Smart Gym Assistant: Posture Monitoring and Rep Counting

Arya Jayasankar
Department of Electronics and Communication Engineering
Amrita School of Engineering, Bangalore
Amrita Vishwa Vidyapeetham, India
bl.en.u4eac22003@bl.students.amrita.edu

Krishnendu M Uday
Department of Electronics and Communication Engineering
Amrita School of Engineering, Bangalore
Amrita Vishwa Vidyapeetham, India
bl.en.u4eac22027@bl.students.amrita.edu

Ishaan Shokeen
Department of Electronics and Communication Engineering
Amrita School of Engineering, Bangalore
Amrita Vishwa Vidyapeetham, India
bl.en.u4eac22022@bl.students.amrita.edu

S. Lalitha
Department of Electronics and Communication Engineering
Amrita School of Engineering, Bangalore
Amrita Vishwa Vidyapeetham, India
s_lalitha@blr.amrita.edu

***Abstract*—Many workout applications offer their users a set of videos to follow, where users have no opportunity to have their form corrected in real time. This approach can lead to unnatural alignment, increase the risk of injury and diminish the usefulness of the exercises. The workout pose corrector with an attached integrated repetition counter is the focus of this paper for improving performance and reducing the possibility of injury while exercising. The system employs state-of-the-art computer vision technology and machine learning algorithms to record video and analyze real-time body position during exercises. It provides error feedback information to the user from the evaluation of skeletal points against that of the ideal pose of a specific exercise. It also contains a counter mechanism which is activated only when the exercise is performed correctly. The information from the experiments revealed that the pose detection system was trained and tested with a precision of 98. 3% during training and 96. 4% during testing, while guaranteeing that the users perform the necessary exercises in the right form.**

***Index Terms*—Convolution Neural Network, deep learning models, real-time monitoring, MediaPipe**

I. Introduction

Recent years have seen a significant increase in home-based fitness routines due to improvements in wearable technology, fitness apps, and online fitness platforms. However, one of the greatest hurdles in this regard is ensuring that every individual performs exercises in the proper form. Poor posture reduces exercise effectiveness and increases the risk of injury. Professional supervision by personal trainers or physiotherapists is then often recommended to correct form, though not available or affordable for all.

To fix this, the paper proposes a computer vision-based workout pose monitoring system with counter that provides real-time exercise form feedback to users.

The system performs pose estimation analysis on the user's body positioning during each workout movement using body landmark detection. It uses body landmark detection to compare the user's pose to the reference module to monitor the correctness [1] [2]. In addition, an automatic repetition counter is incorporated in the system to track and count the number of correctly executed repetitions, ensuring both quality and quantity in exercise performance. The architecture, design choices, and methodologies will be discussed in this paper to construct the model. Associated with these are the evaluations of the experiments. The model is designed to fill the gap between professional training and individual fitness, thus combining real-time pose monitoring and repetition counting to allow safer and more effective workouts.

II. Related Works

Gourangi Taware et al. [3] proposed an artificial intelligence (AI) system that uses Media Pipe's pose estimation method to deliver real-time posture correction while tracking exercise movements. Despite the limited dataset details, the system tracks repetitions, calculates calories burned based on BMI, and reports incorrect workout performance with audio feedback. It also gives personalized dietary recommendations to help customers stay at a healthy weight. The following developments aim to improve posture detection accuracy and conduct more in-depth examinations of posture.

Xiangying Li et al. [4] developed a technique for the classification and calculation of the performing actions from fitness exercises using MediaPipe's BlazePose Model. The algorithm distinguishes between two simple states of the movements: down and up, and the repetitions of movements are estimated and classified using the K-Nearest Neighbors algorithm. Known for its minimum hardware requirements and maximum recognition speed, BlazePose was agreed to be better than other frameworks such as OpenPose and AlphaPose, making it appropriate for on the go fitness applications. This approach improves the election of motion execution in squats and push-ups.

Anubhav Singh et al. [5] proposed, for the first time, the use of a convolutional neural network to estimate the human pose by means of the MPII Human Pose data set. The model effectively computes body joint positions in confidence maps as well as vector fields which communicate the anatomical relationship between the body parts with high accuracy. Though dataset information is scanty, the system has clear potential in activity recognition, human-computer interaction, and safety surveillance. There lends itself another possibility. Future study may aim at equally looking at lesser 3D pose estimation approaches.

