Final Project Report: HAND GESTURE BASED UAV CONTROL

COMPSCI 528: Mobile and Ubiquitous Computing

Team Members: Noshitha Padma Pratyusha Juttu, Janet Reshma Jeyasingh, Sri Harsha Vardhan Prasad Jella

Introduction:

Our project focuses on developing an intuitive and responsive Unmanned Aerial Vehicle (UAV) control system leveraging hand gestures. By integrating the ESP32 S3 microcontroller with the MPU6050 Inertial Measurement Unit (IMU) sensor, we aim to capture and interpret dynamic hand movements to translate them into precise UAV commands. This system enables users to control the UAV's movements such as ascending, descending, turning, and navigating through natural hand gestures, eliminating the need for traditional remote controls.

The motivation behind this project stems from the increasing demand for more natural and user-friendly interfaces in human-machine interactions. Gesture recognition offers a seamless way to interact with devices, enhancing user experience by making control mechanisms more intuitive. Specifically, controlling UAVs through gestures opens up applications in areas where manual controls are impractical or hindered, such as in search and rescue operations, aerial photography in confined spaces, and interactive entertainment. Additionally, this project serves as a practical exploration of integrating embedded systems, sensor data processing, and machine learning for real-time applications.

The proposed gesture-controlled UAV system offers several benefits:

1. **Enhanced User Experience**: Provides a more natural and hands-free method of controlling drones, making operations smoother and more intuitive.
2. **Accessibility**: Facilitates drone control for individuals who may have difficulty using traditional remote controllers due to physical limitations.
3. **Versatility in Applications**: Expands the usability of UAVs in various fields such as emergency response, surveillance, and recreational activities by allowing quick and effortless control.
4. **Innovation in Human-Machine Interfaces**: Pushes the boundaries of current UAV control technologies, contributing to advancements in gesture recognition and embedded system integrations.
5. **Educational Value**: Serves as a comprehensive learning platform for integrating hardware components, programming microcontrollers, and applying machine learning techniques in real-world scenarios.

Project Overview:

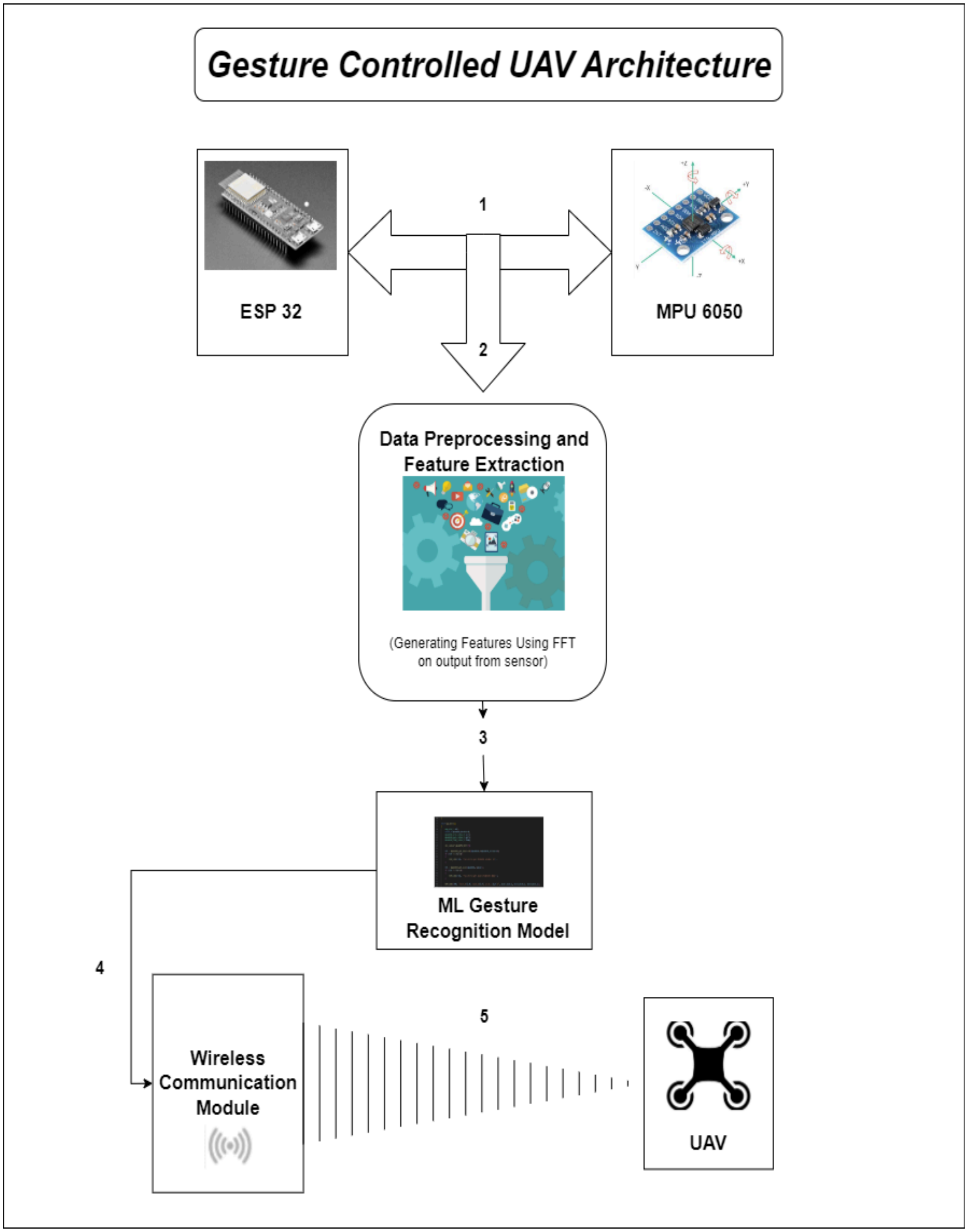
The system architecture of our gesture-controlled UAV comprises the following key components:

1. **Hardware Components**:
    - **ESP32 S3 Microcontroller**: Acts as the central processing unit, handling data acquisition from the IMU sensor, processing gesture data, and communicating with the UAV.
    - **MPU6050 IMU Sensor**: Captures acceleration and gyroscopic data corresponding to hand gestures in three-dimensional space.
    - **Quadcopter Frame**: Provides the structural foundation for the UAV, building the UAV prototype and test flight control systems.
    - **Battery Pack**: Ensure adequate power supply for sustained operations.
2. **Software Components**:
    - **ESP-IDF Framework**: Utilized for programming the ESP32 S3, enabling data collection, preprocessing, and communication functionalities.
    - **Gesture Recognition Model**: A Support Vector Machine (SVM) model trained to classify gestures based on processed IMU data.
3. **Communication Protocols**:
    - **Wireless Communication Module**: Facilitates the transmission of control signals from the ESP32 S3 to the UAV, ensuring real-time responsiveness.
4. **User Interface**:
    - **Gesture-Based Controls**: Users perform predefined hand gestures (Ex: Up, Down, Front, Back, Left, Right, Rotate-left, Rotate-Right, Land) to control the UAV' s movements seamlessly.

Workflow:

1. **Data Collection**: The MPU6050 sensor records acceleration and gyroscopic data corresponding to user gestures (Up, Down, Front, Back, Left, Right, Rotate-left, Rotate-Right, Land).
2. **Data Preprocessing**: Raw sensor data undergoes normalization and data interpolation for missing records.
3. **Gesture Classification**: The preprocessed data is fed into the trained SVM model to identify the performed gesture in real-time.
4. **Command Execution**: Recognized gestures are mapped to specific UAV commands, which are then transmitted to the drone to perform the desired action.
5. **Feedback Loop**: Continuous data collection and gesture recognition ensure that the UAV responds promptly to user inputs, maintaining synchronized control.

Architecture:



1. **Collection of Data:**

   The IMU sensor captures the dynamic acceleration and angular velocity of the hand in 3D space while performing gestures. This data is transmitted to the ESP32 microcontroller.

2. **Data Preprocessing and Feature Extraction:**

   The raw sensor data is preprocessed for consistency and reliability. The relevant frequency domain features are extracted from the preprocessed data using Fast Fourier Transformations (FFT).

3. **Feeding the preprocessed data:**

   The preprocessed data is then passed to the gesture classification model for gesture prediction.

4. **Gesture Classification:**

   A machine learning model is trained using the extracted features and corresponding gesture labels. During real-time prediction, the model processes input features and identifies the gesture being performed.

5. **Communication between the Gesture system and UAV:**

   After the gesture is classified, the corresponding control signal is transmitted wirelessly (e.g., via Bluetooth) from the ESP32 to the UAV.

