Internship Report

June 2024 - August 2024

Sankalp Rajeev

Bsc. Computer Science Graduate from Arizona State University

Incoming MSc.+ MSE in AI,Robotics Engineering from University of Michigan-Dearborn

Drone Software Development Engineer

Project Title : Drone Software Development

Table of Contents

Acknowledgements

I undertook this internship project and completed the internship under the guidance of Rohit Dey. I am grateful to the Mistral staff for their patience and assistance during my training at Mistral.

This internship has been an exceptional learning opportunity, and I am deeply appreciative of the chance to contribute to and learn from such a dynamic and forward-thinking team.

About the company

Mistral Solutions is an ISO 9001:2015, ISO 27001:2013, ISO 14001:2015, ISO 45001:2018 and AS9100D certified technology design and systems engineering company (Embedded System Services) providing end-to-end solutions for product design and application deployment. Mistral is focused on three business domains viz., Product Engineering Services, Aerospace & Defense and Homeland Security. Mistral is among the leading product design and development companies in Bangalore focusing on providing Embedded Technologies and services for global customers. Incepted in 1997, Mistral provides end-to-end embedded product design and development services for global customers.

Internship position

As a Drone Software Development Engineer intern, my main objectives were to acquire hands-on experience in drone technology and Artificial Intelligence . The internship provided a comprehensive exposure to several key areas:

- Software Development: I contributed to developing and refining software solutions for drone systems, focusing on improving functionality and performance.

- Hardware Integration: I was involved in integrating drone hardware components.
- Trials and Testing: I participated in testing drones, including high-altitude trials, to assess performance and reliability under various conditions.
- Learning New Technologies: The internship enabled me to explore new technologies and software, and understand their application in autonomous drone systems.

This experience significantly enhanced my technical skills and provided valuable insights into the practical aspects of drone development and AI.

Internship Milestones

Milestone 1 : Understanding drone architecture

1.1 About the drone:

The Mistral Xherpa Heavy Payload Drone is a rugged, all-weather Military Logistics drone designed to operate in harsh environments and at very high altitudes. It is a dodeca drone with a true hexa configuration. It uses BLDC motors and is a coaxial setup. The flight controller on board is the cube orange plus. The drone is designed to carry up to 50KG of payload.

1.2 Flight Avionics:

During my internship, I was assigned the task of understanding the Cube Orange Plus flight controller and its various connectivity options. The Cube Orange Plus is equipped with several ports, each serving specific functions to facilitate integration and communication within the drone system. The ports on the Cube Orange Plus are as follows:

- 2 CAN Ports: Used for connecting and communicating with various CAN bus devices, such as sensors and external peripherals.
- 2 GPS Ports: Dedicated to connecting GPS modules for accurate positioning and navigation.
- 2 Telemetry Ports: Provide connectivity for telemetry systems to transmit data between the drone and ground stations.
- 1 I2C Port: Facilitates communication with I2C-compatible sensors and devices.
- 1 USB Port: Allows for direct connection to external devices or for data transfer and firmware updates.
- 1 ADC Port: Used for connecting analog sensors to measure various physical parameters.
- 1 SPKT Port: Connects to the Spektrum receiver for radio control signals.

Understanding the functions and applications of these ports was essential for integrating the Cube Orange Plus into the drone system and ensuring effective communication between its components.

1.3 Utilizing Ground Control Software

I was tasked with learning Mission Planner, a ground control software essential for communication with drones. My work with Mission Planner included the following key activities:

- Documentation Review: I thoroughly reviewed the Mission Planner documentation to understand its features, functionalities, and operational procedures.
- Simulated Missions (SITL): I performed simulated missions using Software-in-the-Loop (SITL) to practice and refine mission planning and execution in a virtual environment.

