AN INTELLIGENT COMPUTER APPLICATION TO CORRECT POSTURE IN EXERCISES USING HUMAN MOTION TRACKING

Zhiyao Zha[1], Jonathan Thamrun[2]

[1]Los Osos High School, 6001 Milliken Ave, Rancho Cucamonga, CA 91737
[2]Computer Science Department, California State Polytechnic University, Pomona, CA 91768

ABSTRACT

Our solution addresses the widespread issue of inefficient posture during exercise, a common obstacle many individuals encounter, including myself and those around me. Proper posture during workouts is crucial as it not only enhances the effectiveness of the exercise but also minimizes the risk of injuries associated with incorrect form [1]. Ensuring optimal posture leads to a safer and more productive exercise experience, maximizing benefits and reducing the likelihood of harm over time. To tackle this challenge, the program employs a method of tracking the user's form across various exercises, providing tailored feedback for improvement [2]. This approach proves more effective than traditional methods such as referencing static pictures or videos, as it offers direct, personalized guidance that can adapt to the user's specific needs and progress, thereby improving exercise quality and safety.

KEYWORDS

Intelligent, Computer, Posture, Exercises

1. INTRODUCTION

This program attempts to solve the problem of inefficient posture in specific exercises. There have been many times in my life where I and the people around me have struggled with achieving optimal posture in exercises. Making sure your posture is correct during workouts ensures that you are getting the most efficient and beneficial exercising experience all while lowering the many injury risks that come with bad posture [3]. Focusing on proper form when exercising results in an overall safer and more effective exercise which can maximize benefits and minimize injuries overtime.

My method to solving this problem is through tracking the form of the user in particular exercises that can potentially benefit most when being analyzed from a side-view relative to the camera and giving tips for improvement [4]. This is an effective solution as it allows the user to get direct and personalized feedback instead of having to try and reference things like pictures or videos which can often be hard to properly follow and adjust accordingly.

In the first experiment, I attempted to test the angle analysis system to determine if it worked as intended. It was important to see that the correct posture was actually registered and detected as "correct" by the program. In this experiment, the program was able to successfully determine the

David C. Wyld et al. (Eds): DMSE, CSEIT, NeTCoM, SPM, CIoT, NCS, NLPD, ArIT, CMLA – 2024
pp. 191-198, 2024. CS & IT - CSCP 2024 DOI: 10.5121/csit.2024.141415

correct posture as “correct” and the incorrect posture as “incorrect.” The results came out the way it did due to the predetermined angle values that were put into the code of the analyzer, which were tested to make sure that the angles corresponded to the generally correct posture for each exercise. In the second experiment, I attempted to test if having a second arm in the analysis would cause a disturbance to the analyzer that would negatively affect the accuracy of the results. The finding was that yes, the second arm did actually impact the analysis in a negative way, making me realize a flaw in the current program and the CV2 library where accuracy in the analysis can be compromised by factors such as the addition of a second arm and differences in the angle of the user relative to the camera that is used for the analysis. These results came about due to the analyzer having trouble differentiating one arm from another in a side view.

2. CHALLENGES

In order to build the project, a few challenges have been identified as follows.

2.1. The Information

The component of my program that takes in information about the exercise through the camera has been filled with problems regarding accuracy. There are a lot more potential problems that have occurred such as camera quality, distance between the camera and the user, the different proportions of the user, the lighting difference, camera angle, and camera distortion. To mitigate potential problems I could use an extra calibration step so that the program can adjust to all of these factors and assess the video based on each camera’s different resolution and each user’s body proportions. The other problems could be fixed through a guided alignment step of the camera.

2.2. Feedback

The component that gives feedback based on how well the user is performing the exercise has been another point of numerous errors. The current feedback system is quite basic and can cause the problem of not providing information specific enough to most effectively help the user fix their posture. This could be improved by having the feedback code examinewhich body part is at the incorrect angle and then giving specific feedback to the user to tell them to adjust their body accordingly.

2.3. GUI

The GUI, being the component that makes up the interface of the program, has caused problems that involved issues with getting the analysis tab to work with the correct resolution and ease of use for the user. As I plan to add more exercises into this app the GUI will also need to include more options in a way that is easy to use and navigate, while also being visually pleasing. I could improve visual clarity while maintaining ease of use through the organization of the exercises based on targeted muscle groups, and utilizing things such as dropdown menus or some alternative way of selecting exercises. I would also have to make sure the analysis tab works properly regardless of the different camera resolutions that could be inputted.

