Aivisiontrain: Artificial Intelligence Fitness Trainer

Swaraj Singh Mandley(s5501445)

Master Project

Bournemouth University

M.Sc. Artificial Intelligence for Media

August 14,2023

Bournemouth
University

Acknowledgement

I would like to express my deepest gratitude to my supervisor, Professor Hammadi Nait-Charif, for his unwavering support and guidance throughout the duration of this program. His invaluable suggestions, feedback, and constructive criticism have played a pivotal role in shaping my work. His expertise, patience, and commitment have been instrumental in my academic journey, and for that, I am profoundly thankful.

Table of Contents

Abstract

Integrating artificial intelligence (AI) with fitness training has revolutionized how individuals approach exercise routines. This thesis delves into the creation of Aivisiontrain, an AI-powered fitness trainer that addresses fitness-related inquiries and monitors real-time exercise performance. The implementation involves utilizing Python code to build a chatbot interfaced with OpenAI API for fitness inquiries and employing MediaPipe's library to monitor exercise execution through webcam analysis. The system is further extended to provide feedback and guidance during workout sessions. The Aivisiontrain potential for improving exercise accuracy, providing personalized routines, and promoting healthier lifestyles is explored.

1.Introduction

1.1. Background and Motivation

The fitness industry has undergone numerous transformations, with technological advancements playing a pivotal role. Traditional one-on-one training has been supplemented and, in some cases, even supplanted by digital tools and platforms, offering users personalized workouts, dietary advice, and progress tracking. However, as the ubiquity of Artificial Intelligence (AI) grows, there is an opportunity to further harness its potential in creating a more interactive and responsive fitness experience. The idea of an AI-Fitness trainer empowered to provide answers to fitness-related queries and monitor and assist in exercises actively has significant merit. This is where the blend of advanced conversational models like the Large Language Model (LLM) with human pose estimation can revolutionize personal training. By developing a Python-based chatbot capable of providing expert advice and monitoring exercise techniques in real-time, we offer a personalized training experience and create a paradigm where digital fitness interaction becomes as intuitive and responsive as a human trainer. The motivation is to merge the realms of digital information with physical execution, ensuring that users know what to do and are guided and monitored effectively when they do it.

1.2. Understanding the Reluctance to Exercise

Exercise, often dubbed a "miracle cure," has been long acknowledged as a critical component to achieving and maintaining optimal health. Being physically active offers many health benefits both in the short and long term. To begin with, regular physical activity significantly improves brain health, helping in the management of weight, reduction in the risk of various diseases, strengthening bones and muscles, and enhancing the ability to perform everyday tasks. Immediate advantages after a session of moderate-to-vigorous physical activity can include improved cognition in children aged 6 to 13 years and a notable decrease in feelings of anxiety among adults. Furthermore, being physically active consistently can help maintain cognitive skills, reduce the risks of depression and anxiety, and even improve sleep quality (CDC 2023).

Additionally, exercise can reduce the risk of significant health issues, including coronary heart disease, type 2 diabetes, stroke, and various cancers, ultimately lowering the risk of premature death by up to 30%. Beyond just the physical aspects, regular exercise has been scientifically proven to enhance self-esteem, mood, energy levels, and overall mental well-being. It plays a pivotal role in alleviating the risks associated with stress, clinical depression, dementia, and Alzheimer's disease. (NHS 2023)

In studies, it is found that people often face numerous challenges when trying to integrate regular physical activity into their daily lives. A prominent reason cited is the lack of time, often stemming from demanding job schedules, family obligations, and other day-to-day commitments. Moreover, many individuals cite inconvenience, a lack of self-motivation, and an absence of enjoyment as reasons for avoiding exercise. Others express a sense of boredom with repetitive exercise routines or feel they lack confidence in their ability to engage in physical activities. This low self-efficacy is sometimes coupled with the fear of injury or being hindered by a recent injury (CDC 2022).

Additionally, some individuals may need self-management skills to set, monitor, and reward progress toward personal fitness goals. External factors also play a role. Many highlight the absence of supportive infrastructure such as nearby parks, sidewalks, bicycle trails, or safe walking paths close to their homes and workplaces. Social dynamics can also hinder; family and friends' lack of encouragement, support, or companionship can diminish motivation. (American Heart Association 2023)

Recognizing and understanding these barriers can aid individuals in creating strategies to overcome them and make physical activity a regular part of their daily life.

1.3. Problem Statement

While AI's potential in fitness is widely recognised, there is a need for a holistic system that provides textual guidance and offers real-time feedback on physical exercises. Currently, there need to be more comprehensive tools that combine the capabilities of chatbots with real-time physical activity monitoring.

1.4. Objectives

The core objective of this thesis is to design and implement an AI-driven fitness trainer (Aivisiontrain) utilising Python. This system will possess the following capabilities:

User Interaction: Aivisiontrain can engage users, addressing their questions about exercises, techniques, and fitness principles. This interactivity will be grounded in an extensive database dedicated to physical training knowledge.

Real-time Analysis via Computer Vision: Leveraging computer vision, Aivisiontrain can detect and analyse the user's exercise form through human pose estimation techniques. Upon user request, the system will access the webcam to oversee the exercise, offering instant feedback on form and counting repetitions in real time. Significantly, all user detection will be strictly based on computer vision; we will not employ wearable devices like smartwatches and fitness trackers, chest Straps, clip-on Devices, headphones or any IoT devices to track exercises.

24/7 Availability: The AI fitness trainer ensures consistent guidance around the clock. Users can rely on uninterrupted assistance Whether they prefer daytime or nighttime workouts. The embedded chatbot operates without breaks to address user queries at any time of day.

In sum, the overarching aspiration of this project is to amalgamate the information-rich characteristics of a chatbot with the dynamic analysis offered by human pose estimation, crafting a comprehensive AI-centric fitness tool. By doing so, we aim to enhance accessibility, interactivity, and personalisation in fitness training, tackling obstacles that might deter individuals from maintaining consistent physical activity.