- Waypoint/Fence Point Entry: I utilized the point-and-click interface to set waypoints and geofencing boundaries leveraging Google Maps for accurate placement.
- Mission Commands: I configured and executed mission commands by selecting options from drop-down menus, tailoring the mission parameters to requirements.
- Log Analysis: I downloaded mission log files and learnt how to analyze them.
- Cube Setup: I was responsible for setting up the Cube flight controller from scratch, including hardware configuration and integration.
- Calibration: I performed calibration of the flight controller to ensure accurate sensor readings and reliable operation.
- Pre-Flight Checks: I learnt how to conduct mandatory pre-flight checks to verify the readiness and safety of the drone system before deployment.

These tasks provided me with a solid understanding of ground control operations and simulation techniques.

Milestone 2 : Software and Artificial Intelligence

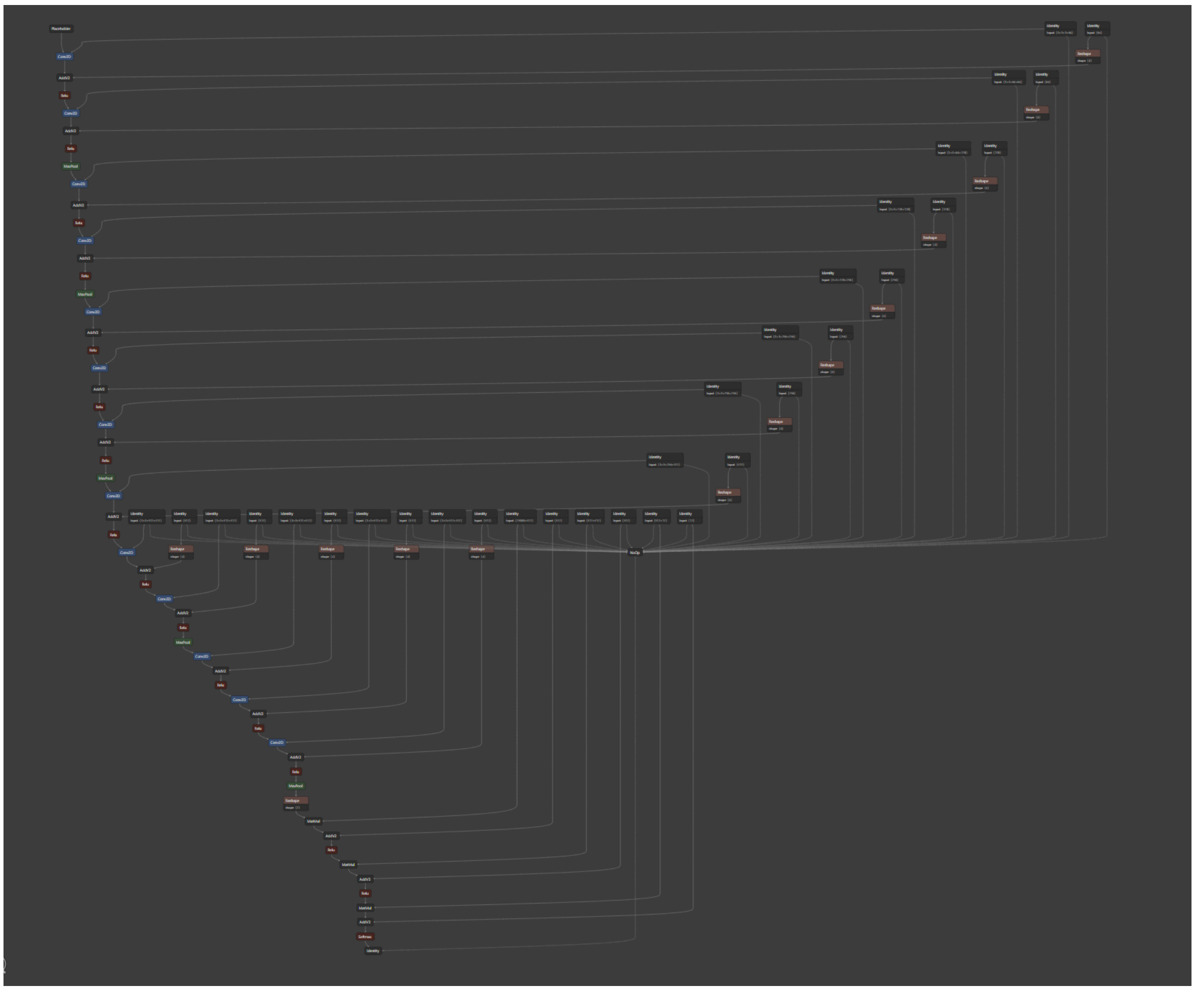
2.1 Creating a custom Image classification and object detection model