Sven Kreiss et al. [6] displayed the PifPaf approach for 2D human position estimation outlined for low-resolution, active settings such as urban portability, . Body parts are distinguished utilizing Portion Escalated Areas (PIF) and connected into entire stances utilizing Portion Affiliation Areas (PAF) in this strategy. Especially in cluttered pictures, PifPaf beats routine strategies like Open Pose and Cover R-CNN by utilizing comprehensive data and a Laplace-based misfortune work. For applications like person on foot discovery in independent frameworks and self-driving automobiles, this breakthrough is basic.

Rahul Ravikant Kanase et al. [7] established a functional system for posing analysis and correction during exercises using an OpenPose model based on CNN. This system pinpoints critical joints during the machine bicep curl or shoulder press, records the angles between joints, and provides corrective feedback in real time. Through computer machine, modeling these directions, or methods, they assist the users in keeping the correct body position thus reducing the risk of injuries. The purpose of this system is to improve the efficiency of the practice by correcting body posture and preventing injuries and providing feedback on the practice of the exercises through video analysis.

Urmi Dedhia et al. [8] developed a virtual fitness instructor supported by a video game with the help of MediaPipe's hardware oriented model BlazePose, which identifies 33 landmarks to control the execution of exercises such as squats and biceps curl. The system analyses the joint angles and applies machine learning algorithms such as SVM and neural networks to determine whether the form is executed correctly and count the number of repetitions. Users are provided with instructional feedback immediately after every similar exercise for them to correct their mistakes. Thanks to the use of BlazePose, it is easy to perform acceleration tracking thus the system is suitable for indoor exercises.

Amritanshu Kumar Singh et al. [9] did a comprehensive study on detecting driver drowsiness using various methods, focusing on real-time detection and prevention of accidents caused by fatigue. Multiple physiological and behavioral indicators are explored, including eye closure rate, yawning frequency, and head movements. We compare machine learning algorithms and computer vision techniques, highlighting their strengths and weaknesses in detecting drowsiness. A combination of facial landmarks and sensor-based data is proposed to enhance accuracy. The study aims to contribute towards creating a more reliable and efficient system for alerting drivers in critical situations, improving road safety overall.

Ainun Syarafana Binti Pauzi et al. [10] proposed a system for tracking body movement using Mediapipe BlazePose to augment labeled skeleton joints onto the body. The system is aimed at applications in physically demanding environments and the sports industry, using deep learning techniques to recognize body joints. Their approach was compared to IMU-based motion capture, with a difference in accuracy of within 10%. The system estimates movement velocity and joint angles, helping to identify risk-prone movements that could lead to injuries over time.

Steven Chen and Richard R Yang [11] discussed their implementation of an application called Pose Trainer, which used a combination of pose estimation and machine learning in order to correct user postures during exercise sessions. Using the OpenPose model for real time keypoint detection, their system assesses exercise quality by matching the joint keypoints and the angles with the standard provided by professionals. However, while the model is efficient in providing feedback on the user's postures during bicep curls and shoulder presses, it still cannot be utilized for other exercises since the accuracy in pose estimation is low at the moment, and so further studies are needed in order to make it possible to extend its use in other applications, including those on mobile platforms.

Hitesh Kotte et al. [12] developed a real-time computer vision system that employs the YOLOv7-pose model to detect a user's posture during fitness exercises. This system not only finds the important human points but also finds out angles of the joints in order to recognize the presented movements and give the user real time feedback in terms of correcting the form. The model reduces retraining by the use of transfer learning and was compared with professional fitness videos. The users' feedback from trials showed encouraging results, however the authors faced challenges of improving elements of the user interface and pose estimation accuracy in the provision of more focused feedback

LA Rakhsith et al. [13] performed a thorough survey on object detection approaches in deep learning for methods like CNN, YOLO, SSD, MobileNet, MTCNN, ResNet and so on. They also pointed out that such methods have been developed and used in other related fields such as image retrieval, security, and surveillance. All the above methods have been compared based on their ability to detect objects, to identify features of objects and to classify objects correctly.

