Project report on

Empowering Advancing Fitness Monitoring Using Raspberry Pi -powered push up counter with Media Pipe pose estimation Technology

Submitted by

DibyaRanjan Sahoo(2241019075)

Abhishek Sahoo(2241019026)

Jaydev Sahoo(2241019078)

Archit Kumar Sahoo(2241019040)

B. Tech. (CSE(IOT) 4th Semester (Section–2241037)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Institute of Technical Education and Research

SIKSHA 'O' ANUSANDHAN

DEEMED TO BE UNIVERSITY

Bhubaneswar, Odisha, India.

(May, 2024)

Abstract

- Fitness tracking and monitoring are crucial for enhancing personal health and athletic performance. This study presents the design and implementation of an advanced push-up counter leveraging a Raspberry Pi and Media Pipe Pose Estimation technology. The proposed system aims to provide an affordable, efficient, and accurate solution for tracking and evaluating push-up exercises.

- The push-up counter system integrates a Raspberry Pi, a low-cost single-board computer, with a camera module to capture real-time video. Media Pipe, a robust machine learning framework developed by Google, is employed for pose estimation, enabling the identification and tracking of key body landmarks. The system processes the video feed to detect push-up movements, count repetitions, and assess the form and technique of the user.

- Key features of the system include real-time feedback, ease of setup, and portability. The use of Raspberry Pi ensures cost-effectiveness and accessibility, making the technology available for a wide range of users, from fitness enthusiasts to professional athletes. Media Pipe Pose Estimation technology provides high accuracy in detecting and analyzing body poses, ensuring precise counting and form evaluation.

- The system's performance was evaluated through extensive testing, demonstrating its reliability and accuracy in various conditions and environments. The results indicate that the Raspberry Pi-powered push-up counter with Media Pipe Pose Estimation technology is a viable tool for enhancing fitness monitoring, offering significant potential for future developments in the field of automated exercise tracking.

- In conclusion, this project illustrates a significant advancement in fitness monitoring technology, combining affordability, precision, and user-friendliness. It paves the way for future innovations in exercise tracking and personalized fitness coaching, contributing to the broader goal of improving health and fitness outcomes.

Contents

Chapter 01: Introduction

1.1. Introduction

With the rising focus on health and fitness, accurate and efficient exercise monitoring has become essential. This project presents a Raspberry Pi-powered push-up counter utilizing Media Pipe Pose Estimation technology. By capturing real-time video through a camera module and employing advanced pose estimation, the system precisely counts push-up repetitions and evaluates form. It offers a cost-effective, portable, and user-friendly solution, providing real-time feedback to users. This innovative approach addresses the limitations of traditional exercise tracking methods, making advanced fitness monitoring accessible to a wider audience and paving the way for future advancements in automated exercise tracking.

1.2. Background-

- ❖ The background of "Empowering Advancing Fitness Monitoring" lies in the growing demand for accessible and accurate fitness tracking solutions.
- ❖ Traditional methods often lack precision and real-time feedback.
- ❖ Leveraging Raspberry Pi and Media Pipe Pose Estimation technology addresses these challenges, enabling efficient push-up counting and performance monitoring.
- ❖ Raspberry Pi's affordability and versatility make it ideal for developing portable fitness devices, while Media Pipe's advanced pose estimation algorithms ensure precise movement detection.
- ❖ This project bridges the gap between advanced technology and fitness monitoring, empowering users to track their progress effectively and cultivate healthier habits with ease.

1.3. Project Objectives

Develop a Raspberry Pi-based push-up counter integrating Media Pipe Pose Estimation for accurate exercise monitoring. Design an intuitive user interface for real-time feedback and data visualization, enhancing user experience. Implement customizable features allowing users to set personal fitness goals and adjust detection sensitivity. Ensure affordability and accessibility by utilizing readily available hardware components and open-source software. Enable data logging and analysis capabilities for performance tracking and long-term progress evaluation. Promote consistent workout routines and healthier habits through accurate monitoring and goal setting, fostering positive impact on user fitness levels and well-being

1.4. Scope

Development of a Raspberry Pi-powered push-up counter integrating Media Pipe Pose Estimation for accurate exercise tracking. Implementation of a user-friendly interface for real-time feedback and data visualization. Customization options allowing users to set personal fitness goals and adjust detection parameters. Data logging and analysis features for performance tracking and progress evaluation. Utilization of affordable and accessible hardware components and open-source software. Promotion of consistent workout routines and healthier habits through accurate monitoring goal setting. Potential for future expansion to include additional exercise tracking capabilities and integration with fitness applications for comprehensive fitness monitoring