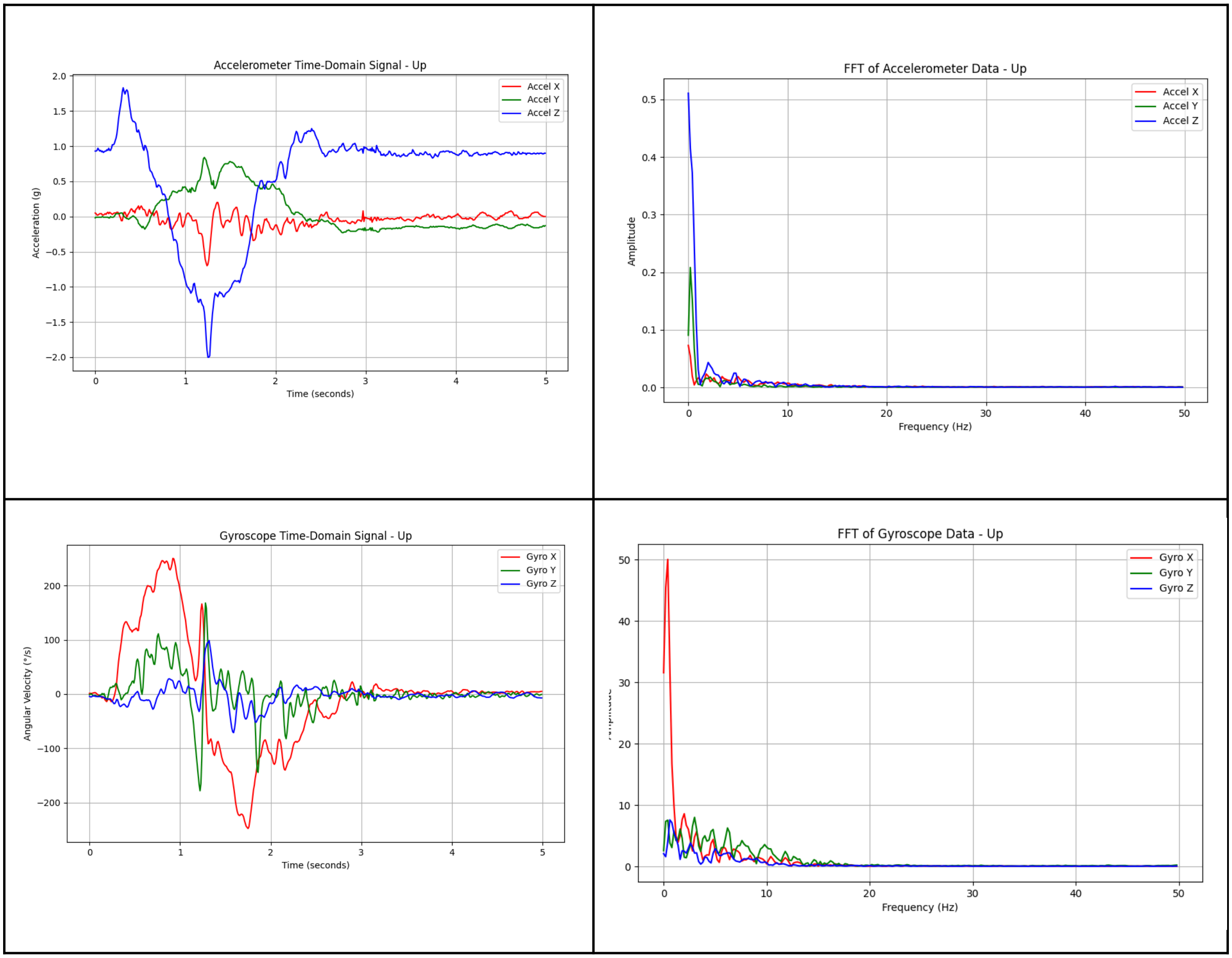
Project Implementation:

Hardware Integration

1. **ESP32 S3 and MPU6050 Setup**:
   - Connected the MPU6050 IMU sensor to the ESP32 S3 via the I2C protocol.
   - Ensured proper wiring and power supply to maintain stable sensor readings.
   - Calibrated the IMU sensor to account for any biases or noise in the accelerometer and gyroscope data.
2. **UAV Assembly**:
   - Connected the ESP32 S3 to the UAV's control system, enabling it to send flight commands based on recognized gestures.
   - Installed the battery pack, ensuring adequate power supply for both the microcontroller and the UAV.

