EXERCISE MONITORING USING MACHINE LEARNING

Febin Rajan[1], Joel Sabu[2], Binil V B[3], Badush[4], Jibin C Baby[5], Anurudh Deepan[6], Ignatius Nobel Chittilappilly[7], Joyal Jaison[8], Thulasi Rajan K[9]

Department of Computer Science and Engineering
Vidya Academy of Science & Technology, Thalakkottukara, Thrissur, Kerala, India - 680 501

Abstract. The goal of this idea is to create an exercise monitoring system that can help users improve their workout routines by providing accurate and personalized feedback on their form and technique.The system uses computer vision techniques to analyse the user's body position and provide personalized recommendations for improvement. Users can use the system to track their workouts and receive feedback on their form and technique. The exercise monitoring system is designed to be adaptable to the user's fitness level and can be customized to their individual needs and goals. The system was evaluated using a series of user tests and received positive feedback for its effectiveness in improving exercise form and technique.

Keywords: Deep Learning, CNN, Pose Landmark Model, Curl counter logic, Computer Vision.

1 Introduction

1.1 General

Incorporating advanced technologies such as computer vision and machine learning into exercise monitoring opens up new possibilities for the fitness industry. It introduces an innovative and interactive approach to fitness tracking, which can help users stay motivated and engaged in their workouts. By integrating the system into various devices and applications, we aim to make exercise monitoring more accessible and convenient for individuals, potentially revolutionizing the way people approach their fitness routines.

The project aims to provide individuals with a reliable, accurate and personalized tool to enhance their exercise form and technique. By improving the effectiveness and safety of exercise, the system aims to help users achieve their fitness goals and maintain a consistent and rewarding workout routine.

1.2 Objectives of the Idea

This project aims to create an exercise monitoring system using computer vision and machine learning. It provides personalized feedback on exercise form and technique, helping users improve their workouts. The system tracks progress, offers real-time recommendations, and can be customized to individual fitness goals. User tests have shown positive feedback on its effectiveness. This technology has implications for the fitness industry and can be integrated into various devices. Future work involves expanding exercise options and integrating with existing fitness tracking systems.

1.3 Motivation for this Idea

Many people struggle with maintaining proper form and technique while exercising, which can lead to sub optimal results or even injuries. By developing an exercise monitoring system, we aim to provide users with accurate and personalized feedback on their form and technique in real-time. This feedback can help users make necessary adjustments and improvements, leading to more effective and safer workouts.

1.4 Methodologies Adopted

A convolutional neural network (CNN) is a type of neural network used in signal processing and image processing. CNN is chosen as a classifier because it gives high accuracy In our proposed system, three phases exist.

The first stage is data preprocessing, where the data set is collected. The data set consists of images of different body postures. The classified data set is further divided into a training data set and a testing data set. An augmentation process is performed on the data set.

The next stage is the training stage, where we train the model with the data set to find Different body postures. The model is developed with the help of machine learning algorithms and other deep learning libraries to provide highly accurate models.

The last phase is the testing phase, where we use the trained model to detect people real time. From the video stream, bodies are detected and extracted for inspection If the body posture is correct or not. Now, search results are shown with precision.

2 LITERATURE REVIEW

[1] Open-Pose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields

This work presents a real-time approach for multi-person 2D pose estimation using Part Affinity Fields (PAFs) to associate body parts with individuals in images. The bottom-up system achieves high accuracy and real-time performance, regardless of the number of people in the image. By refining PAFs only, not both PAFs and body part locations, we achieve significant improvements in runtime performance and accuracy. Moreover, we introduce a combined body and foot keypoint detector, reducing inference time while preserving individual accuracy. The culmination of this effort is OpenPose, the first open-source real-time system for multi-person 2D pose detection, encompassing body, foot, hand, and facial keypoints.

[2] AI Human Pose Estimation: Yoga Pose Detection and Correction

It highlights the challenges faced in using computer vision technology for assessing human posture in the context of yoga. The paper discusses various technologies for pose estimation and suggests the best approaches for an Android app application based on ease of use. The methodology involves passing the image through a CNN classifier to detect people and then employing a pose estimation network to identify joints and limbs. The app displays the image to users with markers indicating different body parts, aiding in understanding and correcting yoga poses effectively.

[3] Body Posture Detection and Motion Tracking Using AI For Medical Exercises and Recommendation System

AI-driven body posture detection and motion tracking are vital for medical exercises. Exercises aid healing and improve overall health. AI and Image Processing enhance workouts without professional supervision. A software-based motion tracker monitors exercises, offering real-time posture feedback. The MediaPipe framework analyzes body movement and stores data for personalized exercise apps. Users can be linked to verified doctors for tailored recommendations using medical history from databases.

