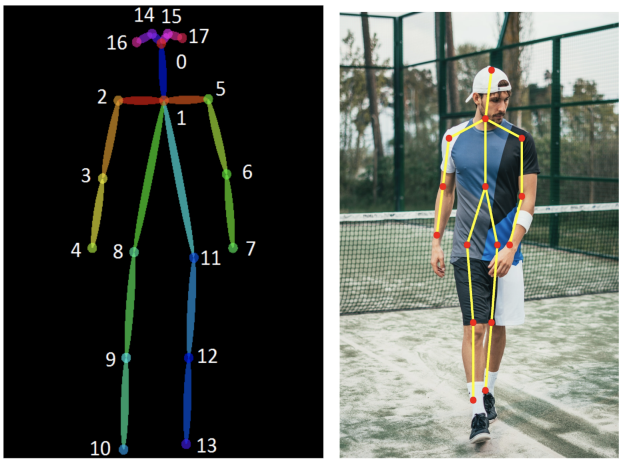
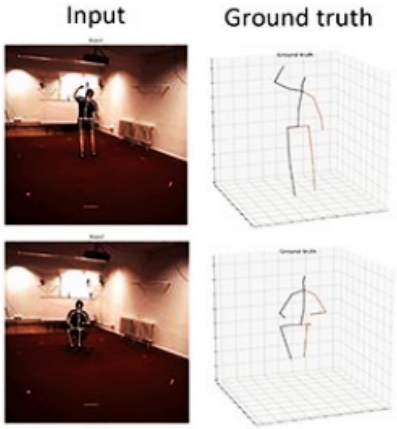


Fig 4.1: 17 keypoints provided by COCO

An example of the annotations (of a single image) included in the COCO keypoints json file for model training:

```
"annotations":[ {
    "segmentation":[[424.01,256.23,..,41.53,
            507.95]],
    "num_keypoints": 17,
    "area": 5823.9286,
    "iscrowd": 0,
    "keypoints": [329,356,2,...,423,569,2],
    "image_id": 289343,
    "bbox": [414.01,505.08,81.84,221.55],
    "category_id": 1,
    "id": 201807
            }  ]
```

For reconstruction of the 2D pose into 3D, VideoPose3D model is used which has been trained on the Human 3.6M Dataset. It contains 3.6 million 3D human poses and corresponding images and is



Human 3.6M 3D skeleton

used for training VideoPose3D to get the 3D keypoints. High-speed motion capture systems are used to obtain accurate 3D joint positions and joint angles. 15 sensors (4 digital video cameras, 1 time-of-flight sensor, 10 motion cameras) were used to capture the data.

Section 4.2: 2D live pose tracking with Movenet

MoveNet is a bottom-up pose estimation algorithm developed by Tensorflow that uses heatmaps to locate human keypoints accurately. There are two parts to the architecture: a feature extractor and a collection of prediction heads. The prediction technique is based on CenterNet, but with a few tweaks that increase speed and accuracy. The TensorFlow Object Detection API is used to train all of the models.

MobileNetV2 with an associated feature pyramid network (FPN) is MoveNet's feature extractor, resulting in a high-resolution (output stride 4) and semantically rich feature map output.

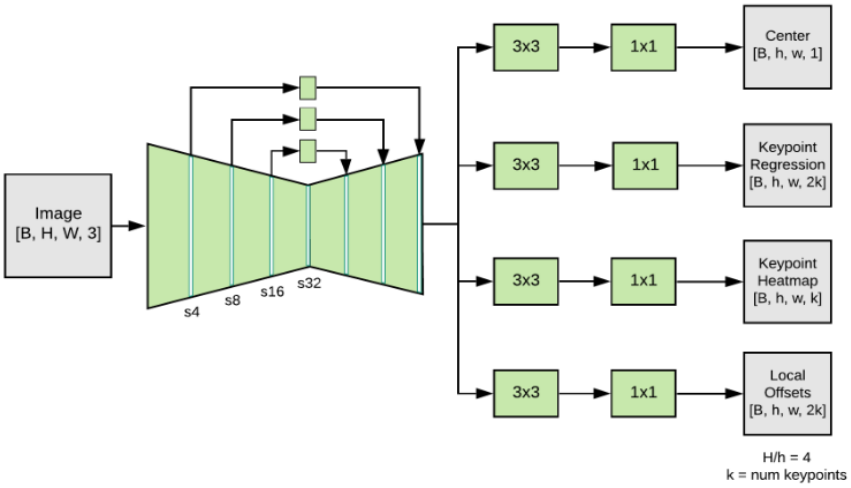


Fig 4.2.1: MoveNet Architecture

Here, B is the batch size (number of images per batch), H is the height of the image, W is the width of the image and C is number of channels (usually uses 3 channels for RGB).The feature extractor is equipped with four prediction heads, each of the heads is at the origin of a step in the sequence of operations that allows the model to function and thus, to define all 17 key points of a posture:

- **Global heat map**: prediction of a person's center of gravity. This data is then taken into account by the other predictive heads.

- **Key points regression field**: Based on the center of gravity, the model predicts all the key points of a person according to his position at time T.
- **Heat map of key points**: Using the regression field, the system then predicts the location of all the key points of a person, taking into account only the person in the foreground.
- **Two-dimensional shift field**: The final set of key points is selected by taking into account local 2D shift predictions to refine the final result.

