RV College of Engineering®
Mysore Road, RV Vidyaniketan Post,
Bengaluru - 560059, Karnataka, India

Go, change the world®

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

Project Report

On

AI-Powered Exercise Tracking and Health Management

Submitted in partial fulfilment of the requirements for the V Semester

ARTIFICIAL NEURAL NETWORK AND DEEP LEARNING

AI253IA

By

1RV22AI032	Nishanth H R
1RV22AI029	Nandeesh C M
1RV22AI039	Preetham N

Department of Artificial Intelligence and Machine Learning
RV College of Engineering®
Bengaluru – 560059

Academic year
2024-25

RV COLLEGE OF ENGINEERING®

(Autonomous Institution Affiliated to Visvesvaraya Technological University, Belagavi)

DEPARTMENT OF

ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

Bengaluru– 560059

CERTIFICATE

This is to certify that the project entitled "**AI-Powered Exercise Tracking and Health Management"** submitted in partial fulfillment of Artificial Neural Networks and Deep Learning (21AI63) of V Semester BE is a result of the bonafide work carried out by (1RV22AI032) Nishanth H R, (1RV22AI029) Nandeesh C M and (1RV22AI039) Preetham N during the Academic year 2024-25

Faculty In charge

Date :

Head of the Department

Date :

RV COLLEGE OF ENGINEERING®

(Autonomous Institution Affiliated to Visvesvaraya Technological University, Belagavi)

DEPARTMENT OF
ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

Bengaluru– 560059

DECLARATION

We, (1RV22AI032) Nishanth H R, (1RV22AI029) Nandeesh C M and (1RV22AI039) Preetham N students of Fifth Semester BE hereby declare that the Project titled **"AI-Powered Exercise Tracking and Health Management**" has been carried out and completed successfully by us and is our original work.

Date of Submission: **Signature of the Student**

ACKNOWLEDGEMENT

We are profoundly grateful to our guide, **Dr. Somesh Nandi,** Assistant Professor, RV College of Engineering, for his wholehearted support, valuable suggestions, and invaluable advice throughout the duration of our project. His guidance and encouragement were instrumental not only in the successful completion of the project but also in the preparation of this report.We also extend our special thanks to **Dr. Anupama Kumar** for her invaluable insights, support, and constructive feedback, which significantly contributed to the improvement of our work.

We would like to express our sincere thanks to our Head of the Department, **Dr. Satish Babu,** for his constant encouragement and for fostering an environment of innovation and learning that greatly aided our progress.

We extend our heartfelt gratitude to our beloved Principal, **Dr. K. N. Subramanya,** for his unwavering appreciation and support for this Experiential Learning Project, which motivated us to give our best.

Lastly, we take this opportunity to thank our family members and friends for their unconditional support and encouragement throughout the project. Their backup and motivation were crucial in helping us overcome challenges and successfully complete our work.

ABSTRACT

The AI-Based Gym Trainer is an advanced fitness management system that combines computer vision, deep learning, and artificial intelligence to enhance workout efficiency and nutritional tracking. The system utilizes MediaPipe Pose and BlazePose for real-time human pose estimation, ensuring users maintain proper posture and reducing the risk of injuries. Additionally, YOLOv8, a cutting-edge object detection model, enables accurate food recognition and composition analysis, allowing users to monitor their dietary intake effectively.

For seamless user interaction, the system incorporates a graphical user interface (GUI) built with Tkinter and Flask, offering an intuitive platform to track exercise performance, BMI calculations, and personalized health recommendations. MySQL is employed for secure data management, ensuring efficient storage and retrieval of user profiles, workout logs, and nutritional data.

The system addresses key challenges in fitness training by providing automated repetition counting, real-time form correction, and customized workout plans tailored to users' BMI and fitness goals. The experimental evaluation highlights the system's high accuracy in pose estimation, effective workout tracking, and precise food composition analysis. While performance may be influenced by factors such as webcam quality and lighting conditions, the AI-Based Gym Trainer remains a cost-effective, scalable, and intelligent solution for modern, data-driven fitness management.

Furthermore, the integration of computer vision and AI-driven analytics ensures that users receive instant feedback on their workouts and dietary habits, bridging the gap between traditional fitness coaching and technology-driven solutions. By leveraging real-time tracking and adaptive learning, the system continuously improves accuracy and personalization, making it a versatile tool for fitness enthusiasts, athletes, and beginners alike. With its robust architecture and user-centric design, the AI-Based Gym Trainer paves the way for a more accessible, data-driven, and efficient approach to personal fitness management.

Table of Contents

List of Figures

Chapter 1 INTRODUCTION

1.1 Project Description

The AI-Based Gym Trainer is an innovative fitness management system that leverages cutting-edge computer vision and artificial intelligence technologies to enhance workout efficiency and personal health management. Designed to bridge the gap between traditional fitness solutions and modern technology, this system provides users with real-time feedback, nutritional analysis, and personalized fitness recommendations.

Using **MediaPipe Pose** and **YOLOv8**, the trainer monitors exercise form, counts repetitions, and analyzes food composition with high accuracy. The system also features a **BMI calculator** that generates personalized workout and dietary plans tailored to users' fitness goals, such as weight loss, maintenance, or muscle gain. Through an intuitive interface powered by **Tkinter** and **Flask**, users can interact seamlessly with the system, receiving instant insights into their fitness progress.

This AI-driven solution addresses common challenges in fitness management, such as incorrect exercise form, lack of real-time feedback, and inadequate nutritional guidance. By combining real-time pose tracking, food analysis, and comprehensive health management, the system aims to promote safer and more effective workouts while making fitness accessible and affordable to all.

Objectives of the Project:

The AI-Based Gym Trainer aims to revolutionize personal fitness and health management by leveraging artificial intelligence, computer vision, and advanced data processing techniques. The key objectives of the system are:

- Provide real-time feedback on exercise form using pose estimation to reduce injury risks and improve posture accuracy.
- Utilize computer vision techniques to count exercise repetitions automatically, streamlining workout tracking.
- Implement YOLOv8 for food detection and composition analysis, offering precise calorie and macronutrient details.
- Calculate and track BMI to provide tailored fitness and dietary recommendations aligned with user goals.
- Process live webcam feeds to track body poses and evaluate exercise performance in real-time.
- Develop a user-friendly GUI with Tkinter and Flask, allowing seamless interaction and feedback visualization.
- Utilize MySQL to ensure safe storage and management of user profiles, workout logs, and health data.

Theory and Concept Relevant to the Project:

1. **Computer Vision in Fitness Tracking**:
   Computer vision techniques enable the detection and tracking of body movements through video input. By employing tools like **MediaPipe Pose** and **BlazePose**, the system identifies human joints, tracks motion, and calculates angles between joints to assess posture and exercise form.
2. **Object Detection with YOLOv8**:
   YOLOv8 (You Only Look Once) is a state-of-the-art object detection model used for identifying food items. This model processes video frames or images, detects objects, and classifies them based on a pre-trained dataset, providing accurate nutritional analysis.
3. **Pose Estimation Algorithms**:
   Pose estimation involves identifying the coordinates of human body joints from video frames. Algorithms like BlazePose and MediaPipe use deep learning to map body landmarks in real-time, enabling precise feedback on exercise form and automatic repetition counting.
4. **Body Mass Index (BMI) Calculation**:
   BMI is a simple metric used to assess body weight in relation to height. It is calculated using the formula:

$$\text{BMI} = \frac{\text{Weight (kg)}}{\text{Height (m)}^2}$$

Figure 1.1:formula for BMI calculation

This helps classify individuals into categories such as underweight, healthy weight, overweight, or obese and forms the basis for personalized health recommendations.

5. **Deep Learning and Neural Networks**:
   Deep learning models like convolutional neural networks (CNNs) are integral to pose estimation and food detection. These models learn patterns from large datasets, enabling accurate classification and regression tasks, which are essential for analyzing user poses and identifying food items.
6. **Nutritional Analysis**:
   Nutritional analysis involves determining the caloric and macronutrient composition of food items. The system integrates YOLOv8 outputs with a food composition database to provide users with detailed dietary insights.
7. **Database Management**:
   The system uses **MySQL** for storing and managing user data securely, including profiles, workout logs, and health statistics. This ensures scalability, data integrity, and accessibility while maintaining user privacy.
8. **Real-Time Video Processing**:
   OpenCV facilitates real-time video capture and processing from a webcam. Frames are pre-processed to enhance clarity and reduce noise before being passed to pose estimation or object detection algorithms.
9. **Graphical User Interface (GUI)**:
   The interface is designed using **Tkinter** and **Flask**, enabling users to interact with the system easily. The GUI displays real-time exercise feedback, nutritional information, BMI calculations, and health recommendations.

1.2 Report Organization

The report is structured to provide a comprehensive understanding of the AI-based gym trainer system. It begins with the **Introduction**, outlining the project's purpose, the problem it addresses, and the significance of real-time exercise form tracking. The **Literature Review** explores previous research on exercise tracking, computer vision-based fitness systems, and nutrition analysis, discussing relevant machine learning algorithms, pose estimation technologies (such as MediaPipe and YOLOv8), and their applications in fitness. The **System Design and Architecture** section explains the selection of tools and technologies like MediaPipe, YOLOv8, MySQL, and Flask, detailing the system's components, including pose detection, repetition counting, food composition analysis, and BMI calculation. The **Software Requirement Specifications** section specifies the system's general and functional requirements, as well as external interfaces.

In the **Implementation** section, the report covers the development process, from dataset preparation and feature extraction to training machine learning models for pose tracking and food detection. Challenges such as lighting conditions and camera quality are discussed, along with the solutions implemented. The **Results and Evaluation** section evaluates the system's performance based on accuracy in pose estimation, repetition counting, food detection, and system response time, comparing it with existing methods and highlighting its strengths. The **Conclusion** summarizes the findings, emphasizing the system's impact on fitness training, personalized health management, and injury prevention, showcasing the potential of AI in transforming gym training. The **Future Work** section suggests improvements, such as enhancing pose estimation, expanding the food database, and integrating mobile platforms. Lastly, the **References** section lists all scholarly sources, books, research papers, and online resources used in the development of the project.

Chapter 2 LITERATURE REVIEW

This chapter presents a literature survey on AI-based fitness systems, summarizing various computer vision, deep learning, and machine learning techniques used across multiple studies to enhance exercise tracking, posture correction, and nutritional analysis. It explores the application of pose estimation, object detection, and real-time feedback mechanisms, evaluating their effectiveness in improving workout efficiency, injury prevention, and personalized fitness recommendations.

2.1 Literature survey:

In [1], the authors present the AI Fitness Coach Solution, which utilizes **MediaPipe Pose** and **BlazePose** to evaluate exercise form in real-time via webcam, providing effective feedback to enhance workout efficiency. However, the accuracy of the system is significantly impacted by the **quality of the webcam** and **lighting conditions**, which are crucial for optimal performance. This limitation highlights the importance of hardware specifications in achieving reliable real-time exercise tracking. Despite these challenges, the system demonstrates the potential of **AI-driven feedback** for improving fitness routines.

Similarly, in [2], **LAZIER's Virtual Fitness Coach** employs **MediaPipe** for **real-time motion tracking**, enhancing workout form and helping users refine their exercises. While effective, the system faces usability challenges in **camera-restricted** or **privacy-sensitive environments**, which limits its applicability for a wide range of users. This research emphasizes the trade-off between technological advancement and real-world usability in fitness applications.

In [3], the authors introduce the concept of **AI-Enhanced Fitness Training** by leveraging **OpenCV** and **Flask** to deliver **real-time feedback** and personalized workout plans. However, the performance of the system is **dependent on high-quality cameras** and **stable internet connections**, making it less adaptable to diverse user environments. This paper underlines the need for **robust and flexible fitness systems** that can function effectively with varying hardware and network conditions.

In [4], the **AI Fitness Trainer** offers customized feedback on exercise form and repetition counts, aiming to improve home workout efficiency. However, the system struggles with **limited exercise variety**, **reliance on technology**, and **privacy concerns** associated with camera usage. These challenges highlight the complexities of creating a comprehensive fitness solution that is both effective and privacy-conscious for home users.

