

# Fitness Freaks: A System for Detecting Definite Body Posture Using OpenPose Estimation



Harshwardhan Pardeshi, Aishwarya Ghaiwat, Ankeet Thongire,  
Kiran Gawande, and Meghana Naik

**Abstract** Nowadays, because of busy schedules, people have no time to go to the gym, and even if they manage to find a gym nearby having a gym trainer besides all the time to correct postures while doing exercise is impossible unless people do not opt for a personal trainer. Even if they assist a personal trainer for them, they would have to adjust their time accordingly, and both of these methods are quite costly, and not everyone can afford it; also, during this global pandemic, people are stuck at home and have no access to go to the gym or can't even take a risk of getting in contact with a personal trainer. Performing exercises let it be exercise or doing Yoga proper body posture is important; if not performed properly, it can lead to crucial problems such as poor joint alignment, increased shear forces on the spine, compression of disks and joints, less space for nerves to course through the body due to compression, and reduced blood flow to prevent such injuries and pains, and to track gym exercises repetitions, we came up with a system called Fitness Freaks. Fitness Freaks is an AI fitness tracker. It tracks users' body movements using human pose estimation. This in turn keeps track of repetitions of gym exercises and detection of wrong body posture while doing Yoga.

**Keywords** Computer vision · Fitness · Gym exercises · OpenPose · Posture · Yoga

---

H. Pardeshi (✉) · A. Ghaiwat · A. Thongire · K. Gawande · M. Naik  
Computer Engineering Department, Sardar Patel Institute of Technology, Andheri (West),  
Mumbai 400059, India  
e-mail: [harshwardhan.pardeshi@spit.ac.in](mailto:harshwardhan.pardeshi@spit.ac.in)

A. Ghaiwat  
e-mail: [aishwarya.ghaiwat@spit.ac.in](mailto:aishwarya.ghaiwat@spit.ac.in)

A. Thongire  
e-mail: [ankeet.thongire@spit.ac.in](mailto:ankeet.thongire@spit.ac.in)

K. Gawande  
e-mail: [kiran\\_gawande@spit.ac.in](mailto:kiran_gawande@spit.ac.in)

M. Naik  
e-mail: [meghana\\_naik@spit.ac.in](mailto:meghana_naik@spit.ac.in)

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2022  
P. K. Singh et al. (eds.), *Futuristic Trends in Networks and Computing Technologies*,  
Lecture Notes in Electrical Engineering 936,  
[https://doi.org/10.1007/978-981-19-5037-7\\_76](https://doi.org/10.1007/978-981-19-5037-7_76)

1061

## 1 Introduction

Development in the field of fitness and well-being of health has grown exponentially in the last decade which includes fit-bands, calorie counter, diet planner, and run tracker. Supporting the advancement in this field, we focused on the problem of getting assistance while doing exercises and focusing on the prevention of injuries. A system to track count of user performed exercises and detect errors in a Yoga pose, with the use of computer vision in which we use OpenCV Python library, OpenPose which uses baseline CNN network, and COCO dataset. We defined rules for different Yoga poses, exercises, and accordingly, we get the results for the count of repetitions of the user-performed exercises and detect error for the Yoga poses; if there is any, feedback is given to users based on the results observed. For this, the user needs to run the system on their laptop, no fitness band or other extra weight is to be carried with the user while performing it. After analyzing the user's exercise, feedback about the exercise is given to the user using a voice assistant. It helps the user to keep track of repetitions and focus more on correcting the body posture while doing exercises.

## 2 Literature Survey

Xiong et al. [1] in 2020 worked on robust vision-based workout analysis. Test results show the prevalence of their proposed 3D posture assessment over the past ones. It identifies incorrect motions but not along with the timing of these motions and hence does not provide timely feedback to users. They could not integrate their model with video tutorials.

Yadav et al. [2] in 2019 approached to accurately recognize various Yoga poses using deep learning algorithms. A dataset of 6 Yoga asanas had been created using 15 individuals. A hybrid deep learning model was proposed using CNN and LSTM for Yoga recognition on real-time videos; the system can be implemented on a portable device for real-time predictions and self-training.