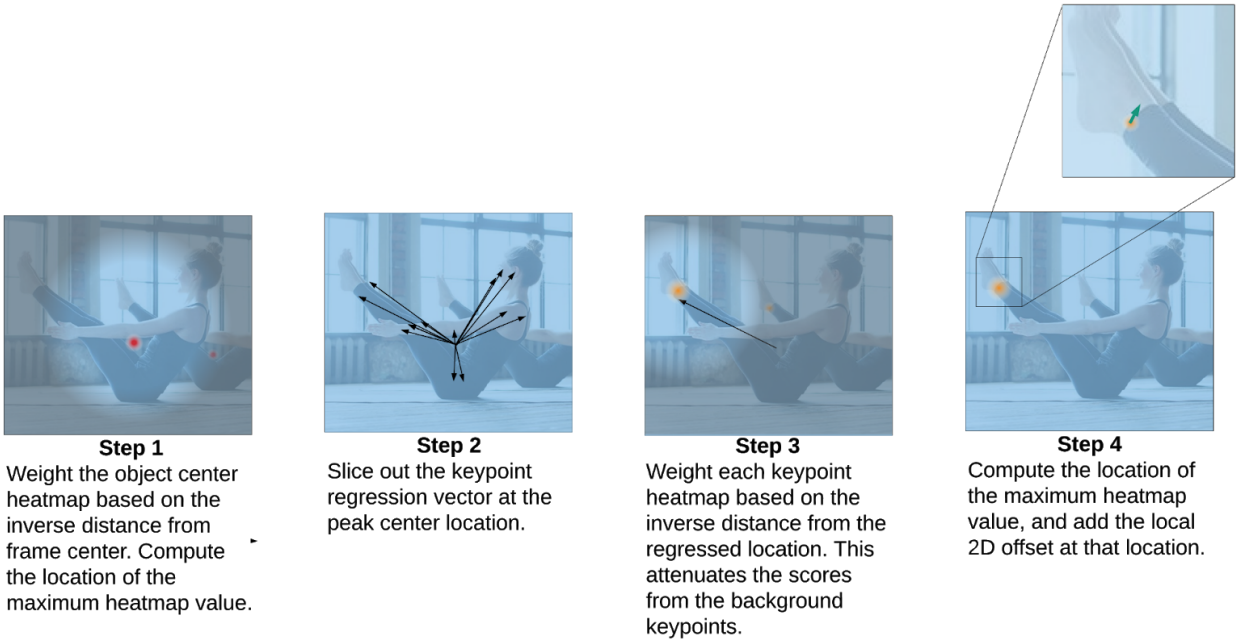


Fig 4.2.2: The four steps of MoveNet Algorithm

Section 4.3: 2D pose tracking with Alphapose

AlphaPose is an accurate multi-person pose estimator, and also the first open-source system that achieves much higher accuracy for yoga pose tracking than Blazepose and Openpose. It follows a top-down approach using a person detector to first identify the person in a frame and then estimate 18 keypoints to draw the skeleton. We would use the Alphapose network for the a better tracking, displaying of mistakes and providin a percentage score for showing pose correctness.

AlphaPose gives the framewise estimated keypoints in a json file as described :

```
{
      "image_id" : string, image_1_name,
      "category_id" : int, 1 for person
      "keypoints" : [x1,y1,c1,...,xk,yk,ck],
      "score" : float
}
```

- *keypoints contains the body part locations (x,y) and detection confidence score (c):*

  x1,y1,c1,x2,y2,c2, ... c is the confidence score in the range [0,1]

- *"score" is the confidence score for the whole person*

Section 4.4: 3D pose reconstruction

VideoPose3D: VideoPose3D is a fully convolutional network for estimating 3D human posture in videos. It employs dilated convolutions and temporal information

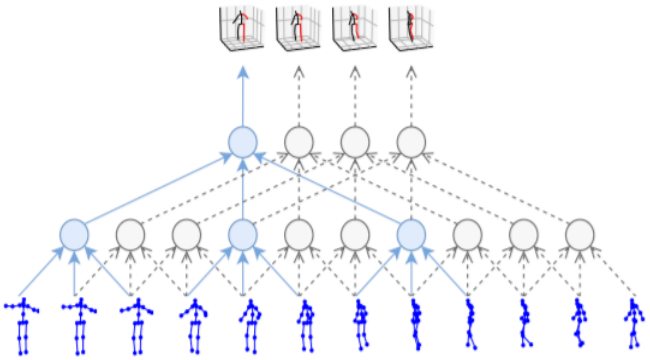


Fig 4.4 Dialated temporal convolutions to capture long term information

across 2D keypoint trajectories. It also employs back-projection in a semi-supervised training technique that regresses the person's 3D trajectory using a sequence of anticipated 2D positions as input.

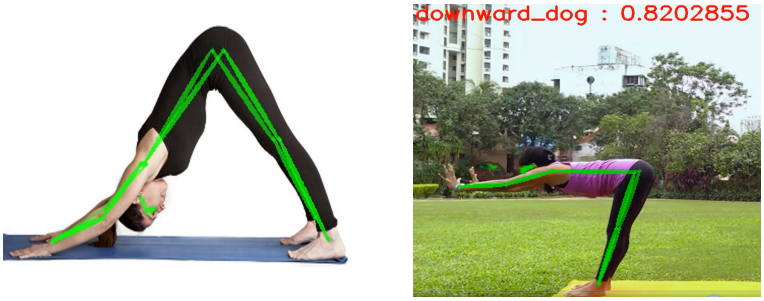
Using the predicted 2D keypoints by AlphaPose, the 3D pose (x,y,z keypoints) will be estimated with the VideoPose3D network and finally back-projected into 2D space.

