

# AI-based Workout Assistant and Fitness guide

Gourangi Taware<sup>[1]</sup>, Rohit Agarwal<sup>[2]</sup>, Pratik Dhende<sup>[3]</sup>,  
Prathamesh Jondhalekar<sup>[4]</sup>, Prof. Shailesh Hule<sup>[5]</sup>

<sup>1,2,3,4,5</sup> Computer Department, Pimpri Chinchwad College of Engineering,  
Pune, India

**Abstract:** Nowadays virtual assistant is playing a very important role in our daily activities and has become an inseparable part of our lives. As per the Clutch survey report that was published in 2019, almost 27% of people are using AI virtual assistants for performing their day-to-day activities. AI is an emerging field that we aim to explore through this project of AI-based workout assistants. In our work, we introduce Fitcercise, an application that detects the user's exercise pose counts the specified exercise repetitions and provides personalized, detailed recommendations on how the user can improve their form. The application uses the MediaPipe to detect a person's pose, and afterwards analyses the geometry of the pose from the dataset and real-time video and counts the repetitions of the particular exercise.

**Keywords:** AI, Virtual assistant, CNN, workout assistant, Pose estimation, BlazePose, OpenCV.

## 1. INTRODUCTION:

In our work, we introduce Fitcercise, an application that detects the user's exercise pose counts the specified exercise repetitions and provides personalized, detailed analysis about improving the user's body posture. This is an AI-based Workout Assistant and Fitness guide to guide people who don't have access to the gym but are still willing to work out at home to maintain their physique and fitness and keep their body in good shape. To help them perform the exercises correctly and prevent them from chronic and immediate injuries. This also provides a personalised health guide and diet plan along with a personalised daily workout calorie count. The application also displays necessary health insurances and policies provided by the government of India for the common people and check the eligibility criteria using API and Web services. Staying at home for long periods of time can become boring, especially when most fun activities are done outdoors, which is difficult considering the current scenario of pandemics and lockdown. But this cannot be a relevant excuse for being unproductive because it is an excellent idea to utilize the extra time we get into our own health.

Most gyms have a wide variety of exercise equipment and also have trainers who guide us about the exercise and its correct posture. But the unavailability of the above equipment and trainers can be an important reason that can stop us from doing exercise at home. We aim to build an AI-based trainer that would help you exercise more efficiently in your own homes. The project focuses on creating an AI algorithm to help you exercise, by determining the quality and quantity of repetitions which is done by using pose estimation running on the CPU.

This project, which will have a non-distractive interface, intends to make exercising more easy and more fun. We are going to see an overview of the contribution of these families, their algorithms, advantages, disadvantages, its efficiency compared to other existing technologies, applications and possible future work.

## 2. LITERATURE REVIEW:

There are numerous applications available in the market which guide the user about the exercises to be performed. But through our application, we not only guide the user regarding which exercise to perform but also about the correct posture and counting the repetitions using computer vision. This application can be considered as the workout assistant which provides real-time posture detection and diet recommendations. The application can not only be used by individuals at homes but by increasing the scope can be used in gyms as smart trainers thus reducing the human intervention.

[1] Their objective was to provide a bottom-up approach for the activity of estimation of the pose of the user and real-time segmentation of the user while using images of the multi-person solution and by implementing an effective single-shot approach.

So the idea they proposed used a CNN that is a convolutional neural network by training it to detect and classify the key points and accordingly give accurate results by studying the relative displacements and thus by clustering or identifying the group of different key points and studying the pose instances.

Using single-scale inference, the model obtained a COCO[6] accuracy of the points of 0.665 and 0.687 using multiple level inference. Using part-based modelling. It depends on the key point level structure in order for training the real-time segmentation activity. In the future, there might be ways to overcome this limitation.