Gu et al. [3] in 2019 adopted deep learning models for human pose estimation and worked on home-based physical therapy with an interactive computer vision system. They could not provide users a side-view option and could not develop an algorithm that gives more detailed feedback on how the patient is doing, i.e., instead of giving feedback based on the overall performance.

Chen et al. [4] in 2018 incorporated computer vision strategies and proposed system that examines the practitioner's stance from both front and side perspectives by separating the body shape, skeleton, dominant axes, and points. Improving or even redesigning the methods of feature point detection and assistant axis generation for some poses can make the system more solid.

Chen and Yang [5] created this application for correcting users' pose by creating ideal movements of exercise. They used deep neural networks and OpenPose for pose estimation and machine learning and heuristic-based models for getting the result of

performance by comparison. The application created works only on the Web which runs on Windows and Linux computers based on GPU.

In this paper, Keshari [6] has used opencv for image processing, SVM and RCNN for detecting the errors. They have created their own dataset for detecting the errors; they distributed the dataset to understand the proper posture and remaining for testing using SVM and RCNN to detect wrong posture.

In this paper, Nagarkoti et al. [7] use a pre-recorded trainer's video using deep learning and opencv. For tracking users body movements, optical flow tracking is used; dynamic time warping is used to sync trainers and users body movements. It only corrects user posture; proper AI assistant for the user should include tracking exercises repetitions, errors, and creating reports.

Agrawal et al. [8] used various classification techniques to detect Yoga poses out of which random forest classifiers gave the best results. It detects different Yoga poses using this application along with identification of the pose; if the accuracy is calculated, then the user will be able to track and improve performance.

In Cao [9] did pose estimation for single person and multi-person. In single person, they perform inference over a combination of local observations on body parts and the spatial dependencies between them. For multi-person, they have used a top-down strategy to first detect people and then estimate the pose of each person independently. It works only on an image and cannot be used on a video.

Kumar et al. in [10] proposed to use OpenPose on the client's ongoing and recorded sessions to distinguish the joint areas by utilizing Part Confidence Maps and Part Affinity Fields. Then, based on the difference in angles, feedback is provided to the user. This works only when images are provided by the user; it does not work in real-time.

In [11] Chiddarwar et al. collect a single ideal image for getting the key points and store these locally in their machine. Then, OpenCV is used to predict the 17 essential keypoints using a pretrained model; the distance between each body part is calculated, using Euclidean distance. The system only detects the Yoga pose and gives the correctness of the pose.

In [12] Dang et al. did a survey, on human pose estimation methods. The single-person estimation is classified into two types: regression-based approach and heatmap-based approach. Multi-person estimation is classified into two categories: top-down approach and bottom-up approach. Now-a-days, the human pose estimation methods are improved significantly and can still be improved to use them for real-world applications. The speed of algorithms is still slow for real-time prediction.

In [13] Sajjad et al. compared different techniques for human pose recognition to identify which is better by calculating accuracy for each and getting better results so that the implementation of the human pose recognition should be correct.

### 3 Proposed Methodology

After identifying the problem statement and the gaps from the research paper, we proposed a solution to build a tech-based personal trainer which will guide and track users' exercises 24/7.

**Fitness Tracker:** The fitness tracker will help maintain users' body posture while doing gym exercises and Yoga poses.

**Gym Exercises:** The system counts the repetitions of the exercises performed so that the user does not have to keep track of the exercises.

**Yoga poses:** The system tells the accuracy of Yoga poses performed and detects an error in users' posture.

We got assistance from 12 professional trainers and Yoga instructors who could help us build a robust system. We asked them to identify all the factors that needed to be considered while doing the 2 simple exercises, one a gym exercise and another a Yoga pose. In gym exercise, we have built a system for bicep curls, whereas in Yoga pose built a system for Warrior Pose. Based on the guidance, we created 10 rules for Warrior Pose as the total number of factors involved for performing that pose were 10 and similarly, for bicep curls, only 2 rules; based on those 2 rules, the user's repetitions are going to be counted. If the user does not satisfy those rules, the repetitions won't be counted.