2. Literature Review

AI-driven chatbots, the analysis of human posture, and computer vision techniques have historically been employed to monitor and recognize physical activity. Over the years, there have been numerous investigations into the categorization and tracking of human movements. Presently, multiple software solutions specialize in posture analysis which can be utilized to monitor individuals engaged in various activities. In the subsequent section, we will delve into previous research in this area and provide a critical evaluation.

In the field of AI-driven fitness chatbots, (Zhang et al. 2020) did research that yielded 108 articles from four databases. A breakdown by year showed: 15 articles from 2020, 26 from 2019, 15 from 2018, 14 from 2017, five from 2016, and 33 from 2015 or earlier. After a thorough review, 101 articles (93.5%) were discarded. The main reasons for exclusion where the articles were commentary or opinion pieces, scoping reviews, or they addressed health domains unrelated to physical activity and diet. For instance, some chatbots in these articles aided in diagnostic tasks or provided mental health interventions or treatment.

From the remaining articles, three chatbots stood out, Tess (Tess 2023). This behavioural coaching chatbot was designed to help adolescent patients manage weight and prediabetes symptoms. Patients found the interaction with Tess meaningful, witnessing positive strides toward their objectives. They found Tess beneficial in their journey; Paola (Anubbe 2023), highlighted in a proof-of-concept study, provided users with educational insights on physical exercise and diet. It also conducted weekly follow-ups and addressed user inquiries. Users of this chatbot experienced significant weight reduction and enhancements in their diet. CoachAI (Coachai 2023): Detailed in a validation study, CoachAI provided personalized and social health coaching. This chatbot proved to be particularly impactful for highly engaged users.

(Zheng et al. 2020) presented a system that utilizes a variety of machine learning and deep learning methodologies to identify activities related to sports equipment in modern gym settings. They employed Decision Trees, Linear SVM, and Gaussian-kernel Naive Bayes as machine learning models. For detecting exercise recognition (DER), they introduced a Deep Neural Network comprised of three hidden layers, each containing 150 LSTM cells, achieving an accuracy of 92%. A significant challenge they faced was the overlapping recognition of exercises targeting similar muscle groups, such as the Fly and Rear Delt and the Pull Down, Lat Pull, and Overhead Press.

(Taware et al. 2021) developed a method that uses pose estimation analysis output as input for an LSTM classifier, which they encapsulated in a toolkit named Action-AI. For video processing, Open-CV was adequate. They employed Open-pose for pose estimation, integrated with popular deep learning platforms like TensorFlow and PyTorch. Users can control yoga sessions with voice commands using the Snips AIR voice assistant, allowing them to start, stop, pause, or restart the activity. The model demonstrated high accuracy. However, it faced challenges in specific scenarios where the angles remained unchanged even when vital points were rotated.

From prior models, it became evident that employing OpenCV, identifying human body key points, and utilizing the COCO dataset led to impressive accuracy levels. Building on this, (Pardeshi et al. 2022) monitored the number of exercises performed by users and identified inaccuracies in yoga postures. Their approach involved the primary CNN network, the OpenPose Python package, and the COCO datasets. While the model showcased commendable accuracy and performance, it lacked variations based on different age groups, failing to distinguish between categories such as men, women, and children.

Recently, (Kwon 2023) developed a program that evaluates the posture in real-time images to guide the training content, such as squats and push-ups, and count the number of workouts using OpenCV and MediaPipe. MediaPipe facilitates real-time analysis, and this model yielded remarkable accuracy and performance. However, the model's limitation is its absence of real-time user support for queries. Additionally, users might feel demotivated when exercising alone.

Infigro (infiGro 2023) is one of the few apps that harness computer vision, accessible directly from your mobile device. Designed to offer real-time coaching and feedback, it swiftly analyzes user performance and suggests actionable improvements. Beyond its advanced training functionalities, Infigro stands out with its unique mental health module, dietary guidance, and meditation techniques.

Together, these works underline the growing potential of leveraging AI, specifically Python-based solutions, to revolutionize the realm of fitness training. Integrating AI chatbots, human pose estimation, and advanced computer vision techniques paints a promising future for personalized and interactive fitness experiences.

3. Methodology

3.1. Pose Estimation

I employed the MediaPipe (Mediapipe 2023) framework for human pose estimation to extract critical information regarding the user's posture. MediaPipe provides a sophisticated approach to detecting key body landmarks. When the user's input is processed through the MediaPipe library, it discerns the significant keypoints on the body, presenting the results as coordinates along the X, Y, and Z axes for 33 primary landmarks. These pinpointed locations elucidate the positioning of major body parts within the provided image. By leveraging these keypoints, it's feasible to recreate an intricate representation of the user's skeletal structure.

For a clearer understanding, refer to Figure 1 where these landmarks, often known as "landmarks," represent major junctions and regions on a human body. These are systematically indexed from 0 through 32, summing up to 33 distinct landmarks. For facial features, landmarks in the range of 0 to 10 are utilized. Landmarks spanning from 11 to 22 are designated for the upper body recognition, encapsulating areas like shoulders, wrists, elbows, and hands. The concluding set of landmarks, from 23 to 32, pinpoints the lower body, comprising of regions such as the hips, knees, legs, and feet.

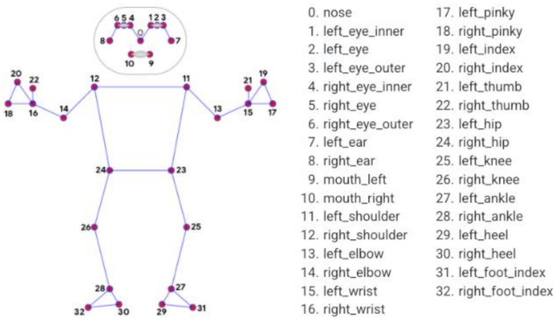


Figure 1: Mediapipe Landmarks (Mediapipe 2023)

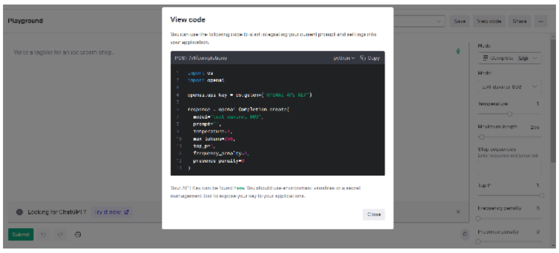
MediaPipe, an open-source and multi-platform library, delivers tailor-made machine learning solutions optimized for real-time streaming media—this encompasses a spectrum of media from audio, videos, to sequential data. Compatible across a wide array of platforms, including iOS, Python, JavaScript, and Android, MediaPipe offers a plethora of tools and solutions. Its repertoire consists of features like pose estimation, hair segmentation, face mesh, and motion tracking.

