ZENPOSE: FITNESS ANALYZER

A PROJECT REPORT

Submitted by

JERPHY MIRACLIN BACKIA D (2116210701096)

KEERTHANAA SP (2116210701119)

KEERTHIGA K (2116210701120)

KIRAN M (2116210701122)

in partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

RAJALAKSHMI ENGINEERING COLLEGE, CHENNAI

ANNA UNIVERSITY:: CHENNAI 600 025

APRIL 2024

RAJALAKSHMI ENGINEERING COLLEGE, CHENNAI

BONAFIDE CERTIFICATE

Certified that this Report titled "**Zenpose: Fitness Analyzer**" is the bonafide work of **"Jerphy Miraclin Backia D (210701096), Keerthanaa SP (210701119), Keerthiga K (210701120), Kiran M (210701122)"** who carried out the work under my supervision. Certified further that to the best of my knowledge the work reported herein does not form part of any other thesis ordissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE

Dr. K. ANAND ME, Ph.D
PROJECT COORDINATOR
Professor,
Department of Computer Science and Engineering,
Rajalakshmi Engineering College,

Submitted to Mini Project Viva-Voce Examination held on ___________________

Internal Examiner **External Examiner**

ABSTRACT

The fitness analyzer is truly emblematic of a groundbreaking leap forward in the realm of health and wellness technology. This project presents the development of a Fit Analyzer application that leverages advanced computer vision technologies, specifically OpenCV and MediaPipe, to assist users in monitoring and improving their fitness routines. The primary focus of the Fit Analyzer is to identify body landmarks, calculate joint angles, and accurately count curls and repetitions during exercise sessions. The application utilizes OpenCV for efficient image processing and MediaPipe for robust pose detection.

By integrating these technologies, the Fit Analyzer can detect key body landmarks in real-time, allowing it to calculate joint angles dynamically. These angles are then analyzed to determine the completion of specific movements, such as curls, enabling the system to count repetitions with high accuracy. This tool is particularly useful for individuals aiming to improve their exercise efficiency, prevent injuries, and achieve better fitness outcomes through precise and automated tracking.Through detailed analysis and real-time feedback, the Fit Analyzer stands as a valuable addition to the fitness technology landscape, offering a practical solution for personal training and exercise monitoring.

ACKNOWLEDGEMENT

Initially we thank the Almighty for being with us through every walk of our life and showering his blessings through the endeavor to put forth this report. Our sincere thanks to our Chairman **Mr. S.MEGANATHAN, B.E, F.I.E.**, our Vice Chairman **Mr. ABHAY SHANKAR MEGANATHAN, B.E., M.S.,** and our respected Chairperson **Dr. (Mrs.) THANGAM MEGANATHAN**, **Ph.D.,** for providing us with the requisite infrastructure and sincere endeavoring in educating us in their premier institution.

Our sincere thanks to **Dr. S.N. MURUGESAN, M.E., Ph.D.,** our beloved Principal for his kind support and facilities provided to complete our work in time.We express our sincere thanks to **Dr. P. KUMAR, Ph.D.**, Professor and Head of the Department of Computer Science and Engineering for his guidance andencouragement throughout the project work. We convey our sincere and deepest gratitude to our internal guide, **Dr.K.Anand ME,Ph.D,** Assistant Professor, Department of Computer Science and Engineering. Rajalakshmi Engineering College for his valuable guidance throughout the course of the project.

Jerphy Miraclin Backia D (210701096)
Keerthanaa SP (210701119)
Keerthiga K (210701120)
Kiran M (210701122)

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

PC	Personal Computer
CV	Computer Vision
AI	Artificial Intelligence
OS	Operating System
RGB	Red Green Blue

CHAPTER 1

INTRODUCTION

1.1 PROBLEM STATEMENT

A fitness analyzer aims to revolutionize personal fitness by integrating data from wearable devices and fitness apps with user inputs to offer personalized fitness assessments, workout plans, and health insights. It analyzes historical and real-time data to generate customized recommendations that dynamically adjust based on the user's progress and goals. Key features include tailored workout plans, nutritional advice, vital sign monitoring, and motivational tools such as gamification and virtual coaching. The system ensures accessibility across devices and prioritizes data privacy and security to protect user information. By providing personalized, data-driven insights, the fitness analyzer can significantly enhance fitness outcomes and user satisfaction, addressing the challenges of data accuracy, personalization, user engagement, and privacy concerns.

1.2 SCOPE OF THE WORK

The scope of work for developing a fitness analyzer includes integrating and analyzing data from wearable devices, fitness apps, and user inputs to create a comprehensive fitness profile. It involves designing algorithms for personalized workout plans and nutritional advice, monitoring vital signs, and providing health insights. The project will also focus on enhancing user engagement through features like gamification and virtual coaching, ensuring accessibility across multiple devices, and implementing robust data privacy and security measures. Additionally, it requires addressing challenges related to data accuracy, personalization, and user motivation to deliver a seamless and effective fitness solution

1.3 AIM AND OBJECTIVE

The aim of our fitness analyzer is to revolutionize personal health and fitness by leveraging advanced artificial intelligence algorithms to provide users with comprehensive, personalized insights into their physical well-being. Our objective is to empower individuals to make informed decisions about their fitness routines, nutrition, to offer tailored recommendations for optimizing health and achieving personal fitness goals. Through intuitive interfaces and real-time feedback, our solution aims to enhance user motivation, adherence to healthy behaviors, and ultimately, contribute to improved health outcomes.