#### 3.1 Requirements

The system is a Web site-based application that runs on a laptop. The user needs enough space to perform the exercise and place the camera so that the whole body of the user is fitted in the camera angle.

#### 3.2 Assembling Data

For both gym and Yoga poses, we have taken a total of more than 150 images of professional trainers and Yoga instructors to build our system that can identify ideal values of the factors involved in the exercises. OpenCV is a Python library. It is used in the system to focus on capturing the video and detecting the face. It is also used for image processing. OpenCV uses the COCO dataset which is the largest dataset for 2D pose estimation. It has around 1.5 million object instances, 80 categories of images consisting of objects, and around 250,000 instances of people. The COCO dataset is also considered a large-scale universal dataset for a lot of tasks related to computer vision.

### 3.3 *Data Preprocessing*

In this computer vision library, OpenCV is used where we use OpenPose for human pose estimation. Human pose estimation tracks and recognizes different main points of posture in different individuals. We use the videos of instructors performing the exercises. First, it analyzes the picture from the user's video, and it acts as an input image. Initially, to extract feature maps of the input, the picture is passed through baseline CNN. It utilizes the VGG-19 network's first 10 layers. To produce the Part Confidence Maps and Part Affinity Field, the feature map is then processed in multistage CNN.

### 3.4 *Algorithm*

A greedy bipartite matching algorithm is used to process the Confidence Maps and Part Affinity Fields that are created above and are used to get the postures for every individual in the picture.

**Certainty Maps:** A Confidence Map is a 2D portrayal of the conviction that a specific body part can be situated in some random pixel.

**Part Affinity Fields:** Part Affinity is a bunch of 2D vector handles that encode the location and direction of body portions of various individuals in the picture. It encodes information as pairwise associations between body parts.

**Multi-Stage CNN:** The above multi-CNN engineering has three significant stages: The original set of stages anticipated the Part Affinity Fields refines Lt from the element guides of base organization.

The second arrangement of stages takes utilizes the yield Part Affinity Fields from the past layers to refine the forecast of certainty maps identification.

The last certainty guides and Part Affinity Field are then passed into the covetous calculation for an additional cycle.

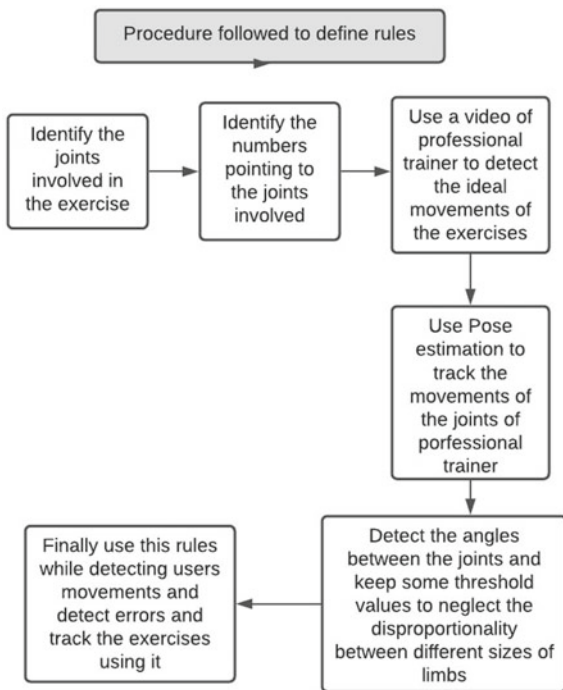
## 4 **Implementation**

The system first takes users video as an input; the user is instructed with a demo video showing how to perform the exercise; user can watch the video and perform the exercise, and from that, we find the 10 stillest points and take the mean of those points using human pose estimation.

More than 150 images are used in training our system; furthermore, 30 user's videos of performing the exercise are taken as testing data for the system.

