AI-Powered Virtual Fitness Assistant Project Submission

Course Project Title: AI-Powered Virtual Fitness Assistant

**Team Members:**

* Dharmik Shah
* Sanskriti Mansotra
* Naimul Islam
* Gurinder Bassi

Table of Content

[1. Abstract 2](#_Toc163231882)

[2. Introduction 2](#_Toc163231883)

[3. System Design 2](#_Toc163231884)

[4. User Interaction Design 2](#_Toc163231885)

[5. Iterative Development 2](#_Toc163231886)

[6. Implementation 3](#_Toc163231887)

[7. Testing and Evaluation 3](#_Toc163231888)

[8. Discussion and Conclusion 3](#_Toc163231889)

[9. References 3](#_Toc163231890)

[Additional Sections (Optional) 3](#_Toc163231891)

[Essential Tips: 3](#_Toc163231892)

# Abstract

The AI-Powered Virtual Fitness Assistant is a comprehensive system developed to track biceps curls, squats, and provide real-time feedback using advanced computer vision techniques, speech recognition, and hardware integration with Sense HAT. The project was developed iteratively over four phases, incorporating software and hardware enhancements at each stage. Challenges were encountered and overcome, resulting in a robust system capable of enhancing workout efficiency and user engagement. Future developments could include additional exercises, mobile app integration, and improved user experience.

# 2. Introduction

**Problem Statement:**  
Fitness enthusiasts and trainers often rely on manual counting or basic fitness trackers, which can be limiting in terms of accuracy and scope. The AI-Powered Virtual Fitness Assistant addresses this gap by providing automated, accurate tracking of multiple exercises, seamless mode switching through voice commands, and real-time feedback, thereby improving workout efficiency and user experience.

**Project Objectives:**

* Accurately track biceps curls and squats using computer vision.
* Integrate speech recognition for seamless exercise mode switching.
* Provide real-time feedback using Sense HAT.
* Ensure the system operates smoothly with minimal lag.

# 3. System Design

**Hardware Architecture**: The system is built around the Raspberry Pi 4, equipped with a Coral AI camera for pose detection and a Sense HAT for displaying real-time exercise counts. The hardware was selected for its balance between cost-effectiveness and processing power, enabling real-time computer vision and feedback.

**Software Architecture**:

The software is divided into three main modules:

* **Pose Estimation:** Implemented using MoveNet, this module tracks 33 key points on the body to detect exercises accurately.
* **Angle Calculation:** Custom algorithms compute the angles between key joints to determine exercise stages.
* **Speech Recognition:** Using the SpeechRecognition library and Gemini API, this module processes voice commands, allowing users to switch exercises hands-free

.

# 4. User Interaction Design

**Interface Design**: The system provides a user-friendly interface via the Sense HAT, displaying exercise counts. Voice commands are used to switch between exercises, enabling hands-free operation.

**Feedback Mechanism**: Real-time feedback is displayed on the Sense HAT, showing the current count of repetitions. The text on the display updates dynamically, ensuring that users are always informed of their progress.

**Command Detection:**  
Voice commands such as "curls" or "squats" are detected and processed by the system, allowing users to switch modes without manual input.

# 5. Iterative Development

# Phase 1: Biceps Curl Tracking In the first phase, we focused on developing a real-time biceps curl tracking system using OpenCV and MoveNet. Challenges included ensuring accurate angle calculation despite variations in camera perspective and body positioning. We refined the angle calculation logic using trigonometric functions and adjusted coordinate normalization to improve accuracy.

A screen shot of a computer code

Description automatically generated

# Phase 2: Squat Detection Enhancement The second phase introduced squat detection. The initial challenge was accurately counting squats, as the angle calculation logic for squats differed significantly from curls. We overcame this by refining the angle thresholds and introducing distance-based calculations to distinguish between squats and other exercises.

A person squatting in a room

Description automatically generated

# Phase 3: Dual-Exercise Tracking and Voice Recognition In this phase, we integrated dual-exercise tracking and introduced voice recognition for mode switching. The challenge here was managing transitions between exercises and ensuring consistent performance. Multi-threading was implemented to handle multiple tasks simultaneously, and we refined the speech recognition logic for better accuracy.