FFT and Spectogram Analysis:-

Accelerometer:



Accelerometer:

- ➔ The **Z-axis** shows a sharp initial upward acceleration until 0.5 seconds, followed by a rapid change to negative values from 0.5 to 1.2 seconds before stabilizing, which signifies the end of the upward motion.
- ➔ This pattern indicates that the gesture involves an initial strong upward motion, followed by stabilization.
- ➔ The Y-axis shows a gradual increase from 0.5 to 1.8 seconds, suggesting a sudden upward movement. This spike reflects the effect of the IMU sensor's changing orientation as the hand moves upward in the horizontal direction.
- ➔ The X-axis exhibits smaller fluctuations, indicating less significant motion in the horizontal direction, as it remains mostly at zero.

Gyroscope:

- ➔ The up gesture shows significant rotational movement primarily along the X-axis, with a peak angular velocity of approximately 250 deg/s at 0.8 seconds, followed by a gradual stabilization to zero around 3 seconds.
- ➔ The Y-axis also contributes to the gesture, peaking at around 150 deg/s close to the same time, indicating coordinated rotational motion between the X and Y axes.
- ➔ In contrast, the Z-axis remains relatively stable with only minor fluctuations, suggesting minimal involvement in this direction.

Spectogram :

Interpretation :

- ***Accelerometer:***
    - From the FFT analysis, we observe that the maximum frequency with significant magnitude is at most 10 Hz across all axes.
    - This is consistent with the spectrogram, where all the bright regions across the X, Y, and Z axes are below 10 Hz.
    - The maximum magnitude for each axis occurs between 0 Hz and 6 Hz, which is why the spectrogram shows bright spots within this frequency range at certain times.
    - For the X-axis, the FFT indicates a peak amplitude of approximately 0.06 at around 0.2 Hz, corresponding to a bright spot on the spectrogram near 1.5 seconds at that frequency range.
    - For the Y-axis, the magnitude fluctuates up to 5 Hz, resulting in all bright spots in the Y-axis spectrogram appearing below 5 Hz.
    - The highest power frequency for the Y-axis occurs around 1.4 seconds, which aligns with the FFT's peak frequency between 0.1 Hz and 0.2 Hz.
    - Similarly, for the Z-axis, the spectrogram's bright regions fall within 0.4 Hz to 0.5 Hz, with some slightly brighter spots up to 6 Hz, but diminishing significantly for frequencies above 6 Hz.
- ***Gyroscope:***
    - From the FFT analysis, we observe that the maximum frequency with significant magnitude is at most 20 Hz across all axes.
    - This is consistent with the spectrogram, where all the bright regions across the X, Y, and Z axes are below 20 Hz.
    - The maximum magnitude for each axis occurs between 0 Hz and 8 Hz, which is why the spectrogram shows bright spots within this frequency range at certain times.
    - For the X-axis, the FFT indicates a peak amplitude of approximately 50 at around 1 Hz, corresponding to a bright spot on the spectrogram near 1 second at that frequency range.
    - For the Y-axis, the magnitude fluctuates up to 1 Hz and has maximum magnitude close to 8, resulting in all bright spots in the Y-axis spectrogram at around 1 second.
    - Similarly, for the Z-axis, when frequency is approximately 4 Hz it has the maximum magnitude close to 8 , because of which we can observe from a spectrogram a bright spot

at around 3 seconds for a frequency of that range and it is less brighter than the brightest spots from X axis spectrogram and Y axis spectrograms.
  - After 15 Hz, the spectrogram shows no bright spots for any axis, reflecting the FFT's near-zero magnitudes for these higher frequencies.

Software Development :

1. **Firmware Programming with ESP-IDF**:
   - Developed firmware to collect real-time data from the MPU6050 sensor.
   - Implemented data preprocessing like normalization and data interpolation for missing records.
   - Integrated the trained SVM model to classify gestures based on processed IMU data.
2. **Gesture Recognition and UAV Control**:
   - Mapped classified gestures to specific UAV commands.
   - Ensured that commands are transmitted wirelessly to the UAV with minimal latency.
   - Developed safety protocols to handle unexpected gestures or communication failures, ensuring the UAV can land safely if needed.
3. **User Interface and Feedback**:
   - Provided visual feedback through serial monitoring to display recognized gestures and corresponding actions.
   - Implemented a fail-safe mechanism where the UAV lands automatically upon detecting a "land" gesture or in case of system interruptions.