1.4 RESOURCES

This project has been developed through widespread secondary research of accredited manuscripts, standard papers, business journals, white papers, analysts' information, and conference reviews. Significant resources are required to achieve an efficacious completion of this project.
The following prospectus details a list of resources that will play a primary role in the successful execution of our project:

- A properly functioning workstation (PC, laptop, net-books etc.) to carry out desired research and collect relevant content.
- Unlimited internet access.
- Unrestricted access to the university lab in order to gather a variety of literature including academic resources (for e.g. Prolog tutorials, online programming examples, bulletins, publications, e-books, journals etc.), technical manuscripts, etc. Prolog development kit in order to program the desired system and other related software that will be required to perform our research.

1.5 MOTIVATION

The Fitness Analyzer project is designed to transform the fitness industry by utilizing artificial intelligence to provide personalized fitness assessments, workout plans, and health insights. This innovative system aims to integrate data from various sources, including wearable devices, fitness apps, and user inputs, to create a comprehensive and dynamic fitness profile for each user. By analyzing both historical and real-time data, the AI algorithms will deliver customized recommendations that adapt to the user's progress and evolving fitness goals. This personalized approach addresses the shortcomings of traditional fitness regimens, which often fail to account for individual variability in fitness levels, goals, and physical conditions.

The AI algorithms developed for this project will be designed to offer dynamic personalization, continuously adjusting workout plans and nutritional advice based on the user's real-time data and progress. This ensures that the fitness recommendations remain relevant and effective as the user's fitness level improves or as their goals change. For instance, if the system detects a plateau in the user's progress, it might suggest new exercises or adjustments to the current routine to overcome it. This level of personalization is key to maintaining user engagement and ensuring sustained progress towards fitness goals.

CHAPTER 2

LITERATURE SURVEY

The systematic review delved into AI-powered software designed [1] for personal fitness trainers, analysing its efficacy and impact. Across 15 studies meeting inclusion criteria, the research highlighted the effectiveness of AI-driven personal fitness trainers in improving physical fitness, bolstering motivation levels, and fostering greater adherence to exercise regimens. These findings underscored the transformative potential of AI in revolutionizing the fitness industry, offering personalized guidance and support to individuals seeking to optimize their health and achieve their fitness goals. Through intelligent algorithms and tailored recommendations, fitness analysers have emerged as invaluable tools for enhancing overall well-being and empowering individuals on their wellness journey.

In developing an AI-based fitness software tailored [2] to senior citizens, this study employed machine learning algorithms to analyse user movement and provide personalized exercise recommendations. Results showcased the software's efficacy in improving muscle power and balance among older individuals. By leveraging advanced technology to cater to the specific needs of seniors, this innovative approach holds promise for enhancing physical well-being and promoting active aging through targeted fitness interventions.

.

[3]This study delved into AI's capacity to personalize fitness instruction, exploring its potential for individualized training. Findings revealed that AI-based fitness trainer applications could deliver tailored and adaptive training programs to meet individual needs. Additionally, the research scrutinized both the opportunities and challenges associated with AI-driven fitness training, shedding light on its promise for revolutionizing the fitness industry while also considering potential limitations and areas for improvement.

A mobile app powered by artificial intelligence that encourages college students to move more[4] . An AI-based smartphone app was created as part of this project to encourage college students to move more. Personalized workout advice was given by the app based on the user's degree of physical activity, time limits, and fitness objectives. The study revealed that the app was successful in boosting pupils' physical activity levels.

This study aimed to address sedentary [5] behavior among overweight or obese individuals by developing an AI-powered virtual personal trainer. The innovative app was designed to motivate users to engage in more physical activity. Results indicated that overweight and obese adults who utilized the app experienced significant increases in physical activity levels and subsequent weight loss. By harnessing the capabilities of AI, this intervention demonstrated promising potential in promoting healthier lifestyles and combating the challenges associated with excess weight.

Amit Nagarkoti et al. [6] proposed an enhanced system aimed at improving prior models' accuracy through vision-based deep learning techniques. Their approach involves utilizing the initial ten layers of the VGG19 network to generate a fixed-size vector representation for a given image. Subsequently, the system employs two multi-step branches, integrating convolutional neural networks (CNN) along with OpenCV for optical flow tracking. However, it's noted that this system is limited to capturing motion in two dimensions.

To improvise the system proposed previously, Steven Chen et al., [7] used deep convoluted neural networks (CNNs) to label RGB images. They made advantage of the trained model, Open-Pose, for pose detection. The model consists of multiple-stage CNN with two branches: one branch is used to learn the part affinity fields, while the other branch is used to learn the confidence mapping of a key point on an image. But this model has its own drawbacks too, i.e it works only for pre-recorded videos.

Danish Sheikh et al., [8] abstracted the technique of exploitation create estimation abstract thought out-put as input for associate LSTM classifier into a toolkit referred to as Action-Al. For video process demo, Open-CV suffices. For create estimation they used Open-pose enforced with in style deep learning frameworks like Tensor-flow and Py-Torch. The user can start, stop, pause, and restart yoga by utilising the voice interface, which uses the Snips AIR voice assistant. This model produced good results with high precision, but in certain cases, when important points are rotated, angles do not change.