Different rules are being written for exercise and Yoga to define the ideal position of that specific pose using OpenPose. A specific procedure is followed while defining the rules in Fig. 1. First of all, the joints involved in the exercise are identified. Then,

**Fig. 1** Procedure followed to define rules

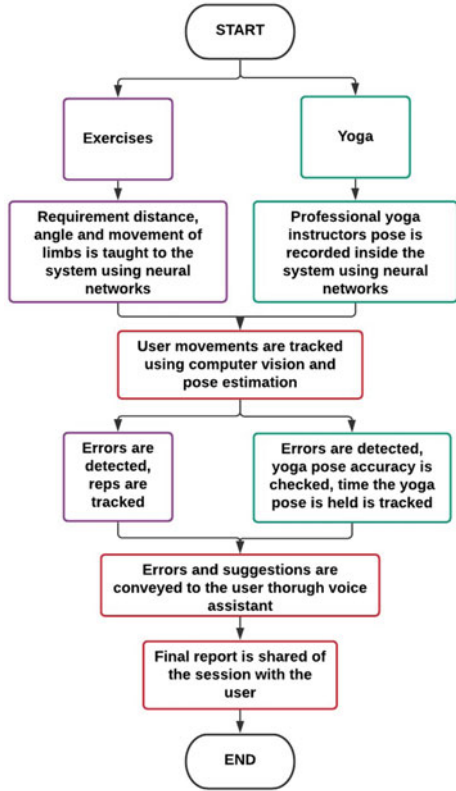


the number pointing to the joints is identified with the help of the COCO human pose estimation model. Then, a video of a professional trainer is used to detect the ideal movements of the exercise and Yoga. Further, pose estimation is used to track the movements of the joints of the professional trainer. The angles between the joints are detected, and some threshold values are kept to neglect the disproportionality caused due to various body types and size depending upon the gender and age.

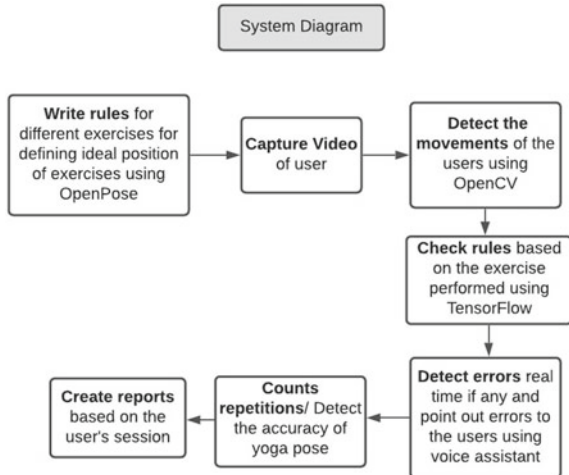
In Fig. 2 there is exercise as well as Yoga detection under one roof. For bicep curl exercise, required distance, angle, and movement of limbs are taught to the system using trainer’s videos, and for warrior II Yoga pose, professional Yoga instructor pose is recorded inside the system. Once the rules are written, (Fig. 3) captured video of the user is passed into the system, and the moments are detected using computer vision, i.e., pose estimation by the means of OpenCV. Rules are being checked based on the exercise performed using TensorFlow. Errors are detected, and repetitions are tracked for the bicep curl exercise. For the Warrior II Yoga pose, errors are detected in real time, i.e., pointed out and real-time feedback is given by a voice assistant to correct the Yoga pose. A feedback report is generated based on the user performance. In this way, using the defined rules, user movements and errors are detected, and track of the exercise and Yoga is kept.

We use OpenPose in our project and write rules needed for calculating the angles, slopes, min, and max ratios. If the conditions are not matching, i.e., if the user is not performing Yoga properly then feedback is given to them on the basis of rules

**Fig. 2** Flowchart of proposed system



**Fig. 3** System diagram



defined. Count of exercise is tracked if conditions are satisfied. The system will provide instructions on how to perform the pose. Once the user is ready, he/she records a video of themselves doing the pose (using a Webcam). Using OpenPose, the video input is processed and an open source; deep learning-based library is used for key-point detection. The app uses the key points obtained from the video to classify different errors that occurred while doing the pose with the help of a rule-based system. Repetitions of exercise are tracked, and feedback is provided to the user so that they can safely improve their pose.