The data that is fed into the MediaPipe library primarily originates from real-time camera captures. Post processing, the library yields a comprehensive list of corresponding keypoints in the Cartesian coordinates (X, Y, and Z). Harnessing these coordinates, one can glean a fairly accurate depiction of the human body's structure and orientation within any given image or live video feed. As stated in MediaPipe's (Mediapipe 2023) official documentation, it operates at a commendable frame rate of 30 frames per second.

3.2. Chatbot

I utilized the OpenAI API(OpenAI 2023) for my AI fitness chatbot, which permits developers to integrate the prowess of models like GPT-3 and GPT-4 into their platforms. The API facilitates communication with OpenAI-hosted models without the need to operate them on personal infrastructure. To interact with it, you typically use HTTP requests, authenticating with API keys provided by OpenAI.

I accessed the API through the OpenAI Playground, a platform crafted for users to experiment with and understand the API's workings. It's a versatile space, predominantly for tech R&D, permitting users to engage with machine learning models. With it, developers can test-drive models like GPT-3 and GPT-4, honing in on functionalities such as text generation and image classification. Figure 2, depicts the process of integrating OpenAI code from the Playground into a personal project; with the provided code and API key, anyone can seamlessly incorporate OpenAI functionalities.



View code

You can use the following code to start integrating your current prompt and settings into your application.

POST /v1/completions

```python
import os
import openai

openai.api_key = os.getenv("OPENAI_API_KEY")

response = openai.Completion.create(
  model="text-davinci-003",
  prompt="",
  temperature=1,
  max_tokens=256,
  top_p=1,
  frequency_penalty=0,
  presence_penalty=0
)
```

Your API Key can be found here. You should use environment variables or a secret management tool to expose your key to your applications.

Figure 2: OpenAI Playground (OpenAI platform 2023)

Discussing the models, GPT-3's "davinci" variant is built on the Transformer framework that uses attention mechanisms to discern input sequences' relevance. The GPT-3 model, packed with 175 billion parameters, stands as one of the grandest in the commercial domain. GPT-3 is trained on extensive web-based text, acquainting itself with grammar, facts, logic, and elements of general wisdom. It's further refined using specific data, often with human oversight, in line with OpenAI's guidelines (Iffort 2023).

For my chatbot, I employed text-davinci-003, with all responses powered by the "davinci" variant. Among GPT-3's offerings, text-davinci-003 is the frontrunner, surpassing models like text-curie-001 and text-babbage-001. It excels in generating refined, extended content, outperforming its precursor, Text-Davinci-002. Specifically crafted for tasks requiring adherence to instructions, text-davinci-003 is adept at producing precise, succinct outputs, even when presented with unfamiliar prompts (Openai 2023).

3.3. Computer Vision

OpenCV, an acronym for Open-Source Computer Vision Library, is a renowned library dedicated to real-time computer vision functions. It’s coded in a diverse set of languages like C, C++, Python, Java, and assembly. To bolster real-time capabilities, GPU acceleration was embedded into OpenCV from 2011 onwards. With a repertoire of over 2500 optimized algorithms, OpenCV stands as a comprehensive suite for both classical and cutting-edge computer vision and machine learning techniques. This encompasses features like face and object recognition, camera motion tracking, image merging, and much more. Tailored for real-time applications, OpenCV leverages MMX and SSE instructions when they're accessible. (OpenCV 2021)

4. Implementation

4.1. Building the Fitness Inquiry Chatbot

4.1.1. Speech recognition

Using SpeechRecognition (Python 2018), I've been able to harness the capability of modern computational systems to transform spoken language into text. First, when someone speaks into the device's microphone, that speech is picked up by the pyaudio library, which serves as a bridge to access audio data from audio input devices on the system. The audio data captured is then passed to the SpeechRecognition library, which, with the help of pyaudio, extracts the sound from the microphone. This raw audio data is sent to Google's Web Speech API (Google 2023) through the method recognize_google(). Powered by sophisticated machine learning algorithms, the API processes the audio, transcribing the speech into text. Once transcribed, the resulting text is received by the software I've developed, which then processes or acts based on what the user initially said.

4.1.2. User Experience

For an enhanced user experience, the Python script is crafted to prioritize usability and user satisfaction. Always on the lookout for instructions, the system springs to action when the user says "wake up", prepping the assistant to attend to the user's demands. Central to its functionality is the main_task_execution() function. This employs the SpeechRecognition library to convert voice prompts into text. To address accidental activations from previous versions, a distinct cue, "fitness", was introduced. Earlier, the system sometimes mistook unrelated chats for commands, giving unrelated responses. Now, only when "fitness" is recognized does the robust main_task_execution() function activate, processing a range of commands. From launching and closing apps, activating camera feed via OpenCV, to playing music from specific folders, it’s adept at multitasking. Users can also explore web commands, visiting popular platforms like YouTube and GitHub or fetching their IP address. The "bye" command offers a smooth exit, with the system expressing gratitude before awaiting its next use.

4.1.3. OpenAI API

The script artfully merges voice-recognition technology with OpenAI's GPT model, with the goal of creating a specialized voice assistant for fitness inquiries. At the heart of this system is the chat_with_openai() function. This enables an interactive dialogue where users can voice their questions. Once a query is posed, the system consults the OpenAI API to fetch pertinent answers, then vocally delivers this information back, ensuring fluid information exchange.