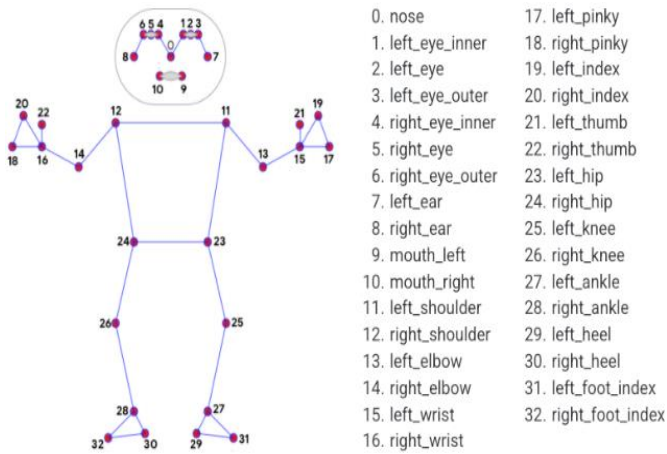


Fig 1: 33 Body Key Points[6]

In research paper[2] Their objective was to create BlazePose, a mobile-optimized lightweight convolutional neural network architecture for human posture prediction. On a Pixel 2 phone, the network creates 33 body key points(as shown in Fig 1) for a single individual during inference and operates at over 30 frames per second. This makes it ideal for real-time applications such as fitness tracking and sign language recognition. Two of our most significant contributions are a novel body posture monitoring method and a lightweight body pose prediction neural network. Both approaches use heatmaps and regression to find the points. They built a robust technique to estimate the posture using BlazePose, which uses CNN and a dataset with up to 25K photos demonstrating distinct body endpoints, enhancing the accuracy. On a mobile CPU, this model runs in near real-time, and on a mobile GPU, it can run in super real-time.

The given algorithm of 33 keypoint topology is efficient with BlazeFace and BlazePalm. In this paper, the authors have developed a system for majorly upper body key points. A solution that shows lower-body analysis of pose will also be integrated.

In the research paper[3], the researchers proposed an efficient solution mainly to tackle the multi-person problem while detecting poses when there are multiple people in the real-time frame. In this approach, the model is trained in such a way that it detects the points of the user and then segregates based on the affinity of different points in the frame. This is considered as the bottom-up approach and is very efficient in accuracy and performance-wise without considering the number of people in the frame as the barrier. For the dataset considering 288 frame images, this approach performs well as compared to the other approaches discussed above by 8.5% mAP. The approach is able to get higher accuracy and precision in real-time. The earlier solutions were redefined in the training stages. The disadvantage of OpenPose is it doesn't return any data about the depth. and also needs high computing power.

In the research paper[4] they aimed to get the precise location of the points by using a deep neural network. In this approach, they presented DNN-based estimators. This allowed an

efficiency in the precision to predict pose. The total efficiency increases by using this approach.

### 3. DATASET USED:

As most of the solutions use the key points and the heatmaps, first we require to pose alignment data for each pose. We can take into consideration the different test cases where if the complete body is visible and there are detectable key points for the body parts. To make sure that the pose detector can perform in heavy occlusions which are some different test cases than normal ones, we can make use of occlusion-simulating augmentation. The training data set has 60000 images with a few images doing the same pose that have different key points and 25000 frames in which the user performs the actual exercise.

### 4. IMPLEMENTATION AND ALGORITHM:

We have used JavaScript node JS and different libraries such as Open CV and MediaPipe which is a library using ML algorithms along with different numerical and algorithms.

The MediaPipe pose estimation tool uses a 33 key points approach wherein it detects the key points and accordingly uses and studying the data set estimates the pose. It tracks The pose from the real-time camera frame or RGB video by using the blaze pose tool that has a Machine Learning approach in pose detection.

This approach uses a double step tracker machine learning pipeline which is efficient in media pipe solutions. Using the tracker locates the region of interest of the activity or posture in the real-time video. It then predicts the key points in the region of interest using the real-time video frame as an input. But the point to be noted is that the tracker is invoked only during the start or whenever the model is unable to detect the body key points in the frame.