To check the accuracy, we have established 10 rules. Table 1 describes the evaluation factors considered for accuracy calculation. We derived a formula to get the final accuracy. For 10 rules, we have 10 variables for users' pose and 10 for the correct Yoga posture which we get by examining the pose of professional Yoga instructors. We derive the formula as

$$\text{Error} = \frac{|\text{Users' value} - \text{Ideal Value}|}{\max(\text{User's value}, \text{Ideal Value})} \tag{1}$$

The individual errors of each rule are recorded using the above formula; we assume the weight for each rule is the same for detecting error, and the mean of all the errors is taken to denote the final error, and accordingly, we get the accuracy of the pose. The 10 rules together check if the warrior pose performed by the user is correct or not, based on which we derived a formula for checking accuracy.

$$\text{Error} = \frac{\sum_{i=1}^{10} \frac{|x_{ui} - x_{pi}|}{\max(x_{ui}, x_{pi})}}{10} \tag{2}$$

where,  $i$ , denotes the rule number among the 10 rules for warrior pose  
 $x_{ui}$ , denotes value of returned by the functions of  $i$ th rule for users pose  
 $x_{pi}$ , denotes value of returned by the functions of  $i$ th rule for professional Yoga instructors pose.

**Table 1** Evaluation factors considered for accuracy calculation

| Rule number | Evaluation factors considered                                  |
|-------------|--|
| 1           | Slope of the arms  |
| 2           | Angle between the hip, knee, and ankle                         |
| 3           | Angle between the hip, knee, and ankle                         |
| 4           | Ratio of distance between eyes to distance between ears        |
| 5           | Slope of left hip and right hip                                |
| 6           | Ratio of arm distance to legs distance                         |
| 7           | Slope between left shoulder to neck and neck to right shoulder |
| 8           | Slope of neck and mid-hip                                      |
| 9           | Slope of neck and mid-hip                                      |
| 10          | Ratio of arm distance to legs distance                         |



**Table 2** Evaluation factors considered for accuracy calculation of exercise: Bicep curl

| Rule number | Evaluation factors considered            |
|-------------|--|
| 1           | Angle between wrist, elbow, and shoulder |

The rules are:

1. Keeping the arms straight and palms faced down
2. Keeping front leg straight perpendicular to the floor
3. Ensuring the front knee doesn't extend beyond the ankle and is inline with the heel
4. Keeping face inline with front hand
5. Distance between the feet is wide enough, so the legs get stretched
6. Keeping shoulders down, stretched out and not lifted toward ears
7. Keep hips and shoulders faced sideways toward camera
8. Keep hips and shoulders in same line so that the rib cage isn't floating forward
9. Keep hips and shoulders in same line so that the rib cage isn't floating backward
10. Place back leg straight and strong and keep short distance between the legs.

Table 2 depicts the evaluation factors considered for the accurate calculation of bicep curl. For bicep curl detection, we have set a threshold range of angle between wrist, elbow, and shoulder depending upon the videos of the trainer and a counter for counting the number of repetitions of the exercise performed.

## 5 Results

To explain the results of the system built, we take an example of some users performing Warrior Pose II. The values of different factors which are mentioned in Table 1 for the users performance are shown in Table 3. From the user's value and ideal value, we calculate the error and accuracy of the user's Yoga pose. Each rule has functions written to measure the values such as the slope of arms, angle between hip, knee, and ankle as we get coordinates of all the joints from which we can define functions and calculate different angles, slopes, and ratios to built the system.

$$\text{Error} = \frac{1.3466}{10} = 0.13466 \tag{3}$$

$$\text{Accuracy} = 100 - \text{Error} * 100 = 86.534\% \tag{4}$$

The accuracy of the example taken of user's pose is 86.534%.