3. SOLUTION

The program follows a very simple pattern and is designed to be as easy to use as possible. It starts with the splash screen on launch, then transitions to the home screen where the selectable exercises are, and finally moves to the analyzer when an exercise is selected where you can then start the analyzer or return to the home screen [5]. The three major components that link my program together are the video analyzer, GUI, and the front-facing camera. Looking at how the program flows, everything starts in the home screen, where there are currently 4 selectable exercises, which are as follows: push-ups, planks, downward-facing dogs, and bicep curls. These are all clickable, which once clicked, will bring you to the analyzer screen. In the analyzer screen, there is a main, large video analyzer that will utilize and display the user′s front-facing camera to track the user′s movements in their selected exercises [6]. The video analysis will begin once the start button is pressed at the bottom left of the window. This window can be exited out where you will then be brought back to the home screen.

To make this program I used Python in Visual Studio Code. The core of the entire program comes from CV2, which operates the video analysis component; CV2 is from the OpenCV library and essentially allows for the tracking of various points on a human body which is achieved through artificial intelligence and machine learning. The other vital component of my program is tkinkter, which is used to make the GUI(Graphical User Interface) of the program.

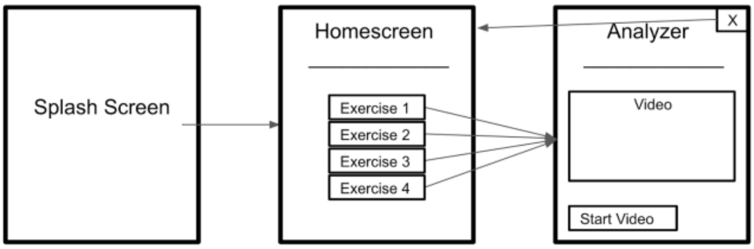

Figure 1. Overview of the solution

The video analysis component utilizes CV2 which serves as the primary processing system for the analysis [7]. CV2 can detect various vital points on the human body, such as the shoulder, elbow, knee, etc. which makes the measuring of angles between 3 points possible through predetermined vital points and angles. This tracking of vital points and taking in the angles is the core system which the program revolves around.

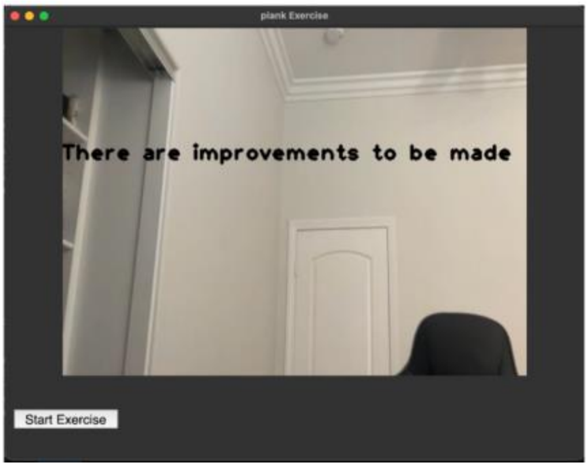

Figure 2. The video analysis component

The screenshot shows the window for the video analysis function which takes in a video input from the user that is then being fed into the program. The code for the video analysis utilizes the

CV2 library; I specify the angles between the various points on the body that are tracked which are used to determine whether or not an exercise is done correctly. This code runs after you select and click on an exercise, enable the usage of the front-facing camera, and start the analysis. There are multiple variables in the video analysis function, such as variables for each of the different measured angles in the code and variables for feedback messages based on the results of the analysis.

Another component is tkinkter, which is what creates the GUI for this program [8]. Tkinkter is a Python library and is used widely for the creation of interfaces. In my program, the main interface elements include the screen, titles, buttons, images, labels, and canvas(webcam feed).

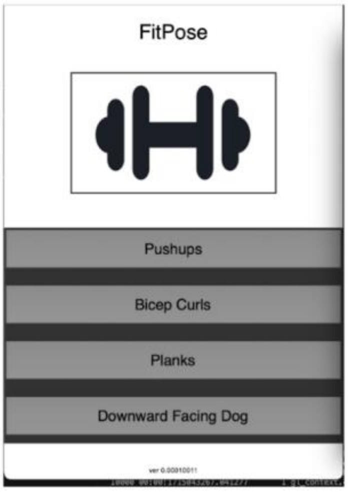

Figure 3. Screenshot of Fitpose

The webcam is another key component that is required for this program to work. The webcam is what takes information from the user and creates a video to be analyzed. The webcam is accessed through Tkinkter by using a canvas for the video feed. Resolutions and framerates are adjusted before the analysis is performed.

4. EXPERIMENT

4.1. Experiment 1

The angle between several body points is used to determine whether a plank is done correctly or not. It is key that this part works well as it is the central component of the program.

To test the accuracy of the analyzer, I will run the program and analyze the plank exercise with myself as the test subject. I will attempt to form a proper plank and see how accurately I can perform the exercise. I will use angle numbers for the analysis that are generally considered “correct” for a plank exercise. I will then perform a plank that should be considered to contain “inproper” form. It should also be noted that the rotation of the body does not affect the accuracy of the analysis, meaning that the plank form can be properly assessed without having to perform the plank on the ground, hence why the planks for this experiment are performed sitting up.