[4] An Efficient Convolutional Network for Human Pose Estimation

In recent years, human position estimation has benefited significantly from deep learning and led to significant performance improvements. However, the progress of measurement accuracy has led to expensive computing network architecture, requiring expensive hardware and large databases to be preformed. This complexity hinders comparison of methods and reproducibility of results. This paper presents an efficient deep network architecture that can be trained without prior training on mid-range GPUs. Despite the low computational requirements, the network compares favorably with more complex models in popular benchmarks for human pose estimation.

[5] Real-time Yoga recognition using deep learning

This study presents a novel approach to accurately identify six yoga asanas using a hybrid deep learning model. A dataset created with a common RGB webcam is made publicly available. The proposed model combines openpos keypoints for temporal predictions and a convolutional neural network (CNN) for feature extraction from long short-term memory (LSTM), achieving a test accuracy of 99.04% on single frames and pooling of 99.38% prediction frames on 4 After doing This end-to-end deep learning pipeline, a first in yoga recognition, shows 98.92% real-time accuracy on a different set of 12 individuals. The results highlight the effectiveness of the system compared to state-of-the-art.

[6] ExNET: Deep Neural Networkfor Exercise Pose Detection

Pose detection estimates human activity in images or video frames using computer vision techniques. It has various applications, including body augmentation, fitness, and animation. ExNET represents a method for detecting human pose from 2D exercise images using Convolutional Neural Networks. In recent times, Deep Learning-based systems enable the detection of exercise poses from images. The model for this task is referred to as ExNET: Deep Neural Network for Exercise Pose Detection. The proposed model has been evaluated on a dataset comprising 2000 images, distributed into 5 classes for training and testing. The experiments on the test dataset achieved an improved performance, with the best accuracy reaching 82.68%.

[7] MobileNetV2: Inverted Residuals and Linear Bottlenecks

This paper introduces MobileNetV2, a novel mobile architecture designed to enhance performance across diverse tasks and benchmarks, catering to various model sizes. The framework includes efficient object detection using SSDLite and the development of mobile semantic segmentation models via a simplified DeepLabv3 variant termed Mobile DeepLabv3. MobileNetV2 employs an inverted residual structure with thin bottleneck layers, in contrast to traditional models, utilizing lightweight depthwise convolutions in the intermediate expansion layer. Notably, the elimination of non-linearities in narrow layers improves representational power. The paper underscores the decoupling of input/output domains from transformation expressiveness, providing a convenient analytical framework.

[8] GHUM & GHUML: Generative 3D Human Shape and Articulated Pose Models

Presented is a statistical, articulated 3D human shape modeling pipeline within a fully trainable, modular deep learning framework. Using high-resolution 3D body scans captured in diverse poses, along with close-ups of head, facial expressions, and hand articulation, and starting with artist-designed gender-neutral rigged quad-meshes, all model parameters are trained, including non-linear shape spaces, pose-space deformation correctives, skeleton joint center predictors, and blend skinning functions. The training loop incorporates over 60,000 diverse human configurations to capture correlations and ensure consistency. The models support facial expression analysis, body (with detailed hand) shape, and pose estimation.

[9] BlazePose: On-device Real-time Body Pose tracking

Presented is BlazePose, a lightweight convolutional neural network architecture designed for real-time human pose estimation on mobile devices. The network achieves over 30 frames per second on a Pixel 2 phone, providing 33 body keypoints for a single person during inference. This makes it well-suited for real-time applications such as fitness tracking and sign language recognition. The contributions include a unique body pose tracking solution and a lightweight neural network for body pose estimation, utilizing both heatmaps and regression for keypoint coordinates.

3 SYSTEM DESIGN

3.1 System Architecture

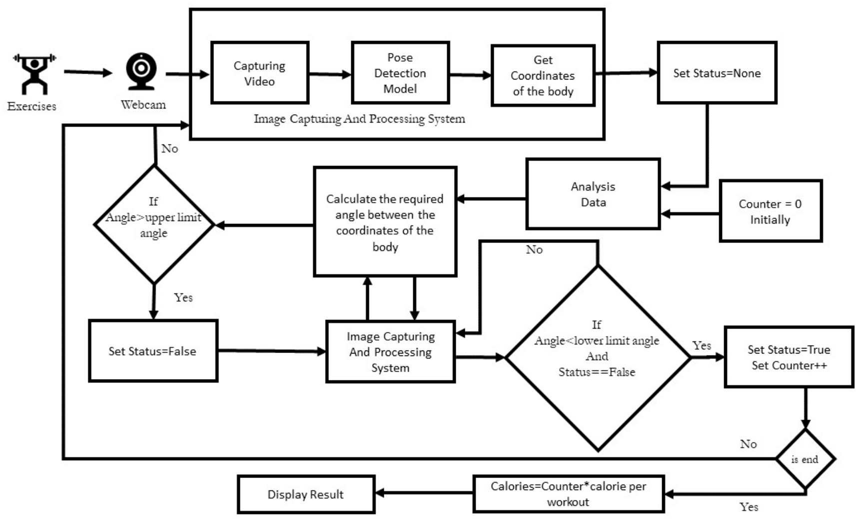

Fig. 1. System Architecture

Exercises : The person starts exercising in front of the webcam.