# Phase 4: Final Integration and Testing The final phase involved integrating all components and conducting extensive testing. The system was fine-tuned to handle different user speeds, lighting conditions, and varying command inputs, ensuring a robust and reliable user experience.

# 6. Implementation

**Computer Vision**: MoveNet was used for pose estimation, tracking the key points necessary for detecting exercises. The following code snippet highlights the angle calculation method used in the system:

A computer screen with text and numbers

Description automatically generated

**Speech Recognition**: Voice commands were processed using the SpeechRecognition library and Gemini API. The system dynamically switched between exercise modes based on user input, enhancing the user experience by providing a hands-free interface.

**System Integration**: The system uses a multi-threaded architecture to handle simultaneous tracking of multiple exercises and the processing of voice commands, ensuring smooth and responsive operation.

# 7. Testing and Evaluation

**Metrics**: We evaluated system performance based on the accuracy of exercise detection, response time to user commands, and overall user satisfaction with the feedback mechanisms.

**Testing Procedures**: The system was tested in various scenarios, including different lighting conditions, varying user speeds, and multiple command inputs. Special attention was given to ensure that the system could accurately distinguish between similar exercises, such as biceps curls and shoulder presses.

# 8. Discussion and Conclusion

**Lessons Learned:**

**Challenges Faced and Solutions Implemented:**

* **Accurate Angle Calculation:** Inconsistent angle measurements due to camera perspective were mitigated by refining the trigonometric functions used for calculation and adjusting the coordinate normalization.
* **Real-Time Performance:** High-resolution video processing introduced lag, which was overcome by reducing input frame resolution and limiting the number of pose landmarks processed per frame.
* **Exercise Stage Consistency:** Ensuring smooth transitions between exercise stages was a challenge. Multi-threading and careful synchronization of data streams ensured consistency in tracking and counting

**Future Work**: Given more time, we would explore adding additional exercises, improving speech recognition accuracy, and developing a mobile application interface for enhanced user interaction. Furthermore, expanding the system to include machine learning models for personalized workout recommendations could significantly increase its utility.

**Ethical Considerations**: The system processes all data locally on the Raspberry Pi, ensuring user privacy. Future work could involve developing strategies to handle and store data securely if cloud integration becomes necessary.

# 9. References

**Google AI MediaPipe Hand Landmarking Example**  
GitHub. (n.d.). *Hand Landmarker Example in Python*. Retrieved from <https://github.com/google-ai-edge/mediapipe-samples/blob/main/examples/hand_landmarker/python/hand_landmarker.ipynb>

**Google AI MediaPipe Examples**  
GitHub. (n.d.). *MediaPipe Samples Repository*. Retrieved from <https://github.com/google-ai-edge/mediapipe-samples/tree/main/examples>

**Coral AI Pose Estimation Models**  
Coral. (n.d.). *Pose Estimation with Coral AI*. Retrieved from <https://coral.ai/models/pose-estimation/>

**MoveNet Pose Estimation Example with Coral**  
GitHub. (n.d.). *MoveNet Pose Estimation using Coral AI*. Retrieved from <https://github.com/google-coral/pycoral/blob/master/examples/movenet_pose_estimation.py>

**Speech Recognition in Python**  
PyPI. (n.d.). *SpeechRecognition Python Package*. Retrieved from <https://pypi.org/project/SpeechRecognition/>

**TensorFlow MoveNet Tutorial**  
GitHub. (n.d.). *MoveNet Tutorial on TensorFlow Hub*. Retrieved from <https://github.com/tensorflow/docs/blob/master/site/en/hub/tutorials/movenet.ipynb>

# 10. Additional Information’s

GitHub Repository Link

[*https://github.com/Nfislam/AI-Powered-Virtual-Fitness-Assistant.git*](https://github.com/Nfislam/AI-Powered-Virtual-Fitness-Assistant.git)