We have created a module named PoseModule.js and defined various functions in it and imported this module to our main project file aiTrainer.js to utilize these functions.

We are basically first detecting the landmark positions on the body in the video with the help of MediaPipe[9]. Then the angle between the points is calculated and a range is determined. This range can be demonstrated by a 0-100 % efficiency bar on the output video frame. We also calculate the number of repetitions of the exercise and display the count in the output video.

Formula for calculating angle formed by 3 points:

$$\text{Angle} = \text{math.degrees}(\text{math.atan2}(y3-y2, x3-x2) - \text{math.atan2}(y1-y2, x1-x2))$$

In the output following data is displayed: fps rate, counter for repetitions, landmark points, the angle between landmark points and status bar.

This project can be implemented on pre-recorded videos as well as in real-time through a webcam.

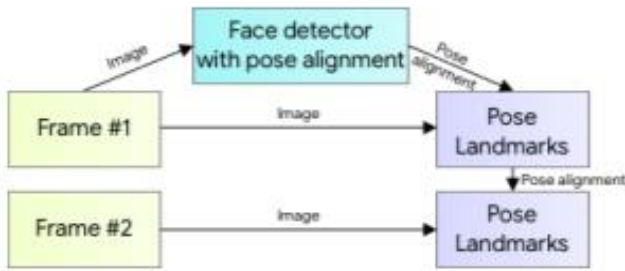


Fig 2: Inference Pipeline

**NEURAL NETWORK ARCHITECTURE:**

The estimator in our application first estimates the position of the 33 key points of the user and later utilises the user alignment. We utilise the combination of heatmap and the regression way. In the training model, we utilise the above approaches and then prune the resultant layers from the test model. We used the heatmap to analyse the light-weight integration and used it by the encoder. The solution is inspired by the Stacked Hourglass solution[12]. We used skip-connections in all levels in order to get a balance in higher and lower characteristics. The slopes or the gradients were not going back to the heatmap in the train set model. For their last post-processing stage, the bulk of current object identification methods use the Non-Maximum Suppression (NMS) algorithm. For hard objects with minimal degrees of freedom, this method works effectively. This algorithm, however, fails in cases that feature highly articulated human postures, such as individuals waving or hugging. This is because the NMS algorithm's intersection over union (IoU) criterion is satisfied by several, confusing boxes. Refer to Fig 3 for the System Implementation plan

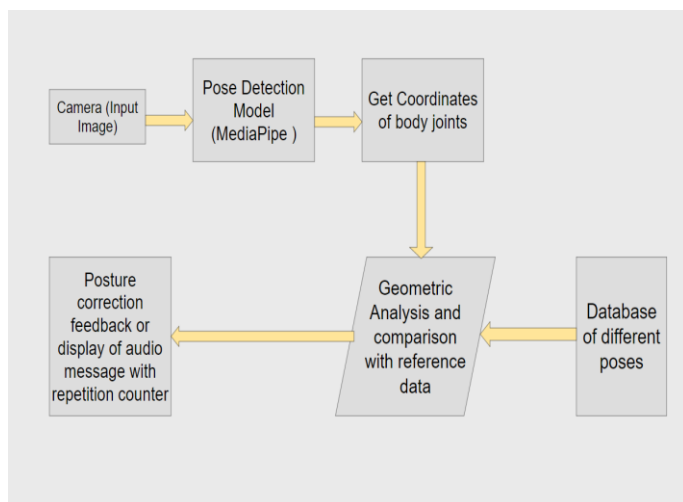


Fig 3: System Implementation plan

**MATHEMATICAL MODELING OF CNN MODEL:  
 KERNEL CONVOLUTIONAL**

It's a method in which we take a small matrix of numbers (known as a kernel or filter), apply it to our image, and then transform it using the values from the filter. The feature map values are determined using the formula below, where f

denotes the input picture and byh denotes our kernel. The indexes of the result matrix's rows and columns are denoted by mand n, respectively.