Webcam : The webcam captures the video of the person exercising.

Capturing Video : This is the initial stage where the video from the webcam is captured for processing.

Pose Detection Model : The captured video is processed using a pose detection model that identifies the person's body posture and movements.

Get Coordinates of the body : After the pose is detected, the coordinates of the body are extracted for further analysis.

Set Status=None : Initially, a status is set to False, indicating that the exercise has not been performed correctly or has not yet started.

Counter = 0 Initially : The counter that tracks the correct repetitions starts at zero.

Image Capturing And Processing System : The system captures and processes images, probably in a loop to continuously monitor the exercises.

If Angle > Upper limit angle : A decision is made to check if the angle of a certain body part (relevant to the exercise) is less than a predefined lower limit angle.

Set Status=False : If the angle is above the lower limit, the status is set to True, indicating a correct movement, and the counter that keeps track of the number of correct repetitions is incremented.

Calculate the required angle between the coordinates of the body : The system calculates the angle between specific body parts to determine if the exercise is being performed correctly.

Analysis Data : The system analyzes the data to monitor the workout progress.

If Angle < lower limit angle and Status==False : Another decision is made to check whether the angle is greater than an upper limit angle and if the previous status was True, indicating the end of a correct movement.

Set Status=True, Set Counter++ : If the angle is less than lower limit, the status is set to True, indicating a correct movement and the counter that keeps track of the number of correct repetitions is incremented.

Is end : A decision is made to determine whether the workout session has ended.

Calories=Counter*calorie per workout : If the workout session has ended, the system calculates the total calories burned by multiplying the number of correct repetitions by the calories burned per workout.

Display Result : Finally, the results, including the correct repetitions and calories burned, are displayed to the user.

Fig.1 describes a system designed to monitor and provide feedback on a person's workout by using computer vision to detect and analyze their movements and postures.

3.2 Data Flow Diagram

A facts glide diagram (DFD) maps out the waft of records for any process or machine. It makes use of described symbols like rectangles, circles and arrows, plus short textual content labels, to show data inputs, outputs, storage points and the routes among each destination. records flowcharts can range from simple, even hand-drawn manner overviews, to in-intensity, multi-level DFD's that dig gradually deeper into how the statistics is treated. they could be used to investigate an current machine or model a new one.

Like all of the best diagrams and charts, a DFD can regularly visually "say" things that would be tough to provide an explanation for in words, and they paintings for both technical and nontechnical audiences, from developer to CEO.That's why DFD's continue to be so famous in spite of everything these years. at the same time as they paintings properly for statistics go with the flow software and structures, they may be much less applicable in recent times to visualizing interactive, actual-time or database-orientated software program or structures.

The Fig.2 maps out the flow of information for any process or system. For attaining the above goal, it uses defined symbols like circles, rectangles, and arrows.The data set initially collected is pre-processed. This pre-processed data set undergoes splitting into two sections, training and testing data. Training is performed using the training data hence the trained model is obtained. The test data is then passed to monitor the success rate and efficiency of the model. The output of the testing stage is the estimation of body posture and pass to monitoring section.