2.1.1 Image classification model

- Objective: Identify and track different types of military vehicles for accurate recognition.
- Model Development: Utilized transfer learning with VGG16 to build a custom image classification model.
- Dataset Collection: Obtained a dataset of 11 types of military vehicles from Kaggle.
- Data Pre-processing: Implemented custom scripts for scanning and cleaning the dataset to address corrupted files and ensure data quality.

- Model Performance:Model V1:
  - Achieved 88% accuracy.
  - Model V3: Improved accuracy to 96% through hyperparameter tuning.
- Integration: Deployed the classification model for use with static images, video files, and live webcam feeds.

This project demonstrated effective use of transfer learning, thorough data preparation, and successful model optimization for real-time vehicle identification and tracking.



Custom Model Architecture

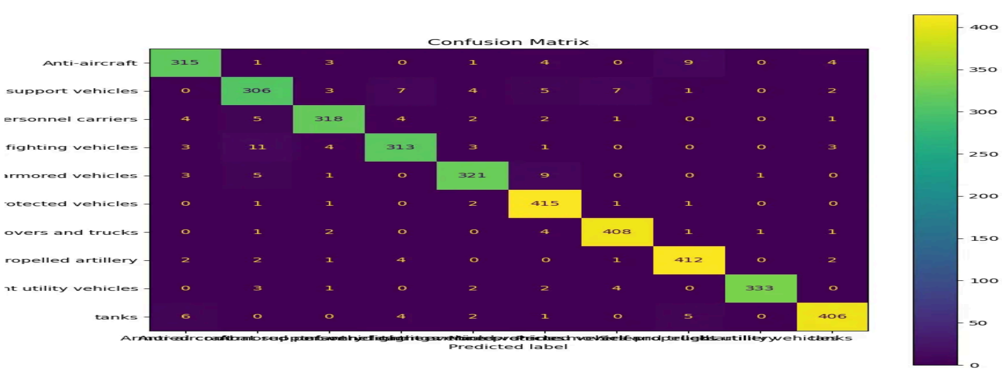
Model Summary

Layer (Type)	**Output Shape**	**Parameters**
input_layer (InputLayer)	(None, 224, 224, 3)	0
block1_conv1 (Conv2D)	(None, 224, 224, 64)	1,792
block1_conv2 (Conv2D)	(None, 224, 224, 64)	36,928
block1_pool (MaxPooling2D)	(None, 112, 112, 64)	0
block2_conv1 (Conv2D)	(None, 112, 112, 128)	73,856
block2_conv2 (Conv2D)	(None, 112, 112, 128)	147,584
block2_pool (MaxPooling2D)	(None, 56, 56, 128)	0
block3_conv1 (Conv2D)	(None, 56, 56, 256)	295,168
block3_conv2 (Conv2D)	(None, 56, 56, 256)	590,080
block3_conv3 (Conv2D)	(None, 56, 56, 256)	590,080
block3_pool (MaxPooling2D)	(None, 28, 28, 256)	0
block4_conv1 (Conv2D)	(None, 28, 28, 512)	1,180,160
block4_conv2 (Conv2D)	(None, 28, 28, 512)	2,359,808
block4_conv3 (Conv2D)	(None, 28, 28, 512)	2,359,808
block4_pool (MaxPooling2D)	(None, 14, 14, 512)	0
block5_conv1 (Conv2D)	(None, 14, 14, 512)	2,359,808
block5_conv2 (Conv2D)	(None, 14, 14, 512)	2,359,808