1.5. Project Management

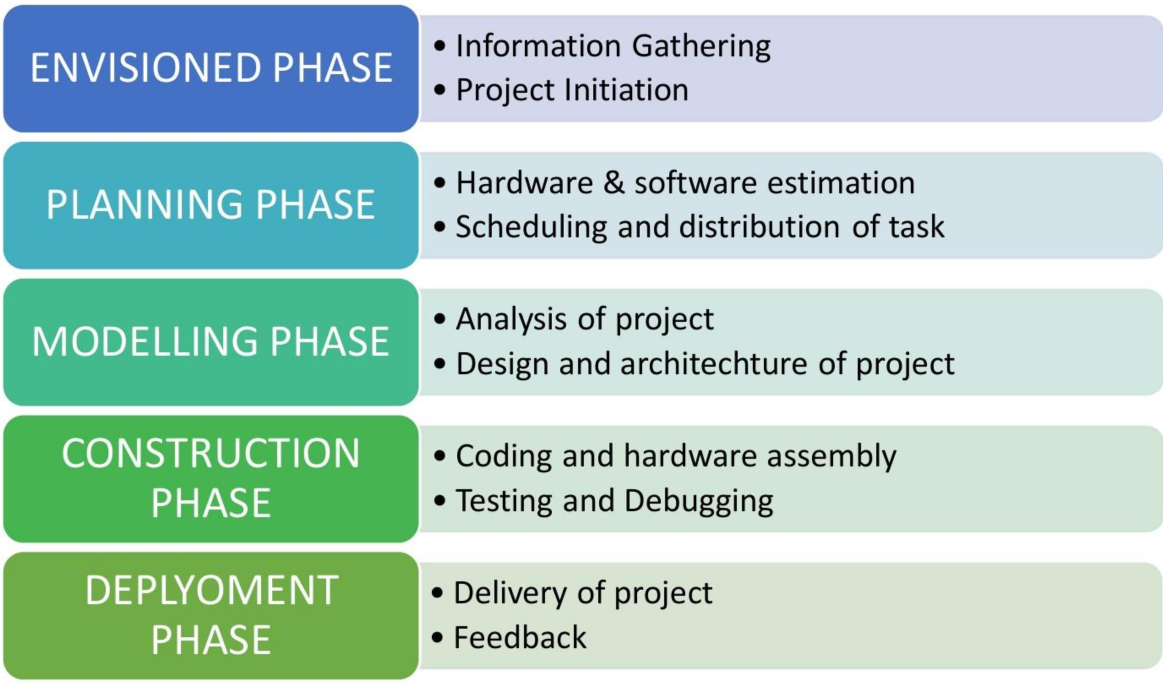


Figure 1. Model of phases in project management.

1.6. Overview and Benefits

This project introduces a push-up counter powered by Raspberry Pi and Media Pipe Pose Estimation technology, designed for accurate, real-time exercise tracking. The system uses a camera module to capture video and advanced pose estimation to count push-up repetitions and assess form. Key benefits include affordability, portability, and user-friendliness, making it accessible to a broad audience. The real-time feedback helps users improve their technique, while the precise tracking enhances workout efficiency. This innovative solution overcomes the limitations of traditional methods, offering a significant advancement in fitness monitoring technology and promoting better health outcomes.

1.7. Organization of the Report

The report is organised into the following chapters. Each chapter is unique on its own and is described with necessary theory to comprehend it.

Chapter 2 deals with background survey and review, Chapter 3 has the description of the theoretical aspects that has been acquired to commence the project work.

Chapter 02: Background Review & Survey

2.1. Related Works

The background review and survey of "Empowering Advancing Fitness Monitoring Using Raspberry Pi-powered Push-up Counter with MediaPipe Pose Estimation Technology" encompassed existing research on fitness monitoring systems, pose estimation technologies, and Raspberry Pi applications. It aimed to understand the current landscape, identify gaps, and assess the feasibility and potential impact of the proposed integration.

Key aspects of the background review included:

- ❖ Fitness Monitoring Systems: Evaluating various methodologies and technologies utilized in fitness monitoring, including wearable devices, camera-based systems, and IoT solutions.

- ❖ Pose Estimation Technologies: Examining state-of-the-art pose estimation algorithms and frameworks, such as MediaPipe, OpenPose, and DeepPose, to understand their capabilities and limitations in tracking human movements.

- ❖ Raspberry Pi Applications: Investigating previous projects and applications utilizing Raspberry Pi for data processing, IoT, and edge computing to gain insights into its suitability for implementing fitness monitoring solutions.

- ❖ Through this comprehensive review and survey, valuable insights were gained to inform the development and integration of the proposed fitness monitoring system, ensuring its effectiveness, scalability, and relevance in empowering users to advance their fitness goals

Chapter 03: Theoretical Aspects

3.1. Internet of Things (IoT)

The Internet of Things (IoT) refers to the network of interconnected devices and objects that can communicate and exchange data over the internet without human intervention. These devices, equipped with sensors and connectivity features, enable remote monitoring, control, and automation across various domains, enhancing efficiency and convenience.