Challenges:

1. **Real-Time Data Processing**:
   - Achieving low-latency processing of IMU data was crucial for responsive UAV control. Optimizing the firmware to handle continuous data streams without significant delays was challenging.
2. **Gesture Classification Accuracy**:
   - Ensuring high accuracy in gesture recognition required extensive data collection and model training. Variations in gesture performance among different users introduced inconsistencies that the model had to accommodate.
3. **Sensor Noise and Calibration**:
   - The MPU6050 sensor is susceptible to noise and drift, which can affect gesture recognition. Implementing effective filtering and calibration techniques was necessary to mitigate these issues.
4. **Wireless Communication Reliability**:
   - Maintaining a stable and reliable wireless connection between the ESP32 S3 and the UAV was essential. Interference and signal degradation posed challenges.
5. **Hardware Integration**:
   - Integrating various hardware components, such as ensuring compatible power levels and secure connections, required meticulous attention to detail to prevent malfunctions or damage to components.
6. **Safety Mechanisms**:
   - Implementing robust safety protocols to handle unexpected scenarios (e.g., gesture misclassification, sudden loss of connection) was imperative to prevent accidents or UAV crashes.

Pros and Cons:

- **Pros:**
  - **Gesture Recognition**:
    - The SVM model successfully classified predefined gestures with high accuracy, enabling precise control over the UAV's movements.
  - **Real-Time Control**:
    - The system achieved real-time responsiveness, allowing the UAV to execute commands promptly in response to user gestures.
  - **Hardware Stability**:
    - The integration of the ESP32 S3 and MPU6050 sensor proved stable, with consistent data collection and minimal sensor drift after calibration.
  - **User Experience**:
    - Users found the gesture-based control intuitive and seamless, enhancing the overall interaction with the UAV.

- **Cons:**
  - **Complex Gesture Recognition**:
    - While basic gestures were accurately recognized, more complex or rapid gestures sometimes led to misclassifications, affecting control precision.
  - **Battery Life**:
    - The UAV's battery life was shorter than anticipated due to the additional power consumption from the ESP32 S3 and continuous sensor data processing.
  - **Environmental Factors**:
    - Variations in lighting and background movements occasionally affected the IMU sensor's ability to accurately capture gestures, especially in dynamic environments.

Evaluation and Demonstration:

Our project was evaluated through a series of tests focusing on gesture recognition accuracy, system responsiveness, and UAV control reliability. The evaluation process included:

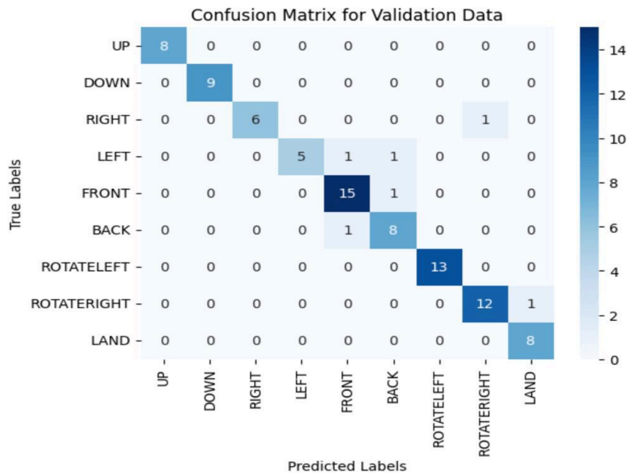
1. **Controlled Environment Testing**:
   - Performed gesture recognition and UAV control tests in a controlled setting to minimize external interference and ensure consistent results.
2. **User Trials**:
   - Engaged multiple team members to perform gestures, assessing the system's ability to generalize across different users and gesture styles.
3. **Stress Testing**:
   - Pushed the system to handle rapid and consecutive gestures to evaluate its performance under high-demand scenarios.
4. **Battery Performance Analysis**:
   - Monitored battery consumption during prolonged operations to identify potential optimizations for power efficiency.

The project's success was determined based on the following criteria:

1. **Gesture Classification Accuracy**:
   - Achieved an accuracy rate of over 90% in recognizing predefined gestures, validated through extensive testing and cross-validation.

```
[16]: # Perform training and testing with SVM
train_and_evaluate_svm(X_train, y_train, X_test, y_test)

SVM Training accuracy: 100.000%
SVM Validation accuracy: 93.333%
```



○

2. **System Responsiveness**:
   - Demonstrated minimal latency between gesture execution and UAV response, ensuring smooth and timely control.
3. **UAV Stability and Maneuverability**:
   - Verified that the UAV responded accurately to control commands, maintaining stability during movements and rotations.
4. **User Feedback**:
   - Received positive feedback from team members regarding the intuitiveness and reliability of the gesture-based control system.