The study in [5] presents the **Robust Intelligent Posture Estimation** system, which combines **OpenCV**, **MediaPipe**, **ChatterBot**, and **Flask** to estimate posture and generate workout plans. While it provides effective results, the system lacks **real-time query capabilities** and motivational features for **user engagement**, limiting its potential for sustained use in fitness tracking applications. This gap points to the importance of incorporating user engagement elements in AI fitness systems.

In [6], the **Virtual Personal Trainer via Kinect** utilizes a **Kinect sensor** to track **20 body joints** during exercises and employs **machine learning algorithms** to provide real-time feedback for posture correction. However, the system's reliance on specialized hardware limits its **accessibility** for general users, especially in non-laboratory settings. This highlights the need for more **universal and affordable fitness solutions** that can be used in everyday environments.

The study in [7], **Human Posture Estimation**, focuses on advanced algorithms for detecting and analyzing body landmarks, ensuring accurate posture estimation for fitness and health applications. The paper emphasizes the importance of precise posture tracking but also notes that **real-world scenarios** often present challenges in achieving consistent accuracy due to varying environmental conditions.

In [8], the **Pose Trainer for Correcting Exercise Posture** leverages **OpenCV**, **TensorFlow**, and **MediaPipe** for accurate pose detection and body landmark identification. The system provides **real-time feedback** on exercise forms to reduce injury risks and improve workout efficiency. However, challenges like **camera quality**, **lighting conditions**, and **user adherence** to system recommendations affect the overall effectiveness, suggesting that further refinement of the system's robustness in real-world settings is needed.

The paper in [9], **BlazePose: Real-Time Body Pose Tracking**, introduces a **deep learning model** for **3D human pose estimation** using **BlazePose**. This architecture is designed to balance high **accuracy** with **computational efficiency**, enabling real-time processing even on devices with **limited hardware capabilities**. The system is highly effective for real-time applications, but the **trade-off between computational load and accuracy** remains a key consideration for further optimization.

In [10], the authors provide a comprehensive survey on the **progress of human pose estimation**, categorizing advancements in **2D human pose estimation models** and exploring their applications in **fitness and health monitoring**. The study emphasizes the **trade-off between accuracy and computational efficiency**, noting that achieving real-time adaptability across various devices and environments continues to be a significant challenge.

In [11], the study on **Human Pose Estimation and Activity Recognition** explores **multi-view video data** for **pose estimation** and **activity recognition** using advanced machine learning techniques. The reliance on **multi-view setups**, however, limits scalability for **real-world, single-camera environments**, making it less practical for everyday fitness tracking applications. This highlights the ongoing challenge of **scalability** and **adaptability** in pose estimation systems.

In [12], the **Pose Trainer** system integrates **TensorFlow** and **MediaPipe** to track body landmarks in real-time, offering corrective feedback for exercise posture. This system aims to **reduce injury risks** and enhance workout effectiveness but faces **limitations due to camera quality**, **lighting conditions**, and user adherence to feedback. These constraints suggest that there is still room for improvement in the **robustness** and **user-friendliness** of pose estimation systems for fitness applications.

These studies collectively demonstrate the growing potential of **AI-powered fitness systems** for real-time exercise tracking and form correction. However, they also highlight the persistent challenges in ensuring **reliable performance** across diverse environments, including the need for **high-quality hardware**, **stable network conditions**, and **user engagement**. Future research should focus on improving **accuracy**, **scalability**, and **usability** to make these systems more accessible and effective for a wide range of users.

2.2 Summary of the literature survey:

Unresolved Issues and Emerging Opportunities:

- **Accuracy in Varied Environments:** The system's performance heavily depends on factors such as webcam quality, lighting, and the user's environment, which can lead to inaccuracies in pose estimation and exercise feedback.
- **Camera Dependency:** Many solutions rely on high-quality cameras for real-time pose tracking, making them unsuitable for environments where camera use is prohibited or limited.
- **Privacy Concerns:** The use of cameras for continuous tracking may raise privacy issues, especially in sensitive or shared spaces.
- **Exercise Variety Limitation:** Existing systems may not support a wide variety of exercises, limiting their application for users with different workout routines.

Conclusion of the Literature Survey:

The literature review on AI-powered fitness systems reveals significant advancements in real-time exercise tracking, posture correction, and personalized workout guidance using computer vision and AI technologies. Systems such as those utilizing MediaPipe Pose, BlazePose, and OpenCV provide real-time feedback on exercise form, offering promising solutions for improving workout efficiency and safety. However, challenges such as camera quality dependency, privacy concerns, and limited exercise variety persist across most solutions. Additionally, issues like hardware requirements, usability in camera-restricted environments, and internet connectivity limitations remain unresolved.

Objectives of the Project:

The objectives of the project are:

1. Real-time Exercise Form Evaluation: To provide accurate, AI-driven real-time feedback on exercise form and repetition counting, ensuring correct posture during workouts to prevent injury.
2. AI-Powered Food Composition Analysis: To develop a system that uses object detection (YOLOv8) to identify and analyze food items for nutritional information, aiding users in managing their diet.
3. Personalized Fitness and Health Recommendations: To offer personalized workout and diet plans based on users' BMI and fitness goals, such as weight loss or muscle gain.
4. User-friendly Interface: To design an intuitive interface using Tkinter and Flask, making it easy for users to interact with the system and track their workout progress, nutritional intake, and health data.
5. Integration of Real-time Feedback: To integrate real-time feedback on exercise performance, such as repetition counting and posture accuracy, helping users optimize their workout efficiency and safety.

2.3 Existing and Proposed System

Existing system:

Problem Statement:

Traditional fitness training methods often lack real-time feedback, making it difficult for users to maintain proper posture, track repetitions, and receive personalized workout guidance. Many existing fitness applications rely on manual logging of exercises, which can be inaccurate and inconvenient. Additionally, nutritional tracking is often limited to text-based food logging, lacking automated food recognition and composition analysis.

Moreover, systems that do offer AI-based exercise tracking and posture correction are typically hardware-dependent, requiring expensive sensors or specialized equipment such as Kinect or wearable motion trackers. Privacy concerns also arise due to the continuous use of webcams and data storage policies. Furthermore, lighting conditions, camera quality, and limited exercise variety affect the accuracy of existing AI-based fitness systems.

Due to these challenges, there is a need for a cost-effective, scalable, and intelligent AI-powered gym trainer that provides real-time exercise tracking, posture correction, repetition counting, and food composition analysis without requiring additional hardware.

Proposed System:

Problem Statement and Scope of the Project

Existing fitness tracking systems lack real-time feedback, automated exercise monitoring, and integrated dietary analysis, making workouts less effective and increasing the risk of improper form and injuries. Many traditional and AI-based fitness applications rely on manual data entry, reducing tracking accuracy and user engagement. Moreover, posture correction and repetition counting in current AI solutions often require expensive sensors or specialized hardware, making them less accessible. Additionally, nutritional analysis in existing systems is either manual or text-based, with no real-time food recognition and composition analysis. Factors such as camera quality, lighting conditions, and privacy concerns further impact the reliability of AI-based fitness tracking. Hence, there is a need for a cost-effective, AI-driven fitness trainer that provides real-time exercise tracking, posture correction, repetition counting, and food composition analysis within an accessible and user-friendly platform.

Scope of the Project:

This project aims to develop an AI-powered fitness system that combines real-time exercise tracking, posture correction, and food composition analysis. By utilizing MediaPipe Pose for posture estimation and YOLOv8 for food recognition, the system enhances workout efficiency and safety. Additionally, it offers personalized workout plans and dietary recommendations based on user health data and goals.

Key Features:

- Real-time feedback on exercise form using MediaPipe Pose and YOLOv8.

- Automatic repetition counting to track workout progress.
- Nutritional analysis of food items via object detection.
- BMI calculation and personalized health recommendations.
- Secure data management and sto rage with MySQL for user profiles and workout logs.

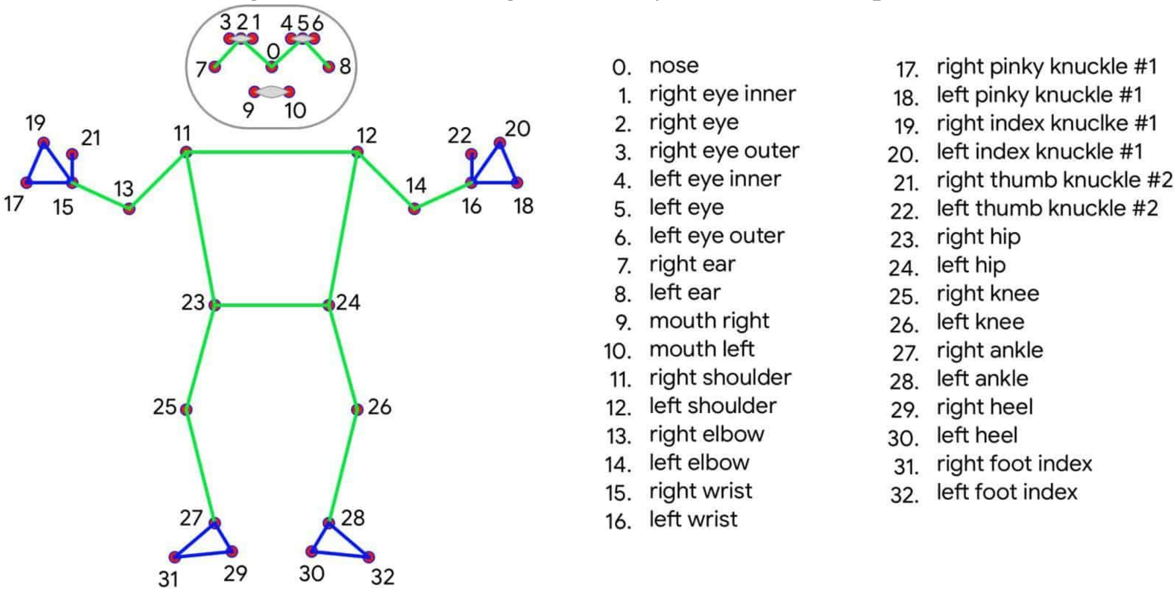

Figure 2.1:mediapipe

Methodology Adopted in the Proposed System:

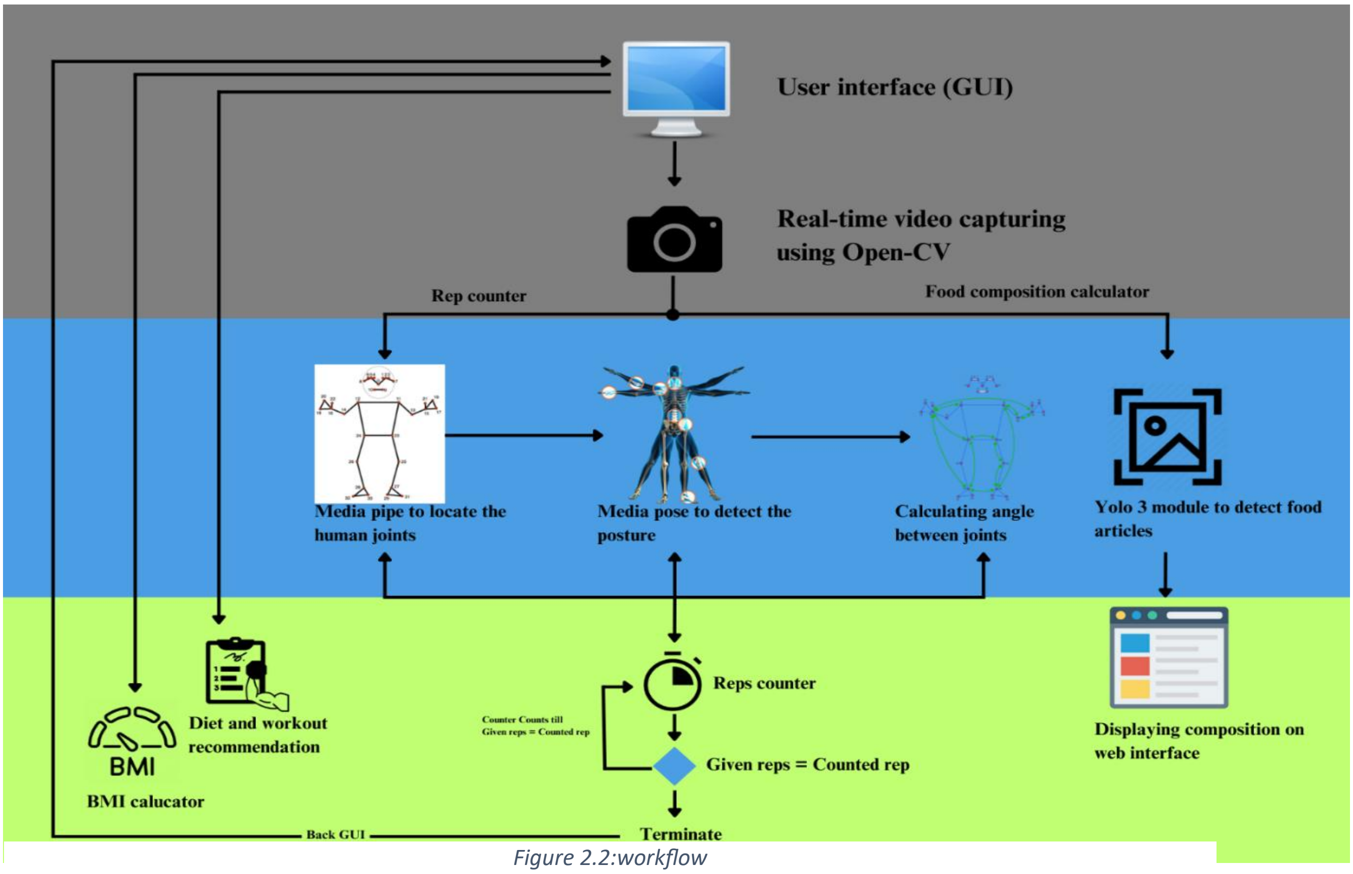

Figure 2.2:workflow

The proposed system follows a structured methodology to ensure accurate exercise tracking, real-time feedback, and personalized recommendations. It integrates computer vision for pose estimation, object detection for food analysis, and a secure database management system for storing user data. Below is a detailed breakdown of the steps involved in the system's methodology:

1. **User Input and Data Collection**:The system begins by collecting basic user information, such as height, weight, and fitness goals (e.g., weight loss, muscle gain). These details are essential for calculating Body Mass Index (BMI) and generating personalized workout and diet plans. The system also gathers exercise-related data in real-time by capturing video from the user's webcam during workout sessions. The webcam serves as the primary tool for collecting visual data of the user's movements, which are analyzed for pose estimation.
2. **Real-Time Pose Estimation**:To assess the user's exercise form and ensure proper posture, MediaPipe Pose is utilized. This technology tracks the user's body in real-time by detecting key joint coordinates and angles. Each joint is identified based on the video input, and its position is continuously updated during the workout. The joint angles and their relative positions help evaluate the accuracy of the exercise form. This step is critical for ensuring that users perform exercises correctly, thus reducing the risk of injury.
3. **Exercise Tracking and Rep Counting**:Once the pose estimation is complete, the system identifies key postures associated with specific exercises (e.g., squats, push-ups). The system continuously tracks the user's body and counts the number of repetitions based on when the user performs the correct movement. For instance, in a squat, when the user bends their knees to a specific angle and stands back up, the system counts it as one repetition. This allows for an automatic tracking of the exercise without requiring manual input, offering a more efficient workout tracking experience.
4. **Nutritional Analysis**:The system includes a food composition calculator powered by YOLOv8, an advanced object detection model. When users capture images of food items with their webcam, YOLOv8 detects the food and classifies it into categories (e.g., fruits, vegetables, meat). Once identified, the system pulls nutritional data such as calories, protein, carbs, and fats from a pre-built database. This feature offers users real-time analysis of their diet and helps in managing nutrition alongside their exercise routine. By integrating object detection with nutritional analysis, the system provides more holistic health tracking.
5. **Data Management and Storage**:All the data collected, including user profiles, workout logs, exercise performance, nutritional data, and BMI calculations, are securely stored in a MySQL database. The use of MySQL ensures that the data is efficiently managed, queried, and updated. For instance, the system can store historical workout logs, track progress over time, and retrieve personalized recommendations based on past performance. The database also allows for easy user profile management and ensures that all the health data is accessible and securely stored for future reference.
6. **Feedback and Recommendations**:As users perform exercises, the system continuously provides feedback on their form and rep count. This feedback is displayed through a graphical interface, allowing users to see real-time accuracy and posture improvement. The system also uses the BMI and personal health goals to generate customized workout and diet recommendations. These suggestions are displayed on a user-friendly interface built using Tkinter and Flask. For example, if a user is trying to gain muscle, the system might recommend a protein-rich diet and strength training exercises. Real-time feedback helps users stay engaged with their fitness journey, making the entire process more interactive and informative.

7. **Integration with Web Interface**:The system integrates with a web interface for easy access to all features. Flask is used for creating the backend, which communicates with the user interface to display results, exercise feedback, and recommendations. The web interface also allows users to interact with the system, view their data, and input new information such as food images and workout logs. Tkinter, a Python library for building graphical user interfaces, enhances the system by providing real-time data visualization, such as the progress bar for pose accuracy and rep counting. This ensures that the user has a smooth and comprehensive experience.
8. **Security and Privacy Considerations**:Since the system involves collecting personal data and video inputs, security is a primary concern. All user information, including health data and exercise logs, is securely stored and managed using MySQL. The system ensures privacy by restricting access to user data, enabling encrypted communication between the frontend and backend. Furthermore, users have control over their data and can choose whether to share specific details or delete records. These security measures protect the integrity of the system and ensure that user data is handled responsibly.

2.4 Tools and Technologies Used

Platform / Tools Used in Implementing the Project

The proposed system leverages a combination of cutting-edge technologies and tools to ensure smooth functionality and high performance in real-time exercise tracking, nutritional analysis, and user engagement. These tools support efficient pose estimation, object detection, database management, and user interface development. Below are the key platforms and technologies used in the implementation of the project:

- **Python**: The primary programming language for the development of the system, enabling integration with libraries for pose estimation, image processing, and web interface.
- **MediaPipe**: A cross-platform framework used for real-time pose estimation to track and evaluate exercise form.
- **OpenCV**: A computer vision library used for handling video feeds, frame extraction, and processing for pose detection and object recognition.
- **YOLOv8**: An object detection model used for real-time food analysis, identifying food items through webcam input for nutritional composition analysis.
- **MySQL**: A relational database management system used for securely storing user profiles, workout logs, health data, and food composition data.
- **Flask**: A web framework used to create the back-end server for user interaction, allowing integration with MySQL for dynamic content delivery and providing an interface for users to view workout results and nutritional data.
- **Tkinter**: A GUI toolkit used to build a simple and user-friendly front-end interface for displaying real-time exercise feedback, BMI calculations, and workout recommendations.
- **TensorFlow / PyTorch (Optional)**: Libraries for deep learning-based pose estimation and food recognition, ensuring robust model performance for exercise tracking and food composition analysis.
- **Numpy and Pandas**: Libraries for numerical operations and data manipulation, supporting processing and analysis of fitness and nutritional data.
- **Chart.js**: A JavaScript library used for rendering interactive charts in the web interface, displaying visualizations of workout and nutritional data over time.

2.5 Hardware and Software Requirements

Hardware Requirements

The hardware requirements for the project are as follows:

- Processor: Intel Core i5 (8th Gen or newer) / AMD Ryzen 5 (or equivalent)
- RAM: 8 GB (16 GB recommended for smoother development and testing)
- Storage: 256 GB SSD (for faster application development and data processing)
- Graphics Card: Integrated graphics (Intel HD or AMD equivalent) should be sufficient. However, a dedicated GPU (e.g., NVIDIA GTX 1050 or higher) can improve MediaPipe performance in real-time pose estimation.
- Webcam: High-resolution webcam (720p or higher) for real-time tracking and pose estimation testing.
- Operating System: Windows 10 or 11 (64-bit), macOS Catalina or newer, or Linux (Ubuntu 20.04 or newer)
- Other Peripherals: Keyboard, Mouse, and Monitor (1080p or higher resolution recommended)

Software Requirements

1. Programming Languages:
   - Python (for implementing the real-time object detection, pose estimation, and database management)
   - HTML, CSS, and JavaScript (for front-end development of the web interface)
2. Libraries/Frameworks:
   - MediaPipe (for real-time pose estimation and exercise tracking)
   - OpenCV (for image processing and computer vision tasks)
   - Flask (for web server development and handling real-time data interaction)
   - NumPy (for numerical data processing)
   - Pandas (for data manipulation, especially in managing the exercise and food logs)
   - TensorFlow/PyTorch (for machine learning-based models if required in advanced analysis)
   - Chart.js (for interactive visualization of nutritional analysis)
3. Database:
   - MySQL (for storing and managing user profiles, workout logs, and food composition data)
   - PHPMyAdmin (for easier database management and visualization)
4. Version Control:
   - Git (for version control and collaboration)
   - GitLab (for repository management and collaboration)
5. Development Environment:
   - VS Code (for code development and debugging)
   - Postman (for testing the Flask API and web server endpoints)
6. Operating System:
   - Windows 10 or 11 (64-bit), macOS Catalina or newer, or Ubuntu 20.04 or newer.

Chapter 3 SOFTWARE REQUIREMENT SPECIFICATIONS

This chapter introduces to definitions, acronyms and abbreviations used in the report , additionally it gives the general description of the product . It also describes the functional ,non functional requirements and external interface requirements.

3.1 Introduction

Definitions, Acronyms, and Abbreviations

- AI: Artificial Intelligence – The simulation of human intelligence processes by machines, particularly computer systems.
- BMI: Body Mass Index – A measurement of body fat based on height and weight.
- GUI: Graphical User Interface – A user interface that includes graphical elements like icons, buttons, and windows.
- ML: Machine Learning – A subset of AI that allows systems to learn from data and make predictions or decisions without being explicitly programmed.
- YOLO: You Only Look Once – A popular real-time object detection system for identifying objects in images and videos.
- Mediapipe: A framework used for building multimodal applied machine learning pipelines, such as pose detection and real-time video processing.
- MySQL: A relational database management system used for storing and managing structured data.
- Flask: A lightweight web framework for building web applications in Python.
- Pose Estimation: The process of detecting and analyzing human body postures from images or video streams.
- API: Application Programming Interface – A set of rules and protocols for building and interacting with software applications.

Overview

StrideSync is an AI-powered Exercise Tracking and Health Management System that integrates AI, computer vision, and web technologies to provide real-time exercise feedback, meal tracking, and personalized workout plans. Using tools like YOLO for object detection, Mediapipe for pose estimation, and MySQL for data management, the system offers a user-friendly web interface for tracking progress and receiving AI-driven insights. StrideSync aims to empower users by improving their fitness routines, offering nutritional guidance, and enabling informed decisions about diet and exercise to promote healthier lifestyles.

3.2 General Description

Product Perspective

StrideSync is an AI-powered fitness solution that combines real-time exercise tracking with nutrition analysis, aimed at providing users with personalized workout plans and meal recommendations. It integrates advanced technologies like YOLOv8 for food detection and Mediapipe for pose estimation, ensuring accurate feedback for both workouts and diet. The system operates on a web-based platform using Flask for the user interface and MySQL for database management. Designed to be accessible across devices, StrideSync aims to offer an all-in-one solution for fitness enthusiasts looking to track their physical activity and nutrition efficiently.