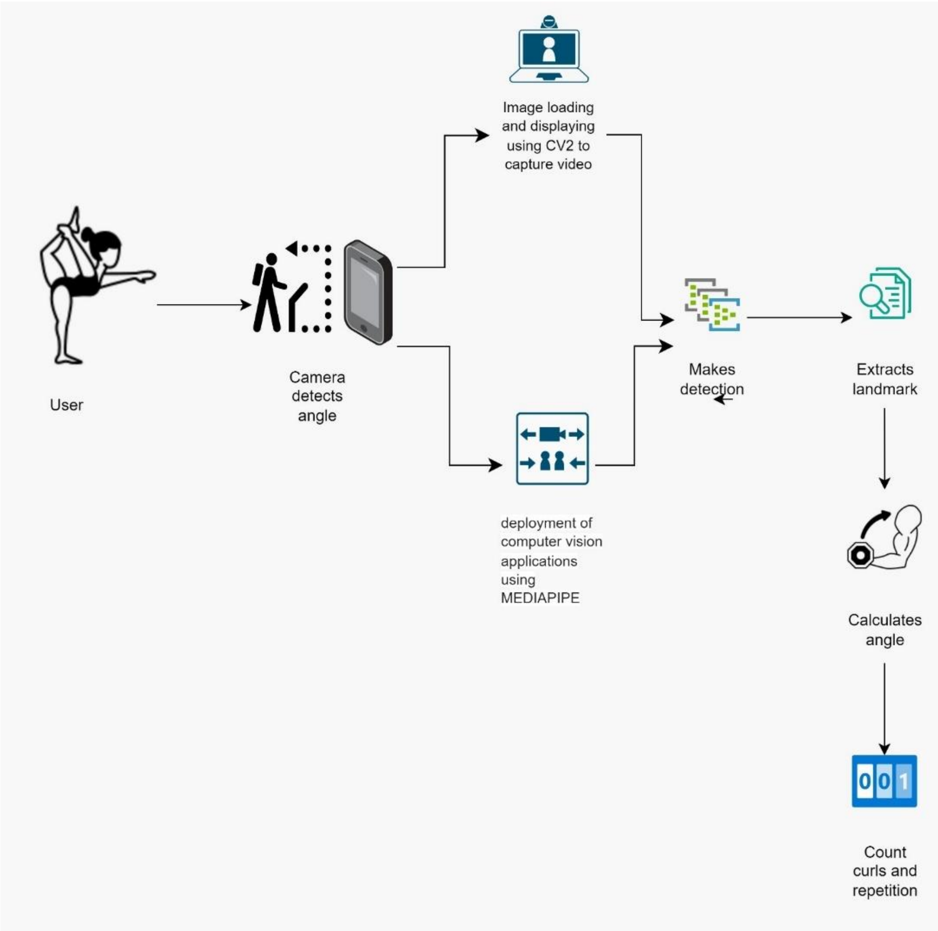
CHAPTER 3

SYSTEM DESIGN

3.1 GENERAL

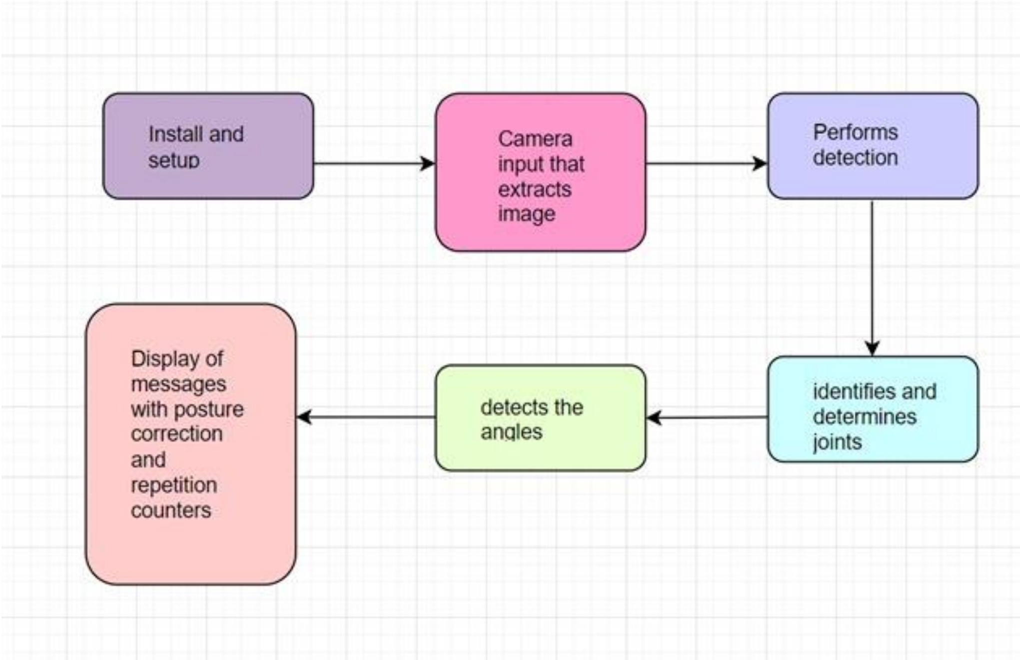
In this section, we illustrate the cohesive functioning of all components when organized together. The integrated process is depicted in the accompanying architecture diagram, offering a visual representation of how each element collaborates and along with a block diagram to represent the functionalities and to achieve the system's objectives.