Section 4.5: Existing pose recognition techniques

- Classifiers based on 2D Skeleton tracking which use the 17 2D extracted keypoints from input frame and compare with a poses keypoints dataset to determine the performed pose. The classifiers are usually deep learning based prediction models using LSTM or ANN, or machine learning based models using SVM or Logistic Regression. 2D tracking based classification is unable to capture body rotations (+90 /-90 degree) i.e. horizontal or vertical positioning for a particular asana

- CNN based Classifier (feature-learning based Image Classification) is another method to determine the poses. CNN based classification method cannot distinguish between left and right directions and requires a very large dataset for transfer learning.

- Two examples are shown here where 2D keypoints based ANN classifier is showing misclassifications. The solution would be to include the depth or z-coordinate for classification.



'Downward_dog' position *Changing to 'Forward_bend' position*

Fig 4.5.1 Example of mis-classification of 'Forward_bend' position as 'Downward_dog' pose

'Salute_with_8_parts' position

Coming to 'Raised_arms' position

Fig 4.5.2 Example of mis-classification of 'Raised_arms' position as 'Salute_with_8_parts' pose

Chapter 5. Solution Methodology

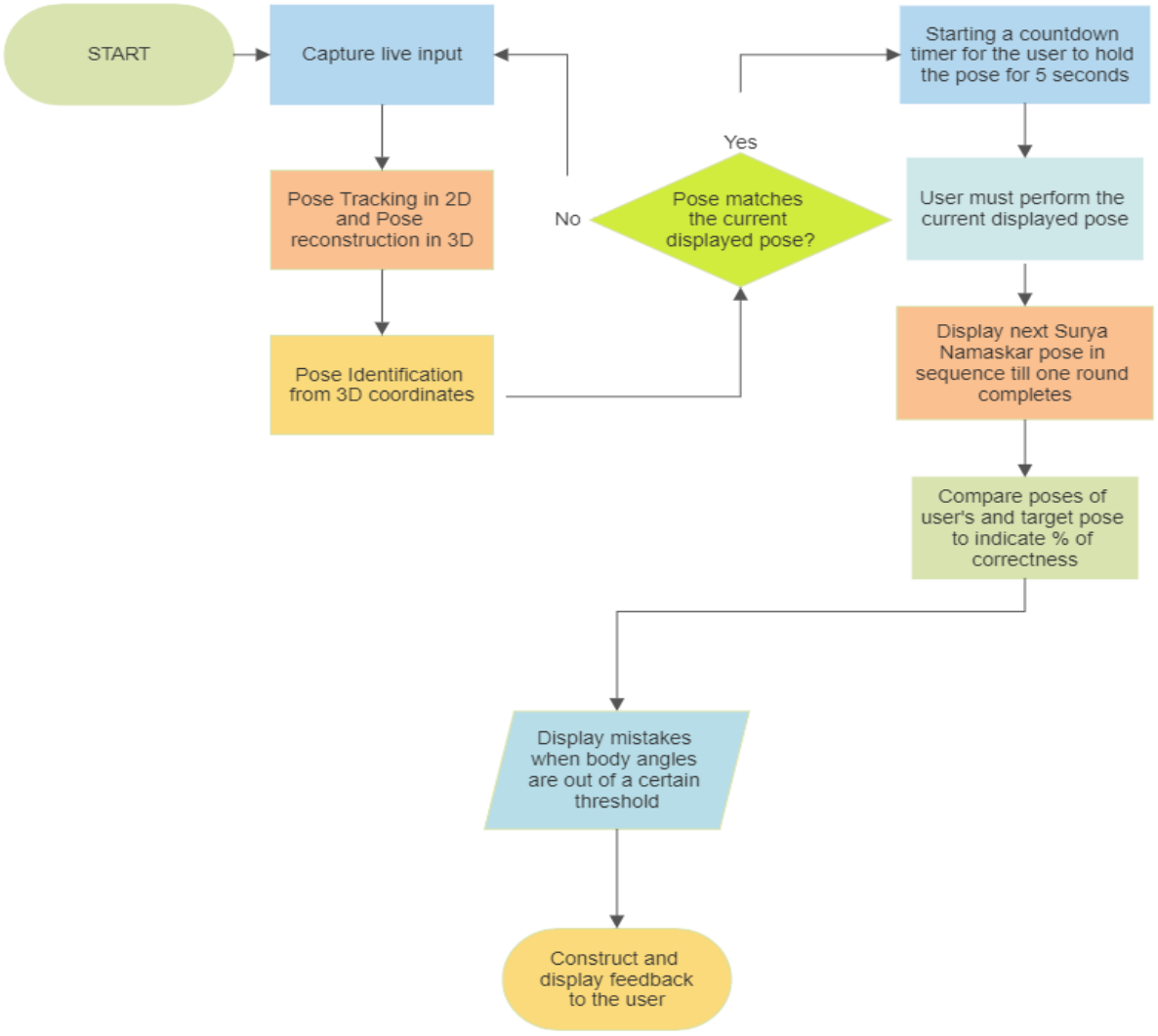


Fig 5: Flowchart of our proposed solution approach

The solution approach proposes to capture live input and extract the 2D skeleton keypoints followed by the 3D pose reconstruction for yoga pose recognition. System checks if the recognised pose matches with the current target pose of Surya Namaskar sequence. A countdown timer starts if pose matches upto a certain threshold and sequentially next pose is displayed on the screen for the user to perform. After completion of one complete Surya Namaskar cycle, the user can check his pose analysis video where errors or mistakes and the percentage of correctness will be shown while the cycle was performed. A feedback is displayed based on the time and sequence of how each pose should be held.