Product Functions

- **Real-Time Exercise Tracking**: Utilizes Mediapipe for pose estimation to monitor and provide feedback on workout form, ensuring users perform exercises correctly.
- **Food Detection and Nutritional Analysis**: Leverages a custom-trained YOLO model to identify food items from camera input and provides detailed nutritional information through an interactive database.
- **Personalized Workout Plans**: Generates customized fitness plans based on user data, offering workout recommendations tailored to individual goals and progress.
- **BMI Calculation and Health Management**: Calculates the user's Body Mass Index (BMI) and offers insights into overall health, helping users manage their fitness journey.
- **Data Storage and Management**: Secures user data (workout logs, food logs, etc.) with MySQL, ensuring that the system retains historical information for continuous improvement and tracking.
- **Interactive Web Interface**: A user-friendly interface, developed using Flask and integrated with a real-time feedback system, enhances the overall experience for tracking exercises and meals.

User Characteristics

- **Fitness Enthusiasts**: Individuals actively engaging in workouts, seeking real-time feedback on their exercise form, performance, and progress to enhance their fitness routines.
- **Health-Conscious Individuals**: Users focused on maintaining or improving their overall health, utilizing the system for BMI tracking, personalized workout plans, and nutritional insights.
- **Beginner and Intermediate Exercisers**: Users who are new to fitness or have intermediate experience, looking for guidance, corrective feedback, and motivation to ensure they perform exercises correctly and avoid injuries.
- **Diet-Conscious Users**: People who need assistance in tracking their food intake, receiving nutritional information, and integrating diet management with their exercise plans for a holistic approach to fitness.

General Constraints

1. **Hardware Dependence**: Performance is influenced by the quality of the webcam, lighting, and the user's system specifications.
2. **Accuracy Issues**: Pose detection and food composition analysis may have inaccuracies, especially with poor camera angles or complex foods.
3. **Privacy Concerns**: Continuous webcam usage raises potential privacy issues, requiring secure data management.

Assumptions and Dependencies

- **Assumed Camera Quality**: The system assumes that the user has access to a webcam with at least 720p resolution for accurate pose tracking and food composition analysis.
- **Accurate Input Data**: The system relies on the user providing accurate height, weight, and food image inputs for accurate BMI calculation and nutritional analysis.
- **External Library Dependencies**: The system assumes the availability of required libraries like MediaPipe, OpenCV, YOLOv8, and MySQL to function properly for pose tracking, image processing, and database management.

3.3 Functional Requirements

1.User Registration and Authentication

Input: Name of the user (8 Characters), Date of birth (DD/MM/YYYY), email id(abc@gmail.com), phone (11 digit), age. During the sign in process and while login use email and password.

Process: After providing the necessary details, the user successfully created their account and can log in using their password.

Output: After the authentication user can login to the page.

2.User Profile Management:

Input: Users can input and update personal details like age, weight(in kg), height(in m), and fitness goals. Store and retrieve user profiles securely. He can update some of these data like weight, height, and fitness goals.

Output: Displays the data of the user.

3.Exercise Tracking

Input: Selection of required exercise type, then specifying the number of reps that he/she wanted perform.

Process: Camera will open after selecting the requirements then it capture and track the exercise.

Output: Displayed on the UI in real-time, showing the number of completed repetitions. Helps the user track their progress through each set of exercises

4.Workout Log Management

Input: Displayed on the UI in real-time, showing the number of completed repetitions. Helps the user track their progress through each set of exercises.

Output: Users can view and update workout logs, including exercises performed, sets, reps, and timestamps. Store workout history and retrieve records for analysis and progress tracking.

5. BMI Calculator and Health Metrics

Input: Users need to enter the height and weight.

Process: Uses the BMI calculation formula: (BMI=height (m)/2weight (kg))

Output: Calculate BMI based on user's input details and display relevant health insights.

6. Nutritional Information Tracking

Input: Users need to scan the food articles using camera and he need to enter the amount (in grams) of the food

Process: Scan the food using open cv and uses yolov8 model to display the nutritional information.

Output: Displays the nutritional information in grams along with the bar graph

3.4 External Interfaces Requirements

1. **Webcam Interface**: The system requires a webcam interface for real-time video capture to track user poses and detect food items for analysis. The webcam feed will be processed using OpenCV for pose estimation and YOLOv8 for food detection.
2. **Database Interface (MySQL)**: The system interacts with a MySQL database to store and retrieve user profiles, workout logs, food composition data, and personalized health recommendations. This interface ensures secure data storage and retrieval for users' health tracking.
3. **User Interface (GUI)**: The user interface is designed to display workout feedback, BMI results, food composition details, and personalized recommendations. It communicates with the system's backend (via Tkinter and Flask) to provide a seamless user experience.
4. **External Libraries and APIs**: The system depends on external libraries such as MediaPipe for pose estimation, OpenCV for video capture and image processing, YOLOv8 for food detection, and Flask for web-based components, which should be integrated with the system for proper functionality.

3.5 Non-Functional Requirements

1. **Performance:** The system should provide real-time feedback during exercises. Database queries is optimized to ensure efficient retrieval and updates, even with large datasets**.**
2. **Scalability:** The MySQL database is able to scale to support a growing number of users and historical workout/nutritional records. The application is able to handle increased load as more features or users are added.
3. **Usability:** The interface is intuitive and easy to navigate, with clear labels and organized sections. The platform provides real-time, understandable feedback for users of all fitness levels.

4. **Security:** Encrypt user data, particularly sensitive information such as login credentials and personal details.

3.6 Design Constraints

1. **Standard Compliance**
   - Follows WHO and ACSM guidelines for health and fitness.
   - Complies with GDPR and HIPAA for data privacy.
   - Adheres to software development and web standards.
2. **Hardware Limitations**
   - Webcam quality affects pose estimation accuracy.
   - Limited by processing power for real-time video analysis.
   - Performance depends on available RAM and storage.

3.7 Other Requirements

- Stable internet connection for MySQL database syncing.
- User privacy must be maintained due to camera usage.
- Secure data handling in compliance with privacy regulations.

Chapter 4 SYSTEM DESIGN

This chapter delves into the architectural design and data flow of the AI-based gym trainer system, outlining the core components and their interactions. The system integrates various modules, including food tracking, exercise tracking, and BMI calculation, to provide users with real-time feedback and personalized recommendations. It leverages advanced computer vision techniques like MediaPipe and YOLOv8 for pose tracking and food detection. Additionally, the chapter presents a detailed dataset overview, preprocessing steps, and the system's block diagram. Finally, it includes Data Flow Diagrams (DFD) to visualize the system's functional structure and data interactions.

4.1 Architectural Design of the Project

The AI-based gym trainer system aims to enhance workout efficiency and safety by providing real-time feedback on exercise form, counting repetitions, and analyzing food composition. It utilizes computer vision technologies like MediaPipe for pose tracking and YOLOv8 for food detection. The system calculates BMI and offers personalized workout and diet recommendations based on user input.

Dataset Overview

The dataset used for training yolov8 and validation is sourced from Roboflow and consists of images related to food items, specifically 12 classes of food. The dataset is structured as follows:

- **Number of Classes**: 12
- **Class Names**: ['Apple', 'Chapathi', 'Chicken Gravy', 'Fries', 'Idli', 'Pizza', 'Rice', 'Soda', 'Tomato', 'Vada', 'banana', 'burger']
- **Number of Training Images**: 2,103
- **Number of Validation Images**: 64
- **Number of Test Images**: 41

The dataset covers various types of food items, including fruits, snacks, and main dishes. Each image is labeled with a corresponding food class, making it suitable for training a YOLOv8 model for food detect.

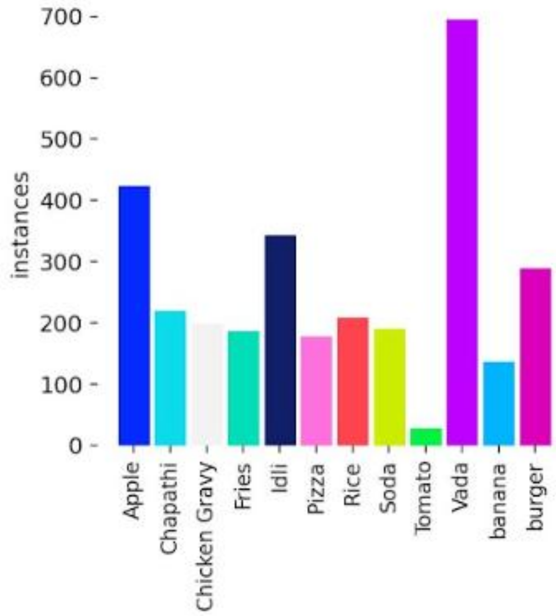

Figure 3.1:classes frequency in all the instances

Data Preprocessing Considerations

1. **Image Resizing**: Images are resized to a uniform size (e.g., 640x640 pixels) to ensure consistency across the dataset, optimizing the model's performance.
2. **Normalization**: Pixel values in images may be normalized to a range of [0, 1] to help the model learn efficiently, reducing the chance of large gradients during training.
3. **Data Augmentation**: Techniques such as flipping, rotation, scaling, and color adjustments could be applied to increase the diversity of the dataset and help the model generalize better.
4. **Annotation Format**: The dataset is likely annotated in the YOLO format, where each object (food item) is labeled with a class and bounding box coordinates.
5. **Splitting**: The dataset is divided into training and validation sets, with a separate set of test images to evaluate model performance post-training.
6. **Class Imbalance**: If there is any imbalance in the number of images per class, strategies like class weighting or oversampling the minority class might be considered to address it.
7. **GPU Utilization**: Training is performed on a T4 GPU depending on the available resources, to speed up model training.

By performing these preprocessing steps, the dataset is optimized for the YOLOv8 model, ensuring effective and efficient training and validation for food detection tasks.

Block Diagram and Module Specification

1. Food Tracking Module

- **Purpose**: Tracks the consumption of food and collects information about the weight of the food consumed.
- **Functionality**:
  - Prompts the user to input the type and weight of the food.
  - Tracks the intake to maintain a food log or calculate caloric and nutritional intake.
  - Likely integrates with the BMI module or a broader health tracking system to analyze dietary habits.

2. Exercise Tracking Module

- **Purpose**: Monitors physical activities and records details about exercises performed.
- **Functionality**:
  - Prompts the user to select the type of exercise.
  - Asks for specific inputs like **weight lifted**, **number of repetitions (reps)**, or **sets**.
  - Tracks exercise progress until the provided input matches the expected output (e.g., target reps or sets).
  - Provides feedback or updates based on the user's activity to ensure goals are being met.

3. BMI Calculation Module

- **Purpose**: Calculates the Body Mass Index (BMI) to assess an individual's health status.
- **Functionality**:
  - Uses standard BMI formula: BMI=Height (m)2Weight (kg)
  - Requires inputs such as the user's weight and height.
  - Outputs the BMI index and categorizes it (e.g., underweight, normal weight, overweight, obese).
  - Integrates with food and exercise modules to provide insights into the overall fitness and health goals.