$$G[m, n] = (f * h)[m, n] = \sum_j \sum_k h[j, k]f[m - j, n - k]$$

Relu:

$$\text{ReLU} = \begin{cases} 0, & \text{if } x < 0, \\ x, & \text{if } x \geq 0. \end{cases}$$

$$O_i = \frac{e^{z_i}}{\sum_{i=1}^M e^{z_i}}$$

Padding:

$$O_i = \frac{e^{z_i}}{\sum_{i=1}^M e^{z_i}}$$

Output Matrix Dimension:

$$n_{out} = \left\lfloor \frac{n_{in} + 2p - f}{s} + 1 \right\rfloor$$

Resultant Tensor After Multiple Filters:

$$[n, n, n_c] * [f, f, n_c] = \left[ \left\lfloor \frac{n + 2p - f}{s} + 1 \right\rfloor, \left\lfloor \frac{n + 2p - f}{s} + 1 \right\rfloor, n_f \right]$$

Activation:

$$\mathbf{Z}^{[l]} = \mathbf{W}^{[l]} \cdot \mathbf{A}^{[l-1]} + \mathbf{b}^{[l]} \quad \mathbf{A}^{[l]} = g^{[l]}(\mathbf{Z}^{[l]})$$

Backpropagation:

$$d\mathbf{A}^{[l]} = \frac{\partial L}{\partial \mathbf{A}^{[l]}} \quad d\mathbf{Z}^{[l]} = \frac{\partial L}{\partial \mathbf{Z}^{[l]}} \quad d\mathbf{W}^{[l]} = \frac{\partial L}{\partial \mathbf{W}^{[l]}} \quad d\mathbf{b}^{[l]} = \frac{\partial L}{\partial \mathbf{b}^{[l]}}$$

$$d\mathbf{Z}^{[l]} = d\mathbf{A}^{[l]} * g'(\mathbf{Z}^{[l]})$$

$$d\mathbf{A}^{[l]} = \sum_{m=0}^{n_h} \sum_{n=0}^{n_w} \mathbf{W} \cdot d\mathbf{Z}[m, n]$$

**5. BLOCK DIAGRAM:**

1. User Login: The user has to enter valid credentials and login into the system and save the personal data of the user into the respective account.
2. Exercise Routines: The application contains different exercise routines which have different exercises that the user can do in real-time and has different pose correction and set repetition counter tools.
3. Repetition counter: It counts the set of repetitions the user does of a particular exercise in real-time by identifying the position of the user.

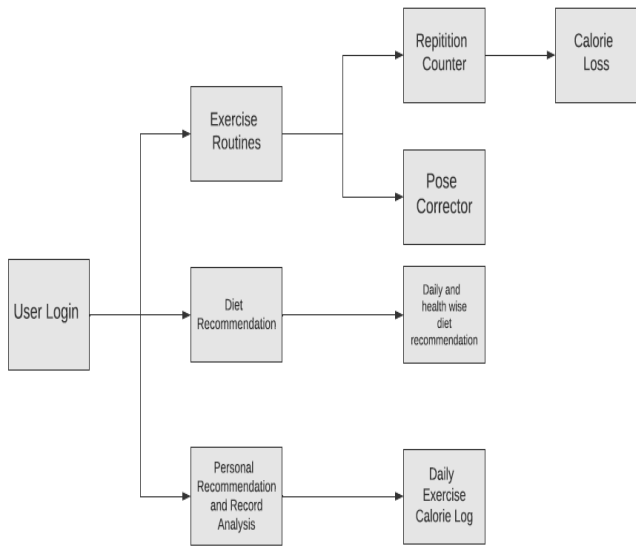


Fig 4: Block Diagram of the system

4. Pose corrector: it helps the user to detect and correct the poses or the exercise posture of the user in real-time by using different pose detecting algorithms and computer vision techniques.
5. Diet Recommendation: The system prepares a diet plan for the user depending upon the health Issues and the area of interest.
6. Personal Recommendation and record log: The system monitors the daily calorie loss and the exercise count repetitions of the user and does the research of the data to give relevant reports to the person and thereby increase the precision in the recommendations.