3.2. Features of IoT

a) Intelligence

The intelligence of this system lies in its integration of MediaPipe pose estimation, OpenCV, and pyttsx3 on a Raspberry Pi platform. This enables real-time analysis of push-up movements, accurate counting, form assessment, and instant voice feedback. Such intelligence enhances fitness monitoring, guiding users to improve their workout techniques effectively.

b) Connectivity

The system's connectivity enables seamless communication between the Raspberry Pi-powered push-up counter and external devices or networks. Through wireless or wired connections, data collected from push-up monitoring can be transmitted for analysis, feedback, or storage, enhancing the overall functionality and accessibility of the fitness monitoring technology.

c) Dynamic Nature

The dynamic nature of this system arises from its ability to adapt and respond to changes in user behavior and environmental conditions during push-up exercises. Through real-time analysis and feedback mechanisms, it continuously adjusts to provide accurate push-up counts and form assessments, ensuring effective and personalized fitness monitoring.

d) Enormous Scale

The enormous scale of this technology lies in its potential to impact a wide range of users and fitness environments. With the Raspberry Pi-powered push-up counter and MediaPipe pose estimation, it can be deployed in homes, gyms, and rehabilitation centers, revolutionizing fitness monitoring and improving health outcomes on a large scale.

e) Sensing

The sensing capabilities of this technology enable precise detection and analysis of body movements during push-up exercises. Leveraging MediaPipe pose estimation and Raspberry Pi camera, it accurately senses and interprets key landmarks to count repetitions and assess form, providing valuable feedback for improving workout performance and fitness monitoring.

f) Heterogeneity

The heterogeneity of this technology refers to its ability to accommodate diverse users, environments, and fitness levels. With MediaPipe pose estimation and Raspberry Pi-powered push-up counter, it can adapt to different body types, exercise styles, and settings, ensuring inclusivity and effectiveness in advancing fitness monitoring and improvement.

g) Security

Security in this system involves safeguarding user data and ensuring the integrity of the device and communication channels. Measures such as data encryption, secure authentication protocols, and regular software updates are implemented to protect against unauthorized access and potential vulnerabilities, ensuring user privacy and system reliability.

3.3. Advantages of IoT

a) Communication

Communication in this system facilitates data exchange between the Raspberry Pi-powered push-up counter and external devices or networks. It enables real-time transmission of push-up data, feedback, and updates, enhancing the functionality and accessibility of the fitness monitoring technology for users, trainers, and healthcare professionals alike.

b) Automation and Control

Automation and control mechanisms enable seamless operation and management of the fitness monitoring system. Through programmed algorithms and feedback loops, the system autonomously analyzes push-up movements, adjusts parameters, and provides real-time guidance, enhancing user experience and facilitating efficient fitness monitoring and improvement.

c) Information

Information processing involves analyzing data collected from push-up exercises using MediaPipe pose estimation on Raspberry Pi. This includes tracking body movements, counting repetitions, assessing form, and providing feedback. By processing this information in real-time, the system empowers users to monitor and enhance their fitness effectively.

d) Monitoring

Monitoring entails continuous observation and analysis of push-up performance using MediaPipe pose estimation on Raspberry Pi. It involves tracking repetitions, evaluating form, and providing real-time feedback to users. This enables effective oversight of fitness progress, facilitating adjustments and improvements to optimize workout routines and achieve fitness goals.

e) Efficiency

Efficiency in this technology ensures optimized performance and resource utilization. Through real-time analysis of push-up movements using MediaPipe pose estimation on

Raspberry Pi, the system accurately tracks repetitions and assesses form, providing immediate feedback. This enhances workout effectiveness and streamlines fitness monitoring, maximizing user productivity and results.

3.4. Disadvantages of IoT

a) Compatibility

Compatibility in IoT refers to the seamless integration and interoperability of diverse devices, protocols, and platforms within an ecosystem. It ensures that connected devices can communicate and collaborate effectively, enabling data sharing, automation, and scalability across different hardware and software environments for efficient IoT deployment and operation.

b) Complexity

The complexity in IoT arises from managing vast amounts of data, diverse devices, and heterogeneous networks. It involves challenges in data processing, security, interoperability, and scalability. Additionally, integrating sensors, actuators, and cloud services while addressing privacy concerns and regulatory requirements further adds to the complexity of IoT implementations.

c) Privacy/Security

Privacy and security in IoT are paramount due to the interconnected nature of devices and data exchange. Measures like encryption, authentication, and access control safeguard sensitive information. Regular updates, intrusion detection systems, and privacy-by-design principles mitigate risks, ensuring user trust and compliance with data protection regulations in IoT ecosystems.

d) Safety

Safety in IoT involves ensuring the reliability and integrity of connected devices and systems to prevent accidents and hazards. It includes measures such as fail-safes, remote monitoring, and predictive maintenance to detect and mitigate potential risks. Compliance with safety standards and regulations ensures the protection of users and assets in IoT environments.

3.5. Application areas of IoT

IoT finds application in diverse sectors including healthcare, agriculture, smart cities, and industrial automation. It enables remote monitoring, predictive maintenance, and data-driven decision-making by connecting devices and systems to the internet, improving efficiency, productivity, and quality of life in various domains.