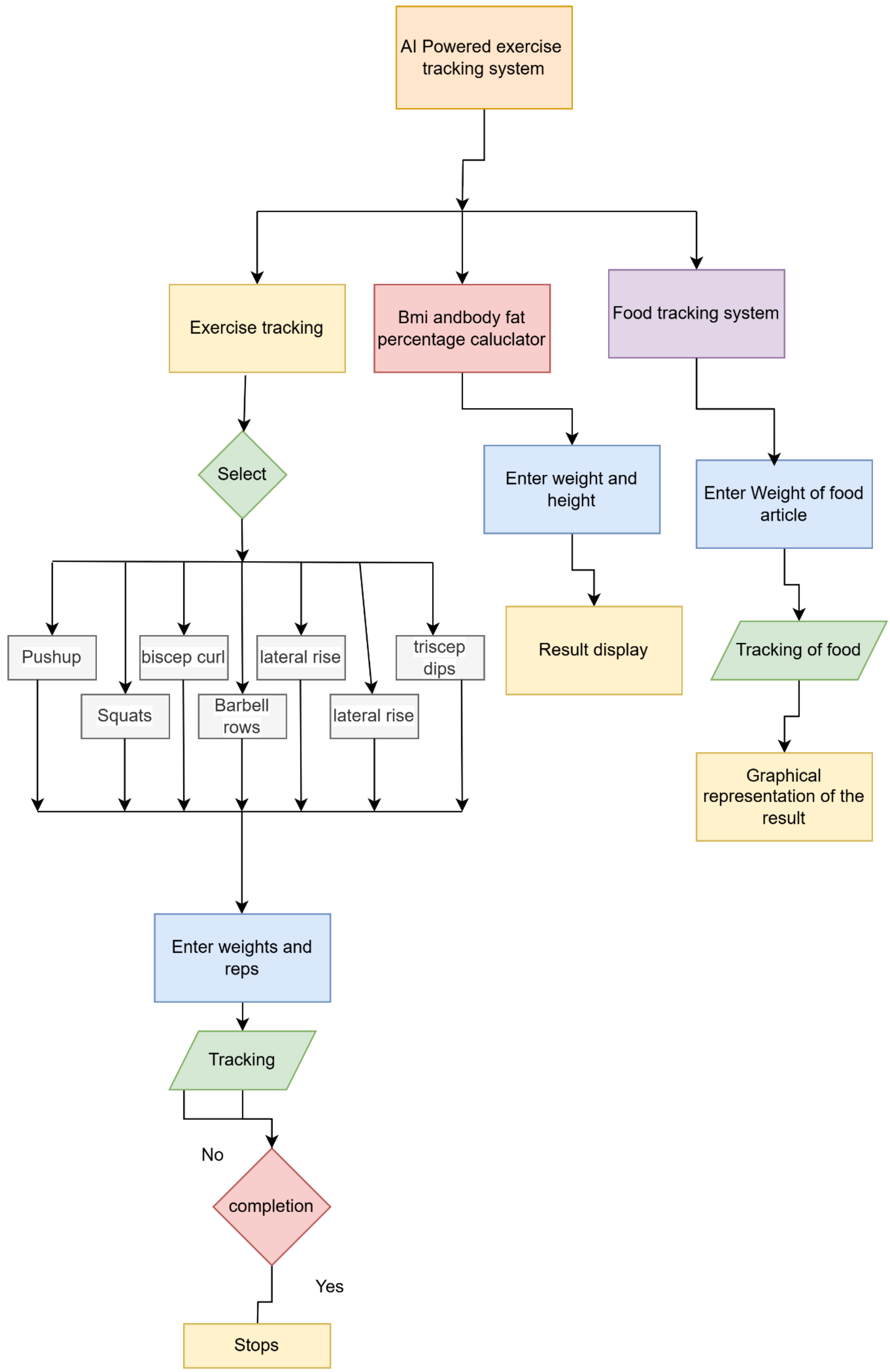

Figure 4.2 block diagram

4.2 Data Flow Diagram

Level 0 of Data flow diagram

At Level 0 Data Flow Diagram (DFD) provides a high-level overview of a system, depicting the system as a single process with its major interactions with external entities. t focuses on the system's inputs and outputs without detailing the internal workings.

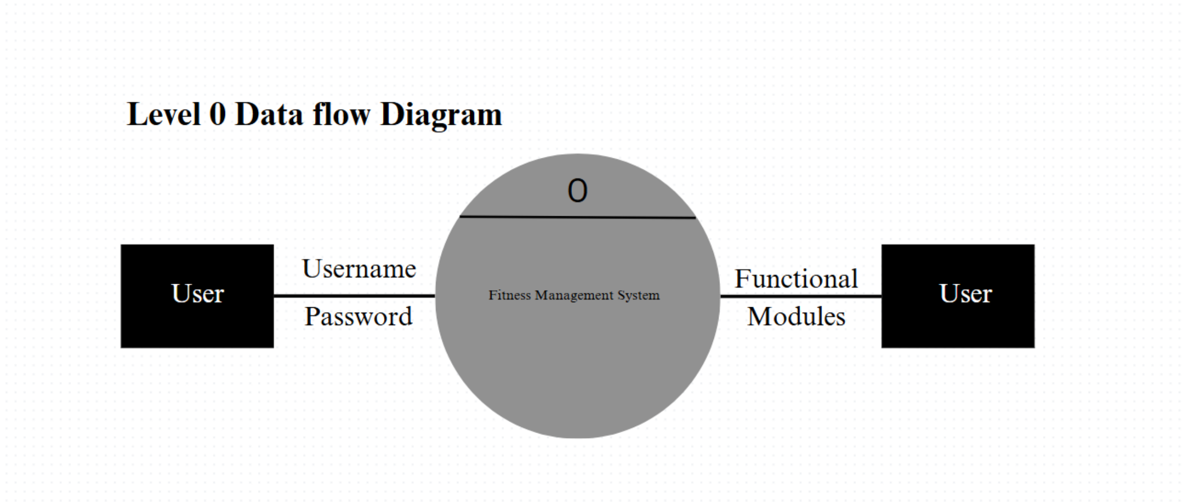

Figure 4.3: Level 0 data flow diagram

- **User Input:** The user provides their **username** and **password** to the Fitness Management System.
- **Processing:** The Fitness Management System processes the username and password.
- **Output:** The system returns **functional modules** to the user.

Level 1 of Data flow diagram

At Level 1 Data Flow Diagram (DFD) provides a high-level overview of a system's functions. It breaks down the main process into sub-processes and illustrates the flow of data between them.

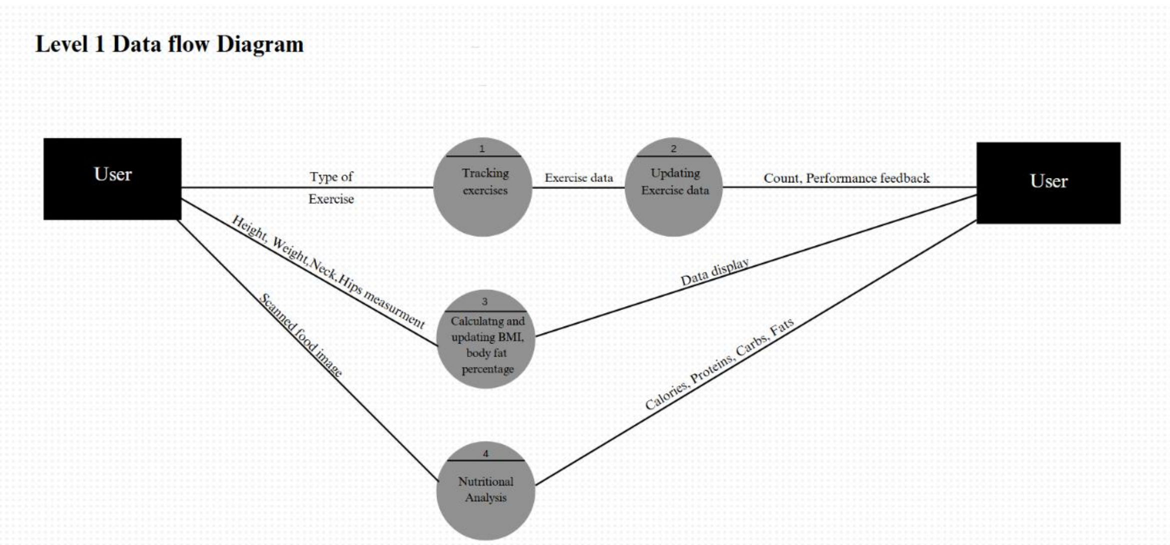

Figure 4.4: Level 2 data flow diagram

1. Tracking Exercises

- **Input:** Type of Exercise selected by the user.
- **Process**: The system tracks exercise-related information such as repetitions, time, and other performance metrics.
- **Output:** Exercise data is generated and passed to the "Updating Exercise Data" process.

2. Updating Exercise Data

- **Input:** Exercise data from the "Tracking Exercises" process.
- **Process**: The system updates the user's exercise history and performance metrics in the database.
- **Output**: Provides feedback to the user in terms of counts and performance metrics.

3. Calculating and Updating BMI and Body Fat Percentage

- **Input**: User's height, weight, neck, and hips measurements.
- **Process:** The system calculates BMI and body fat percentage using the provided measurements. Updates these values for the user profile in the system.
- **Output:** Data is displayed for user review or analysis.

4. Nutritional Analysis

- **Input:** Scanned food image uploaded by the user.
- **Process**: The system performs nutritional analysis on the scanned food image, identifying key metrics such as calories, proteins, carbohydrates, and fats.
- **Output**: Displays the nutritional data for the user, providing insights into their dietary intake.

Chapter 5: IMPLEMENTATION

This chapter details the implementation of the StrideSync system, focusing on real-time workout tracking, exercise form validation, BMI calculation, and food composition analysis. It discusses the libraries and frameworks used, such as OpenCV and MediaPipe, for video processing and pose detection. The YOLOv8 model is employed for tracking exercise movements and validating form. Testing results showcase the system's ability to provide feedback on various exercises. Lastly, the food composition analysis feature enables real-time detection and nutritional tracking of food items.

5.1 Code Snippets

5.1.1 Importing Libraries for workout tracking

```
import cv2
import mediapipe as mp
import numpy as np
import tkinter as tk
from tkinter import simpledialog, messagebox
```

```
import cv2
import mediapipe as mp
import numpy as np
```

Figure 5.1 Code snippet for importing libraries

StrideSync leverages cutting-edge AI technologies to provide **real-time exercise tracking, posture analysis, and performance monitoring**. The following libraries play a crucial role in its AI-powered fitness management system:

1. **OpenCV (cv2) – Real-Time Video Processing**
    - Captures live workout footage from a webcam or mobile camera.
    - Enhances images for better pose detection and analysis.
2. **MediaPipe (mediapipe as mp) – AI-Powered Pose Detection**
    - Identifies key body landmarks to track form and movement.
    - Helps detect incorrect postures and provides feedback for improvement.
    - mp.solutions.pose is used for skeletal tracking, while mp.solutions.drawing_utils visualizes key points.
3. **NumPy (numpy as np) – Precision in Motion Analysis**
    - Computes angles between body joints to assess posture accuracy.
    - Assists in tracking reps, movement consistency, and range of motion.
4. **Tkinter (tk) – User-Friendly Interface & Feedback System**
    - Allows users to configure workout preferences and receive AI-generated feedback.
    - simpledialog gathers input (e.g., exercise type, difficulty level), while messagebox provides form correction alerts.

5.1.2 Importing Libraries for YOLO model training

```
pip install torch torchvision torchaudio
pip install ultralytics
```

Figure 5.2 libraries for yolo model

1. PyTorch (torch, torchvision, torchaudio)

- torch: Provides deep learning capabilities, enabling the training and deployment of AI models for food detection and nutritional analysis.
- torchvision: Supports image processing tasks such as transformations, dataset handling, and model architectures, enhancing food image recognition accuracy.
- torchaudio: Facilitates audio processing, though it is not primarily used in food detection.

2. Ultralytics YOLO (YOLOv8)

- A deep learning-based object detection model used for recognizing food items in real-time.
- Efficiently detects multiple food items using pre-trained or custom-trained weights.
- Processes images in a single pass, making it fast and highly suitable for live food composition analysis.

Yolo model implementation

1. **Importing Required Libraries**:The implementation starts by importing the **Ultralytics YOLO** framework using from ultralytics import YOLO. This ensures that the necessary functions and modules for YOLO-based object detection are available for training and evaluation.
2. **Model Initialization**:A pre-trained **YOLOv8m** model is loaded using Yolo8m medium model.This initializes the medium-sized YOLOv8 model, which serves as the foundation for training on a custom dataset.
3. **Dataset Preparation**:The dataset is defined in **data.yaml**, which specifies the paths to images and labels. The dataset is stored in /content/drive/MyDrive/DATASET FOOD 2K/. This file contains information about class labels, image paths, and dataset splits (train, validation, and test).
4. **Configuring Training Parameters**:The training process is customized with the following parameters:
   - **epochs=30** → The model iterates through the dataset 30 times to refine its learning.
   - **batch=16** → Each training step processes 16 images at a time, optimizing memory and speed.
   - **imgsz=640** → Images are resized to 640x640 pixels to maintain consistency.
   - **device=0** → Training runs on **GPU** for faster processing (if available).
5. **Training Execution**:The model.train() function starts the training process. The model learns object features and refines detection accuracy through backpropagation and optimization. The process includes loss calculation, weight adjustments, and validation performance evaluation.
6. **Saving Results**:Training results, including model weights, logs, and performance metrics, are saved in the **project directory** /content/drive/MyDrive/DATASET FOOD 2K/. A subfolder named

"YOLOv8_Training" stores session-specific outputs, ensuring organized tracking of multiple training runs.