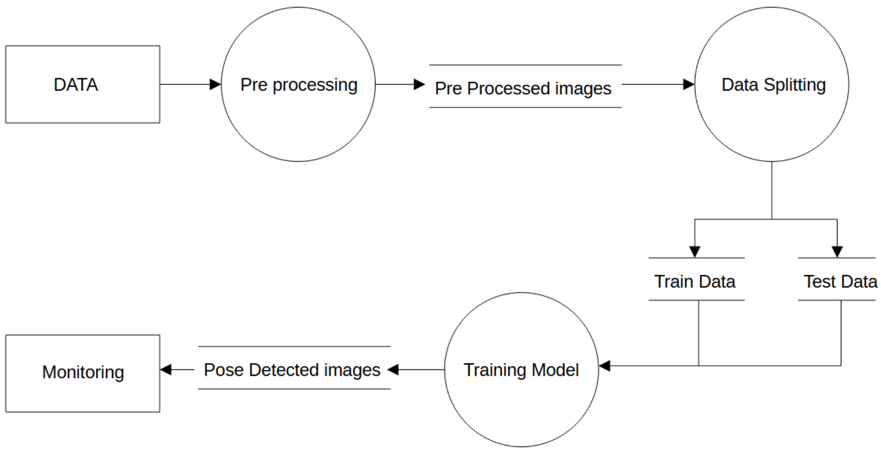

Fig. 2. Data Flow Diagram

3.3 Use Case Diagram

One impactful use case for the exercise monitoring system is its application in self-guided workouts. Users can leverage the system to receive real-time feedback on their exercise form and technique without the need for a personal trainer. The AI-powered system analyzes the user's body position using computer vision techniques and provides personalized recommendations for improvement. Users can track their workouts, monitor their progress over time, and receive accurate feedback on their form, helping them optimize their workout routines and minimize the risk of injuries. This use case empowers individuals to take control of their fitness journey, offering them a virtual fitness companion that guides and supports their exercise sessions. With the exercise monitoring system, users can enhance their workout experience, improve their technique, and achieve their fitness goals with confidence and independence.

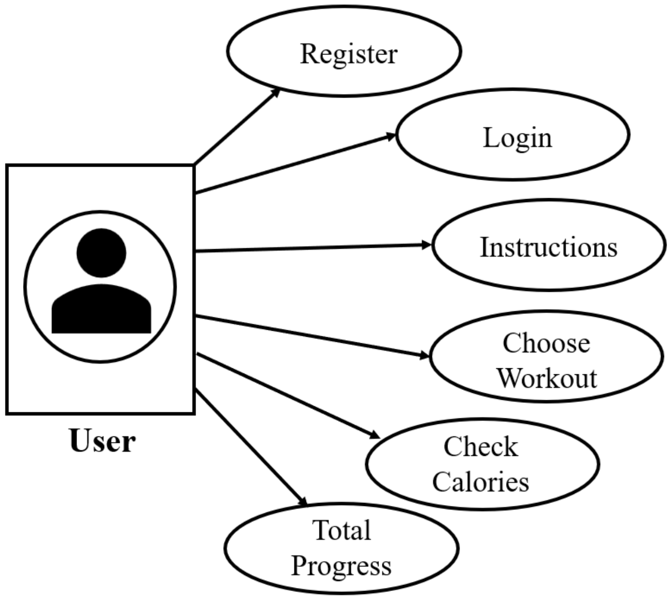

Fig. 3. Use Case Diagram

Register: This is the process where a new user creates an account with the application. The user would typically provide personal information such as name, email, and create a password to set up their profile.

Login: After registration, users can access their accounts by entering their credentials, usually an email or username and password. This step is necessary for authentication and to access personalized features of the app.

Instructions: This feature likely provides users with guidelines on how to use the app effectively. It may include tutorials, FAQs, or helpful tips for navigating the app and making the most of its features.

Choose Workout: Here, users can select from various workout routines or exercises provided by the app. It could offer different levels of intensity, workout durations, or targeted areas of the body to customize the user's exercise experience.

Check Calories: 5. This function allows users to monitor their caloric intake and expenditure. Users might be able to input foods they've consumed to track calories, and the app could also estimate the number of calories burned during workouts.

Total Progress: This feature likely provides an overview of the user's fitness journey. It could show progress towards goals, improvements over time, or comparisons of performance metrics. This function helps users stay motivated and informed about their health and fitness achievements.

4 METHODOLOGY

4.1 Human Body Detection

Human body detection using deep learning and Convolutional Neural Networks (CNNs) is a state-of-the-art computer vision technique. It involves training CNNs on labeled datasets to accurately identify and locate human bodies in images or videos. The trained model can be applied to various applications like surveillance and gesture recognition, revolutionizing computer vision systems and enabling practical solutions in different industries.

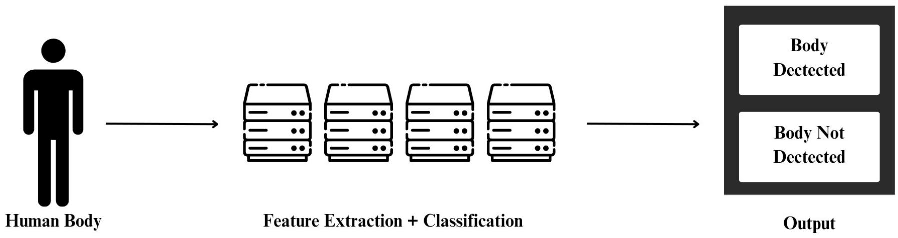

Fig. 4. Deep Learning Process

4.2 Pose Landmark Detection

Identify the key body location and render visual effects on them. We have to train our system to recognize the body. This task uses machine learning algorithm that can work with single images or continuous stream of images. The task outputs body pose landmarks in image coordinates and in 3D world coordinates.