Chapter 6: Experimental details

Section 6.1: Proposed setup

- Live video feed in real time would be taken through visual camera.
- Display screen would show the GUI, the sequence of poses to be performed and a video of the analysis of mistakes performed.

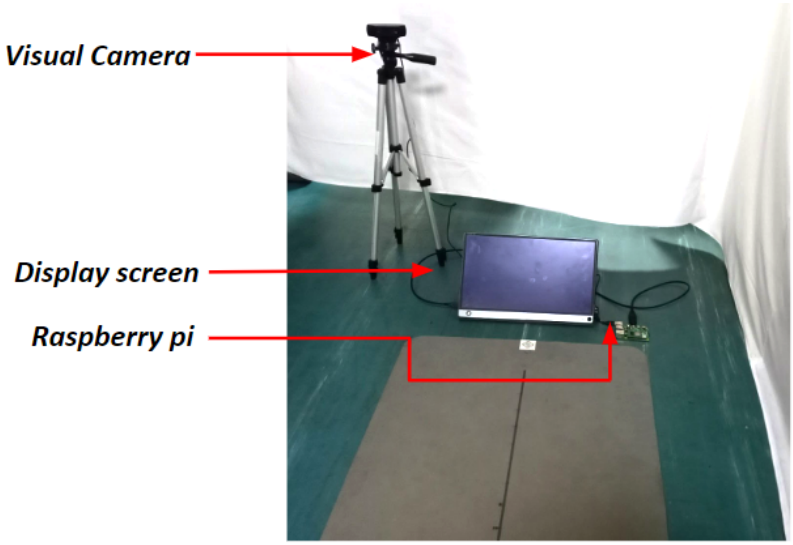


Fig 6.1.1: Proposed setup

- A feedback would be provided based on percentage of pose correctness and time duration of each pose performed.
- Raspberry pi with 4gb RAM is the edge device where all the processing takes place.

Fig 6.1.2:Webcams used (within Rs. 2000)

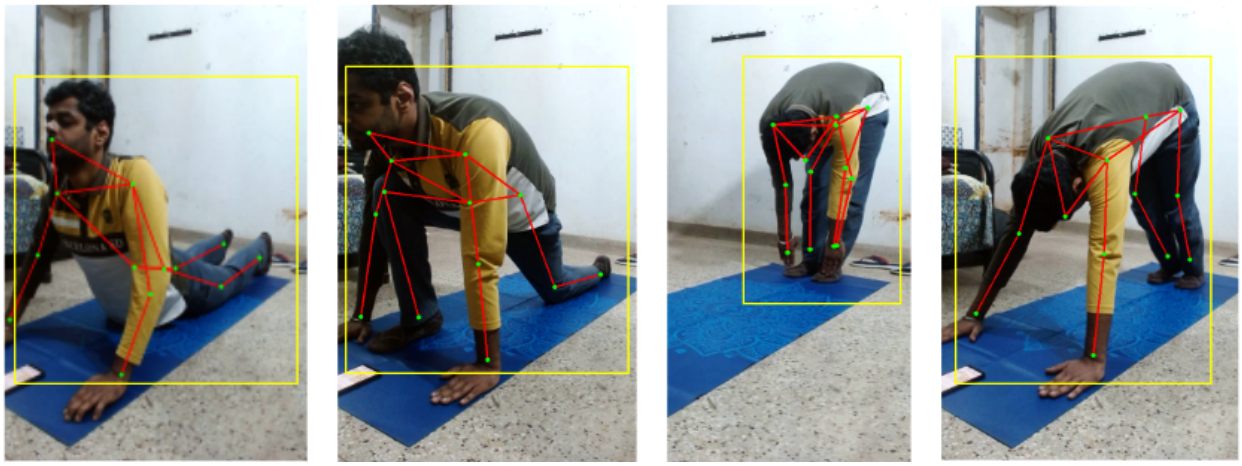


Fig 6.1.3: Examples of 2D Live keypoints tracking and human detection by MoveNet

The MoveNet algorithm first resizes each frame into 256*256 size and runs the prediction model on the image. A [1, 1, 17, 3] shaped float numpy array is returned by the algorithm representing the predicted keypoint coordinates and scores. 17 keypoints in the format of [y,x,c] where x,y are the normalized coordinates and c is the confidence score for each set of keypoints. It also gives the bounding box coordinates for the human detector. The indexing of the keypoints in the array determines which body part or joint it represents. Indexing is same as that of COCO keypoints format.