Figure 4. Figure of experiment 1

Looking at the results of the experiment, when I held out a plank position with a generally correct posture, the program was able to detect that posture as “good form” while when I held out a generally bad form, the program detected that the posture had “improvements to be made”. This is what I expected to happen as this is the intended result of the program and shows that the program should be generally accurate, else there would be a problem somewhere in the analysis code. What most contributed to this result is the specific angle values that were given to the analyzer, where angles for generally correct forms were given and used to detect correct vs incorrect posture.

4.2. Experiment 2

There could be problems when the AI tracks one arm but the other arm being shown in the video starts messing with the AI. This is part of the accuracy of the AI analysis [9].

To test this, I will run the program and see the difference between doing an exercise with a single arm vs both arms. This is done to see if the use of two visible arms instead of a single arm will have a significant impact on the results of the analysis. This setup can see if the program consistently allows for exercises to be performed with both arms while keeping accuracy, or if another factor such as rotation of the arms in relation to the camera is affecting the analysis.

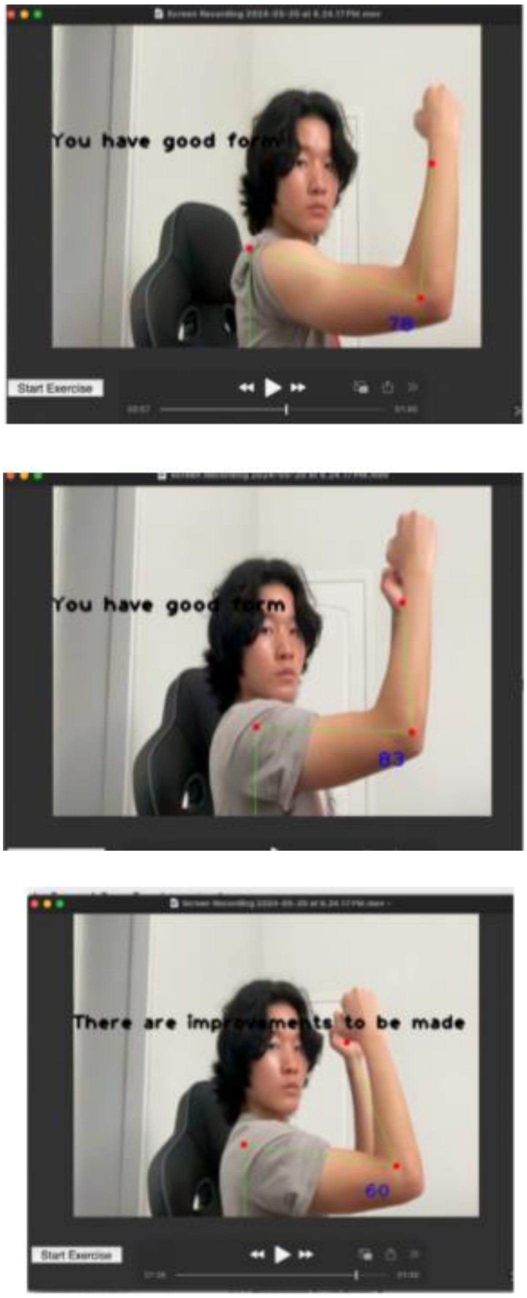

Figure 5. Figure of experiment 2

Looking at the results of the experiment, it is clear that using both arms in an exercise does have an impact on the results of the analysis. In the one-armed test, it was the most consistently accurate analysis of all. Adding a second hand has caused the analysis to wrongly assign points to the wrong hands, resulting in inaccurate analysis [10]. This problem is emphasized when using the program live, as you can easily see the program occasionally struggles to determine which arm is which. This creates a problem for the accuracy of the analysis and makes it so that, in order to maximize the accuracy of the program, either the use of the program should be limited to a single arm for the most consistent results, or move both your arms in a way where the arm not being analyzed is perfectly covered by the one that is being analyzed. Note that the program can still give accurate results with the use of two arms, however, the analysis can become jittery in its vital point tracking and give more inconsistent results.

5. RELATED WORK

This first solution uses a program called YOLO(You Only Look Once) which is a ″real-time object detection system" built from deep learning algorithms [11]. A shortcoming of this solution comes from the fact that the effectiveness and accuracy of the analysis largely depend on the complexity of the input material. This means that that simple exercises work best and produce the most accurate results while the more complex exercises that involve extra movements of the entire body will experience greater difficulty in tracking with sufficient accuracy.