Model output includes Landmarks normalized coordinates and WorldLandmarks world coordinates for each landmark.

The pose landmarker model tracks 33 body landmark locations, representing the approximate location of the following body parts:
0 - nose
1 - left eye (inner)
2 - left eye
3 - left eye (outer)
4 - right eye (inner)

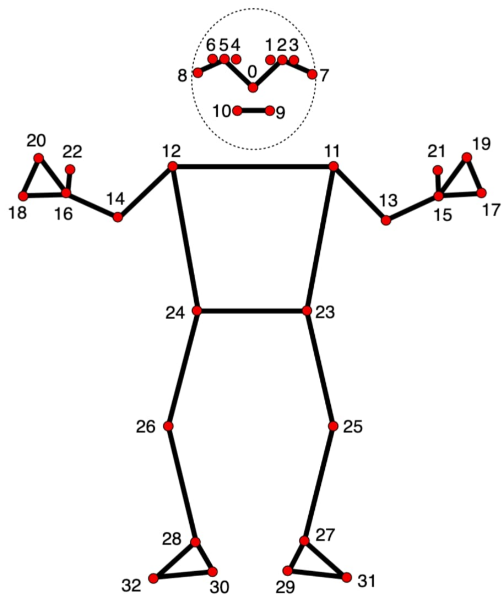

Fig. 5. Landmarks of the Body [15].

5 - right eye
6 - right eye (outer)
7 - left ear
8 - right ear
9 - mouth (left)
10 - mouth (right)
11 - left shoulder
12 - right shoulder
13 - left elbow
14 - right elbow
15 - left wrist
16 - right wrist
17 - left pinky
18 - right pinky
19 - left index
20 - right index
21 - left thumb
22 - right thumb
23 - left hip
24 - right hip
25 - left knee

26 - right knee
27 - left ankle
28 - right ankle
29 - left heel
30 - right heel
31 - left foot index
32 - right foot index [15]

The Pose Landmarker utilizes a tandem of models to infer pose landmarks. The initial model identifies human bodies within an image frame, accompanied by a handful of key pose landmarks. Subsequently, the second model precisely localizes landmarks on the bodies.

Packaged within a downloadable model bundle are:

1. Pose detection model: This model discerns the presence of bodies along with essential pose landmarks.

2. Pose landmarker model: Enhancing the mapping of the pose, this model furnishes a comprehensive set of 33 3-dimensional pose landmarks.

Employing a convolutional neural network akin to MobileNetV2 [7], this bundle is tailor-made for on-device, real-time fitness applications. This variant of the BlazePose model [9] integrates GHUM [8], a 3D human shape modeling pipeline, to accurately estimate the full 3D body pose in images or videos.

The biceps curls and push ups use angles between the landmarks {16, 14, 12}(right hand) and {11,13,15}(left hand), squats use angles between the landmarks {28,26,24}(right hand) and {27,25,23}(left hand) and sit ups use angles between the landmarks {26,24,12}(right hand) and {25,23,11}(left hand)

4.3 Calculation of angles

Calculate the angle between three points in a 2D space using the arctan2 function

1. Convert three sets of coordinates (a, b, and c) into arrays.
2. Calculate the angle between the vectors ba and bc using the arctan2 function.
3. Convert the angle from radians to degree.
4. If the angle is greater than 180 degrees, adjust it to be in the range [0, 180) by subtracting it from 360 degrees.

radians = arctan(c[1] - b[1], c[0] - b[0]) -arctan(a[1] - b[1], a[0] - b[0])
angle = abs(radians * 180.0 /π)
if angle > 180.0:
    angle = 360 - angle

Arc tangent of 2 numbers or 4 quadrant inverse tangent arctan(x,y) returns arc tangent of x and y. This is similar to y/x arc tangent, except that the sign of both arguments is used to find the quadrant. The result is the angle in radians.

4.4 Curl counter logic

The initial status is set to none, and the counter is set to zero. If the angle is greater than a upper limit, then we have to set status FALSE. If the angle is less than lower limit and status is FALSE, then set status TRUE. Whenever the status become TRUE we have to increment the counter by 1. The variable "counter" is significant in the exercise monitoring system as it tracks the number of correct repetitions during a workout. When the system detects a correct movement, the counter is incremented by 1. This allows the system to monitor the user's progress and provide feedback based on the number of correct repetitions performed.