III. Dataset Description

For the training and testing, a semi-custom dataset consisting of around 11,000 images of both correct and incorrect postures has been used. The dataset incorporates video dataset of correct and incorrect postures of push-up from kaggle. The video dataset is then converted into images using cv2 and imwrite for precise training. [14]

TABLE I. Classification results for push-up videos

LSTM Exercise Classification: Push Up Videos		
Dataset	**Correct**	**In-Correct**
Dataset 1	5500	5200

IV. Methodology

The Smart Gym Assistant for posture correction and rep counting proposed in the paper undertakes the task of physical exercises video analysis using deep learning methods. For this system, a Convolution Neural Network (CNN) has been utilized which is widely used in image related tasks such as classification and recognition due to its success. The approach focuses on identifying and correcting exercise forms that are prone to errors. In this paper, the only form analyzed is push-ups. The methodology is divided into several stages which includes data preprocessing, model training, and real-time video analysis.

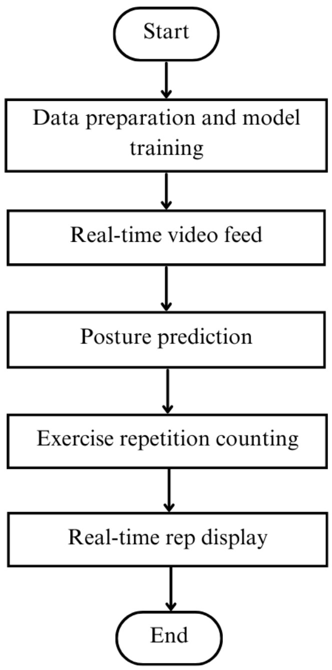

Fig. 1. Methodology

A. Data Preprocessing

Data cleaning is an important process which prepares the dataset before the model is trained. The dataset is composed of a combination of 11,000 images of different workouts showing the person doing the exercises correctly or wrong. Before training, all images have to be normalized so that the shape and size are consistent across the board. This kind of standardization is important to prevent the Convolutional Neural Networks (CNN) model from failing during the training as a result of varied shaped image sizes. The images are resized and remolded with the help of the OpenCV (cv2) library and NumPy available in python.

Here's a brief idea of the methodology according to the block diagram:

1) Start: The process begins with system initialization and the preparation of image data input. Model loads the required libraries and it sets the parameters for further computations.

2) Input Image: Images from the dataset are loaded into the model. The dataset comprises a range of images that are useful in training of the model, mostly the patterns, features or actual objects in relation to the task undertaken.

3) Image Resizing: After the images are loaded into the system, they are resized to a standard dimension of 100x100 using efficient array operations from the NumPy libraries. This resizing aims at standardization of all input images to ensure those are processed efficiently by the model in order to bring down the computation for large data sets.

4) Grayscale Processing: The images are then normalized by resizing and then changed to grayscale format to reduce the dimensions of each image to one color channel, which thus increases the efficiency and decreases the burden on the GPU. Since relatively more images are available in grayscale, most structural and textural features remain intact, sometimes even enough for the model to learn important features when color is not a major concern.

5) Training phase: The images which have been processed are then loaded into the deep learning model. In this stage, the model acquires an ability to find features that are relevant with the image and classify the patterns that are important. This training process is iterative, where the model changes the parameters to make the accuracy of the model ideal. To reduce errors and make the model optimal for identification during the inference stage, methods such as backpropagation and gradient descent are used.

6) End: After the training is completed, the process ends. This happens when the trained model is available for deployment, which means that it can make prediction or classification of the new input images.

Such functions are intended for capacity of images to be adjusted to the required input dimension of the model, further described in (224, 224, 3), which stand for height, width and color channel respectively. Aside from the resizing function, the images are adjusted by scaling the pixels to [0, 1] range, which in turn helps in better convergence during training time.

Besides above techniques, other data augmentation techniques such as rotating, zooming or flipping images are also used in order to make a better and more rugged model. These transformations help in controlling overfitting by adding variety in the training set without collecting more data. This stage is most crucial considering the relatively small size of the dataset as it is known that CNN model does better with bigger datasets.

B. Model Training

After data preparation, CNN Model is then used in this case to do the actual training. Convolutional neural networks entail convolutional layers which are responsible for the scanning of images and identifying key features structural and non-structural forms from the image.The dataset is classified into two categories namely correct and incorrect postures.

The basic structure of the CNN includes many channels of CNN which are then followed by the layers of max-pooling and fully connected layers. Each of the convolutional layers processes the input pictures through a number of filters generating feature maps from the images. The fully connected and pooling layers assist in reducing the size of the created feature maps hence allowing the model to be less complicated and yet maintaining essential information. For the binary classification of correct for inaccurate and incorrect posture the model uses a binary cross entropy loss function which is appropriate in two class problems. Due to the adaptive nature of the Adam optimizer, the model performs faster training and reaches convergence efficiently. The performance measure that the model is able to produce is Accuracy, which refers to the ratio of the number of images correctly classified to the total number of images that were presented.