3.2 SYSTEM ARCHITECTURE DIAGRAM



3.2.1 ARCHITECURE DIAGRAM

3.3 BLOCK DIAGRAM



3.3.1 BLOCK DIAGRAM

3.4 DEVELOPMENTAL ENVIRONMENT

3.4.1 HARDWARE REQUIREMENTS

The hardware requirements may serve as the basis for a contract for the system's implementation. It should therefore be a complete and consistent specification of the entire system. It is generally used by software engineers as the starting point for the system design

Table 3.1 Hardware Requirements

COMPONENTS	SPECIFICATION
PROCESSOR	Intel Core i5
RAM	8 GB RAM
GPU	NVIDIA GeForce GTX 1650
MONITOR	15" COLOR
HARD DISK	512 GB
PROCESSOR SPEED	MINIMUM 1.1 GHz

3.4.2 SOFTWARE REQUIREMENTS

The software requirements document is the specifications of the system. It should include both a definition and a specification of requirements. It is aset of what the system should rather be doing than focus on how it should be done. The software requirements provide a basis for creating the software requirements specification. It is useful in estimating the cost, planning team activities, performing tasks, tracking the team, and tracking the team’s progressthroughout the development activity.

Python IDLE, and **chrome** would all be required.

CHAPTER 4

PROJECT DESCRIPTION

4.1 METHODOLODY

The innovative evolution in fitness tracking and analysis is poised to undergo a significant transformation through the integration of an AI-powered fitness analyzer that harnesses the advanced capabilities of Mediapipe and OpenCV. By leveraging Mediapipe's cutting-edge Pose Estimation module, this revolutionary system empowers users to monitor the intricacies of exercise form and movement in real-time by tracking critical points on the human body. This real-time tracking feature plays a pivotal role in offering users immediate feedback on their exercise techniques and postures, thereby enhancing workout efficiency and reducing the likelihood of injuries.

Not only does Mediapipe excel in tracking various body movements, but it also excels in capturing hand gestures with its exceptional Hand Tracking module, which proves invaluable for analyzing activities involving weights or resistance bands. Complementing the prowess of Mediapipe, OpenCV steps in to augment the system with its state-of-the-art image and video processing capabilities, amplifying the capacity to gather and analyze fitness data with precision and efficiency.

By synergizing real-time data acquisition and the cutting-edge capabilities of these technologies, the AI-powered fitness analyzer is equipped to deliver personalized fitness assessments and dynamically adjustable training plans tailored to each individual's objectives and progress. This seamless integration of Mediapipe and OpenCV enables the system to furnish users with pertinent insights and recommendations that cater to their unique fitness goals, thereby fostering healthier and more effective workout routines. In essence, the amalgamation of Mediapipe and OpenCV within the AI fitness analyzer heralds a new era in fitness tracking, elevating the user experience by providing tailored guidance and data-driven strategies for optimal workout performance and results.

4.2 MODULE DESCRIPTION:

Making Detections:

To make detections using an fitness analyser with Mediapipe and OpenCV, first, Python environment is set. Use OpenCV to capture video input from your camera or video file. Next, utilize Mediapipe for pose estimation, converting each frame to RGB format for processing. Apply the pose estimation model to detect and track key body landmarks, which are then visualized on the frame using OpenCV. Additionally, you can incorporate Mediapipe's hand tracking module to monitor hand movements and interactions, enhancing the analysis of exercises involving equipment. This setup allows for real-time monitoring of exercise form and movement, providing users with immediate feedback and personalized fitness recommendations based on their performance.

Determine joint:

To determine joints using a fitness analysers with Mediapipe and OpenCV, begin by setting up your Python environment with the required libraries. Capture video input using OpenCV from your camera or video file. Utilize Mediapipe's Pose Estimation module to process each frame and detect key body landmarks in real-time. By analysing these coordinates, you can accurately determine the positions of each joint. This capability allows for precise monitoring of exercise form and movement, providing users with immediate feedback on their posture and technique.

Calculation Of Joint Angle:

Extract the coordinates of specific landmarks such as shoulders, elbows, and wrists from the pose estimation results. Use these coordinates to calculate the joint angles using trigonometric functions. For example, you can calculate the angle between the vectors formed by the shoulder, elbow, and wrist to determine the arm's orientation. Display these calculated angles on the video frame using OpenCV, providing users with immediate feedback on their posture and technique.