However, it was noted that the default OpenAI API responses were lengthy and tedious to listen to. To address this, the maximum token number was trimmed from its default of 256 down to 10. Moreover, the model's randomness hyperparameter was tweaked to match the desired chatbot output. A higher setting yields more unpredictable results, while a lower one offers more consistent replies.

A significant component is the is_fitness_keyword_present() function. It serves to examine user queries for fitness-specific terms. If such terms are detected, the system precisely tunes its responses to the fitness realm. Conversely, if a fitness keyword isn't found, or if a user inquiry about something outside the scope of fitness, the assistant will respond that it lacks information on the subject.

Enhancing the overall user experience is the generate_ai_response() function. If a user's query incorporates the term "make", this function prompts a response from the OpenAI API. The system then stores this response in a text file located in an "Openai" directory, highlighting the system's trust in the OpenAI API for intricate query solutions.

In summary, this meticulously designed script is a testament to the vast potential of OpenAI's API. By seamlessly blending cutting-edge voice technology with the robust capabilities of the GPT model, it signals the emergence of specialized voice assistants. In this case, the assistant offers specialized guidance in the comprehensive realm of fitness.

4.2. Real-time Exercise Monitoring and Counting

In our detailed examination of computer vision and exercise monitoring, we developed four unique Python scripts: PoseModule5, Bicep_Curls_Exercise, Bicep_Curls_Exercise, and Bicep_Curls_Exercise. The posemodule script primarily focuses on estimating human posture, while the other three are dedicated to monitoring specific exercises like bicep curls, jumping jacks, and squats. All these scripts harness the 'blazepose' functionality from the Mediapipe framework for pose estimation, seamlessly integrating with the OpenCV library to enable real-time video streaming from a webcam.

Inside posemodule.py, there are crucial class methods. The get_pose method processes an input image by converting it to RGB, identifying pose landmarks, and optionally annotating these landmarks with their connections on the image while displaying detection confidence. In contrast, the get_landmark_position method predicts all 33 keypoints, each with 3 attributes: x and y coordinates and visibility. Additionally, it considers two virtual alignments. (Figure 3) shows the landmarks with distinct circles, where attributes like color and size indicate the detection confidence of each landmark.

Adding to the script's functionality, it can calculate the frame rate (FPS) by measuring the time differences between consecutive frames. This comes with a handy overlay feature that displays essential metrics on the live video, such as the real-time FPS.

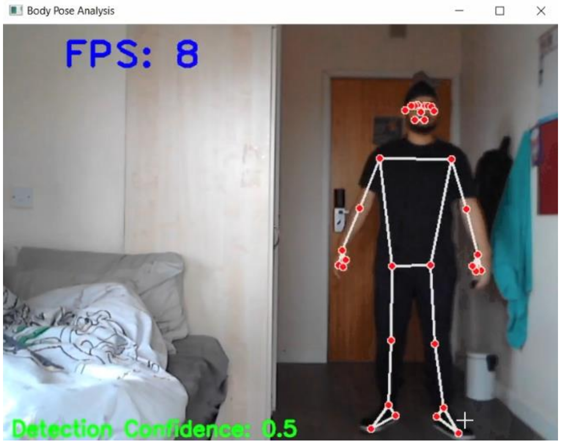


Figure 3: get_landmark_position method Detecting all 33 Keypoints

4.2.1. Calculating Angles

Initially, we determine the coordinates of the three essential joints to compute the angle. Using NumPy, we then deduce the slopes of these joints. While angles are primarily computed in radians, they can easily be converted into degrees.

The formula Angle = math.degrees(math.atan2(y3 - y2, x3 - x2) - math.atan2(y1 - y2, x1 - x2))

allows us to find the angle between two lines directly without the need to determine the slope. Depending on the joints selected, this method discerns the inter-joint angle. For my left bicep curl application, I focused on the elbow angle, as it's a critical indicator of the correct form for a left bicep curl.

4.2.2. Repetition Counting

To count the repetitions, the algorithm monitors the probability of a target pose class. When the probability of the “down” pose class passes a certain threshold for the first time, counter increases by one. Then successful transition of hand from down or up increases counter. Given the landmarks are detected, in the case of bicep curls script calculates the angle of the right arm using specific landmarks denoting the shoulder, elbow, and wrist. This angle is further interpolated to determine its percentage representation, which is instrumental in identifying the stage of the dumbbell curl. A bar visualization is rendered on the image to depict this percentage, with its height and color indicating the extent and stage of the curl respectively. A full curl is denoted by the angle reaching 100%, resulting in the color turning green and incrementing the repetition count (Figure 4.1). When the arm returns to the starting position (angle at 0%), the count is once again incremented, thus completing one full repetition(Figure 4.2).

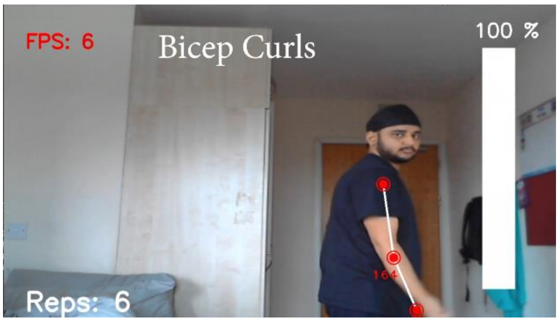


Figure 4.1 Bicep Curl Starting Position

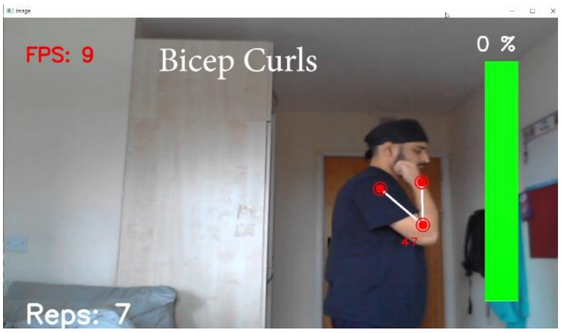


Figure 4.2 Bicep Curl End Position

4.2.3. Bicep curls

Bicep curls (Rogers 2007), sometimes referred to as arm curls or dumbbell curls, are a cornerstone exercise in weight training, targeting the biceps in the upper arm. The exercise is performed by standing upright with dumbbells in hand, palms facing outwards. Keeping the upper arms anchored to the side, the weights are raised by flexing the elbow, bringing the dumbbells towards the shoulders. In a full curl, the dumbbells might reach the level of the eyes or even the forehead. While the primary muscle worked is the biceps of the upper arm, the exercise also engages the brachialis and brachioradialis in the lower arm.