For Push ups and Biceps;
if angle > 160:
    status as "FALSE"
if angle < 70 and status == 'FALSE':
    set status as "TRUE"
    increment counter

For Squats;
if angle > 105:
    set status as "FALSE"
if angle < 55 and status == 'FALSE':
    set status as "TRUE"
    increment counter

For Sit Up;
if angle > 160:
    set status as "FALSE"
if angle < 30 and status == 'FALSE':
    set status as "TRUE"
    increment counter

4.5 Calculation of calories

The calculation of calories per workout is based on the counter value. To calculate the total calories burned during a workout, the counter value is multiplied by the calories per workout. This provides an estimate of the total energy expended during the exercise session.

For example, if a user performs 20 correct repetitions of a particular exercise and the calories per workout for that exercise is 5, then the total calories burned would be:

Total calories = counter * calories per workout
Total calories = 20 * 5
Total calories = 100
In this example, the user would have burned an estimated 100 calories during the workout session based on the number of correct repetitions and the calories per workout value.

4.6 Data Set

The dataset consists of labeled examples of various exercise poses performed by individuals. Each example includes image or video data capturing different angles and perspectives of the exercises, along with corresponding annotations indicating the correct form and technique.

Fig. 6. Data Set

The figures depict the data set and various exercises, showcasing the system's capability to provide personalized feedback on exercise form and technique. The number of images or poses required to provide recommendations on form and technique for an individual can vary based on the complexity and diversity of exercises being analyzed. However, a comprehensive and precisely labeled dataset, as mentioned in the document, is essential for the exercise monitoring system to learn from a diverse range of exercise examples and provide accurate feedback on users' exercise form and technique.

The dataset consists of labeled examples of various exercise poses performed by individuals, capturing different angles and perspectives of the exercises, along with corresponding annotations indicating the correct form and technique. This diverse dataset includes samples from individuals with different body types, fitness situations, and variations in form. By leveraging this comprehensive dataset, the exercise monitoring system can provide accurate and personalized feedback, empowering individuals to optimize their workout routines and achieve their fitness objectives.

5 Experimental Results

Experimental results of the fitness tracker demonstrate its effectiveness in providing accurate, personalized feedback on fitness and style. The system uses computer vision technology and machine learning algorithms to instantly analyze exercise and provide users with useful information. Rigorous testing, including user testing, has received positive feedback, especially regarding the improvement of form and structure. The system is evaluated with clear and precise data, allowing it to learn from a variety of examples and provide accurate feedback. Additionally, future development of the system aims to expand its functionality and enable real-time and effortless tracking with smart devices such as smartwatches or fitness trackers for feedback.

Ultimately, experimental results confirm exercise tracking's ability to provide accurate and personalized feedback, allowing people to optimize their exercise and achieve their health goals. The integration of artificial intelligence and wearable devices offers a new way to increase the convenience and efficiency of sports viewing, providing users with solutions to problems in accessing sports.

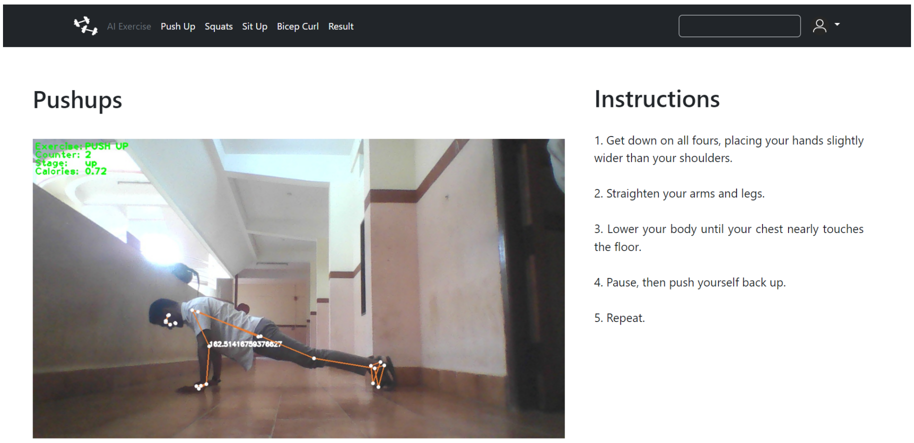

Fig. 7. Push Up Page of the System.