block5_conv3 (Conv2D)	(None, 14, 14, 512)	2,359,808
block5_pool (MaxPooling2D)	(None, 7, 7, 512)	0
flatten (Flatten)	(None, 25088)	0
dense (Dense)	(None, 512)	12,845,056
dropout (Dropout)	(None, 512)	0
dense_1 (Dense)	(None, 10)	5,130

Total Parameters: 67,425,632 (257.21 MB)
Trainable Parameters: 19,930,122 (76.03 MB)
Non-trainable Parameters: 7,635,264 (29.13 MB)
Optimizer Parameters: 39,860,246 (152.05 MB)

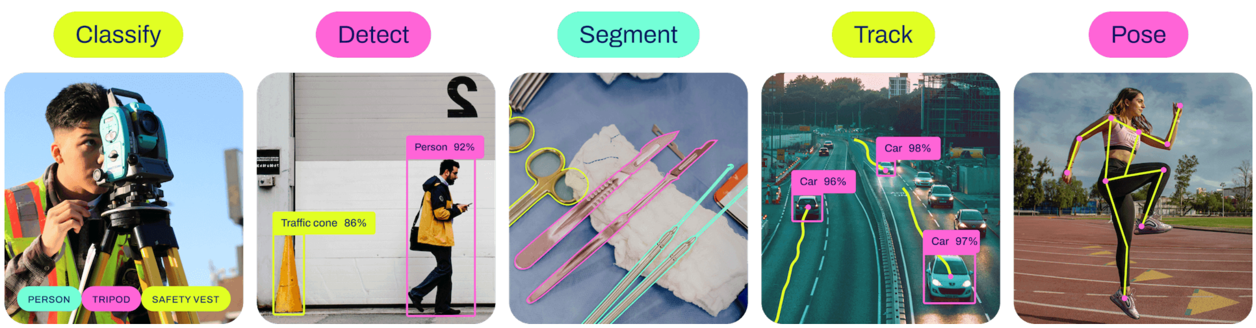
This model includes convolutional layers for feature extraction, pooling layers for dimensionality reduction, and dense layers for classification. The Dropout layer is used to reduce overfitting during training.



Confusion Matrix

2.1.2 Creating object detection and tracking model

- Integrated YOLOv8: Implemented YOLOv8 for advanced object detection, enabling the system to identify and locate objects within images effectively.
- Added ByteTrack: Incorporated ByteTrack, an object tracking algorithm, to follow and track the movement of detected objects across frames.
- Unified Model: Combined YOLOv8 for object detection and ByteTrack for tracking with the existing image classification model. This integration enables a comprehensive system that classifies objects, detects them, and tracks their movement in real-time.



2.1.3 Creating Custom GUI

Developed a Custom GUI for Model Evaluation with Integrated CI/CD Pipeline

Implemented a custom graphical user interface (GUI) using Flask to facilitate the evaluation and refinement of image classification models, integrated with a continuous integration/continuous deployment (CI/CD) pipeline for automated model updates.

System Overview:

- Automated Image Classification Handling: The system automatically saves classified images into a specified directory. Each image is named based on the model's prediction (e.g., an image classified as a tank is saved as tank_01.jpg).
- GUI for Manual Verification and Correction: The GUI provides a platform for verifying the accuracy of model predictions. Features include:
- Confirmation of Classifications: Users can review and confirm whether the model’s classification is accurate.

- Error Correction: If a classification is incorrect, users can modify the label using a dropdown menu. For instance, an incorrectly classified tank can be re-labeled as a drone and saved as drone_01.jpg in a new directory.
- Retraining Process: Images that are corrected and re-labeled are stored in a new folder, which is subsequently used to retrain the model. This iterative process ensures that the model continuously improves by incorporating corrected data and updated labels.