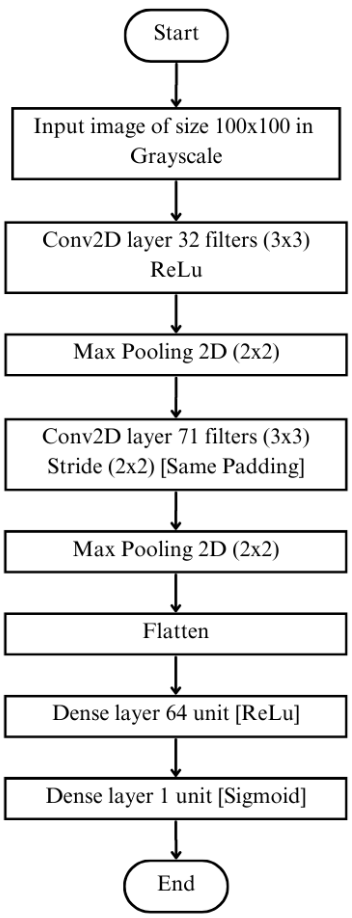

Fig. 2. CNN Model [15]

Number of samples processed at a time is 32 and the model is trained for 50 epochs in order to help it perform better on new data. Figure 2 explains how the model is implemented right from the input data to the resulting classification. The model's output layer is a single neuron, with a sigmoid function in it which predicts the probability of the response being yes or no. In other words it predicts the correctness of the posture, whether it is correct or wrong. To enhance the generalization of the developed model, some measures such as dropout and batch normalization are applied. The Dropout technique is used in solving overfitting problem by taking a portion of some of the neurons to be zero at one training instance which makes the over all model learn powerful features. In addition, Batch normalization is used to ensure that the training is efficient during the training of the network by reducing the output of each layer, enabling efficient training with bigger learning rate.

C. Real Time Analysis

In the context of real-time monitoring and feedback, posture estimation using the trained model of the CNN and computer vision techniques is performed during exercises. In this stage there are two main tasks: postural adjustment and counting of training repetitions. Using MediaPipe, a frameworks for real-time perception, we obtain human key points which include joints and limbs. These includes the body markers which help in getting the angles between the various body parts which are important in ascertaining if the user is in correct posture [16].

In push-ups critical angles include the angle of the flexed forearm in relation to the flexed upper arm as well as the angle of the abdomen in relation to the thigh bone. Figure 3 shows how these angles are worked out based on the perceived landmarks of the body. The ongoing measurement of the angle is performed with the use of mathematical functions from python library. Each angle has a threshold value set within a certain limit; when a user attempts a certain posture which approaches in exceeding these thresholds, the system marks the posture as an incorrect one. This feedback loop allows for the absence of a pause regarding error detection, whereby suggestions on ways of rectifying a users' form are made instant. For instance, in the event where an angle of the forearm in flexion of the upper arm is satisfactorily straightened, the program invites that the upper trunk be flexed more towards the ground.

So that they can avoid miscounting when performing repetitions, such angles are measured at various intervals and compared to determine any changes. Each complete cycle of variation of the angle, for example lowering and raising the body weight during a push-up, constitutes one repetition of the angle variation. This allows for timely rep counting with the system promoting correct bodys haping by the users as they gym. The counting of repetitions and the adjustment of posture carry on at the same time in order to analyze the performance of the workout.

D. Combining of Models

At this level the last step is combining the CNN model in a way that integrates it with the real time analytics pipeline. While the CNN model assesses the correctness of the user's posture, the real time analytics recognizes the body positions and the number of repetitions. The two models run at the same time and in the same setting in order for an accurate guidance to be provided to the user in real time. A detailed explanation about the architecture of the combined model is presented in Figure 4.

A push-up analytical system which extends to other exercises has the potential to be simple and extendable by redeploying the body markers and the angle calculations at the base of the system. The combination of MediaPipe and the CNN model makes it easy for the use of the system even in a non-clinical setting as long as there is a camera, hence the feasibility of users training even at home and monitoring their form. In other words, this particular methodology allows for the efficient implementation of the principles of deep learning alongside real-time computer vision to develop an effective and convenient portable posture correction and rep counting device. Since there is both sight and mathematical judgment use in the system, which is quite commendable, users get to receive prompt feedback thus helping them enhance their style and get the desired level of fitness.