6. FUNCTIONAL REQUIREMENTS:

1. Pose estimator has the ability to detect the pose and count the repetitions along with the posture guide
2. Personalised calorie counter depending upon the exercises performed.
3. Diet planners exhibit different diets depending upon the health conditions and calorie intake.
4. A platform to display different health insurances and policies provided by the Indian government along with the benefits and eligibility criteria.
5. Display different exercise routines according to the health conditions and focus majorly on being fit and weight loss.

7. SYSTEM ARCHITECTURE:

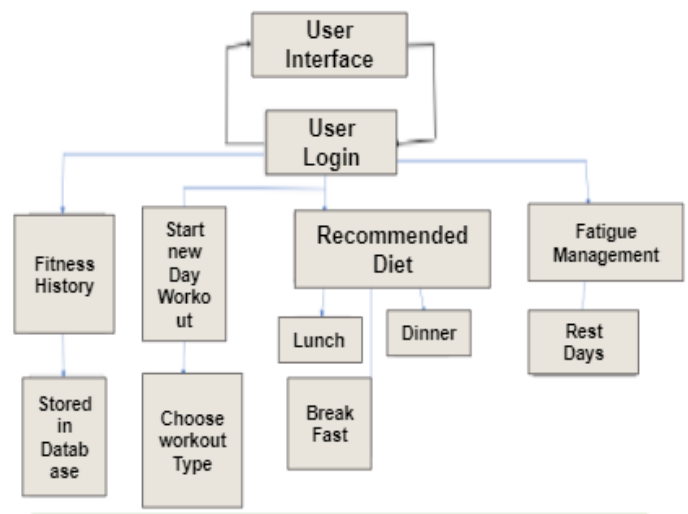


Fig 5: System Architecture

8. ADVANTAGES:

1. There are numerous applications available in the market which guide the user about the exercises to be performed. But through this application, we not only guide the user regarding which exercise to perform but also about the correct posture and counting the repetitions using computer vision.
2. Monitor the user in real-time keeping track of the quality repetitions of a particular exercise, thus keeping his form intact and correct throughout their workout. This will educate newbies about different exercise routines and their correct postures to prevent injuries.
3. The application also offers personalised health advice and nutrition ideas while keeping the daily calorie log in the database.
4. The application can not only be used by individuals at homes but by increasing the scope can be used in gyms as smart trainers thus reducing the human intervention.
5. Our main motive is to spread awareness about the importance of good health and fitness among common people.

9. LIMITATIONS:

1. The application can estimate the poses and count repetitions for a limited number of exercises as pose estimation using computer vision for some exercises and postures can be difficult.
2. The application is developed as a cross-web application and is not used as a mobile android/ios application.
3. The application cannot capture multiple people in the frame in the real-time system.

## 10. APPLICATION:

The application can be used indoors at home or in the gyms to get pose detection and correction suggestions. It can also be used to keep the daily log of calories of each user and suggest changes and exercises accordingly.

Apart from this, the application can be used to spread awareness about the different health-related government schemes and different health insurance-related information.

## 11. CONCLUSION AND FUTURE WORK:

Nowadays our life is becoming busier and we hardly find time in our schedules to be healthy and fit and exercise daily. This has caused many diseases and health issues. Implementation of Artificial Intelligence in the field of fitness can solve many problems. The health-related applications and devices are making our lives easier and eases our fitness journey. Individuals can use this application in their own workouts, hence making them more efficient are less error-prone. In this process, we learnt how to use the OpenCV library and package and how the application of machine learning can be beneficial to humans.