Example output of keypoints tracking by MoveNet on a single image:

```
array([[[[0.34483454, 0.57337326, 0.41690227],
         [0.3344181 , 0.57997423, 0.65108603],
         [0.3362406 , 0.5662141 , 0.5623235 ],
         [0.3426203 , 0.59123063, 0.49262577],
         [0.34783185, 0.559141  , 0.51432425],
         [0.4153647 , 0.6032493 , 0.6649319 ],
         [0.40409812, 0.54917276, 0.62941784],
         [0.4914355 , 0.6082857 , 0.5462451 ],
         [0.47926578, 0.5475651 , 0.6033933 ],
         [0.5044184 , 0.6060807 , 0.2381939 ],
         [0.48617446, 0.5838926 , 0.40250626],
         [0.52801514, 0.5884073 , 0.6880612 ],
         [0.5269756 , 0.5518017 , 0.6209302 ],
         [0.5580775 , 0.5893122 , 0.5947206 ],
         [0.5474108 , 0.5600575 , 0.39507794],
         [0.68671334, 0.59031665, 0.6156337 ],
         [0.69810486, 0.56098324, 0.41984144]]]], dtype=float32
```

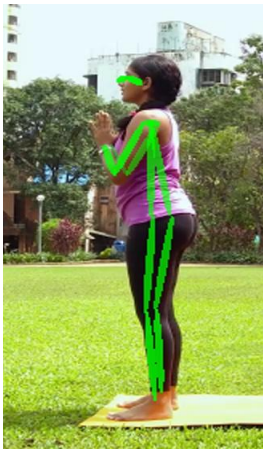
Section 6.2: Surya Namaskar Pose Recognition with 3D Keypoints

For creating Surya Namaskar Pose Recognition Dataset, images of the 9 distinct poses were collected from various Open source datasets mentioned below:

- Yoga 82 Dataset: a large-scale yoga pose recognition dataset with 82 classes with many complex poses. Labels for yoga poses are provided based on the body configuration of the pose.
- Kaggle Yoga Poses Dataset: contains images for 5 well-known yoga poses like downward dog, goddess pose, tree pose, plank pose and warrior pose.
- Kaggle Yoga Pose Image classification dataset: contains 107 different asanas for classification
- Google Images

3D (x,y,z) skeleton landmarks were reconstructed from the 2D (x,y) positions extracted by Alphapose for each image in the Surya Namaskar poses dataset created. These 3D positions were stored in a '.csv' format along with the labelled pose as class.

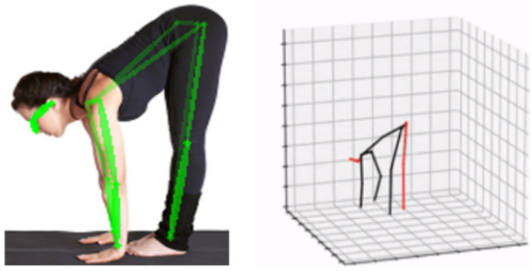
Example output of keypoints tracking by AlphaPose for the image at left:



```
{

"image_id": "1.jpg",
"category_id": 1,
"keypoints": [ 306.754, 647.575, 0.575,301.331,......]
"score": 2.656543

}
```



2D Pose *3D reconstruction*

Fig 6.2 3D reconstruction from 2D keypoints

Section 6.3: KNN for pose classification

K-nearest neighbors algorithm (k-NN): k-NN is used as the classifier. The algorithm determines the pose class based on the top 10 closest samples in the labelled dataset. Frame wise pose classification is done based on the top 10 nearest neighbours by calculating the euclidean distance for each sample set of 3D skeleton coordinates with reach of the records in the dataset. The label appearing the maximum number of times among the nearest 10 neighbours, will be the assigned 'pose' for the current video frame. This method shows much improved classification accuracy and solve the issue of identifying body rotations.

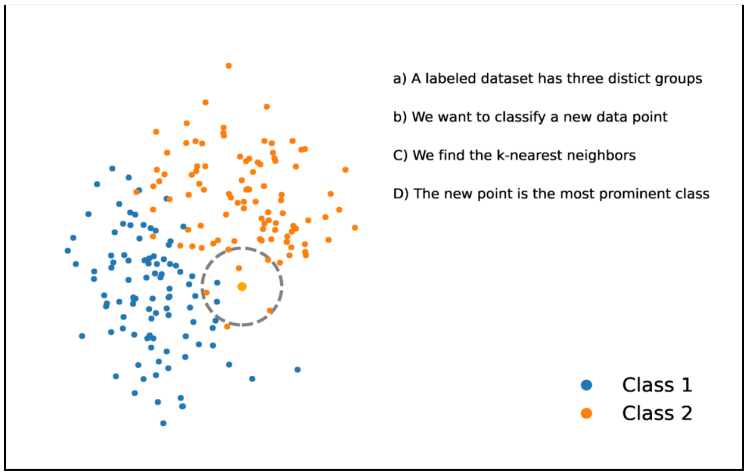
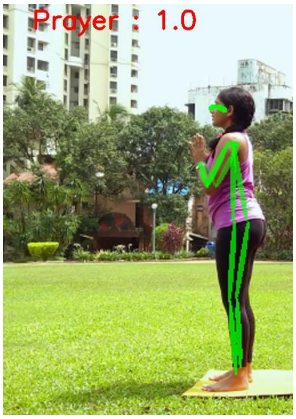


Fig 6.3.1 An illustration of KNN algorithm using two classes

<u>Correct Pose Classification examples from live as well as pre-recorded video frames:</u>