3.6. IOT Technologies and Protocols

a) Bluetooth

Bluetooth is a wireless communication technology used for short-range data exchange between devices. It enables connections between smartphones, laptops, IoT devices, and peripherals, facilitating tasks like file sharing, audio streaming, and device pairing. Bluetooth Low Energy (BLE) extends battery life, making it suitable for IoT applications and wearables.

b) Zigbee

Zigbee is a low-power wireless communication protocol designed for IoT and home automation applications. It operates on the IEEE 802.15.4 standard, enabling reliable and efficient data transmission between devices. Zigbee's mesh networking capability allows devices to communicate over long distances and provides robustness against interference and network disruptions.

c) Z-Wave

Zigbee is a low-power wireless communication protocol designed for IoT and home automation applications. It operates on the IEEE 802.15.4 standard, enabling reliable and efficient data transmission between devices. Zigbee's mesh networking capability allows devices to communicate over long distances and provides robustness against interference and network disruptions.

d) Wi-Fi

Wi-Fi, short for Wireless Fidelity, is a widely used wireless communication technology that enables devices to connect to the internet and local area networks (LANs) wirelessly. Operating on the IEEE 802.11 standard, Wi-Fi facilitates high-speed data transmission over short to medium distances, typically within homes, offices, and public spaces.

e) Cellular

Cellular technology provides wireless communication using cellular networks operated by mobile carriers. It enables devices to connect to the internet and make voice calls over long distances, utilizing radio frequency signals. Cellular networks offer wide coverage and high reliability, making them suitable for mobile devices and IoT applications requiring remote connectivity.

f) NFC

NFC (Near Field Communication) is a short-range wireless technology that enables communication between devices when they are in close proximity, typically within a few centimeters. It facilitates contactless data exchange, such as mobile payments, ticketing, and pairing devices, by simply tapping or bringing devices close together.

g) LoRaWAN

LoRaWAN (Long Range Wide Area Network) is a low-power wireless communication protocol designed for long-range IoT applications. It operates on unlicensed radio frequency bands and enables devices to transmit small data packets over long distances, making it suitable for applications like smart agriculture, asset tracking, and remote monitoring.

3.7. Project Layout

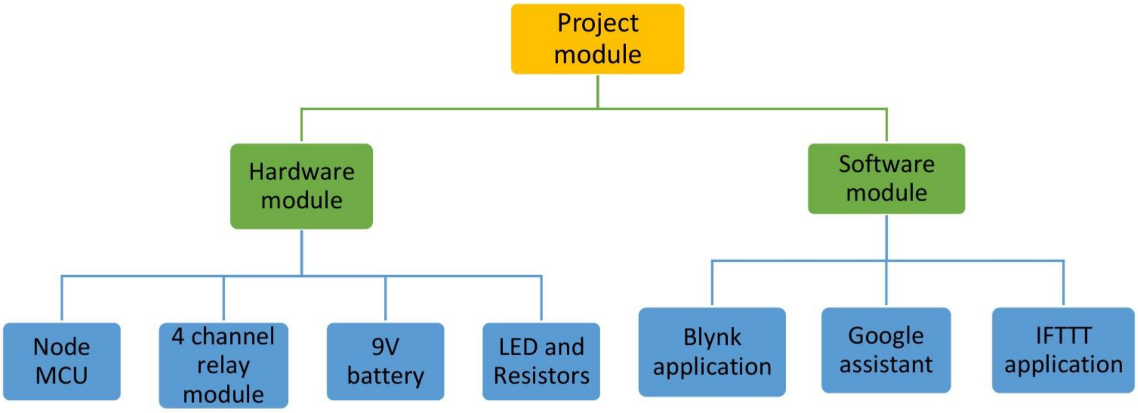


Figure 2. Layout of project module

3.7.1. Brief Description

- ❖ **Project Title: Empowering Advancing Fitness Monitoring**
- ❖ **Technology: Raspberry Pi-powered push-up counter**
- ❖ **Key Feature: MediaPipe pose estimation technology**
- ❖ **Functionality: Accurately counts push-ups by analyzing body movements**
- ❖ **Purpose: Enhances fitness tracking and performance assessment**
- ❖ **Benefits: Affordable, real-time monitoring, and improves workout efficiency**