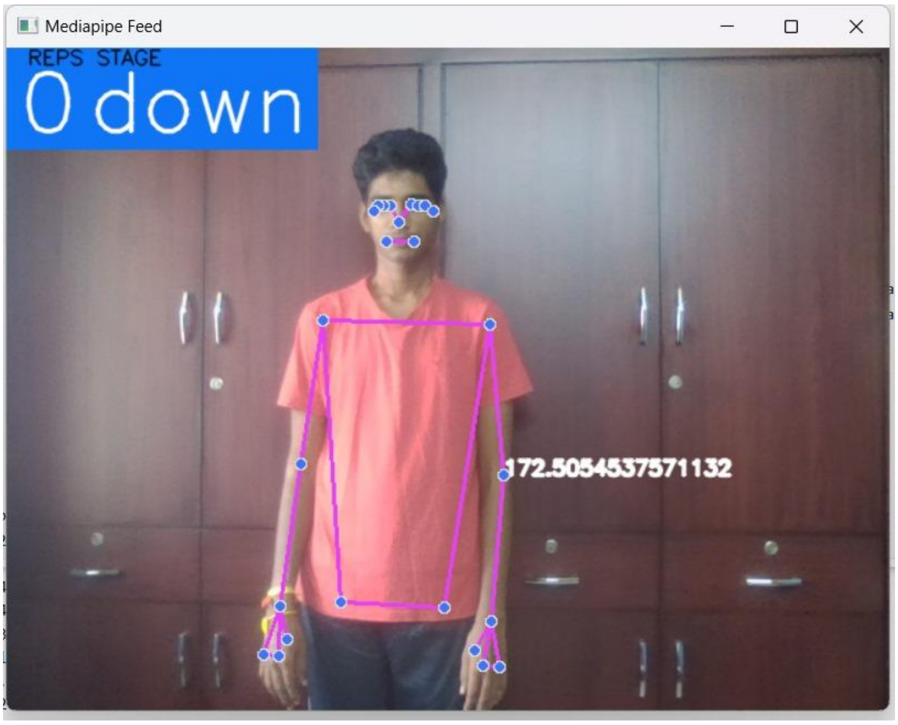
Curl Counter:

To implement a curl counter using a fitness analyzer with Mediapipe and OpenCV, follow these steps. First, set up your Python environment with OpenCV and Mediapipe libraries. Capture video input from your camera using OpenCV. Utilize Mediapipe's Pose Estimation module to detect and track key body landmarks in real-time. Implement logic to detect when the wrist landmark moves closer to the shoulder landmark in a curl motion. Count each curl and display the count on the screen using OpenCV. Reset the count after a certain number of seconds or after a set number of curls.