This second solution uses a program called OpenPose to detect and analyze the user's pose. It tracks key points on the human body such as joints and hands and uses its estimation algorithms to evaluate the input video source[12]. The model is based on a database of over 100 exercise videos using their ″geometric-heuristic and machine learning algorithms." A limitation of this method is the requirements for the video camera placement and positioning of the user relative to the camera, often needing quite a long distance between the user and the camera, which can make it hard for some users to use the correction tool properly and most effectively and accurately.

This third solution utilizes a pose detection program from Mediapipe, which analyzes and corrects exercise poses based on multiple set points on critical parts of the body [13]. This program is largely effective in giving accurate results and providing proper feedback, as long as it is given proper camera alignment and positioning relative to the user. The main limitation is that there are requirements for camera positioning and that there are 4 available exercises that can be analyzed.

6. CONCLUSIONS

Some of the current limitations of the project include the AI software capabilities, the limited amount of exercise options, the limited personalized feedback, the need for the user to be positioned with their side facing the camera, and my overall knowledge of Python. There have been a few issues with Python software updates and the functionality of the program, where updates would cause the code to not run properly. A part of the program that could be fixed and improved on is the implementation of AI [14]. This includes the angles of measurement in which an exercise is counted as correct. I will try to introduce a calibration step into the program to adjust for possible different camera resolutions and to make sure the camera is positioned at a suitable angle. I will also attempt to improve on the lacking feedback function so that it offers more specific, helpful, and personalized instructions on how the user′s posture can be improved.
This project started from a random idea I had when I was struggling with my posture in an exercise and it has been quite a journey to attempt and bring that idea to life [15]. There are still many things I would like to improve on overall, but this project has improved my knowledge of coding by large margins and has been an amazing experience overall.

REFERENCES

[1] Chae, Han Joo, et al. "An Artificial Intelligence Exercise Coaching Mobile App: Development and Randomized Controlled Trial to Verify Its Effectiveness in Posture Correction." Interactive Journal of Medical Research 12.1 (2023): e37604.

[2] Polero, Patricia, et al. "Physical activity recommendations during COVID-19: narrative review." International journal of environmental research and public health 18.1 (2021): 65.

[3] Kotte, Hitesh, Miloš Kravčík, and Nghia Duong-Trung. "Real-Time Posture Correction in Gym Exercises: A Computer Vision-Based Approach for Performance Analysis, Error Classification and Feedback. " (2023).

[4] Joo, Sun-Young, et al. "Feasibility and effectiveness of a motion tracking-based online fitness program for office workers." Healthcare. Vol. 9. No. 5. MDPI, 2021.

[5] Chen, Steven, and Richard R. Yang. "Pose trainer: correcting exercise posture using pose estimation." arXiv preprint arXiv:2006.11718 (2020).

[6] Dhanish, KJ Mohamed, et al. "USABILITY STUDY ON VIRTUAL FITNESS APPLICATION USING ARTIFICIAL INTELLIGENCE AND DEEP LEARNING."

[7] Sonawane, Mr Rutvik, et al. "Fitness Trainer Application Using Artificial Intelligence."

[8] Novatchkov, Hristo, and Arnold Baca. "Artificial intelligence in sports on the example of weight training." Journal of sports science & medicine 12.1 (2013): 27.

[9] Venkatachalam, Parvathy, and Sanjog Ray. "How do context-aware artificial intelligence algorithms used in fitness recommender systems? A literature review and research agenda." SInternational Journal of Information Management Data Insights 2.2 (2022): 100139.

[10] Challagundla, Yagnesh, et al. "A Multi-Model Machine Learning Approach for Monitoring Calories Being Burnt During Workouts Using Smart Calorie Tracer." EAI Endorsed Transactions on Pervasive Health and Technology 10 (2024).

[11] Long, Warren, and Yee-Hong Yang. "Log-tracker: an attribute-based approach to tracking human body motion." International Journal of Pattern Recognition and Artificial Intelligence 5.03 (1991): 439-458.

[12] Faisal, Abu Ilius, et al. "Monitoring methods of human body joints: State-of-the-art and research challenges." Sensors 19.11 (2019): 2629.

[13] Morasso, Pietro, and Vincenzo Tagliasco, eds. Human Movement Understanding: from computational geometry to artificial intelligence. Elsevier, 1986.

[14] Lu, Lei, et al. "Effective assessments of a short-duration poor posture on upper limb muscle fatigue before physical exercise." Frontiers in Physiology 11 (2020): 541974.

[15] Hung, Jui-Sheng, Pin-Ling Liu, and Chien-Chi Chang. "A deep learning-based approach for human posture classification." Proceedings of the 2020 2nd International Conference on Management Science and Industrial Engineering. 2020.

[16] Hung, J.-S., Liu, P.-L., & Chang, C.-C. (2020). A Deep Learning-based Approach for Human Posture Classification.