This system enhances model performance through real-time feedback and iterative improvements, ensuring higher accuracy and reliability in image classification tasks.

2.2 Lua scripting

2.2.1 Reading AHRS values from drone

Developed a Lua script for the extraction and display of Attitude and Heading Reference System (AHRS) values from the drone's onboard sensors. This script facilitates real-time monitoring of the drone's orientation and movement.

2.2.2 Integrating Temperature sensor

Integrated the DS1820 temperature sensor with the cube. Developed a Lua script to read temperature data from the sensor and transmit the information to the ground control station.

2.2.3 Working with Lidar

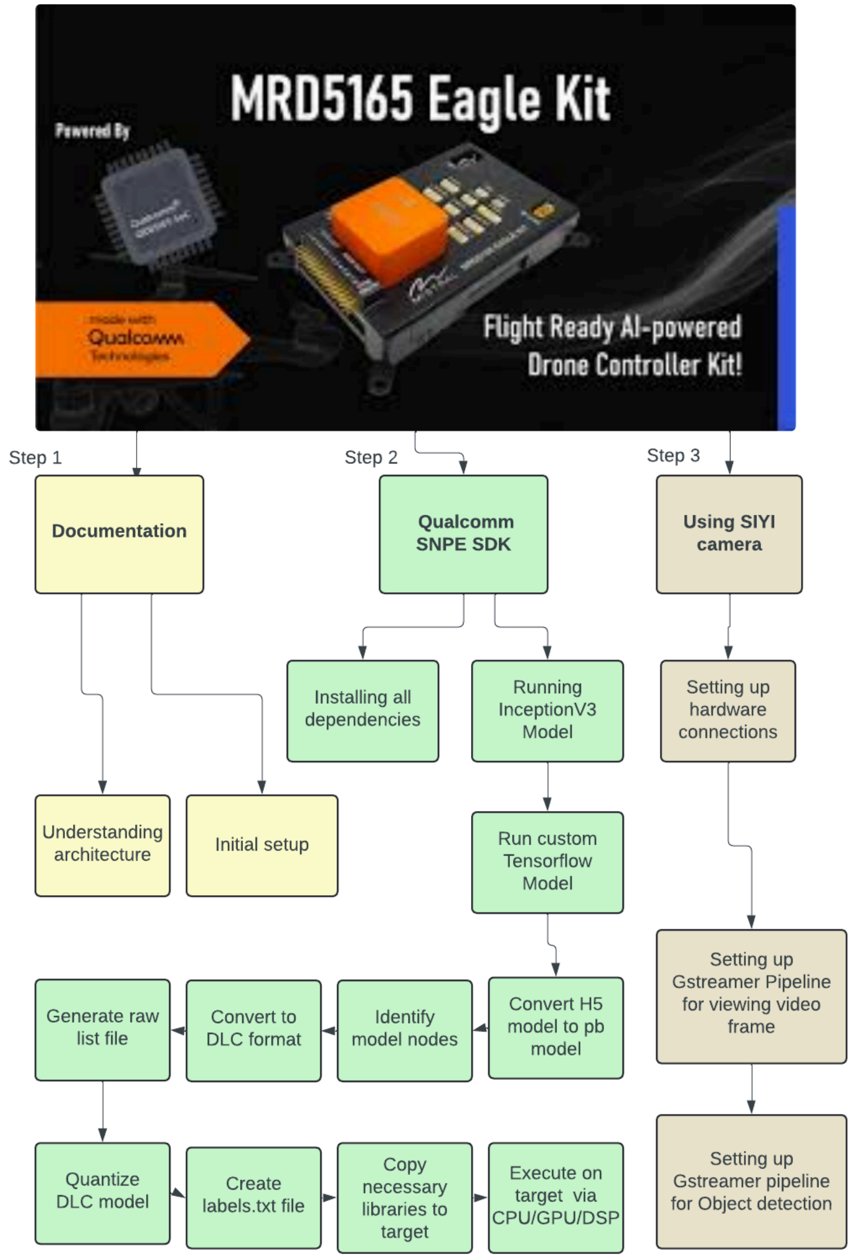
Implemented functionality for the Lightware LW20/C Lidar sensor within the drone's system. Developed a Lua script to enable basic obstacle avoidance capabilities. The script ensures that if an object is detected within a 100-meter range of the Lidar, the drone automatically transitions into Loiter or Return-to-Launch (RTL) mode. Additionally, it triggers a warning to the Ground Control Station (GCS) operator, providing real-time alerts to enhance operational safety and response.