Chapter 04: Hardware Requirements

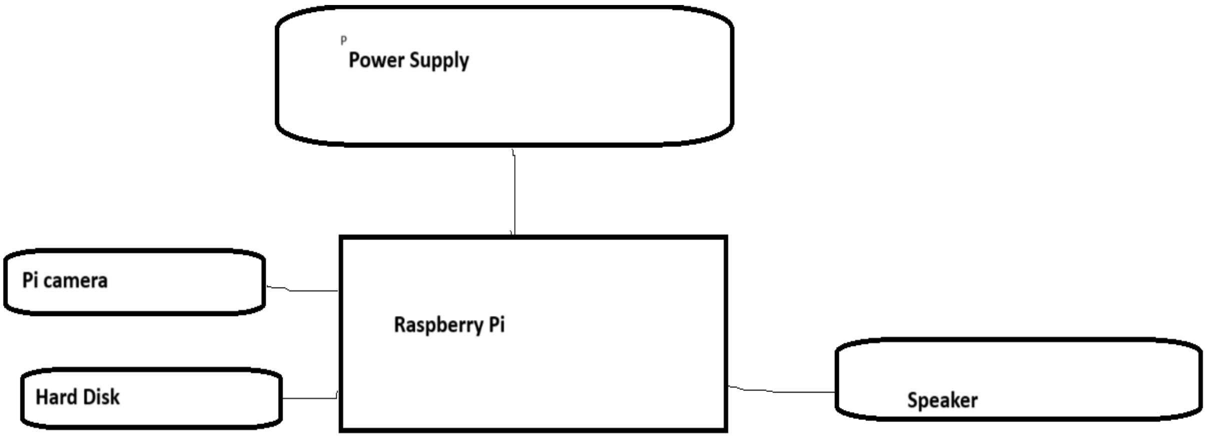
4.1. Raspberry Pi

Raspberry Pi is a credit card-sized single-board computer developed by the Raspberry Pi Foundation. It features a Broadcom system-on-chip (SoC), typically with ARM-based processors, various connectivity options like Wi-Fi and Bluetooth, GPIO pins for interfacing with external devices, HDMI output for display, and USB ports for peripherals. Available in different models with varying specifications, Raspberry Pi runs on Linux-based operating systems like Raspbian and supports a wide range of applications, from educational projects to media centers and IoT devices. Its affordability, versatility, and vibrant community support have made it a popular platform for learning, prototyping, and DIY projects.

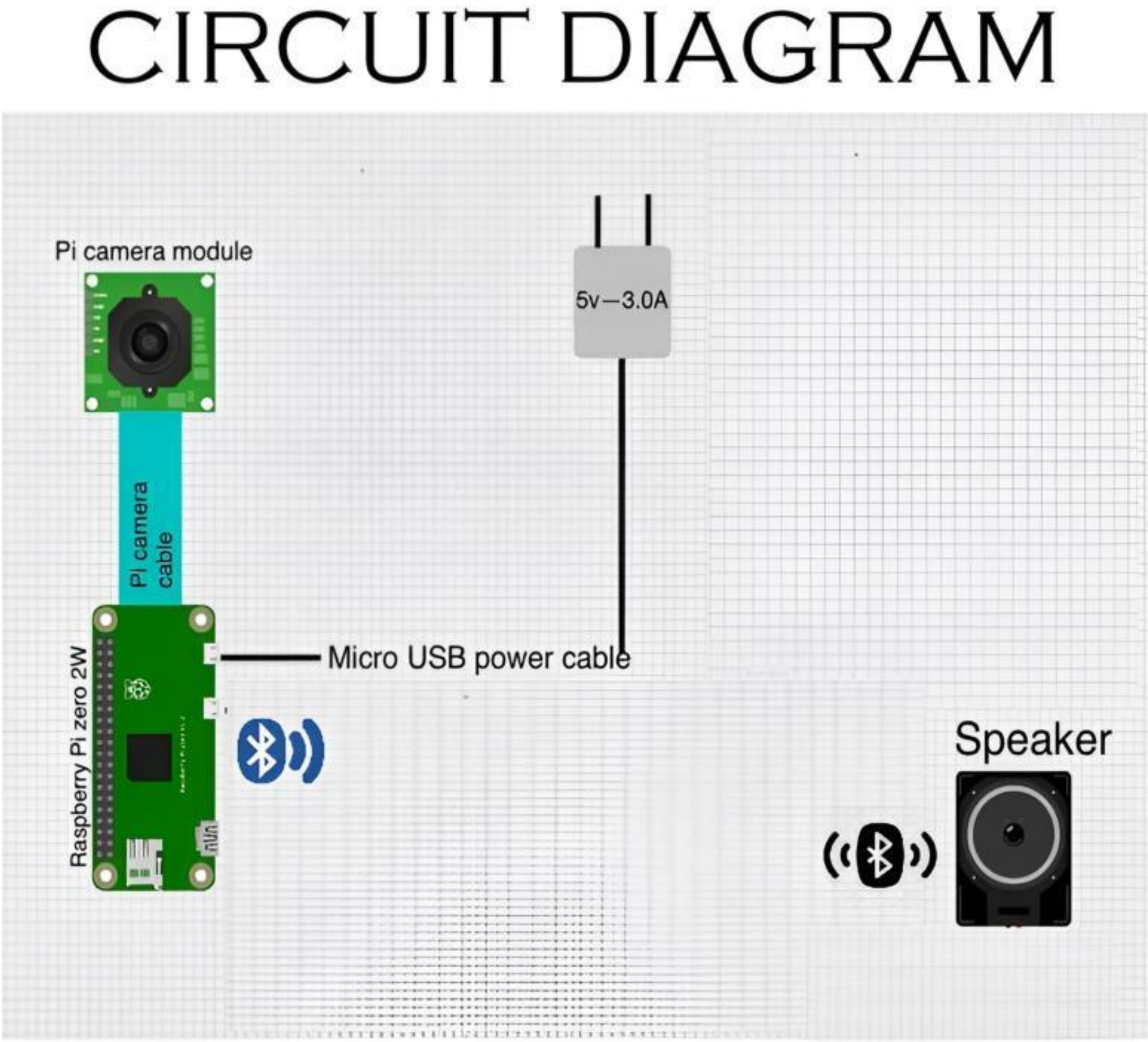
4.1.1. Features

- **Real-time push-up counting**
- **Accurate body movement analysis with MediaPipe pose estimation**
- **Affordable Raspberry Pi integration**
- **User-friendly interface**
- **Detailed performance tracking**
- **Customizable workout goals**
- **Progress visualization**
- **Enhances workout efficiency and effectiveness**

4.5. Block diagram of the proposed system



4.5.2. Circuit Diagram



4.5.3. Components Required

Table 1. Component listing.