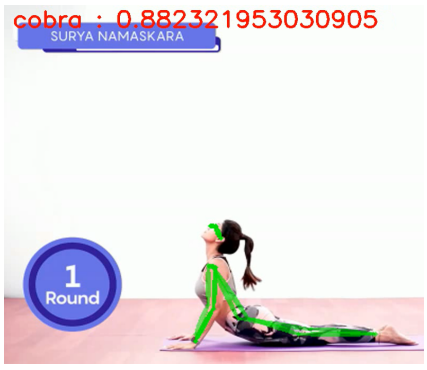


Fig 6.3.2 : Examples of pose classification outputs with both Alphapose and Movenet

Section 6.4: Pose sequence tracking with time duration

Once webcam starts capturing user's live video, 2D tracking by Movenet starts and simultaneously 3D reconstruction and pose identification runs on the background. The target Surya Namaskar poses are displayed on the screen according to the sequence which should be followed.

User must perform the current displayed pose on the screen. When the system identifies that the user is in the displayed pose and the pose classification score is within a certain confidence threshold, a countdown timer of five seconds will start and the user has to hold the position for five seconds. If user changes the pose

meanwhile, the countdown stops and restarts on coming back to the displayed pose

Fig 6.4.1: Display of timer while performing 'prayer' pose live

again. Once timer reaches to zero, the next pose in sequence will be displayed on the screen and the above process is repeated. Timer lets us keep track if the user is performing the poses in the correct Surya Namaskar sequence as well. Accordingly, feedback will be constructed based on the sequence maintained, poses not performed or missed and duration held for each pose.

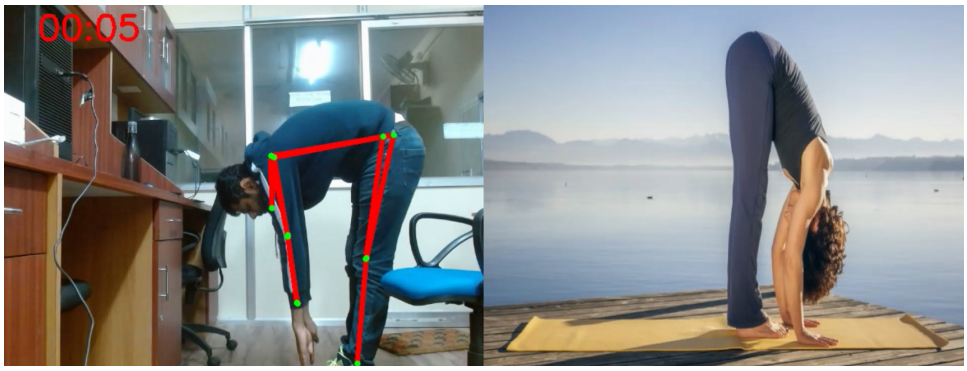


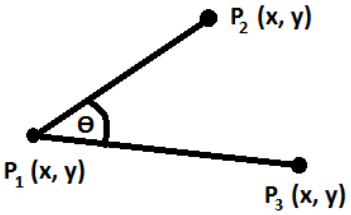
Fig 6.4.2: Display of timer while performing 'forward bend' pose live

Section 6.5: Indicators for feedback on individual pose geometry

For analysing the correctness of each pose performed, first Alphapose will be used for tracking on the recorded live video stored separately. As Alphapose has a higher keypoints tracking precision, the pose comparison results and displaying of mistakes / errors while performing yoga would be more accurate.

For comparison of each target pose with the practitioner's pose, the angles

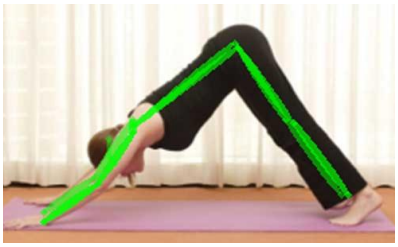
calculated are : hip (left and right), knee (left and right), elbow (left and right), shoulder (left and right). Based on a threshold range for each angle, the display is done in red/green for wrong/right angles respectively.

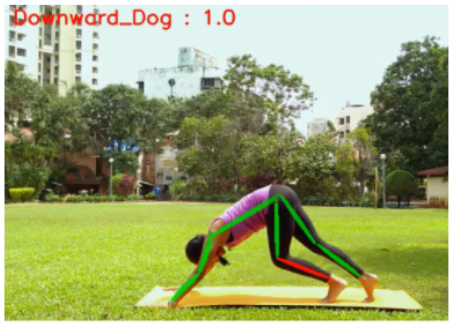


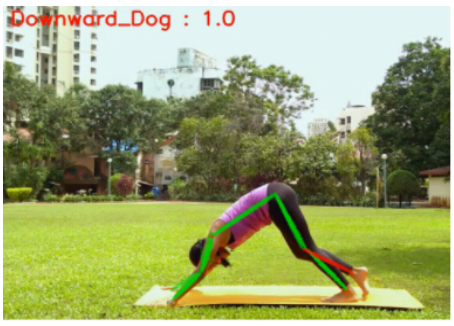
Formula used to calculate angle between any 3 keypoints:

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

Example output:
With 'Downward dog' as target pose⇒







Displaying limb positions with wrong angles in 'red' and correct ones in 'green'

Display on achieving the target pose ⇒

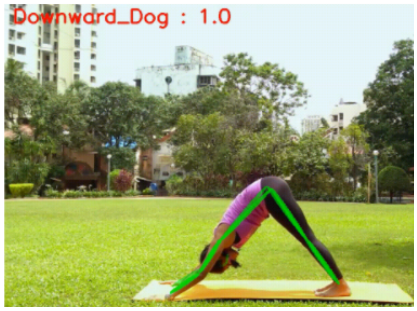


Fig 6.5 : Example output for indicators of mistakes/errors

Section 6.6: GUI for self-training, pose analysis and feedback

In order to build a user-friendly application, a python GUI has been developed. It contains options for starting the live surya namaskar training, for visualizing the results and feedback for improvement.