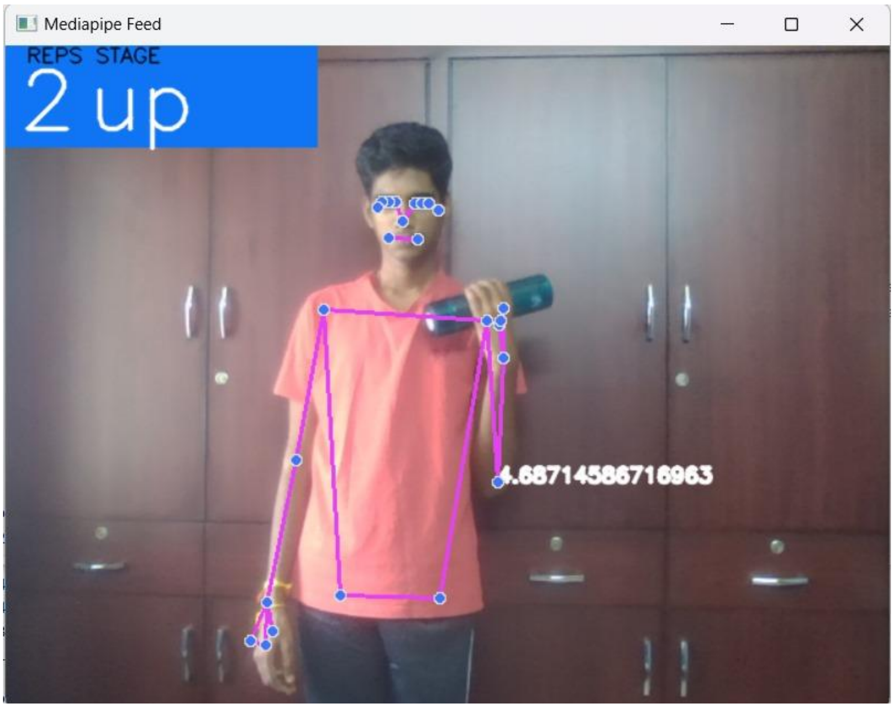
CHAPTER 5

RESULTS AND DISCUSSION

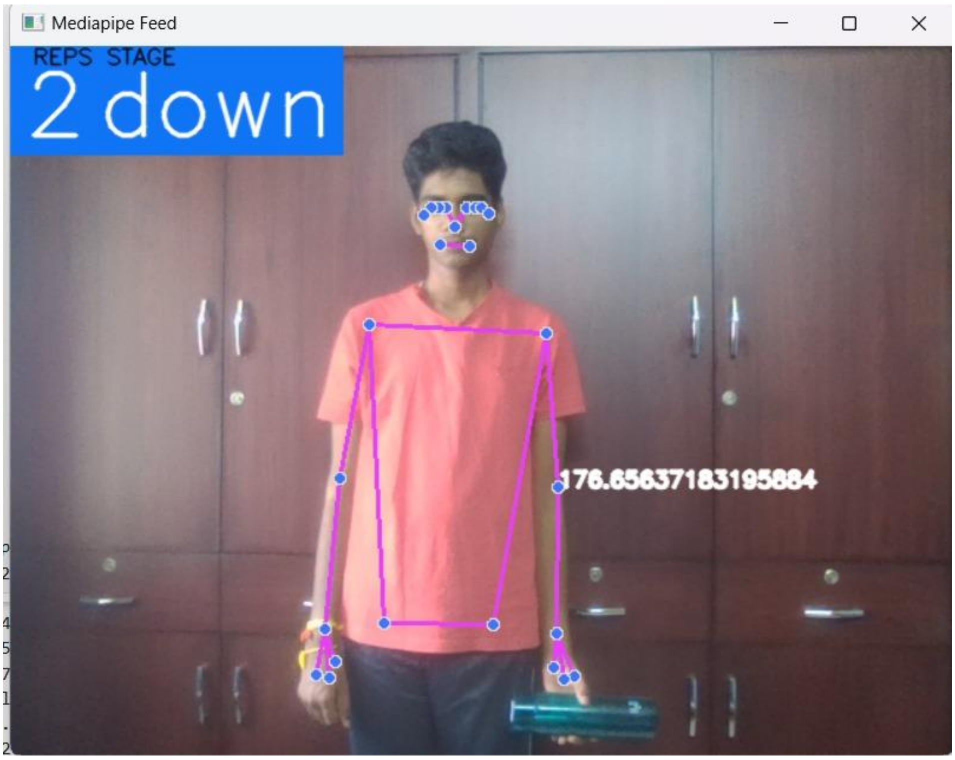
5.1 OUTPUT SCREENSHOTS



5.1.1 Initial Position



5.1.2 Bicep Curl Up Position



5.1.3 Bicep Curl Down Position

5.2 RESULT

An artificial intelligence-enabled fitness analyzer uses AI to evaluate and improve personal exercise regimens. With its ability to measure heart rate, sleep patterns, and levels of physical activity, the analyzer can provide real-time feedback and modify workout regimens accordingly. By customizing advise to each user's individual needs and progress, this technology makes sure that users have a more productive and enjoyable workout experience and helps them reach their fitness objectives more quickly.

CHAPTER 6
CONCLUSION AND FUTURE ENHANCEMENTS

6.1 CONCLUSION

In conclusion, the transformative impact of AI fitness analyzers on personal fitness and health management cannot be overstated. These advanced systems excel in offering highly personalized insights and data-driven suggestions, leveraging cutting-edge technology to revolutionize the way individuals approach their wellness goals. By delivering real-time feedback and tailored workout plans, AI fitness analyzers empower users to optimize their health objectives effectively. These devices seamlessly integrate with wearable technology and apps, providing a comprehensive solution for health monitoring and management.

Moreover, the incorporation of features such as gamification and virtual coaching sets AI fitness analyzers apart by enhancing user motivation and engagement. Through gamified elements and interactive coaching sessions, these products not only drive adherence to exercise regimens but also make the entire fitness experience more engaging and enjoyable. As technology continues to evolve, AI fitness analyzers are poised to become even more indispensable, offering a level of health monitoring and support that promotes improved overall well-being and preventive healthcare.

Despite their potential to revolutionize personal health management, AI fitness analyzers still face challenges that need to be addressed for widespread adoption. Issues such as data privacy concerns, accuracy in data analysis, and ensuring accessibility to diverse user groups are critical areas where further development and refinement are needed. By overcoming these hurdles, AI fitness analyzers can fulfill their promise of enhancing personalized health management and encouraging healthier lifestyles for all.

6.2 FUTURE ENHANCEMENTS

Significant developments are anticipated for fitness analyzers in the future, including increased personalization through deeper data integration, tighter interface with healthcare systems, and the creation of cutting-edge wearable equipment for more precise health monitoring. Wider acceptance will also be fueled by the addition of predictive analytics, enhanced data privacy and security protocols, and augmented and virtual reality for immersive fitness experiences. Further establishing AI fitness analyzers as all-inclusive instruments for holistic health and wellness management will come from increased accessibility and the addition of mental health support.