**Table 3** Accuracy calculation of Yoga: Warrior Pose II

| Rule number | Evaluation factors considered                                  | User value | Ideal value |
|-------------|--|------------|-------------|
| 1           | Slope of the arms  | 0.267      | 0           |
| 2           | Angle between the hip, knee and ankle                          | 101        | 90          |
| 3           | Angle between the hip, knee and ankle                          | 101        | 90          |
| 4           | Ratio of distance between eyes to distance between ears        | 0.67       | 0.5         |
| 5           | Slope of left hip and right hip                                | -0.12      | 0           |
| 6           | Ratio of arm distance to legs distance                         | 0.692      | 0.75        |
| 7           | Slope between left shoulder to neck and neck to right shoulder | 6.52       | 4.25        |
| 8           | Reverse slope of neck and mid-hip                              | 0.012      | 0           |
| 9           | Reverse slope of neck and mid-hip                              | 0.012      | 0           |
| 10          | Ratio of arm distance to legs distance                         | 0.692      | 0.75        |

**Table 4** Accuracy calculation of exercise: bicep curls

| Rule number | Evaluation factors considered            | User value  | Ideal value                        |
|-------------|--|-------------|------------------------------------|
| 1           | Angle between wrist, elbow, and shoulder | 216 and 289 | Less than 230 and greater than 280 |

For Warrior II Yoga pose, we have set 10 rules and some ideal values and based on that system provides feedback to the user if the user performs Yoga properly, no correction will be shown to user, and user can continue performing the Yoga without having any concern about getting muscle strain, joint pain, or any sort of injuries which is the main problem our system focuses on. There may be no errors in the user's posture, but the accuracy is not 100% because accuracy is calculated by comparing the user's pose with the ideal pose. Accuracy will increase when the user repetitively practices Yoga and increase flexibility.

For bicep curl detection, Table 4 shows the ideal values set for a threshold range of angle between wrist, elbow, and shoulder depending upon the videos of the trainer and a counter for counting the number of repetitions of the exercise performed.

Figure 4 shows the Yoga: Warrior II Pose. On the left side of the system, the video of the professional trainer is shown so that the user can refer to it while performing the Yoga pose. On the right side of the system, the user pose evaluation is done and according to the proposed rules and feedback is given.

Figure 5 shows the exercise: bicep curl. On the left side of the system, the video of the professional trainer is shown so that the user can refer to it while performing the Yoga pose. On the right side of the system, the user pose evaluation is done and according to the proposed rules and no of reparations are counted.