7. **Post-Training Evaluation**:After training, the model can be tested on unseen images to assess its detection accuracy. Fine-tuning may be performed by adjusting hyperparameters or augmenting the dataset for better performance.

Exercise Form validation model implementation

The system is designed to integrate multiple components that track exercise repetitions, provide workout recommendations, calculate BMI, and analyze food composition. The architecture is divided into three main sections: User Interface (GUI), Exercise Tracking and Analysis, and Food Composition Analysis.

1. Real-time Video Capturing

Real-time video capturing is a critical component, enabling exercise tracking and food composition analysis. Using OpenCV, the system captures live video from the user's camera, which is processed to detect exercise movement and food items. This dynamic real-time feedback ensures users get instant updates on their exercise repetitions and food composition, enhancing the user experience.

2. Exercise Tracking

The exercise tracking subsystem uses Mediapipe, a framework for real-time human joint detection, to ensure proper exercise form and accurate repetition counting. This subsystem detects key joints such as shoulders, elbows, hips, and knees from the video feed, enabling real-time exercise monitoring and feedback.

3. **Logic and Implementation:**

- **Pose Estimation and Joint Detection:** Mediapipe tracks the user's key joints through the real-time video feed, extracting the joint positions needed for exercise analysis. For example, in a shoulder press, Mediapipe detects the positions of the shoulder, elbow, and wrist to evaluate movement correctness.

- **Angle Calculation:** The system calculates angles between key joints to assess the correctness of exercise movements. The calculate_angle() function computes the angle formed between three points (joints) using trigonometric equations.

- **Repetition Counting Logic:** The system counts repetitions based on detected joint angles. For instance, during lateral raises, it tracks arm angles. A "full extension" (angle > 170 degrees) registers the "up" stage, while a "return" (angle < 140 degrees) marks the "down" stage and increments the repetition counter.

- **Form Validation:** The system ensures proper exercise form by validating joint angles against predefined thresholds. For example, during a shoulder press, it checks if the arm angles match correct form throughout the movement, reducing the risk of injury.

The system supports multiple exercises, each with specific logic for form validation and repetition counting:

- **Lateral Raise:** Tracks shoulder and arm angles.
- **Dumbbell Curls:** Monitors arm flexion.
- **Barbell Curls:** Focuses on elbow and wrist alignment.
- **Push-Ups:** Analyzes elbow and shoulder positions.
- **Squats:** Monitors hip and knee angles.
- **Shoulder Press:** Tracks shoulder and arm alignment.
- **Triceps Dips:** Verifies elbow extension.

This subsystem ensures accurate tracking and real-time feedback, helping users maintain proper form and safely perform their workouts.

5.2 Results and Discussions

The application’s home screen serves as a well-designed central hub, offering users an intuitive and engaging interface to manage their fitness journey. The two primary options, "Track Your Exercise" and "Track Your Food," are visually represented with clear and vibrant icons, making it easy for users to immediately understand their purpose. The "Track Your Exercise" feature allows users to monitor their workout activities, while the "Track Your Food" option focuses on dietary tracking, enabling a comprehensive approach to health and fitness.

Below these options, the interface includes additional buttons for "Workout Logs" and a "BMI Calculator," further enhancing the application’s functionality. The "Workout Logs" section allows users to record and review their past workout sessions, ensuring they can track their progress over time. The "BMI Calculator" offers a quick and accessible way for users to assess their body mass index, providing valuable insights into their overall health status.

The clean layout and user-friendly navigation prioritize simplicity and engagement, ensuring that users of all experience levels can easily interact with the application. The combination of exercise and food tracking, along with supplementary tools like the workout logs and BMI calculator, positions this platform as an all-in-one solution for personalized fitness management.

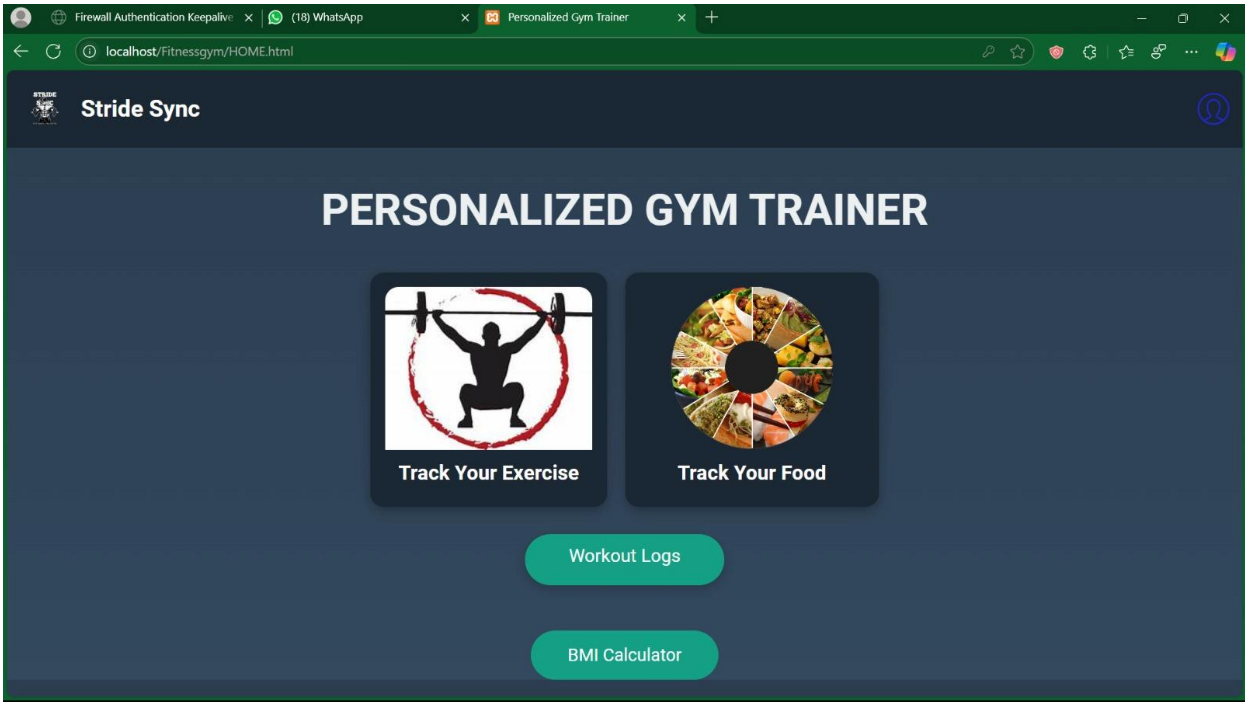

Figure 5.3 Home screen

The application provides users with access to a range of exercises, including Lateral Rise, Alternative Dumbbell Curls, Barbell Row, Push-Up, Squats, Shoulder Press, and Tricep Dips. This comprehensive selection caters to various workout preferences, ensuring users can easily choose exercises that align with their fitness goals.

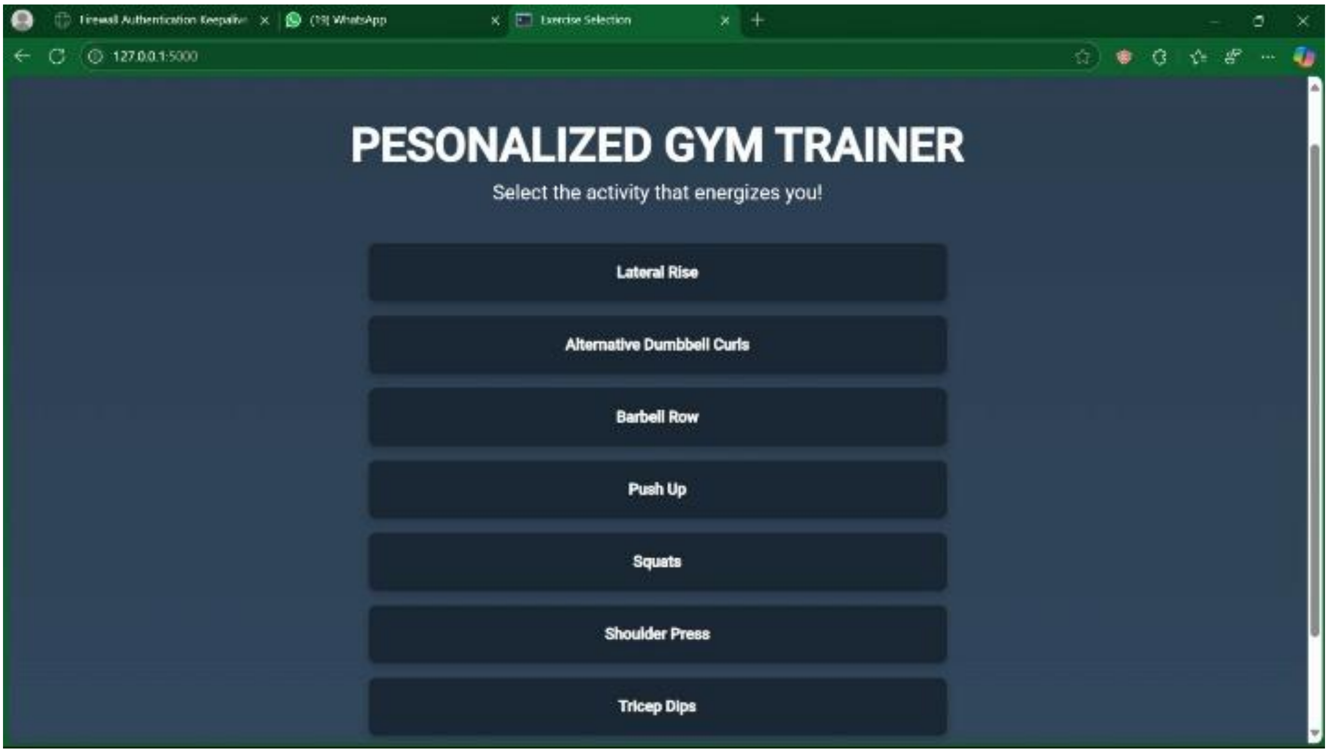

Figure 5.4 exercise window

A. Workout window

The StrideSync system was rigorously evaluated to assess its effectiveness in real-time exercise tracking, BMI calculation, and food composition analysis. The results from the implementation and testing phases highlight the system's capabilities and performance across its core functionalities.

1. barbell curls

A curl counter logic is implemented to track the user's repetitions based on the angles formed by the arms. A "down" state is registered when the arm is fully extended (angle > 160 degrees), and an "up" state is registered when the arm is fully curled (angle < 30 degrees)—the transition from "down" to "up" increments the repetition counter.

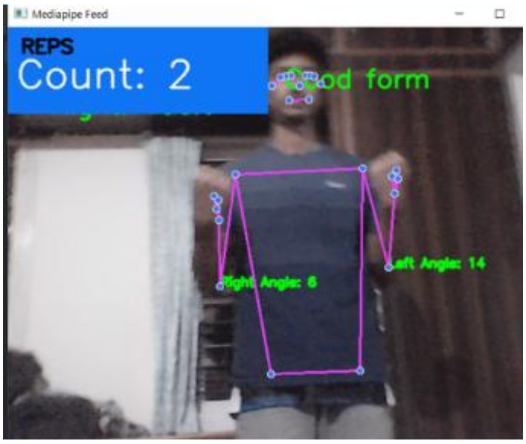

Fig 5.5 Correct form detection of barbell curls

The system provides real-time feedback to the user about their form. If the difference between the angles of the left and right arms exceeds a certain threshold, a "Correct your form" message is displayed, as shown in Fig 7. Otherwise, a "Good form" message is provided, as shown in Fig 6, encouraging the user to maintain the correct posture.