APPENDIX

SOURCE CODE:

```
!pip install mediapipe opencv-python
import cv2
import mediapipe as mp
import numpy as np
mp_drawing = mp.solutions.drawing_utils
mp_pose = mp.solutions.pose

cap = cv2.VideoCapture(0)
while cap.isOpened():
    ret, frame = cap.read()
    cv2.imshow('Mediapipe Feed', frame)

    if cv2.waitKey(10) & 0xFF == ord('q'):
        break

cap.release()
cv2.destroyAllWindows()

cap = cv2.VideoCapture(0)
## Setup mediapipe instance
with mp_pose.Pose(min_detection_confidence=0.5, min_tracking_confidence=0.5) as pose:
    while cap.isOpened():
        ret, frame = cap.read()

        # Recolor image to RGB
        image = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
        image.flags.writeable = False

        # Make detection
        results = pose.process(image)

        # Recolor back to BGR
        image.flags.writeable = True
        image = cv2.cvtColor(image, cv2.COLOR_RGB2BGR)

        # Render detections
        mp_drawing.draw_landmarks(image, results.pose_landmarks, 
mp_pose.POSE_CONNECTIONS,
                                mp_drawing.DrawingSpec(color=(245,117,66), thickness=2, 
circle_radius=2), 
                                mp_drawing.DrawingSpec(color=(245,66,230), thickness=2, 
circle_radius=2) 
                                 )

        cv2.imshow('Mediapipe Feed', image)

        if cv2.waitKey(10) & 0xFF == ord('q'):
            break
```

```
    cap.release()
    cv2.destroyAllWindows()


mp_drawing.DrawingSpec??

cap = cv2.VideoCapture(0)
## Setup mediapipe instance
with mp_pose.Pose(min_detection_confidence=0.5, min_tracking_confidence=0.5) as pose:
    while cap.isOpened():
        ret, frame = cap.read()

        # Recolor image to RGB
        image = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
        image.flags.writeable = False

        # Make detection
        results = pose.process(image)

        # Recolor back to BGR
        image.flags.writeable = True
        image = cv2.cvtColor(image, cv2.COLOR_RGB2BGR)

        # Extract landmarks
        try:
            landmarks = results.pose_landmarks.landmark
            print(landmarks)
        except:
            pass


        # Render detections
        mp_drawing.draw_landmarks(image, results.pose_landmarks,
mp_pose.POSE_CONNECTIONS,
                                mp_drawing.DrawingSpec(color=(245,117,66), thickness=2,
circle_radius=2),
                                mp_drawing.DrawingSpec(color=(245,66,230), thickness=2,
circle_radius=2)
                                 )

        cv2.imshow('Mediapipe Feed', image)

        if cv2.waitKey(10) & 0xFF == ord('q'):
            break

    cap.release()
    cv2.destroyAllWindows()


len(landmarks)
for lndmrk in mp_pose.PoseLandmark:
    print(lndmrk)
landmarks[mp_pose.PoseLandmark.LEFT_SHOULDER.value].visibility
landmarks[mp_pose.PoseLandmark.LEFT_ELBOW.value]
landmarks[mp_pose.PoseLandmark.LEFT_WRIST.value]
```

```
def calculate_angle(a,b,c):
    a = np.array(a) # First
    b = np.array(b) # Mid
    c = np.array(c) # End

    radians = np.arctan2(c[1]-b[1], c[0]-b[0]) - np.arctan2(a[1]-b[1], a[0]-b[0])
    angle = np.abs(radians*180.0/np.pi)

    if angle >180.0:
        angle = 360-angle

    return angle

shoulder =
[landmarks[mp_pose.PoseLandmark.LEFT_SHOULDER.value].x,landmarks[mp_pose.PoseLa
ndmark.LEFT_SHOULDER.value].y]
elbow =
[landmarks[mp_pose.PoseLandmark.LEFT_ELBOW.value].x,landmarks[mp_pose.PoseLandm
ark.LEFT_ELBOW.value].y]
wrist =
[landmarks[mp_pose.PoseLandmark.LEFT_WRIST.value].x,landmarks[mp_pose.PoseLandmar
k.LEFT_WRIST.value].y]

shoulder, elbow, wrist
calculate_angle(shoulder, elbow, wrist)
tuple(np.multiply(elbow, [640, 480]).astype(int))

cap = cv2.VideoCapture(0)
## Setup mediapipe instance
with mp_pose.Pose(min_detection_confidence=0.5, min_tracking_confidence=0.5) as pose:
    while cap.isOpened():
        ret, frame = cap.read()

        # Recolor image to RGB
        image = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
        image.flags.writeable = False

        # Make detection
        results = pose.process(image)

        # Recolor back to BGR
        image.flags.writeable = True
        image = cv2.cvtColor(image, cv2.COLOR_RGB2BGR)

        # Extract landmarks
        try:
            landmarks = results.pose_landmarks.landmark

            # Get coordinates
            shoulder =
[landmarks[mp_pose.PoseLandmark.LEFT_SHOULDER.value].x,landmarks[mp_pose.PoseLa
ndmark.LEFT_SHOULDER.value].y]
            elbow =
[landmarks[mp_pose.PoseLandmark.LEFT_ELBOW.value].x,landmarks[mp_pose.PoseLandm
```

```
ark.LEFT_ELBOW.value].y]
            wrist = 
[landmarks[mp_pose.PoseLandmark.LEFT_WRIST.value].x,landmarks[mp_pose.PoseLandmar
k.LEFT_WRIST.value].y]

            # Calculate angle
            angle = calculate_angle(shoulder, elbow, wrist)

            # Visualize angle
            cv2.putText(image, str(angle),
                           tuple(np.multiply(elbow, [640, 480]).astype(int)),
                           cv2.FONT_HERSHEY_SIMPLEX, 0.5, (255, 255, 255), 2, cv2.LINE_AA
                                )

        except:
            pass


        # Render detections
        mp_drawing.draw_landmarks(image, results.pose_landmarks,
mp_pose.POSE_CONNECTIONS,
                                mp_drawing.DrawingSpec(color=(245,117,66), thickness=2,
circle_radius=2),
                                mp_drawing.DrawingSpec(color=(245,66,230), thickness=2,
circle_radius=2)
                                 )

        cv2.imshow('Mediapipe Feed', image)

        if cv2.waitKey(10) & 0xFF == ord('q'):
            break

    cap.release()
    cv2.destroyAllWindows()


cap = cv2.VideoCapture(0)

# Curl counter variables
counter = 0
stage = None

## Setup mediapipe instance
with mp_pose.Pose(min_detection_confidence=0.5, min_tracking_confidence=0.5) as pose:
    while cap.isOpened():
        ret, frame = cap.read()

        # Recolor image to RGB
        image = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
        image.flags.writeable = False

        # Make detection
        results = pose.process(image)

        # Recolor back to BGR
```

```
        image.flags.writeable = True
        image = cv2.cvtColor(image, cv2.COLOR_RGB2BGR)

        # Extract landmarks
        try:
            landmarks = results.pose_landmarks.landmark

            # Get coordinates
            shoulder =
[landmarks[mp_pose.PoseLandmark.LEFT_SHOULDER.value].x,landmarks[mp_pose.PoseLa
ndmark.LEFT_SHOULDER.value].y]
            elbow =
[landmarks[mp_pose.PoseLandmark.LEFT_ELBOW.value].x,landmarks[mp_pose.PoseLandm
ark.LEFT_ELBOW.value].y]
            wrist =
[landmarks[mp_pose.PoseLandmark.LEFT_WRIST.value].x,landmarks[mp_pose.PoseLandmar
k.LEFT_WRIST.value].y]

            # Calculate angle
            angle = calculate_angle(shoulder, elbow, wrist)

            # Visualize angle
            cv2.putText(image, str(angle),
                        tuple(np.multiply(elbow, [640, 480]).astype(int)),
                        cv2.FONT_HERSHEY_SIMPLEX, 0.5, (255, 255, 255), 2, cv2.LINE_AA
                            )

            # Curl counter logic
            if angle > 160:
                stage = "down"
            if angle < 30 and stage =='down':
                stage="up"
                counter +=1
                print(counter)

        except:
            pass

        # Render curl counter
        # Setup status box
        cv2.rectangle(image, (0,0), (225,73), (245,117,16), -1)

        # Rep data
        cv2.putText(image, 'REPS', (15,12),
                    cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0,0,0), 1, cv2.LINE_AA)
        cv2.putText(image, str(counter),
                    (10,60),
                    cv2.FONT_HERSHEY_SIMPLEX, 2, (255,255,255), 2, cv2.LINE_AA)

        # Stage data
        cv2.putText(image, 'STAGE', (65,12),
                    cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0,0,0), 1, cv2.LINE_AA)
        cv2.putText(image, stage,
                    (60,60),
                    cv2.FONT_HERSHEY_SIMPLEX, 2, (255,255,255), 2, cv2.LINE_AA)
```

```
        # Render detections
        mp_drawing.draw_landmarks(image, results.pose_landmarks,
mp_pose.POSE_CONNECTIONS,
                                mp_drawing.DrawingSpec(color=(245,117,66), thickness=2,
circle_radius=2),
                                mp_drawing.DrawingSpec(color=(245,66,230), thickness=2,
circle_radius=2)
                                 )

        cv2.imshow('Mediapipe Feed', image)

        if cv2.waitKey(10) & 0xFF == ord('q'):
            break

    cap.release()
    cv2.destroyAllWindows()
```