Demo Video Link : The demo video link is provided in the link below: Final Demo

GitHub Repository : Our project's source code, documentation, and related resources are available on GitHub: HandGestureUAV

Repository Contents:

- **ESP32 Firmware**: Source code for data collection, preprocessing, gesture recognition, and UAV command execution.
- **Data**: Labeled datasets used for training and testing the SVM model.
- **Machine Learning Scripts**: Code for data preprocessing, feature extraction, model training, and evaluation.
- **Communication module script:** The python script that integrates the ML model , UAV and Microcontroller.
- **Documentation**: Comprehensive documentation of project proposal and final report
- **Demonstration Videos**: Links and references to demonstration videos showcasing the system's capabilities

.

Team Contribution:

Noshitha Padma Pratyusha Juttu:

- **Data Collection and Labeling**: Coordinated the collection of gesture data, ensuring comprehensive coverage of all predefined gestures.
- **Data Preprocessing**: Applied the machine learning preprocessing techniques like normalization, data-interpolation to prepare sensor data for model training.
- **Hardware Integration**: Collaborated in the interfacing of the ESP32 S3 microcontroller with the MPU6050 IMU sensor, ensuring seamless communication via the I2C protocol.
- **Model Training and Evaluation**: Worked in designing, training, and validating the SVM model for accurate gesture classification to enhance model robustness.
- **Gesture Classification Integration**:Collaborated in integrating the pre-trained SVM model into the firmware, enabling real-time gesture prediction based on processed IMU data.
- **Collaborative Integration**: Collaborated with Janet and Harsha in integrating the trained model with the ESP32 firmware, ensuring seamless operation between gesture recognition and UAV control.

Sri Harsha Vardhan Prasad Jella:

- **Data Collection and Labeling**: Coordinated the collection of gesture data, ensuring comprehensive coverage of all predefined gestures.
- **Hardware Integration**: Collaborated with Noshitha to connect the ESP32 S3 microcontroller with the MPU6050 IMU sensor, ensuring seamless communication via the I2C protocol.
- **UAV Assembly**: Led the assembly of the quadcopter frame, integrating brushless motors, ESCs, propellers, and power distribution boards.
- **Firmware Integration**: Collaborated with Janet to integrate the machine learning model with the ESP32 firmware, ensuring accurate translation of gestures into UAV commands.
- **Testing and Troubleshooting**: Managed hardware testing phases, identifying and resolving integration issues to maintain system stability.
- **Data preprocessing for ML model:** Collaborated with Janet to analyze optimal data preprocessing techniques, implementing normalization and data interpolation to achieve the best results.

Janet Reshma Jeyasingh:

Contributions:

- **Data Collection and Labeling**: Coordinated the collection of gesture data, ensuring comprehensive coverage of all predefined gestures.
- **Firmware Development**: Focused on programming the ESP32 S3 using the ESP-IDF framework, developing code for real-time IMU data collection and preprocessing.
- **Data Processing Analysis**: Implemented Fast Fourier Transformations (FFT) and Spectogram for feature extraction from raw sensor data.
- **Gesture Classification Integration**: Integrated the pre-trained SVM model into the firmware, enabling real-time gesture prediction based on processed IMU data.

- **Control Signal Transmission**: Programmed the ESP32 S3 to transmit control signals wirelessly to the UAV, ensuring reliable command execution.
- **User Interface Development**: Developed serial monitoring tools to provide real-time feedback on recognized gestures and executed commands.
- **Collaborative Testing**: Worked closely with Harsha to test firmware performance and optimize system responsiveness.

References:

1. Kim, M., Cho, J., Lee, S., & Jung, Y. (2019). IMU Sensor-Based Hand Gesture Recognition for Human-Machine Interfaces. *Sensors (Basel, Switzerland)*, *19*(18), 3827. https://doi.org/10.3390/s19183827
2. Babiuch, Marek & Foltynek, Petr & Smutný, Pavel. (2019). Using the ESP32 Microcontroller for Data Processing. 1-6. 10.1109/CarpathianCC.2019.8765944.
3. Djemal, Achraf & Hellara, Hiba & Barioul, Rim & Ben Atitallah, Bilel & Ramalingame, Rajarajan & Fricke, Ellen & Kanoun, Olfa. (2022). Real-Time Model for Dynamic Hand Gestures Classification based on Inertial Sensor. 10.1109/CIVEMSA53371.2022.9853648.