There is a lot of scope of development in this project. The project can be upgraded to support more exercises. A User interface can be added for easy navigation through the exercises. The data collected by the AI trainer can be saved and processed for the next sessions. Daily steps tracker can also be added. The trainer will suggest you workout plan and its intensity according to your body type and weight. This application can be developed into a complete android/ios application for ease of use.

From the brief insight provided above, it shows that “AI-based workout assistant and fitness guide” uses some concepts of blaze pose, requires a camera to capture the body pose as input to the system generated and with the help of pose estimator, will provide the stats of calories burnt and exercise count as output in human-readable form.

Future work may include the movement of the camera vertically and horizontally to capture another wide variety of exercises or it may include the use of multiple cameras to capture the body pose from various angles in order to feed the template of other exercises.

## 12. REFERENCES:

1. “PersonLab: Person Pose Estimation & Instance Segmentation with a Bottom-Up, Part-Based, Geometric Embedding Model” G.Papandreou, T.Zhu, L.-C.Chen, S.Gidaris, J.Tompson, K.Murphy.
2. “BlazePose: On-device Real-time Body Pose tracking.” V.Bazarevsky, I.Grishchenko, K.Raveendran, T.Zhu, F. Zhang, M.Grundmann.
- [2] “OpenPose: Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields” Z Cao, G Hidalgo, T Simon, S-E Wei, Y Sheikh.
- [3] “DeepPose: Human Pose Estimation via Deep Neural Networks (August 2014)” A.Toshev, C.Szegedy (Google) 1600 Amphitheatre Pkwy Mountain View, CA 94043.
- [4] COCO 2020 Keypoint Detection Task.
- [5] “Deep Learning-based Human Pose Estimation using OpenCV” By V Gupta.
- [6] “Pose Trainer: Correcting Exercise Posture using Pose Estimation”. By S.Chen, R.R. Yang Department of CS., Stanford University.

- [7] “BlazeFace: Sub-millisecond Neural Face Detection on Mobile GPUs” V.Bazarevsky, Y.Kartynnik, A.Vakunov, K.Raveendran, M.Grundmann.
- [8] “MediaPipe Hands: On-device Real-time Hand Tracking.” F.Zhang, V.Bazarevsky, A.Vakunov, A.Tkachenka, G.Sung, C.L. Chang, M.Grundmann.
- [9] 10.“Composite fields for human pose estimation” by S Kreiss, L Bertoni, and A Alah, IEEE Conference on Computer Vision and Pattern Recognition pages 11977–11986, 2019. 1.
- [10] “Common objects in context” by T Y Lin, M Maire, S Belongie, J Hays, P Perona, D Ramanan, P Dollar, and C Lawrence ´ Zitnick. Microsoft coco: Springer, 2014. 2, 3.
- [11] “Stacked hourglass networks for human pose estimation” by A Newell, K Yang, and J Deng. In European conference on computer vision, pages 483–499. Springer, 2016. 1, 2.
- [12] 13.“Robust 3d hand pose estimation in single depth images: from single-view CNN to multi-view CNNs” by L.Ge, H.Liang, J.Yuan, and D.Thalmann. IEEE conference on computer vision and pattern recognition, 2016.
- [13] “Feature pyramid networks for object detection” by T Yi Lin, P Dollar, R. Girshick, K He, B Hariharan, and S J Belongie. CoRR, abs/1612.03144, 2016.
- [14] “Robust articulated-icp for real-time hand tracking” by A.Tagliasacchi, M.Schroder, A.Tkach, S.Bouaziz, M.Botsch, and M.Pauly. In Computer Graphics Forum, volume 34 Wiley Online Library, 2015.
- [15] “Associative embedding: End-to-end learning for joint detection and grouping” by Newell A, Deng J NIPS. (2017).