V. Results

For the test efficiency of the model it is run through various test samples to provide a better evaluation of the system. The model has been evaluated on basis of its ability to monitor posture and provide an accurate repetition Fig 3 to 8. shows the output analysis of the model run through an inputted data that includes conversion of raw image into compressed form, monitoring of the posture, evaluation on basis of posture correction and repetition count using body landmarks.

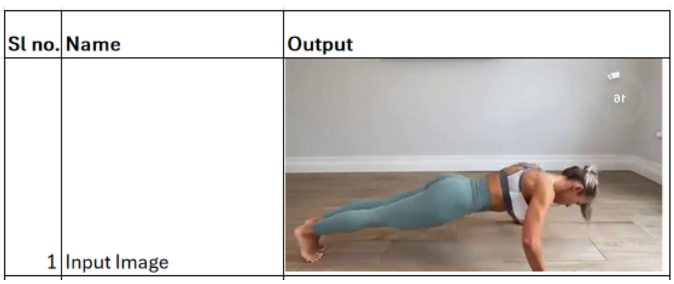

Fig. 3. Input image

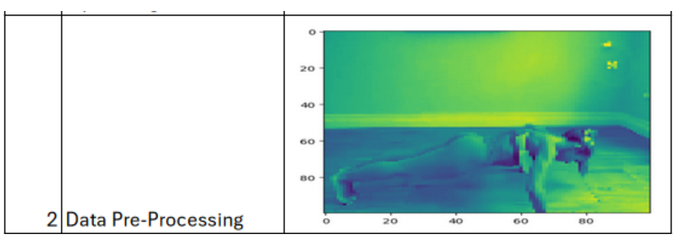

Fig. 4. Data Pre-Processing

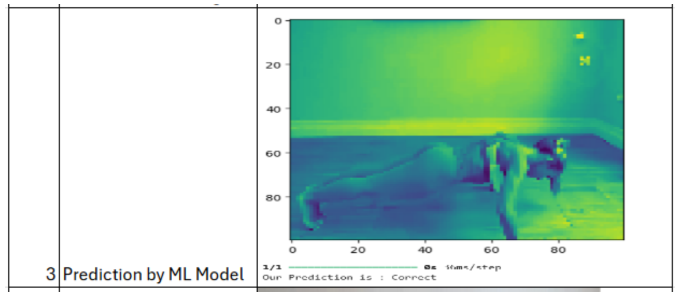

Fig. 5. Prediction by ML Model (Correct)

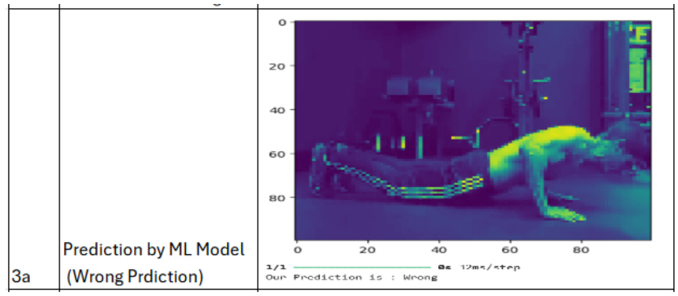

Fig. 6. Prediction by ML Model (Incorrect)

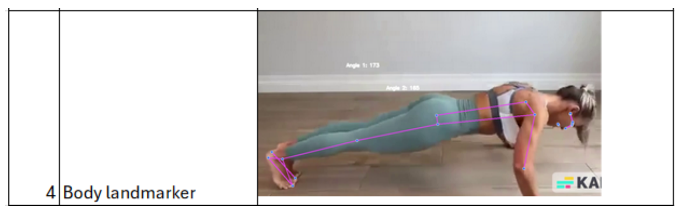

Fig. 7. Body landmarkers

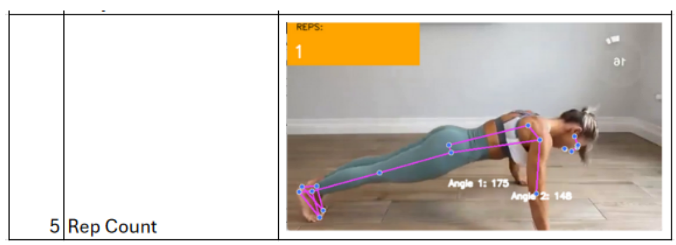

Fig. 8. Rep counter

After training of the model, it is able to successfully generate an accuracy of **98.3%** while training and **96.4%** while testing. An error percentage of only 2.97 % was observed. The promising results of the train and test data help us to apply the model to the real-time function which helps in counting the number of reps, and correctly analyze the correctness of the posture.