2.3 Working with Mistral's MRD5165 Eagle Kit

2.3.1 About the MRD5165 Eagle kit

The MRD5165 Eagle Kit is a sophisticated development platform designed for applications in embedded systems and electronics. It features a high-performance microcontroller and a range of connectivity options, including UART, SPI, I2C, and GPIO pins, which support seamless integration with external devices. The kit includes support for various sensors and actuators, making it ideal for robotics, IoT projects, and real-time data acquisition. With its modular design, user-friendly development environment, and comprehensive documentation, the MRD5165 Eagle Kit accelerates the development process and offers flexibility for a wide array of experimental and industrial applications.

Mistral's MRD5165 Eagle kit with Cube orange plus



My workflow with the MRD5165

2.3.2 Working with the Eagle Kit

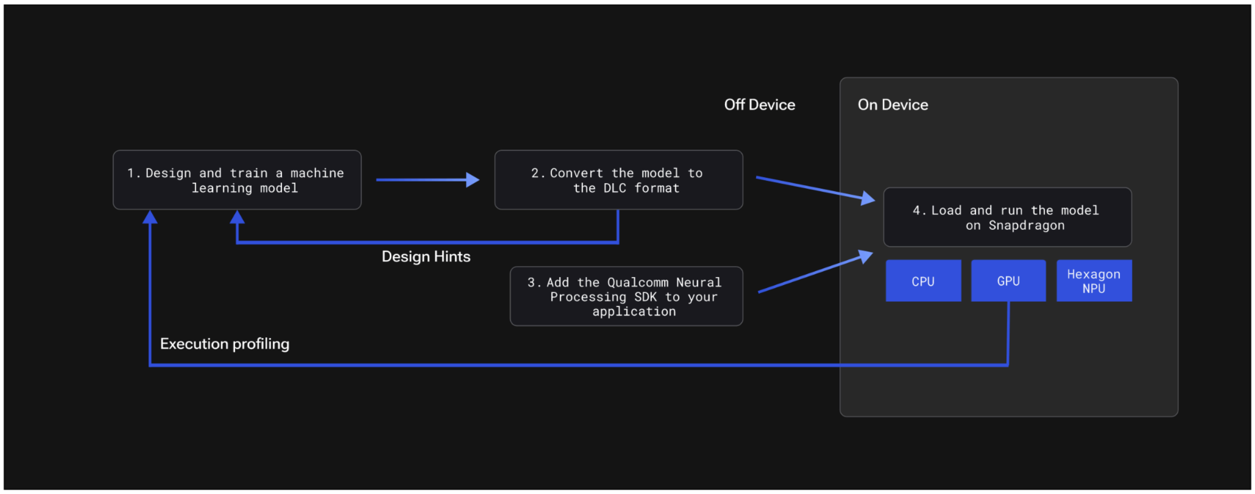
- Documentation Review and Initial Setup:
  I began by thoroughly reviewing the documentation to understand the architecture of the system. After familiarizing myself with the technical details, I proceeded with the development work on the board.

- SNPE SDK Integration
  Consulted Qualcomm's documentation and decided to use the Qualcomm Neural Processing SDK for AI. I carefully followed the installation instructions, successfully installing the SDK on the board. This set the stage for running machine learning models.