Fig 6.6.1: GUI display screen

Clicking on 'Start' will start the webcam video capturing and simultaneously show the Surya Namaskar poses to be performed one after another in proper sequence.

Fig 6.6.2: First Surya Namaskar pose on display screen with countdown displayed on upper left corner

Fig 6.6.3: Second Surya Namaskar pose on display screen, sequence tracking done with the help of countdown timer, 3D reconstruction and pose recognition algorithm running together

After 1 cycle of Surya Namaskar is completed, the webcam capture stops and a separate video is stored for pose comparison and displaying of mistakes. Clicking on 'Preview Video' will first run the pose analysis program on the recorded video where Alphapose is used for tracking and simultaneously analysis based on pose geometry for each recognized pose. A pose accuracy percentage is displayed based on the number of angles within threshold range for a fixed target pose.

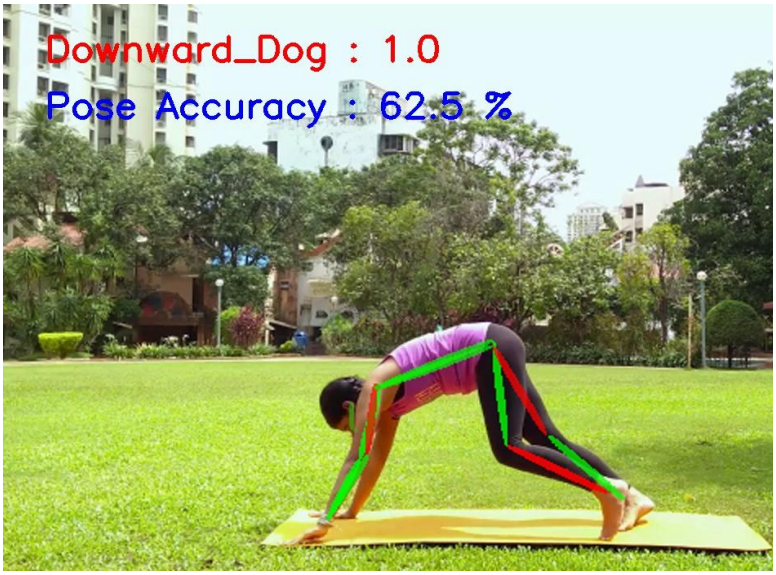


Fig 6.6.4 : Frame showing when limb angles are not within threshold while performing 'Downward Dog' pose

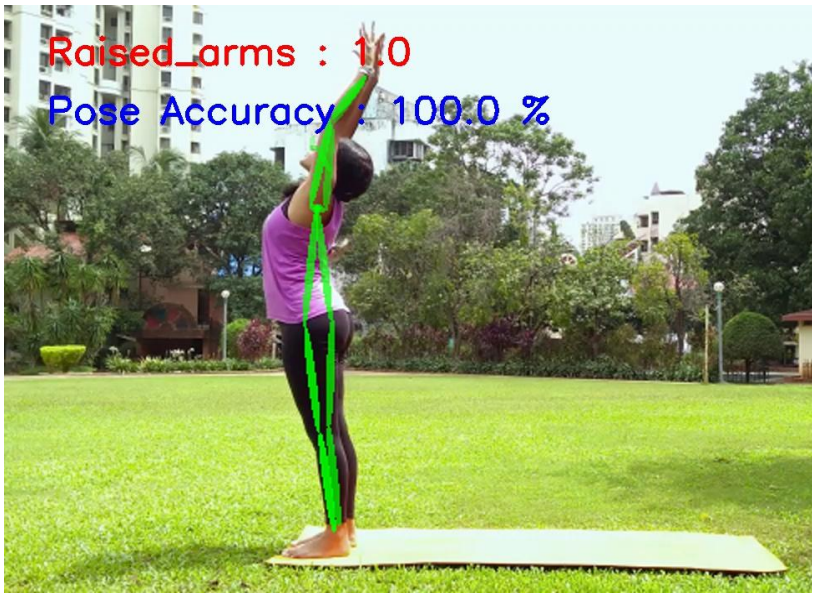


Fig 6.6.5 : Frame showing when limb angles within threshold while performing 'Raised arms' pose

Feedback window informs the user whichever poses has been missed or has not been performed for the entire five seconds duration. It also displays the poses that has been performed at the proper sequence and the associated pose accuracy.