Comparative analysis has been done based on the strength and applications of the model, and the models of Fitness Action Counting's and Aatma Yoga Automation's papers. The model achieved 98.3% accuracy in training and 96.4% accuracy in testing. This is the reason why it can be applied in real-time posture correction and repetition counting during workout sessions. The Fitness action counting paper model uses the BlazePose model for more straightforward exercises, such as squats and push-ups, proves to be reliable in lightweight, mobile-friendly applications with a very high accuracy rate of 98.65%. On the other hand, Aatma Yoga Automation paper's model is about personalized yoga pose correction and recommendation, with relatively modest accuracy in using EfficientNet at 75.4%.

For real time analysis of the model, some functions needs to be incorporated into the program this includes the calculation of reps, angle and linear marking of body co-ordinates. This can be achieved by using media pipe and body markers. For exercises like push ups, calculation of the angle includes those in between the forearm and the upper arm and calculation of the angle between the abdominal area and the femur.

Every cycle of appropriate angle change can help in counting each and every rep. Using body markers file available in media pipe, the model is able to mark the area present in the entire body and calculate the above-mentioned angle using mat functions.

Real time analysis includes the integration of both models in a synchronous manner and help the user detect the margin of errors in the posture and specify a solution related to fixing the posture related issues in live feedback as shown in fig 9 and 10.

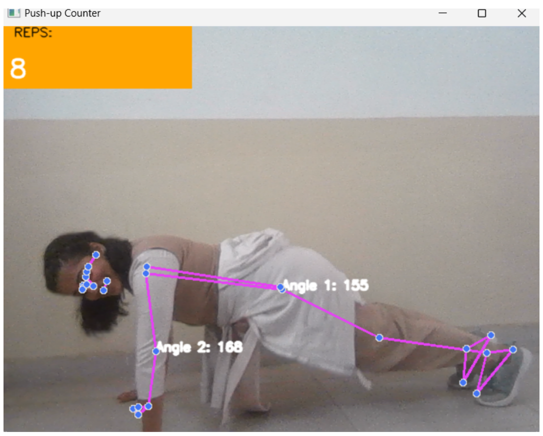

Fig. 9. Real-time Posture 1

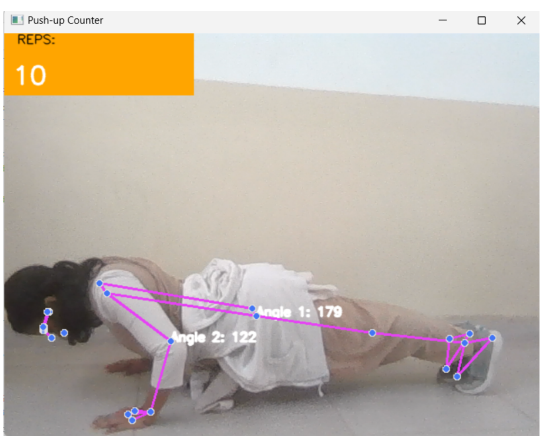

Fig. 10. Real-time Posture 2

A. Limitations:

The current model is still an attempt at more push up exercises only and is restricted to specific and rather simple movements. Extensions such as yoga or active training may necessitate further development, especially with regard to the range of motion and positions of the joints which do not have such rigid postures. In addition, real time performance, while reasonably satisfactory, is likely to have latency issues due to the capabilities of the hardware running the model. It has also relied on the 2D pose estimation techniques despite their limitations in that cut focusing on any exercise via 2D view works effectively on simple postures only.

VI. Conclusion

This project aims at providing real-time warning/feedback to users during personal workout sessions, possibly using tech to correct posture. The purpose of the machine is to enable a person to exercise in body privacy without the help of a trainer, that is why it is a flexible exercise system. Using machine learning algorithms for analyzing posture of workouts from videos and images is that users may get feedback on their performance within seconds, knowing whether they have a proper form while workout, which is important for avoiding possible injuries and receiving the maximum outcome of exercising.