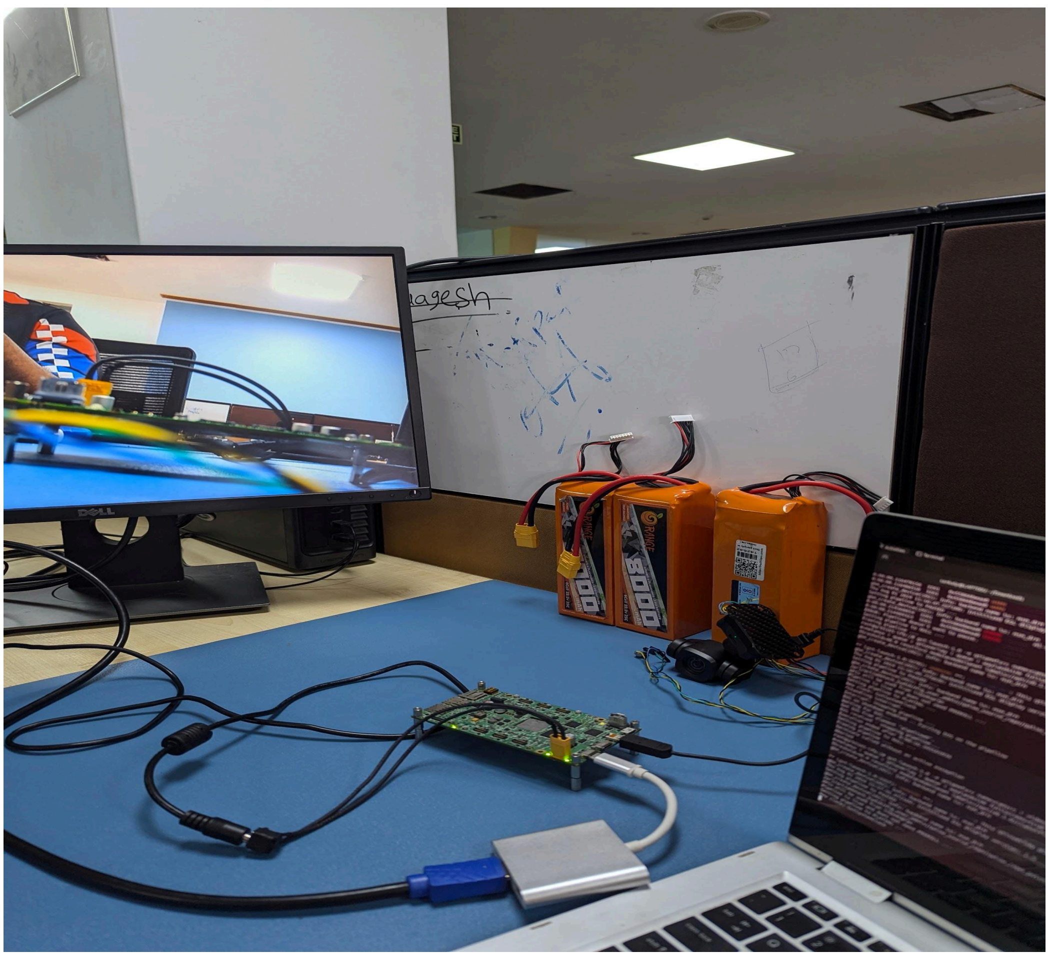
- Model Deployment and Testing:
  Focused on deploying the InceptionV3 model, following Qualcomm's pipeline closely. After extensive debugging and troubleshooting, I successfully ran the InceptionV3 model on the board, processing image inputs as intended.

- Custom AI Model Integration:
  Next, I loaded my custom AI model onto the board. Ensured that the model adhered to the required format and successfully executed it for image classification tasks.

- Documentation:
  Prepared thorough documentation on the process and implementation of the model integration, providing detailed instructions and insights to assist other team members.