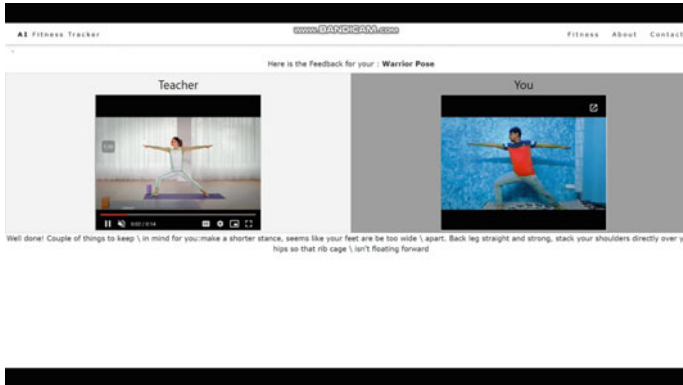


Fig. 4 Yoga: warrior II

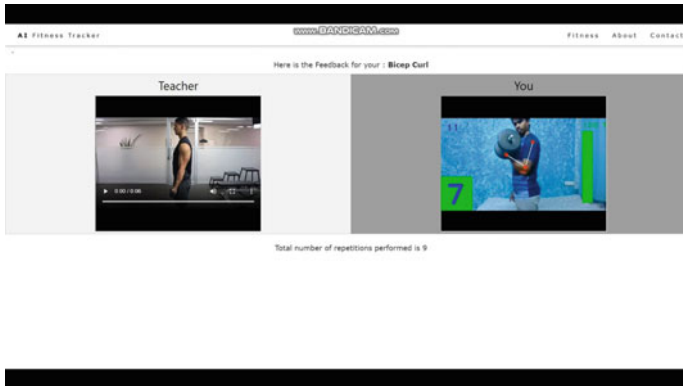


Fig. 5 Exercise: bicep curl

## 6 Conclusion

The system will combine fitness and technology to successfully bring advancement in tracking and acting as a personal trainer to the users without actually requiring the help of any actual trainer. It brings full-fledged flexibility to the user to perform exercise anytime throughout the day as per their availability of free time. The constant need for the attention of personal trainers in gyms and Yoga classes while performing the exercise is drawn out by AI fitness tracker. Constant monitoring of users' body movement and joints helps correct users' body posture which is one of the most fundamental parts while doing exercises. The existing methodology primarily focuses on the time for which user exercises and not on the user's correct posture, instead focuses on the time for which exercise is performed. The system has found promising results after testing on more than 50 different users. The ideal values are calculated by taking the mean of values of slopes, ratio, and angles from a dataset of more than 100 trainers performing Warrior Pose II.

## 7 Future Scope

A system like these can replace personal trainers for correcting body posture and the need for their constant attention while performing exercises or Yoga poses. The system can be featured by adding more exercises and Yoga poses to the system. We can create a user portal that keeps a track and record of the data of their previous exercises and Yoga performed in the system's database. Variation can be brought by adding sections for men, women, and children depending on their age group and segregating different sections based on the difficulties of the exercises. The system with all these features can be added as a daily routine to prevent injuries and muscle strains while performing exercises also track the counts of all exercises, and keep a report of the users' performance.

## References

1. Xiong H, Berkovsky S, Sharan RV, Liu S, Coiera E (2020) Robust vision-based workout analysis using diversified deep latent variable model. In: 2020 42nd annual international conference of the IEEE engineering in medicine biology society (EMBC), pp 2155–2158
2. Yadav S, Singh A, Gupta A, Raheja J (2019) Real-time yoga recognition using deep learning. *Neural computing and applications*, vol 31, <https://link.springer.com/article/10.1007/s00521-019>
3. Gu Y, Pandit S, Saraee E, Nordahl T, Ellis T, Betke M (2019) Home-based physical therapy with an interactive computer vision system. In: 2019 IEEE/CVF international conference on computer vision workshop (ICCVW), pp 2619–2628
4. Huang C, He Y-Z, Hsu C-C (2018) Computer-assisted yoga training system. *Multimed Tools Appl* 77:09
5. Chen S, Yang R (2018) Pose trainer: correcting exercise posture using pose estimation
6. Keshari P (2020) Wrong posture detection using opencv and support vector machine
7. Nagarkoti A, Teotia R, Mahale AK, Das PK (2019) Realtime indoor workout analysis using machine learning amp; computer vision. In: 2019 41st annual international conference of the IEEE engineering in medicine and biology society (EMBC), pp 1440–1443
8. Agrawal Y, Shah Y, Sharma A (2020) Implementation of machine learning technique for identification of yoga poses. In: 2020 IEEE 9th international conference on communication systems and network technologies (CSNT), pp 40–43
9. Cao Z, Hidalgo G, Simon T, Wei S, Sheikh Y (2021) OpenPose: realtime multi-person 2d pose estimation using part affinity fields. *IEEE Trans Pattern Anal Mach Intel* 43:172–186 Jan
10. Kumar D, Sinha A (2020) Yoga pose detection and classification using deep learning. *Int J Sci Res Comput Sci Eng Inf Technol*, 11
11. Chiddarwar G, Ranjane A, Chindhe M, Deodhar R, Gangamwar P (2020) Ai-based yoga pose estimation for android application. *Int J Innov Sci Res Technol* 5:1070–1073
12. Dang Q, Yin J, Wang B, Zheng W (2019) Deep learning based 2d human pose estimation: a survey. *Tsinghua Sci Technol* 24(6):663–676
13. Sajjad F, Ahmed AF, Ahmed MA (2017) A study on the learning based human pose recognition. In: 2017 9th IEEE-GCC conference and exhibition (GCCCE), pp 1–8