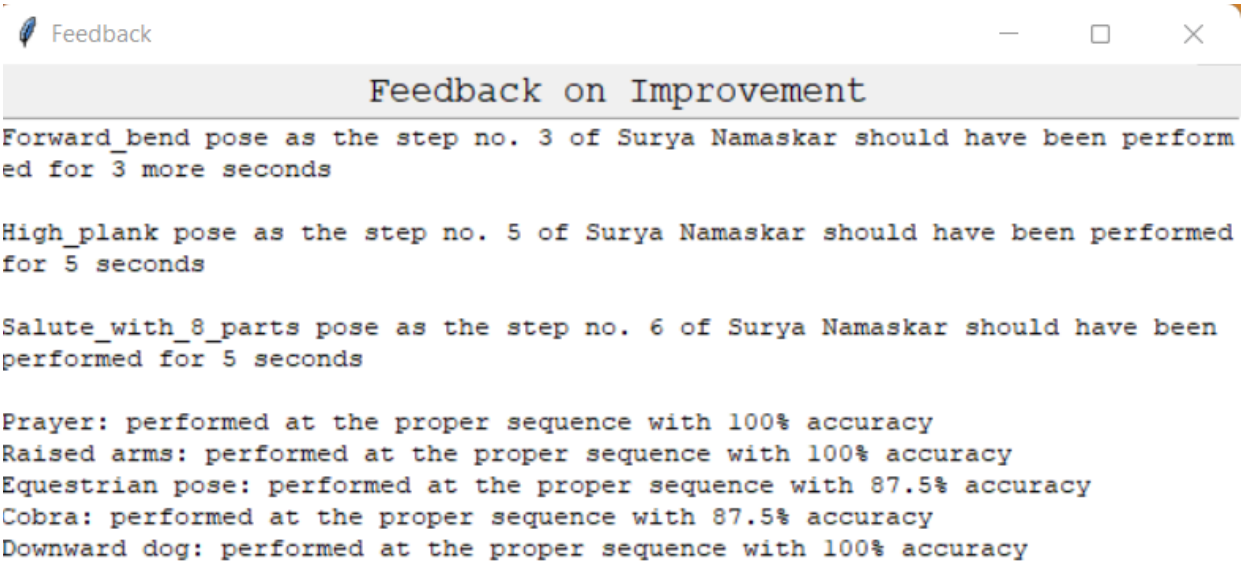


Fig 6.6.6 : Feedback window displayed on clicking 'Feedback' in the GUI screen

Chapter 7: Conclusions

The system which has been proposed for self training of Surya Namaskar enables the users to learn how to perform each pose accurately step by step in real-time. An analysis of correctness of each pose is also shown with certain indicators for displaying errors/mistakes and accordingly a feedback is constructed. The proposed system is capable of performing the above using a CPU or a Raspberry-pi edge device.

2D pose tracking, 3D pose reconstruction and recognition, geometry based pose comparison and sequence analysis are the algorithms running on the background. For certain yoga poses, highly occluded body parts may not be properly tracked by MoveNet or AlphaPose due to lack of occluded keypoint annotations (for yoga poses) in the COCO and Active training dataset. For improving tracking accuracy and making the models more robust for tracking other yoga poses, we need to train the network with more annotated yoga images.

Clearer backgrounds give better tracking results as compared to busy backgrounds, which is again, due to the lack of similar images in the dataset. The COCO dataset mainly contains high contrast and clear background images. According to the desired use cases, images for training need to be annotated. Higher resolution cameras can be used for filtering out the noise and getting a clear image.With the help of GPU processor we can achieve realtime speeds with Alphapose for the self-training mode as well.

For directly obtaining 3D pose coordinates from camera, Azure Kinect Body Tracking SDK can be used which contains 3D depth sensors, RGB camera and I.R. transmitter. Directly pose classification can be done without the use of the Videopose3D algorithm which would give faster processing and results, however the cost of the overall system increases.

Chapter 8: References

1. Shailesh et al. (2021), Yoga Asana Identification: A Deep Learning Approach, IOP Conference Series: Materials Science and Engineering
2. Fang et al. (2016), RMPE: Regional Multi-person Pose Estimation, Computer Vision and Pattern Recognition
3. Bazarevsky et al. (2020), BlazePose: On-device Real-time Body Pose tracking, Computer Vision and Pattern Recognition
4. Cao et al. (2019), OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields, IEEE Transactions on Pattern Analysis and Machine Intelligence
5. Pavllo et al. (2019), 3D human pose estimation in video with temporal convolutions and semi-supervised training, Computer Vision and Pattern Recognition
6. Yadav et al. (2019), Real-time Yoga recognition using deep learning, Neural Computing and Applications
7. Thar et al. (2019), A Proposal of Yoga Pose Assessment Method Using Pose Detection for Self-Learning, International Conference on Advanced Information Technologies
8. Agrawal et al. (2020), Implementation of Machine Learning Techniques for Identification of Yoga Poses, IEEE 9th International Conference on Communication Systems and Network Technologies (CSNT)
9. Kothari et al. (2020), Yoga Pose Classification Using Deep Learning, SJSU ScholarWorks
10. Chaudhari et al. (2021), Yog-Guru: Real-Time Yoga Pose Correction System Using Deep Learning Methods, IEEE International Conference on Communication information and Computing Technology (ICCICT)
11. Chiddarwar et al. (2020), AI-Based Yoga Pose Estimation for Android Application, International Journal of Innovative Science and Research Technology
12. Rishan et al. (2020), Infinity Yoga Tutor: Yoga Posture Detection and Correction System, IEEE
13. Next generation pose detection with MoveNet