When the script runs, it activates the default camera, streaming live video and leveraging the PoseModule3 for pose detection. Within a consistent loop, the script captures individual frames, adjusting them to a consistent size of 1280x720 pixels. The pose detection mechanism then processes these frames to identify specific pose landmarks, which correspond to distinct anatomical points on the body.

In the case of bicep curls, the code intends to calculate the angle of the right arm, focusing on the elbow joint. The method calculate_angle of the pose_detector is then invoked with the image and landmarks of points 12, 14, and 16, these points represent the right shoulder, right elbow, and right wrist respectively.

The dynamics of the elbow angle during a bicep curl are central to the tracking process. As one engages in the exercise, the elbow's angle starts more open when the arm is straight and narrows as the arm curls. This fluctuating angle provides insights into the phase of the curl, facilitating the counting of repetitions. Rep counts are deduced from the angle's degree of closure and the direction of arm movement, whether upward or downward.

4.2.4. Squat

A squat is a core exercise that involves standing upright, then lowering the hips and returning to the starting position. During the descent, the hip and knee joints bend while the ankle moves upward; conversely, as one rises, these joints straighten, and the ankle moves downward. This movement engages primarily the quadriceps, adductor magnus, and gluteus maximus muscles. Moreover, the erector spinae and abdominal muscles provide stabilizing support. Squats are essential for developing lower body muscle endurance and mass and fortifying core stability. They're pivotal in powerlifting and numerous workout regimens(Ribeiro 2023).

The functions, get_right_leg_angle and get_left_leg_angle, procure body reference points using the get_landmark_positions function. These reference points, termed landmarks here, are distinct body markers recognized by the pose detector. For the right leg, landmarks 24, 26, and 28 correspond to the hip, knee, and ankle respectively.

After obtaining the landmarks, the angles of both legs are deduced with the calculate_angle function. These angles are imperative to determine the squat's phase (either rising or descending). In the process_image function, the angles from both legs are combined to deduce the cumulative squat angle, which helps in rectifying any disparity or imbalance in the user's motion. This combined angle is subsequently relayed to the draw_workout_info function.

The draw_workout_info function showcases the squat's progress numerically and through a visual bar, using the average leg angle to gauge this progression. Repetitions are also monitored by observing angle directional changes. If the mean angle indicates a full standing posture and the last movement was descending, it counts as half a repetition. Likewise, transitioning from standing to a squatting stance also accounts for half a repetition. Therefore, a complete squat (down then up) equals one full repetition, added in 0.5 increments.

The get_bar_color function modifies the progress bar's hue to green during the squat's peak points (fully upright or deep squat) and alters the direction indicator. For any in-between angles, the bar appears red. For each video frame, the pose is recognized, leg angles are determined, repetitions are counted, and the related data is superimposed on the video, ready for display.

4.2.5. Jumping Jacks

Jumping jacks are a classic full-body workout enhancing cardiovascular health. Starting with feet close and arms down, you jump and spread your feet past shoulder-width, raising arms overhead. This action reverses with another jump to the original pose. This exercise increases heart rate, fosters cardiovascular stamina, and works the legs, core, and arms. The synchronized limb movement improves coordination. As no special gear is needed, it's a handy workout for various locales. Those new to this or with health issues should be careful and might seek medical advice for safety (Baharuddin 2023).

The get_hand_position() method retrieves the vertical coordinates of both wrists, essential for gauging hand positions during the exercise. The get_ankle_distance() method calculates the gap between the ankles, indicating if the legs are spread or together.

The core logic for repetition counting is in the detect_jump_repetition() method. This method checks both wrist positions and ankle spacing to identify the jumping jack's stage. If ankle separation surpasses 150 (showing legs are wide) and both wrists' vertical positions are below 300 (hands raised), it marks the jumping jack's initial phase. In contrast, if ankles are closer than 100 (legs close) and both wrist positions exceed 400 (hands lowered), the jumping jack concludes one round, increasing the repetition tally.

Lastly, the draw_workout_info() method superimposes the exercise's details onto the image. This includes an ankle gap visual, total repetition number, and cues on the current jumping jack phase ("Jump!" or "Keep Going!"). The system leverages body pointers, like wrist vertical positions and ankle spacing, to recognize the exercise's stages and keep a repetition tally.

4.3. How did combined OpenAi and Mediapose

Using voice commands to run and then halt a Python file was a challenge I faced. After deep investigation, I stumbled upon the notion of Process ID (PID) within the Windows environment (DOMARS 2023). Leveraging this knowledge, I developed the following solution:

The program listens for specific voice prompts like "Exercise One" for bicep curls, "Exercise Two" for squats, and "Exercise Three" for jumping jacks.

Upon hearing "Exercise One", and if no other exercise script is currently active, it launches the Python script at file location. The script's Process ID (PID) is then saved for future reference. If the exercise script is already active, the user is alerted with "Gym exercise is already open."

However, if the commands "close one", "close two", or "close three" are recognized and the associated exercise script is running, that process is ended. For example, the command "close one" prompts a user notification: "Closing Bicep curls exercise".