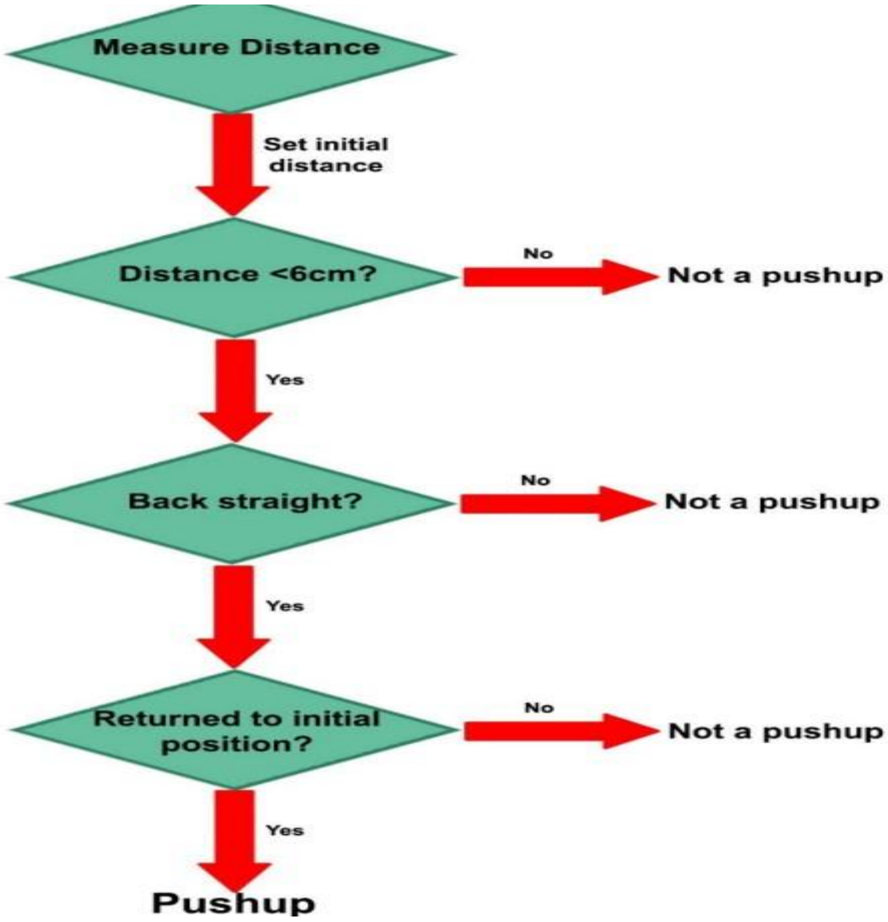
Sl. No.	Component and Specification	Quantity
1.	Raspberry Pi Zero 2w	1
2.	Pi Camera Module	1
3.	Pi Camera Cable	1
4.	Micro USB Card	1
5.	USB HUB	1
6.	Speaker(wired/wireless)	1
7.	Audio Adapter	1

Chapter 05: Software Requirements

5.1. Thonny IDE (Python 3.0)

Thonny IDE is a Python development environment designed for learners and educators, supporting Python 3.0. It features an intuitive interface with tools like code highlighting, debugging, and an interactive shell, facilitating code writing and testing. Thonny's simplicity makes it accessible for beginners, providing a gentle learning curve for those new to programming. With its built-in Python interpreter, users can execute code directly within the IDE, enabling rapid prototyping and experimentation. Additionally, Thonny offers step-by-step debugging functionality, helping users identify and resolve errors in their code. Overall, Thonny IDE serves as a valuable resource for individuals seeking to develop Python skills effectively.

5.2. Logic and Flowchart



Chapter 06: Project development & Testing Aspects

6.1. Project development for "Empowering Advancing Fitness Monitoring Using Raspberry Pi-powered Push-up Counter with MediaPipe Pose Estimation Technology" involved several crucial steps. Initially, the team focused on integrating MediaPipe for accurate pose estimation, OpenCV for video processing, and pyttsx3 for voice feedback within the Raspberry Pi environment. This integration required extensive coding, debugging, and optimization to ensure real-time performance.

Throughout the development process, rigorous testing was paramount. Testing procedures included evaluating push-up counting accuracy, assessing system responsiveness under various conditions, and collecting user feedback. Different scenarios and environments were simulated to ensure the system's robustness against factors like lighting variations, camera angles, and user movements.

Additionally, user experience testing was conducted to refine the interface design and optimize usability. Continuous iteration and refinement were essential to enhance the system's overall functionality, reliability, and user satisfaction before deployment.

Chapter 07: Conclusion & Future Scope

7.1. Result

The result of this system is an advanced, real-time fitness monitoring solution that accurately counts push-ups and assesses form using MediaPipe pose estimation on a Raspberry Pi. Enhanced with OpenCV for video processing and pyttsx3 for voice feedback, it offers an effective, interactive, and user-friendly workout experience.