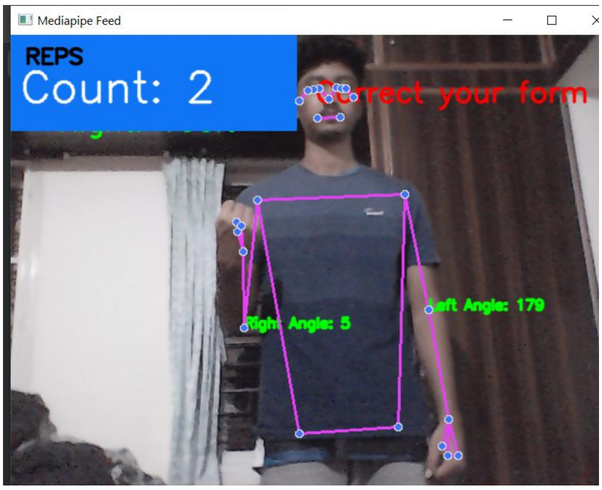

Fig 5.6. Feedback for Incorrect form detected in barbell curls

2. Shoulder press

The shoulder press in Fig 8 component tracks arm movements by measuring the angles at the elbow joints. It detects when the arms are in the "up" position (full extension) with an angle greater than 160 degrees and in the "down" position (lowered) with an angle less than 90 degrees. The joints considered are the shoulder, elbow, and wrist. A repetition is counted when both arms are simultaneously in the "up" position.

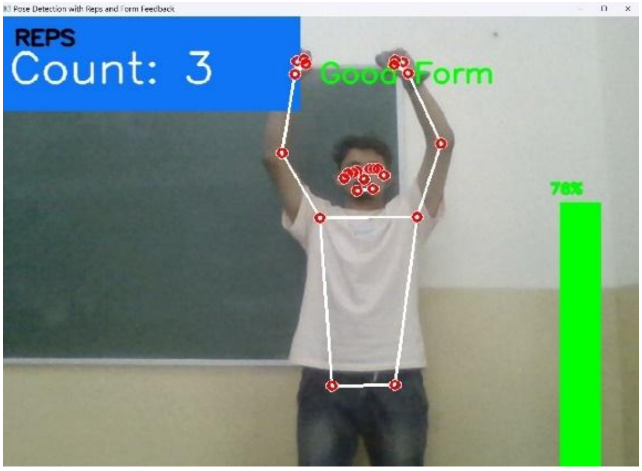

Fig 5.7. Shoulder press demonstration

3. Lateral raises

The lateral raises in the Fig 9 component tracks arm movements by measuring the angles at the elbow joints. It detects when the arms are in the "up" position (fully extended) with angles greater than 170 degrees and in the "down" position (lowered) with angles less than 140 degrees. The joints considered are the shoulder, elbow, and wrist. A repetition is counted when both arms are in the "down" position after being in the "up" position.

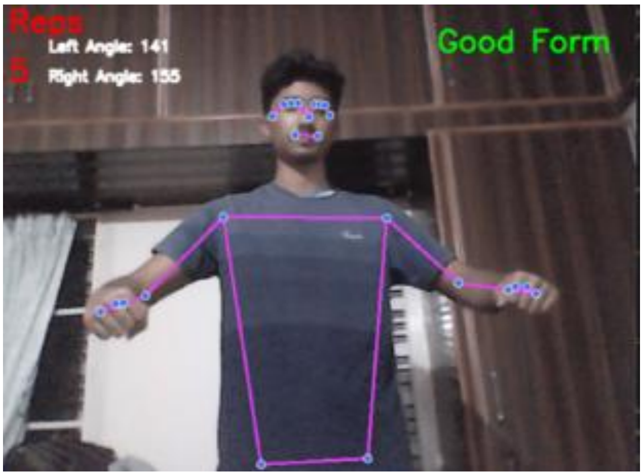

Fig 5.8. Lateral raises demonstration

4. Alternative dumbbell curls

In the alternate dumbbell curls in Fig 110, the component tracks arm movements by measuring the angles at the elbow joints. It detects when the arm is in the "up" position (full flexion) with an angle less than 30 degrees and in the "down" position (extended) with an angle greater than 160 degrees. The joints considered are the shoulder, elbow, and wrist. A repetition is counted for each arm when transitioning from the "down" to the "up" position. The exercise alternates between left and right arms, ensuring that each arm's movement is counted individually.

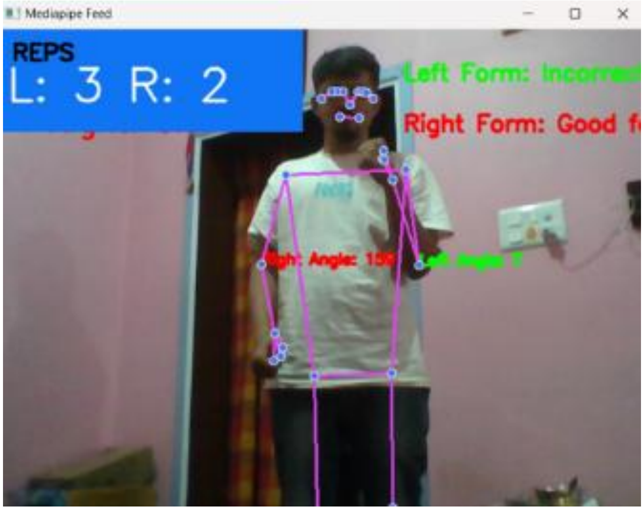

Fig 5.9. Alternative dumbbell curls

5. Push-ups

The push-up in Fig 11's tracking system calculates the angle of the left arm at the elbow joint to determine if the arm is in the "up" (fully extended) or "down" (bent) position. It counts a repetition when the arm transitions from the "down" to the "up" position and displays real-time feedback on the number of push-ups completed, the percentage of arm extension, and the frames per second.

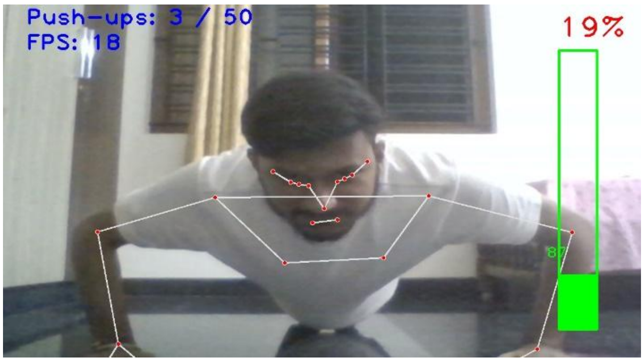

Fig 5.10. Push up demonstration

6. Squats

In the squat in Fig 12, the tracking system monitors leg movement by measuring the angle at the knee joint, counts a repetition when transitioning from an entirely "down" (bent) to a fully "up" (extended) position, and provides real-time feedback on the number of squats completed, the percentage of leg extension, and the frames per second.

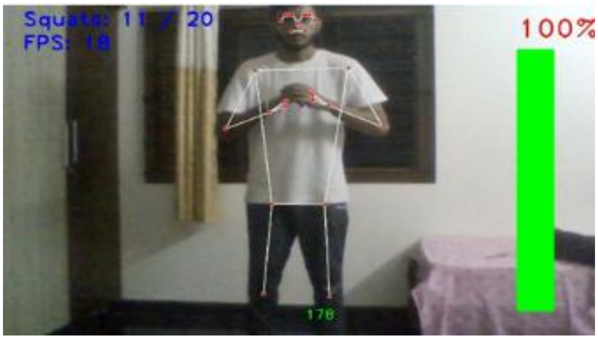

Fig 5.11. Squat demonstration

7. tricep dips

The tricep Fig 13 dips tracker monitors arm movements by calculating the angles at the elbows. It counts repetitions based on the transition between a fully "down" (angle less than 90 degrees) and a fully "up" (angle greater than 150 degrees) position and provides real-time feedback on the number of dips completed.

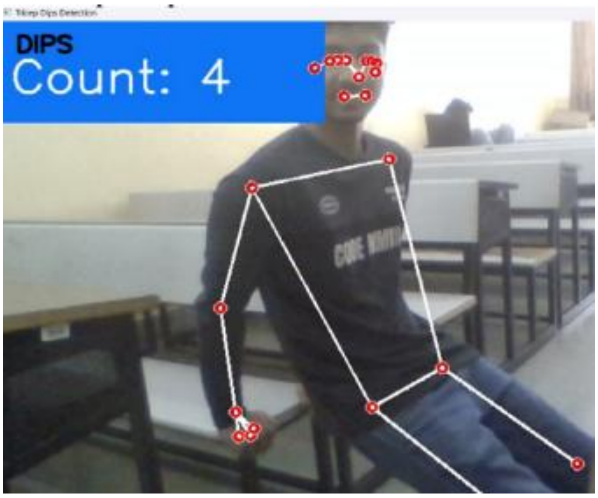

Fig 5.12. Tricep dips demonstration

B. Food Composition Analysis

Opening the Application

When you open the Food Composition Calculator in your web browser, you'll see a clean interface divided into two main sections. On the left side, you'll find information about the detected food item, including its nutritional breakdown. On the right side, there's a bar chart that visually displays the nutritional composition of the food.

Detecting the Food Item

The system automatically detects the food item using a real-time food detection model. Every few seconds, the application checks for the latest detected food and updates the display. Once a food item is detected, its name will appear in the "Detected Food Item" section, and the nutritional information will be displayed.

Nutritional Information

The nutritional values of the detected food item are shown in grams (g) for macronutrients like carbohydrates, proteins, fats, lipids, cholesterol, and calories. These values are based on a standard serving size (usually 100 grams), and the application will recalculate them based on the weight you input.

User Input for Weight

You can input the weight of the food item in grams. As soon as you enter a weight, the system recalculates and displays the new nutritional values according to the weight you entered. This real-time update also adjusts the bar chart.

Bar Chart Visualization

The nutritional breakdown is displayed in the form of a bar chart. This chart updates whenever the detected food or its weight changes, helping you visually compare the different components (carbs, proteins, fats, etc.).

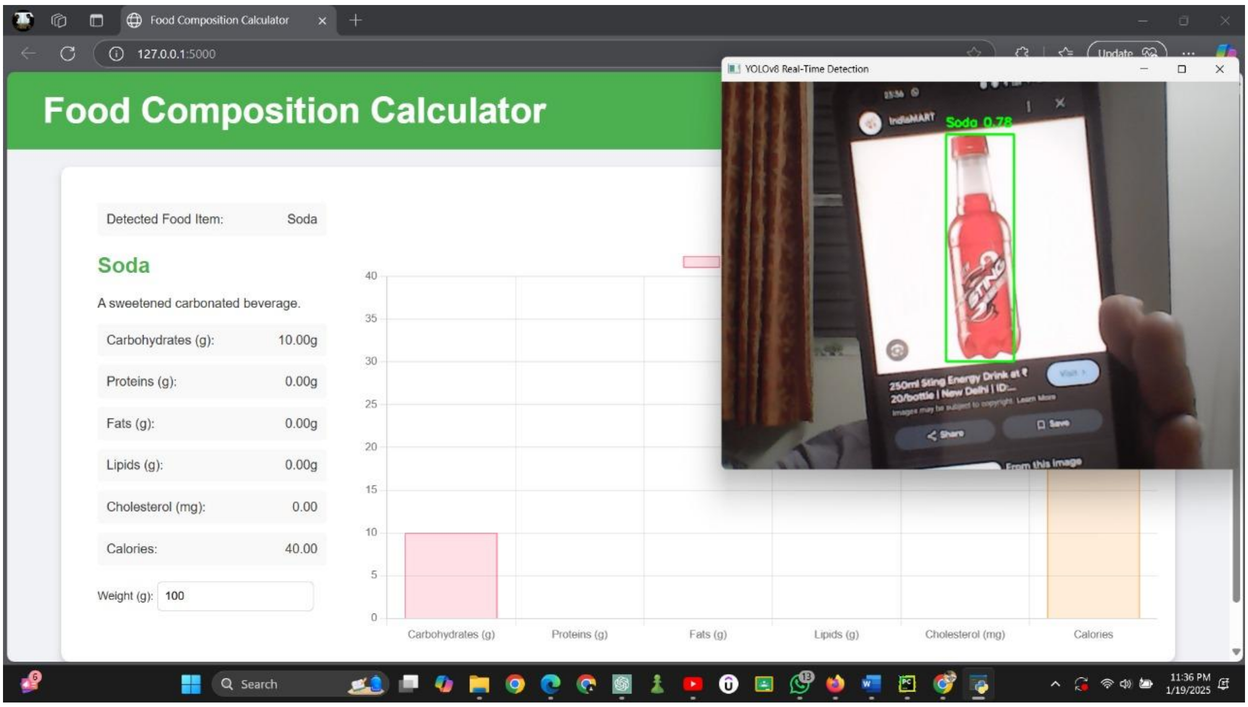

Figure 5.13 demonstration of yolo model

Figure 5.14 sample demonstration of food detected

The system can detect a variety of food items, including Apple, Banana, Chapathi, Chicken Gravy, Fries, Idli, Pizza, Rice, Soda, Tomato, Vada, and Burger. Each food item is identified using the YOLO model, which allows the system to provide real-time nutritional information based on the detected food. The list can be expanded to include more items in the future.