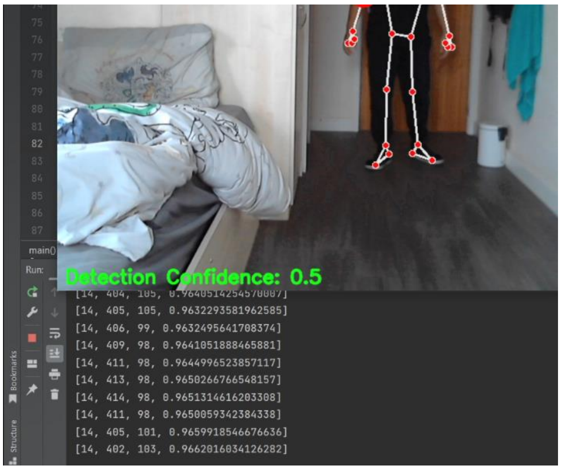
This solution provides an intuitive and effective way to manage a Python application using vocal cues.

5. Results and Discussion

5.1. Pose Estimation Accuracy

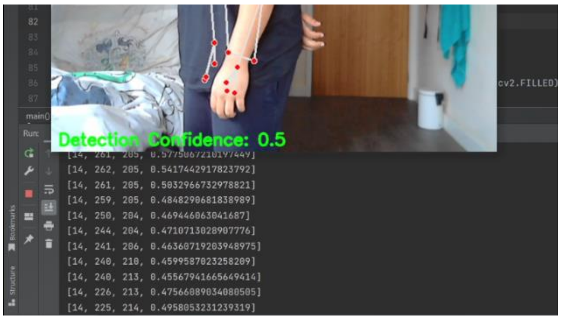
In the process of rigorously testing MediaPipe's pose detection for the right elbow landmark, we identified a clear correlation between detection accuracy and the subject's distance from the camera. When subjects were positioned at an optimal distance, the model exhibited a robust accuracy range of 90% to 95%(Figure5.1), reflecting its high reliability. In contrast, when subjects were too close to the camera, accuracy sharply declined to between 40% and 50%(Figure5.2). This significant variation underscores the model's sensitivity to the subject's positioning relative to the camera.

The min_detection_confidence parameter plays a crucial role in this process. It sets the baseline for pose detection, meaning that if it's set to 0.5, only poses the model has at least 50% confidence in will be recognized. In our tests, a confidence level of 0.5 was achieved at the optimal distance, while nearly no confidence was observed when too close to the camera.



```
[14, 404, 105, 0.9640514254570007]
[14, 405, 105, 0.9632293581962585]
[14, 406, 99, 0.9632495641708374]
[14, 409, 98, 0.9641051888465881]
[14, 411, 98, 0.9644996523857117]
[14, 413, 98, 0.9650266766548157]
[14, 414, 98, 0.9651314616203308]
[14, 411, 98, 0.9650059342384338]
[14, 405, 101, 0.9659918546676636]
[14, 402, 103, 0.9662016034126282]
```

Figure5.1 Accuracy range of 90% to 95% when far from camera



```
[14, 261, 205, 0.5775067210197449]
[14, 262, 205, 0.5417442917823792]
[14, 261, 205, 0.5032966732978821]
[14, 259, 205, 0.4848290681838989]
[14, 250, 204, 0.469446063041687]
[14, 244, 204, 0.4710713028907776]
[14, 241, 206, 0.4636071920394897 5]
[14, 240, 210, 0.4599587023258209]
[14, 240, 213, 0.4556794166564941 4]
[14, 226, 213, 0.4756608903408050 5]
[14, 225, 214, 0.4958053231239319]
```

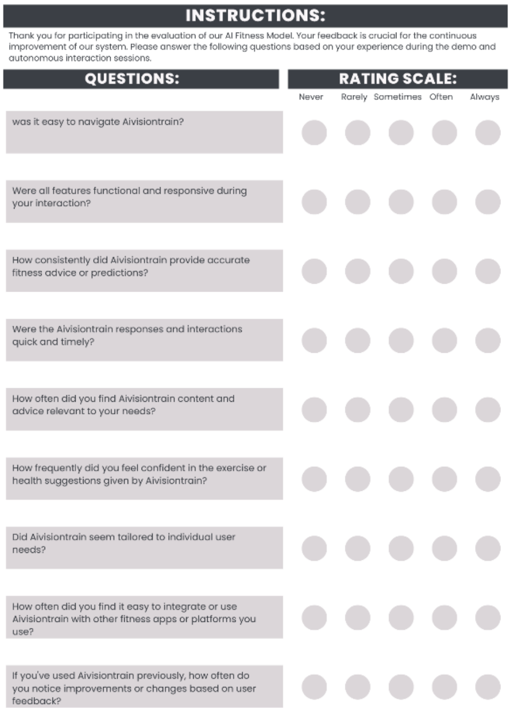
Figure5.2 Accuracy range of 40% and 50% when near to camera

It's essential for users to understand the model's confidence as an indication of its predictive reliability. However, it's equally crucial to understand that these confidence metrics aren't direct measures of accuracy. For an accurate assessment, a comparison between the model's predictions and ground truth labels on a benchmark dataset would be required. Future iterations of the model might prioritize enhanced consistency across varied distances and real-time feedback for users to adjust their positioning.

5.2. Survey

A survey was conducted where six participants were invited to evaluate Aivisiontrain. The evaluation process was divided into two parts. Initially, participants were guided on how to use the model, with me assisting them step-by-step.

This can be seen as a demo session. Subsequently, participants were given the autonomy to interact with the model unassisted. For feedback collection, participants were provided a feedback form (refer to Figure 3) to complete anonymously. The feedback was compiled, and the results can be found in Table 1.



INSTRUCTIONS:

Thank you for participating in the evaluation of our AI Fitness Model. Your feedback is crucial for the continuous improvement of our system. Please answer the following questions based on your experience during the demo and autonomous interaction sessions.