7.2. Conclusion

In conclusion, the Raspberry Pi-powered push-up counter, leveraging MediaPipe pose estimation, OpenCV, and pyttsx3, represents a significant advancement in fitness monitoring technology. It provides accurate, real-time feedback and guidance, enhancing workout efficiency and effectiveness, and demonstrates the potential of integrating AI and computer vision in personal fitness routines.

7.3. Limitations

Limitations of this system include potential inaccuracies in pose estimation under poor lighting conditions, dependency on camera placement for optimal performance, and computational constraints of the Raspberry Pi, which may affect real-time processing speed. Additionally, variations in body types and push-up styles can challenge the system's universality.

7.4. Further Enhancement and Future Scope

Future enhancements could involve improving pose estimation accuracy with advanced machine learning models, optimizing software for better performance on Raspberry Pi, and incorporating diverse exercises. Integration with cloud services for data analytics and personalized fitness recommendations can also expand the system's capabilities and user experience.

References -

- https://www.youtube.com/watch?v=RLQWon4Zmao&list=PLPJQlEokVM71UiZzp1g
  N3leFBiutiq06t&index=6
- https://circuitdigest.com/microcontroller-projects/push-up-counter-using
raspberry-pi-4-and-mediapipe

Appendix 01

A01.2. Main Code

```
import cv2
import mediapipe as mp
import os
import pyttsx3
mp_drawing = mp.solutions.drawing_utils
mp_pose = mp.solutions.pose
counter = 0
stage = None
create = None
opname = "output.avi"
x=0
engine = pyttsx3.init()
def findPosition(image, draw=True):
   lmList = []
   if results.pose_landmarks:
      mp_drawing.draw_landmarks(
         image, results.pose_landmarks,
mp_pose.POSE_CONNECTIONS)
      for id, lm in enumerate(results.pose_landmarks.landmark):
         h, w, c = image.shape
         cx, cy = int(lm.x * w), int(lm.y * h)
         lmList.append([id, cx, cy])
   return lmList
cap = cv2.VideoCapture(0)
with mp_pose.Pose(
      min_detection_confidence=0.7,
      min_tracking_confidence=0.7) as pose:
```

```
 while cap.isOpened():
success, image = cap.read()
      image = cv2.resize(image, (640, 480))
 if not success:
         print("Ignoring empty camera frame.")
         continue
image = cv2.cvtColor(cv2.flip(image, 1), cv2.COLOR_BGR2RGB)
      results = pose.process(image)
      image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
      lmList = findPosition(image, draw=True)
 if len(lmList) != 0:
         cv2.circle(image, (lmList[12][1], lmList[12][2]), 20, (0, 0, 255),
cv2.FILLED)
         cv2.circle(image, (lmList[11][1], lmList[11][2]), 20, (0, 0, 255),
cv2.FILLED)
         cv2.circle(image, (lmList[12][1], lmList[12][2]), 20, (0, 0, 255),
cv2.FILLED)
         cv2.circle(image, (lmList[11][1], lmList[11][2]), 20, (0, 0, 255),
cv2.FILLED)
if (lmList[12][2] and lmList[11][2] >= lmList[14][2] and lmList[13][2]):
            cv2.circle(image, (lmList[12][1], lmList[12][2]), 20, (0, 255, 0),
cv2.FILLED)
            cv2.circle(image, (lmList[11][1], lmList[11][2]), 20, (0, 255, 0),
cv2.FILLED)
            stage = "down"

if (lmList[12][2] and lmList[11][2] <= lmList[14][2] and lmList[13][2])
and stage == "down":
            stage = "up"
            counter += 1
            counter2 = str(int(counter))
            print(counter)
            x+=1
           engine.say(f"{x} Push-up completed")
            engine.runAndWait()  # Wait for the announcement to
complete
      text = "{}:{}".format("Push Ups", counter)
```

```
    cv2.putText(image, text, (10, 40), cv2.FONT_HERSHEY_SIMPLEX
1, (255, 0, 0), 2)
    cv2.imshow('MediaPipe Pose', image)

    if create is None:
      fourcc = cv2.VideoWriter_fourcc(*'XVID')
      create = cv2.VideoWriter(opname, fourcc, 30, (image.shape[1],
image.shape[0]), True)
    create.write(image)

    key = cv2.waitKey(1) & 0xFF

    if key == ord("q"):
      break

cv2.destroyAllWindows()
```

A01.3. Libraries -

- Numpy-

NumPy plays a crucial role in fitness monitoring by providing efficient numerical operations for data processing. In a Raspberry Pi-powered push-up counter utilizing Media Pipe pose estimation, NumPy enables real-time calculations and manipulations of pose data, ensuring accurate and swift performance analysis, thereby enhancing the overall effectiveness and reliability of the system.

- Mediapipe-

MediaPipe enhances fitness monitoring by providing robust pose estimation technology in a Raspberry Pi-powered push-up counter. It accurately tracks body movements, ensuring precise push-up counts and form assessment. This real-time capability empowers users to improve their fitness routines, making workouts more efficient and results-oriented.