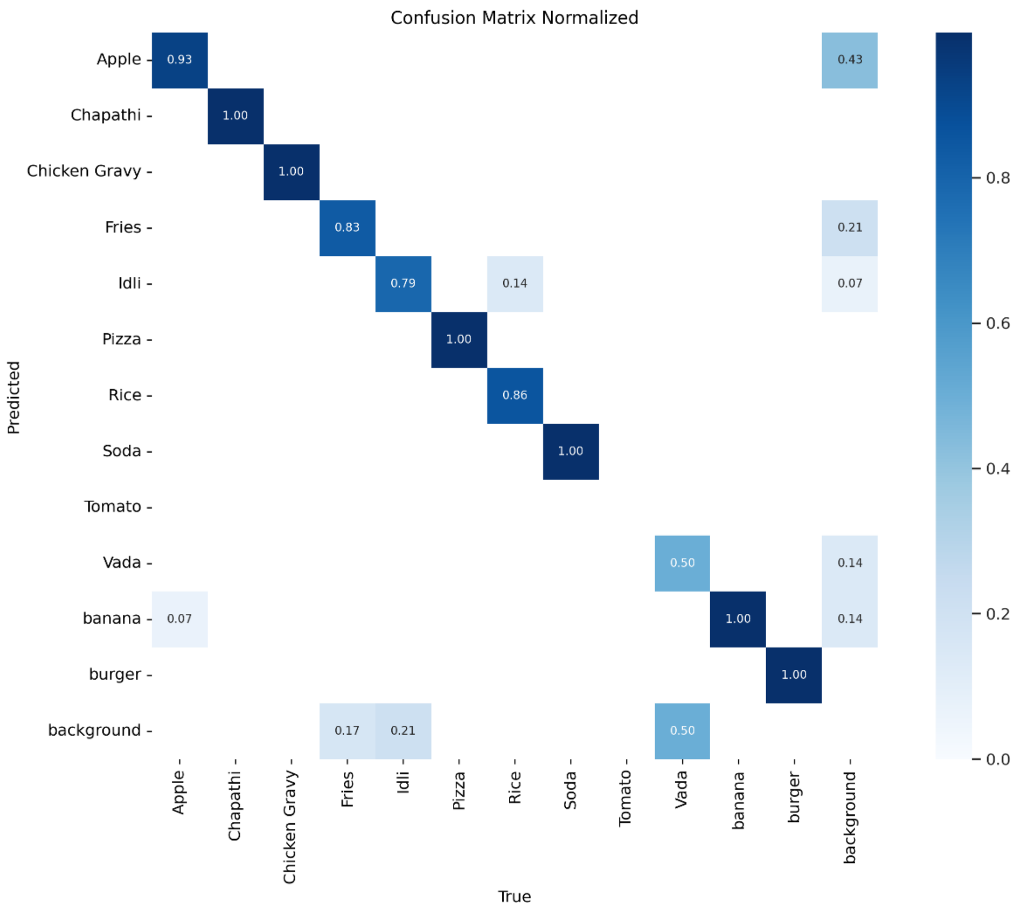

Figure 5.15 confusion matrix

The normalized confusion matrix provides an insightful representation of the classification performance across various categories. High classification accuracy is observed for classes such as Chapathi, Chicken Gravy, Pizza, and Burger, each achieving a precision of 1.00. Moderate performance is noted for categories like Idli (0.79), Rice (0.86), and Fries (0.83). However, certain classes exhibit lower precision, such as Banana, which has a true positive rate of only 0.07, with notable misclassifications into categories like Vada and Background. Additionally, there are instances of significant misclassification, such as Fries being confused with Background (0.21) and Idli also misclassified into Background (0.14). These results highlight areas of strength in the classification model while also pointing out categories that require further optimization to improve overall performance.

Chapter 6 CONCLUSION

The AI-powered exercise tracking and health management system presented in this report demonstrates a significant advancement in fitness technology by integrating artificial intelligence, computer vision, and deep learning techniques. By leveraging MediaPipe Pose for real-time posture estimation and YOLOv8 for food composition analysis, the system effectively addresses common challenges in workout tracking, including incorrect exercise form, lack of real-time feedback, and inaccurate nutritional assessment. The system's ability to provide instant feedback on exercise posture, count repetitions, and analyze food intake enhances both workout efficiency and dietary management. Additionally, the incorporation of a BMI calculator and personalized fitness recommendations based on user goals makes this solution a comprehensive tool for health monitoring. The integration of Tkinter and Flask for an intuitive user interface ensures seamless interaction, while MySQL facilitates secure storage and management of user data, making the system scalable and efficient for long-term use. Despite challenges such as dependency on webcam quality, lighting conditions, and privacy concerns related to real-time tracking, this AI-driven solution offers a promising alternative to traditional fitness methods by making personalized health management accessible and data-driven.

Moving forward, further improvements in pose estimation accuracy, expansion of the exercise database, and integration of additional health metrics like heart rate and sleep patterns could significantly enhance the system's effectiveness. Addressing limitations such as privacy concerns and dependency on external hardware through privacy-focused data handling and software optimizations will be crucial for broader adoption. The potential extension of the system to mobile platforms could improve accessibility, allowing users to benefit from AI-powered fitness tracking on the go. Additionally, refining the food detection model to include a more extensive variety of food items will improve nutritional assessment accuracy. By continuously improving the system's AI algorithms, expanding its feature set, and optimizing its usability, this project has the potential to revolutionize fitness tracking, making workouts safer, more efficient, and tailored to individual health needs. Ultimately, this AI-based gym trainer serves as an innovative step toward integrating technology with fitness, promoting healthier lifestyles through smart automation and real-time monitoring.

Chapter 7 FUTURE WORK

- **Integration of Wearable Devices**: Incorporating data from wearables such as smartwatches or fitness trackers could enhance the system by providing real-time metrics on heart rate, steps, calories burned, and sleep patterns. This would offer a more holistic view of a user's health and fitness progress.
- **AI-Powered Personalized Fitness Plans**: Develop a feature that uses machine learning to generate personalized fitness plans based on user data, goals, and performance trends over time. The system could adapt these plans dynamically to ensure they remain aligned with the user's progress.
- **Real-time Injury Prevention**: Enhance the system's ability to detect improper form and predict potential injuries based on movements or repeated actions. This feature could notify users when they are at risk of overexertion or muscle strain, allowing them to adjust their exercises before injury occurs.
- **Virtual Coach with Voice Feedback**: Add a virtual coaching assistant that provides voice guidance during workouts. This could help users maintain focus by delivering motivational prompts, exercise instructions, or corrective feedback on posture in real time.
- **Nutrition Tracking with Barcode Scanning**: Implement barcode scanning for food items to streamline the process of logging meals. This would allow users to quickly input nutritional information by scanning packaging, making the system more efficient and user-friendly.
- **Group Fitness and Social Sharing**: Integrate a feature that allows users to join group workouts or share progress with friends, creating a social aspect to the fitness tracking experience. This could foster a sense of community and motivation among users.
- **Cloud Syncing and Data Backup**: Enable cloud syncing and automatic backups of user data, ensuring that progress, workout logs, and food records are securely stored and accessible across multiple devices.

REFERENCES

[1] V. Singh, A. Patade, G. Pawar and D. Hadsul, "trAIner - An AI Fitness Coach Solution," 2022 IEEE 7th International conference for Convergence in Technology (I2CT), Mumbai, India, 2022, pp. 1-4, doi: 10.1109/I2CT54291.2022.9824511.

[2] R. Bi, D. Gao, X. Zeng and Q. Zhu, "LAZIER: A Virtual Fitness Coach Based on AI Technology," 2022 IEEE 5th International Conference on Information Systems and Computer Aided Education (ICISCAE), Dalian, China, 2022, pp. 207-212, doi: 10.1109/ICISCAE55891.2022.9927664.

[3] Kannan, Mr & Mohan, Mr & Gokul, Mr & Kumar, Mr. (2023). Enhancing Fitness Training with AI. International Journal of Research Publication and Reviews. 4. 5418- 5423. 10.55248/gengpi.234.4.38631.

[4] Ashraf, Ahsan & Haq, Areebul & Kumar, Kantesh & Alam, Muhammad & Shahid, Talha. (2024). AI Fitness Trainer. 10.13140/RG.2.2.29212.30089.

[5] V. S. P. Bhamidipati, I. Saxena, D. Saisanthiya and M. Retnadhas, "Robust Intelligent Posture Estimation for an AI Gym Trainer using Mediapipe and OpenCV," 2023 International Conference on Networking and Communications (ICNWC), Chennai, India, 2023, pp. 1-7, doi: 10.1109/ICNWC57852.2023.10127264.

[6] Xin Jin *et al.*, "Virtual Personal Trainer via the Kinect Sensor," *2015 IEEE 16th* International Conference on Communication Technology *(ICCT)*, Hangzhou, 2015, pp. 460-463, doi: 10.1109/ICCT.2015.7399879.

[7] Y. Fang, Z. Han, Z. Hu and Z. Wang, "Human Posture Estimation," 2021 3rd International Conference on Machine Learning, Big Data and Business Intelligence (MLBDBI), Taiyuan, China, 2021, pp. 220-234, doi: 10.1109/MLBDBI54094.2021.00050.

[8] "Pose Trainer: Correcting Exercise Posture using Pose Estimation". By S.Chen, R.R. Yang Department of CS., Stanford University

[9] "BlazePose: On-device Real-time Body Pose tracking." V.Bazarevsky I.Grishchenko, K.Raveendran, T.Zhu, F. Zhang, M.Grundmann

[10] T. L. Munea, Y. Z. Jembre, H. T. Weldegebriel, L. Chen, C. Huang and C. Yang, "The Progress of Human Pose Estimation: A Survey and Taxonomy of Models Applied in 2D Human Pose Estimation," in IEEE Access, vol. 8, pp. 133330-133348, 2020, doi: 10.1109/ACCESS.2020.3010248.

[11] M. B. Holte, C. Tran, M. M. Trivedi and T. B. Moeslund, "Human Pose Estimation and Activity Recognition From Multi-View Videos: Comparative Explorations of Recent Developments," in IEEE Journal of Selected Topics in Signal Processing, vol. 6, no. 5, pp. 538-552, Sept. 2012, doi: 10.1109/JSTSP.2012.2196975.

[12] Chen, Steven and Yang, Richard. (2018). Pose Trainer: Correcting Exercise Posture using Pose Estimation.