The next version of this system could add an alert system that goes off when an improper posture is maintained for some time. This would act as an anticipatory measure against the incidence of injuries or accidents that might occur due to maintaining a wrong posture during exercises. It would alert the users and in the same breath help them learn the correct form through the prompts, which would be part of the alarm.

It can go further than merely patterns of push-ups, as the system suggests. In future developments, more types of workouts like yoga or weight lifting might be included because the execution of such workouts requires the right form. This would ensure that the solution would be useful in complying with a variety of fitness activities, thus providing more customers, and encouraging consumption from all angles of fitness.

A Mobile Application can be developed to make use of this technology easier. The experience will allow the user to monitor workouts, receive real-time posture updates, and have a workout history. It could also encompass other elements that, e.g., convey tutorials on exercises that are offered, individual schedules with meal plans, and workout instructions besides displaying results after consecutive months.

Another interesting addition would be to add a countdown for the number of repetitions, so that the user could input the goal of a specific number of repetitions for that particular exercise. If the number is set, a buzzer could sound, which will give a sound signal that the proper amount has been set. Especially valuable for those training to achieve particular results, this feature would allow avoiding the necessity to monitor repetitions, supporting the overall convenience of using the application.

References

[1] S. Maganti, M. Yalamanchili, P. S. Bandy, and S. K. J, "Height and Weight Estimation of an Individual from Virtual Visuals," 2022 International Conference on Inventive Computation Technologies (ICICT), Jul. 2022

[2] Vijaya Raghava Duppala et al., “Aatma Yoga: Automation of Yoga Pose Recognition and Recommendation using Deep Learning,” 2022 International Conference on Inventive Computation Technologies (ICICT), Apr. 2024.

[3] G. Taware, R. Kharat, P. Dhende, P. Jondhalekar, and R. Agrawal, “AI-Based Workout Assistant and Fitness Guide,” IEEE Xplore, Aug. 01, 2022.

[4] X. Li, M. Zhang, J. Gu, and Z. Zhang, “Fitness Action Counting Based on MediaPipe,” IEEE Xplore, Nov. 01, 2022.

[5] A. Singh, S. Agarwal, Preeti Nagrath, A. Saxena, and N. Thakur, “Human Pose Estimation Using Convolutional Neural Networks,” Feb. 2019

[6] S. Kreiss, L. Bertoni, and A. Alahi, “PifPaf: Composite Fields for Human Pose Estimation,” 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Jun. 2019.

[7] R. Kanase, A. Kumavat, R. D. Sinalkar, and S. Somani, “Pose Estimation and Correcting Exercise Posture,” ITM Web of Conferences, 2021.

[8] U. Dedhia, Pratham Bhoir, Prateek Ranka, and P. Kanani, “Pose Estimation and Virtual Gym Assistant Using MediaPipe and Machine Learning,” Sep. 2023.

[9] A. Singh, Vedant Arvind Kumbhare, and K. Arthi, “Real-Time Human Pose Detection and Recognition Using MediaPipe,” pp. 145–154, Jan. 2022.

[10] A. S. B. Pauzi et al., “Movement Estimation Using Mediapipe BlazePose,” Advances in Visual Informatics, pp. 562–571, 2021.

[11] S. Chen and R. R. Yang, “Pose Trainer: Correcting Exercise Posture using Pose Estimation,” arXiv.org, Jun. 21, 2020.

[12] H. Kotte, F. Daiber, Milos Kravcik, and Nghia Duong-Trung, “FitSight: Tracking and Feedback Engine for Personalized Fitness Training,” Jun. 2024.

[13] R. L. A, A. K. S, K. B. E, A. N. D, and K. K. V, “A Survey on Object Detection Methods in Deep Learning,” IEEE Xplore, Aug. 01, 2021.

[14] M. Ashraf, “LSTM Exercise Classification: Push Up Videos,” Kaggle.com, 2023. https://www.kaggle.com/datasets/mohamadashrafsalama/pushup

[15] T. Singh, Prakash Duraisamy, K. N. Reddy, M. T. Reddy, and K. V. Yeswanth, “Emotion Recognition in Images: CNN-based Classification of Happy and Sad States,” 2022 13th International Conference on Computing Communication and Networking Technologies (ICCCNT), pp. 1–5, Jun. 2024.

[16] P. Govindarajan, A. K. K, A. S. Pramod, and Abebe Tesfahun, “Yoga in Digital Age: A Hybrid Approach Using ML and Computer Vision,” pp. 16–23, Nov. 2023.