- OpenCV-

OpenCV is integral to the Raspberry Pi-powered push-up counter, enabling video capture and processing for fitness monitoring. It works with MediaPipe pose estimation to analyze body movements in real time, ensuring accurate push-up counts and feedback. This synergy enhances the system's efficiency and reliability in tracking and improving fitness routines.

- Pyttsx3-

pyttsx3 enhances the Raspberry Pi-powered push-up counter by providing real-time voice feedback. Integrating with MediaPipe pose estimation, it announces push-up counts and form corrections, offering users immediate auditory guidance. This feature improves the workout experience by allowing users to stay focused on their form without needing to check a screen.

Appendix 02

A02.1. Project Proposal Form

The project proposal form was prepared and duly signed from our Faculty-in-Charge Dr. Biswaranjan Swain. The same is attached at the last of this report.

A02.2. Project Management

SR. NO	COMPONENT	Individual Contributions in %				TOTAL
		DibyaRanjan	*Abhishek*	*Jaydev*	*Archit*	
1	Planning	70%	10%	10%	10%	100%
2	Background Research and Analysis	85%	5%	5%	5%	100%
3	Hardware Design	70%	10%	10%	10%	100%
4	Software Design	85%	5%	5%	5%	100%
5	Testing	40%	20%	20%	20%	100%
6	Final Assembling	55%	15%	15%	15%	100%
7	Project Report Writing	60%	30%	5%	5%	100%
8	Presentation	25%	25%	25%	25%	100%
9	Logistics	85%	5%	5%	5%	100%

A02.3. Bill of Material

Table 1. Component listing.

#	Component	Specification	Unit Cost	Quantity	Total
1.	Raspberry Pi	Zero 2 w	1800.00	1	2585.00
2.	Camera Module	5MP	380.00	1	380.00
3.	Speaker	5Watt Power	250.00	1	250.00
4.	Memory Card	32GB	420.00	1	420.00
5.	Adapter	5V 3Amp	500.00	1	500.00
6.	USB Cable	5V 3A	200.00	1	200.00
7.	Sun board	4mm thick	100.00	1	100.00
8.	Other accessories	Gum,Colouring paper	100.00	1	100.00
				Grand Total	4535.00

Appendix 03

A03.1. Data Sheets

1. Raspberry Pi Zero 2 W
Datasheet: The Raspberry Pi Zero 2 W is a compact, affordable, and powerful single-board computer.

Specifications:
- CPU: Broadcom BCM2710A1, quad-core 64-bit Arm Cortex-A53, 1 GHz
- RAM: 512MB LPDDR2
- Wireless: 2.4 GHz IEEE 802.11 b/g/n wireless LAN, Bluetooth 4.2, BLE
- Ports: Mini HDMI, Micro-USB OTG, Micro-USB power, HAT-compatible 40-pin header, CSI-2 camera connector
- Power: 5V via Micro-USB or GPIO header
- Datasheet Link: Raspberry Pi Zero 2 W Datasheet

2. Raspberry Pi Camera Module
Datasheet: The official Raspberry Pi Camera Modules are designed to interface with the Raspberry Pi via the CSI interface.
Specifications:
- Camera Module V2: 8-megapixel Sony IMX219 sensor, fixed focus lens, 3280 x 2464 pixel static images, supports 1080p30, 720p60, and 640x480p90 video
- Camera Module 3: 12-megapixel Sony IMX477 sensor, adjustable focus lens, supports up to 12MP still images and various video modes

Datasheet Links:
-Camera Module V2 Datasheet
-Camera Module 3 Datasheet

3. Speaker (Wired/Wireless)
Datasheet: Varies by model and manufacturer. Typical speakers for Raspberry Pi include small USB-powered speakers or Bluetooth speakers.
Specifications:
- Look for power rating, impedance (usually 4 or 8 ohms), and connection type (3.5mm jack, USB, Bluetooth).

- Example Datasheet: For a small USB-powered speaker:
- Model: Mini USB Speaker
- Power Output: 3W
- Connection: USB for power, 3.5mm audio jack for input

4. USB Hub:
Datasheet: A USB hub expands the number of USB ports available on the Raspberry Pi.
Specifications:
- Ports: Typically 4 to 7 USB 2.0 or 3.0 ports
- Power: Bus-powered or self-powered options
- Example Model: Anker 4-Port USB 3.0 Hub
- Datasheet Link: Anker 4-Port USB 3.0 Hub Datasheet

5. Micro USB Card:
Datasheet: Standard micro USB cables are used for data transfer and power delivery.
Specifications:
- Length: Typically 1 to 3 meters
- Current Rating: Up to 2.4A for power cables
- Data Transfer: USB 2.0 standard

6. Micro USB Power Cable with Adapter:
Datasheet: Power supply for Raspberry Pi Zero 2 W.
Specifications:
- Voltage: 5V
- Current: 2.5A or higher recommended
- Connector: Micro USB
- Example Model: Official Raspberry Pi Micro USB Power Supply
- Datasheet Link: Official Raspberry Pi Power Supply Datasheet