QUESTIONS:	RATING SCALE:				
	Never	Rarely	Sometimes	Often	Always
was it easy to navigate Aivisiontrain?	○	○	○	○	○
Were all features functional and responsive during your interaction?	○	○	○	○	○
How consistently did Aivisiontrain provide accurate fitness advice or predictions?	○	○	○	○	○
Were the Aivisiontrain responses and interactions quick and timely?	○	○	○	○	○
How often did you find Aivisiontrain content and advice relevant to your needs?	○	○	○	○	○
How frequently did you feel confident in the exercise or health suggestions given by Aivisiontrain?	○	○	○	○	○
Did Aivisiontrain seem tailored to individual user needs?	○	○	○	○	○
How often did you find it easy to integrate or use Aivisiontrain with other fitness apps or platforms you use?	○	○	○	○	○
If you've used Aivisiontrain previously, how often do you notice improvements or changes based on user feedback?	○	○	○	○	○

Figure 6: Feedback Survey Form

Participant	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9
Person A	often	sometimes	always	sometimes	often	always	always	rarely	often
Person B	always	always	always	often	always	often	always	never	always
Person C	sometimes	often	always	often	always	sometimes	often	sometimes	always
Person D	often	often	always	sometimes	often	often	often	rarely	often
Person E	often	sometimes	often	often	often	often	often	sometimes	often
Person F	often	always	often	sometimes	often	sometimes	often	sometimes	often

Table 1: Anonymous Feedback Survey Results

Post-evaluation, I interviewed each participant to understand what they appreciated and what they found lacking in the model.

Positive Feedback:

1. Participants appreciated the Aivisiontrain capability to count reps in real-time as they exercised.

2. The real-time feedback, especially related to maintaining the correct form during exercises, was commended. They liked that Aivisiontrain could determine the appropriate angle required to successfully complete a rep.

3. A highlighted feature was the Aivisiontrain "make" command. Participants could provide a prompt, and the Aivisiontrain 's responses were saved as a .txt file for later review.

4. The Aivisiontrain 's engaging responses were appreciated, especially since it remained focused on fitness queries, ensuring users stayed aligned with their fitness objectives.

Areas of Improvement:

1. Participants noted that the Aivisiontrain required very clear articulation. Any mispronunciation often meant they had to repeat their question, leading to frustration.

2. One user pointed out a flaw in the pose estimation; it inadvertently shifted to another person walking in the background. This raised concerns about using Aivisiontrain in crowded settings like gyms.

3. Participants reported a slight delay when requesting Aivisiontrain to initiate an exercise.

Personal Observations:

1. Many participants spoke softly and had pronunciation variations, leading to the Aivisiontrain inability to comprehend their queries.

2. During exercises, certain participants stood too close to the camera, presumably to be closer to the microphone. As a solution, I've added a prompt in the code instructing users to step back three paces from the camera before beginning an exercise.

3. I noted that the rep counter and pose estimator faced difficulties with taller users. For instance, a participant who was 6'5" had to stand considerably far from the camera to be detected.

4. Overall, participants seemed to prefer the exercise and rep counting features over the chatbot. The misinterpretations due to varied accents and pronunciations suggest that training the model for multiple accents might be beneficial.

6. Future Work

As I delve deeper into my project, I see endless possibilities for refinement and broadening its scope. My primary aim is to cater to a wider audience and optimize the user experience. Here's what I'm planning for the next phase of my AI fitness project:

1. I've received feedback about challenges users face due to accent variations. Thus, I intend to expand my dataset to cover a global spectrum of accents, ensuring better communication and understanding.
2. I've noted concerns about accurate pose detection in populated settings. In response, I'm working on a specialized algorithm to maintain focus on the primary user, effectively reducing background distractions.
3. I'm keen on exploring the benefits of wide-angle lens cameras. The idea is to cater to users of all sizes, ensuring they don't have to make spatial adjustments frequently.
4. Speed is of the essence. With that in mind, I'm looking at ways to reduce Aivisiontrain response times, especially during exercise initiation.
5. I believe in continuous improvement. So, I'm designing a system that lets users share feedback in real-time, helping me make immediate adjustments.
6. I envision Aivisiontrain to go beyond the current exercise catalog. My plan is to include a diverse range of exercises, suitable for various fitness levels.
7. Given that some users interact in softer tones, I'm considering integrating more sensitive microphones or voice amplification tools to ensure clear command recognition.
8. I'm aiming to transition from just guiding users to proactively planning their workouts. By leveraging sophisticated AI algorithms, I hope to offer dynamic workout plans that adapt to users' changing goals.
9. Personalization is paramount in fitness. I'm working on user profiles that offer individualized fitness recommendations and track progress.

I genuinely value the feedback from my initial phase. It has given me a clear direction for my future work. My goal remains steadfast: to hone my AI fitness model to the pinnacle of user-centricity and effectiveness.

7. Conclusion

In conclusion, the development of the AI-powered fitness trainer, leveraging Python code and the OpenAI API Playground for chatbot interactions and real-time exercise monitoring through Mediapipe, marks a pivotal stride in the convergence of cutting-edge technology and personalized fitness guidance. This innovative system's initial implementation has showcased its potential to adeptly address user queries, offer exercise assistance, and count repetitions. By seamlessly amalgamating AI and fitness training, it responds to the contemporary demand for tailored and interactive fitness experiences, fostering improved user engagement, motivation, and adherence. Beyond its immediate impact, this fusion holds implications for revolutionizing the fitness industry, potentially democratizing access to expert guidance and paving the way for individually curated workout plans. Looking ahead, identified avenues for enhancement—including broadening fitness knowledge, diversifying exercise options, implementing user-centric features, integrating AI-generated routines—underscore the promising trajectory of AI-powered fitness trainers. Ultimately, this marriage of technology and fitness not only augments workout experiences but also envisions a future where individuals achieve their fitness aspirations with unprecedented precision and empowerment, reshaping wellness and transforming lives.

Reference

(American Heart Association 2023)
American Heart Association, 2023. Breaking down barriers to fitness [online]. www.heart.org. Available from: https://www.heart.org/en/healthy-living/fitness/getting-active/breaking-down-barriers-to-fitness [Accessed 14 Aug 2023].