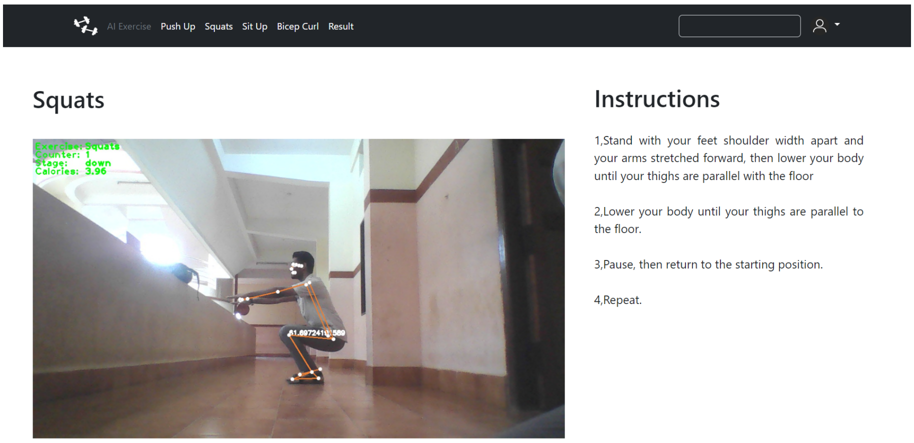

Fig. 8. Squats Page of the System.

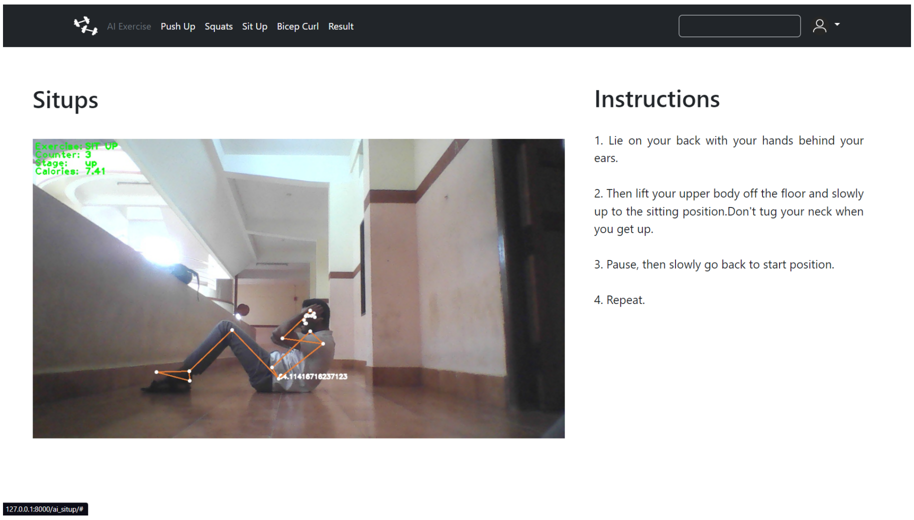

Fig. 9. Sit Up Page of the System.

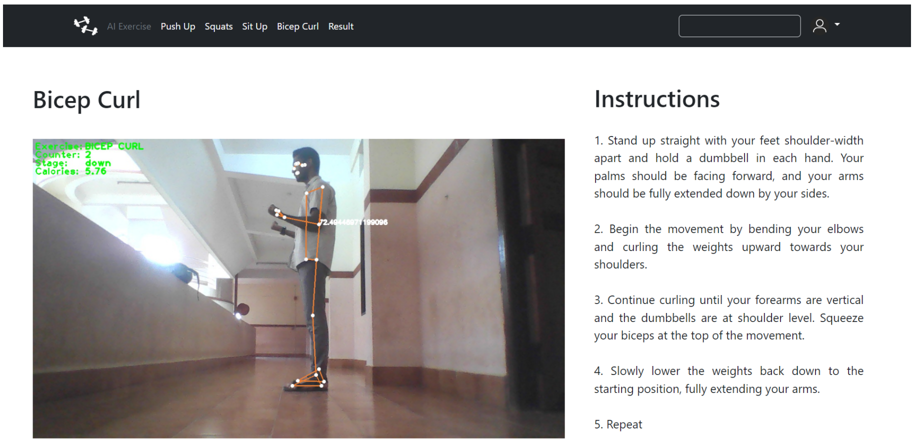

Fig. 10. Bicep Curl Page of the System.

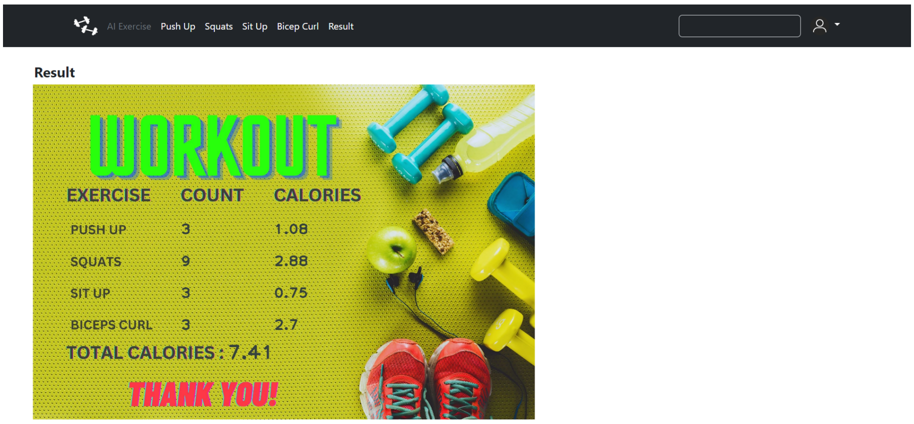

Fig. 11. Result Page of the System.