REFERENCES

[1] A report titled "AI-based Fitness Trainer Market Size, Share & Trends Analysis Report by Type (Hardware, Software), By End User (Home Users, Gym & Health Clubs), By Region (North America, APAC), And Segment Forecasts, 2019 - 2025" was published in 2018. (2019) Grand View Research, Inc.

[2] Forbes (2020) article titled "How AI is Changing the Fitness Industry" Capabilities And Challenges Of AI-Enabled Fitness Applications https://www.forbes.com/sites/forbestechcouncil/2023/09/13/capabilities-and-challenges-of-ai-enabled-fitness-applications/?sh=2ccf20e863ce

[3] According to PCMag (2021), "The Best AI-Based Fitness Apps for Home Workouts" https://www.pcmag.com/picks/best-workout-apps

[4] Built In (2021): "The Rise of AI in Fitness: From Virtual Coaches to Personalized Workouts" AI applications such as virtual coaches, personalized workout recommendations, and data analysis for optimizing fitness routines. https://www.freebeatfit.com/blogs/brand-story/ai-fitness-coach

[5] "AI-Powered, Personalized, and Surprisingly Discreet: The Future of Fitness" 2021 - Fast Company https://www.ijirset.com/upload/2024/april/25_AI.pdf

[6] Gourangi Taware, Rohit Agrawal, Pratik Dhende, Prathamesh Jondhalekar, Shailesh Hule(2021).Al Based Workout Assistant and Fitness Guide.

[7] Jamie Shotton, Andrew Fitzgibbon, Mat Cook, Toby Sharp, Mark Finocchio, Richard Moore Alex Kipman, Andrew Blake (2011). Real-Time Human Pose Recognition in Parts from Single Depth https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/BodyPartRecognition.pdf

[8] Yeji Kwon, Dongho Kim (2022).Real-Time Workout Posture Correction using OpenCV and MediaPipe. https://www.researchgate.net/publication/358421361 R ealTime Workout Posture Correction using OpenCV and MediaPipe