(Anubbe 2023)
Anubbe, 2023. Paola, el chatbot de ventas ideal para tu negocio [online]. Paola, el chatbot de ventas ideal para tu negocio. Available from: https://www.paola.chat/ [Accessed 14 Aug 2023].

(Baharuddin 2023)
Baharuddin, M. Y., 2023. [online]. Researchgate.net. Available from: https://www.researchgate.net/publication/344535731_Biomechanics_analysis_on_muscle_activation_during_jumping_jack_exercise [Accessed 14 Aug 2023].

(CDC 2022)
CDC, 2022. Overcoming barriers to physical activity [online]. Centers for Disease Control and Prevention. Available from: https://www.cdc.gov/physicalactivity/basics/adding-pa/barriers.html [Accessed 14 Aug 2023].

(CDC 2023)
CDC, 2023. Benefits of physical activity [online]. Centers for Disease Control and Prevention. Available from: https://www.cdc.gov/physicalactivity/basics/pa-health/index.htm [Accessed 14 Aug 2023].

(coachai 2023)
coachai, 2023. CoachAi [online]. Coachai.com. Available from: https://www.coachai.com/ [Accessed 14 Aug 2023].

(DOMARS 2023)
DOMARS, 2023. Finding the Process ID - Windows drivers [online]. Microsoft.com. Available from: https://learn.microsoft.com/en-us/windows-hardware/drivers/debugger/finding-the-process-id [Accessed 14 Aug 2023].

(Google 2023)
Google, 2023. Speech-to-text [online]. Google Cloud. Available from: https://cloud.google.com/speech-to-text [Accessed 14 Aug 2023].

(Iffort 2023)
Iffort, 2023. GPT-3 vs GPT-3.5: Key differences and applications [online]. Iffort. Available from: https://www.iffort.com/2023/03/31/gpt-3-vs-gpt-3-5/ [Accessed 14 Aug 2023].

(infiGro 2023)
infiGro, 2023. InfiGro [online]. Infigro. Available from: https://www.infivolve.com/ [Accessed 14 Aug 2023].

(Kwon 2023)
Kwon, Y., 2023. [online]. Researchgate.net. Available from: https://www.researchgate.net/publication/358421361_Real-Time_Workout_Posture_Correction_using_OpenCV_and_MediaPipe [Accessed 14 Aug 2023].

(Mediapipe 2023)
Mediapipe, 2023. Pose landmark detection guide [online]. Google for Developers. Available from: https://developers.google.com/mediapipe/solutions/vision/pose_landmarker [Accessed 14 Aug 2023].

(NHS 2023)
NHS, 2023. Benefits of exercise [online]. nhs.uk. Available from: https://www.nhs.uk/live-well/exercise/exercise-health-benefits/ [Accessed 14 Aug 2023].

(OpenAI 2023)
OpenAI, 2023. OpenAI platform [online]. Openai.com. Available from:

https://platform.openai.com/docs/introduction [Accessed 14 Aug 2023].

(Openai 2023)
Openai, 2023. How do text-davinci-002 and text-davinci-003 differ? [online]. Openai.com. Available from: https://help.openai.com/en/articles/6779149-how-do-text-davinci-002-and-text-davinci-003-differ [Accessed 14 Aug 2023].

(OpenAI platform 2023)
OpenAI platform, 2023. OpenAI platform [online]. Openai.com. Available from: https://platform.openai.com/playground [Accessed 14 Aug 2023].

(OpenCV 2021)
OpenCV, 2021. OpenCV - open Computer Vision library [online]. OpenCV. Available from: https://opencv.org/ [Accessed 14 Aug 2023].

(Pardeshi et al. 2022)
Pardeshi, H., Ghaiwat, A., Thongire, A., Gawande, K. and Naik, M., 2022. Fitness freaks: A system for detecting definite body posture using OpenPose estimation. In: Futuristic Trends in Networks and Computing Technologies. Singapore: Springer Nature Singapore, 1061–1072.

(Python 2018)
Python, R., 2018. The ultimate guide to speech recognition with Python [online]. Realpython.com. Available from: https://realpython.com/python-speech-recognition/ [Accessed 14 Aug 2023].

(Ribeiro 2023)
Ribeiro, A. S., 2023. [online]. Researchgate.net. Available from: https://www.researchgate.net/publication/357897278_A_Brief_Review_on_the_Effects_of_the_Squat_Exercise_on_Lower-Limb_Muscle_Hypertrophy [Accessed 14 Aug 2023].

(Rogers 2007)
Rogers, P., 2007. How to do biceps curls: Proper form, variations, and common mistakes [online]. Verywell Fit. Available from: https://www.verywellfit.com/how-to-do-the-biceps-arm-curl-3498604 [Accessed 14 Aug 2023].

(Taware et al. 2021)
Taware, G., Agrawal, R., Dhende, P., Jondhalekar, P. and Hule, S., 2021. AI-based Workout Assistant and Fitness guide. International journal of engineering research & technology (Ahmedabad) [online], 10 (11). Available from: https://www.ijert.org/ai-based-workout-assistant-and-fitness-guide [Accessed 14 Aug 2023].

(Tess 2023)
Tess, 2023. Tess IBH [online]. X2ai.com. Available from: https://www.x2ai.com/uprisehealth [Accessed 14 Aug 2023].

(Zhang et al. 2020)
Zhang, J., Oh, Y. J., Lange, P., Yu, Z. and Fukuoka, Y., 2020. Artificial intelligence chatbot behavior change model for designing artificial intelligence chatbots to promote physical activity and a healthy diet: Viewpoint. Journal of medical internet research [online], 22 (9), e22845. Available from: https://www.jmir.org/2020/9/e22845/ [Accessed 14 Aug 2023].

(Zheng et al. 2020)
Zheng, C., Wu, W., Chen, C., Yang, T., Zhu, S., Shen, J., Kehtarnavaz, N. and Shah, M., 2020. Deep learning-based human pose estimation: A survey. arXiv [cs.CV] [online]. Available from: http://arxiv.org/abs/2012.13392 [Accessed 14 Aug 2023].