6 CONCLUSION AND SCOPE OF FUTURE WORK

6.1 Conclusion

In this project, an exercise monitoring system was developed to provide accurate and personalized feedback on exercise form and technique. By leveraging computer vision techniques and machine learning algorithms, the system analyzes exercise poses in real-time, offering valuable insights to users.The exercise monitoring system offers accurate and personalized feedback, empowering individuals to optimize their workout routines and achieve their fitness objectives.

6.2 Future Scope

In terms of unborn compass, the exercise monitoring system developed in this design holds immense eventuality for farther advancements and expansions. originally, the system can be extended to include a broader range of exercises, accommodating colorful fitness rules and preferences. This expansion would enable druggies to admit comprehensive monitoring and feedback for a wide array ofexercises.The integration of the exercise monitoring system with wearable bias, similar as smartwatches or fitness trackers, presents another instigative avenue. By connecting the system directly to these bias, druggies can admit real- time feedback and track their exercises seamlessly, enhancing convenience and availability.

References

1. Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. "OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields." 30 May 2019. https://ieeexplore.ieee.org/document/8765346
2. Rutuja Gajbhiye, Snehal Jarag, Pooja Gaikwad, Shweta Koparde. "AI Human Pose Estimation: Yoga Pose Detection and Correction." 412101, Ravet, Pune, Maharashtra. Volume 7, Issue 5, May 2022. https://ijisrt.com/assets/upload/files/IJISRT22MAY619.pdf
3. Anuj Patil, Darshan Rao, Kaustubh Utturwar, Tejas Shelke, and Ekta Sarda. "Body Posture Detection and Motion Tracking using AI for Medical Exercises and Recommendation System." ITM Web of Conferences 44, 03043 (2022).
4. U. Rafi, B. Leibe, J.Gall, and I. Kostrikov, "An efficient convolutional network for human pose estimation", British Mach. Vision Conf ., 2016.
5. S. Yadav, A. Singh, A. Gupta, and J. Raheja, "Realtime yoga recognition using deep learning", Neural Comput. and Appl., May 2019. [Online]. Available: https://doi.org/10.1007/s00521-019-04232-7
6. S. Haque, A. Rabby, M. Laboni, N. Neehal, and S. Hossain, "ExNET: deep neural network for exercise pose detection", Recent Trends in Image Process. and Patter Recog., 2019.
7. Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, Liang-Chieh Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks" 2019.

8. Hongyi Xu, Eduard Gabriel Bazavan, Andrei Zanfir, William T. Freeman, Rahul Sukthankar, Cristian Sminchisescu, "GHUM & GHUML: Generative 3D Human Shape and Articulated Pose Models" 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).
9. Valentin Bazarevsky, Ivan Grishchenko, Karthik Raveendran, Tyler Zhu, Fan Zhang, Matthias Grundmann, "BlazePose: On-device Real-time Body Pose tracking" 2020.
10. Conner, C., & Poor, G. M. (2016, May). Correcting exercise form using body tracking. In Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems (pp.30283034).
11. Andriluka, Mykhaylo, Leonid Pishchulin, Peter V. Gehler and Bernt Schiele. "2D Human Pose Estimation: New Benchmark and State of the Art Analysis." 2014 IEEE Conference on Computer Vision and Pattern Recognition (2014): 3686-3693
12. A. Gupta, T. Chen, F. Chen, and D. Kimber, "Systems and methods for human body pose estimation", U.S. patent, 7,925,081 B2, 2011.
13. Babu, Sudharshan Chandra. "A 2019 Guide to Human Pose Estimation with Deep Learning." AI & Machine Learning Blog. AI & Machine Learning Blog, August 5, 2019. https://nanonets.com/blog/human-pose-estimation-2d-guide/.
14. Voth, E. C., Oelke, N. D., & Jung, M. E. (2016). A theory-based exercise app to enhance exercise adherence: a pilot study. JMIR mHealth and uHealth, 4(2), e4997
15. https://developers.google.com/mediapipe/solutions/vision/pose_landmarker