SNPE SDK workflow



MRD5165 Connected with Gimbal camera

Milestone 3 : Hardware Integration and Validation

3.1 Understanding Arduino

- Gained proficiency with the Arduino Uno microcontroller and its development environment.
- Became familiar with the Arduino Integrated Development Environment (IDE), which facilitated writing, uploading, and debugging code.

This experience provided a solid foundation in using Arduino.

3.2 Integrating Sensors

- Integrated the DS1820 temperature sensor with an Arduino and established I2C communication between the Arduino and the Cube. Since the DS1820 sensor was incompatible with the Orange Cube, developed custom Arduino code and a Lua script to bridge the communication gap, ensuring proper data transfer and compatibility.
- Successfully integrated the Lightware LW20/C Lidar sensor with the Cube using I2C communication, facilitating accurate distance measurement and obstacle detection.
- Integrated the SIYI gimbal camera with the MRD5165 Eagle Kit using a GStreamer pipeline. This integration involved configuring the GStreamer framework to handle video streaming from the gimbal camera, enabling real-time video capture and processing.

3.3 Basic integration of hardware onto the drone

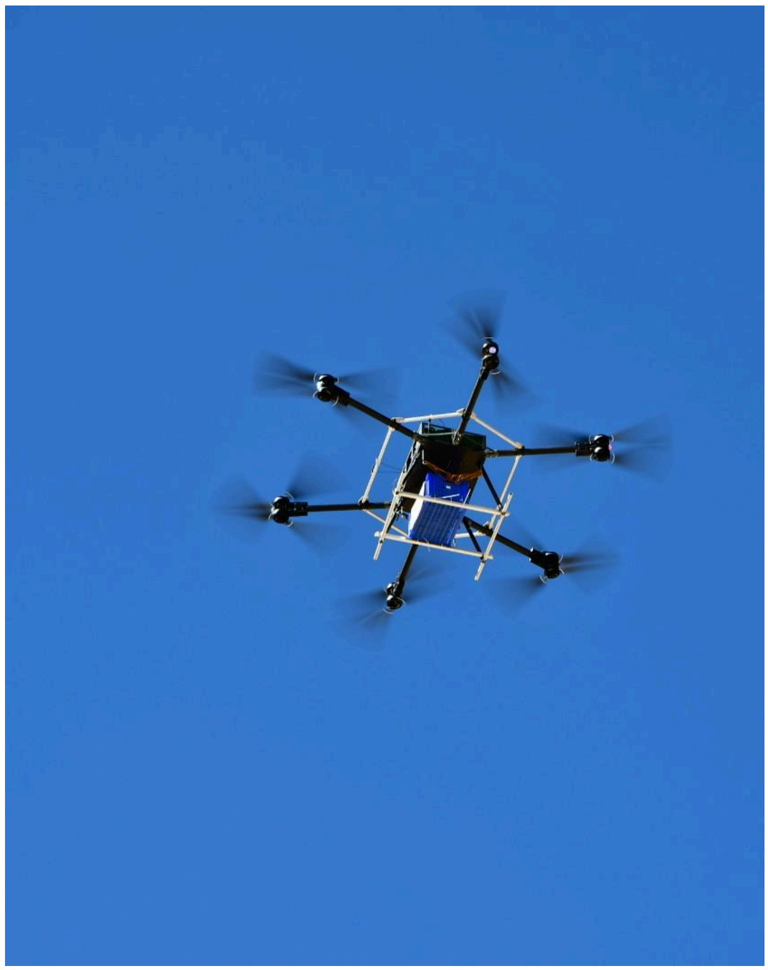
Assisted with the disassembly and reassembly of the drone, gaining hands-on experience with its components and mechanisms. Provided support during field operations, acquiring proficiency with essential tools and understanding basic operational procedures.
Learnt how to perform thrust calculations and selecting motors,propellers and batteries to estimate total flight time

Conclusion

The two-month internship program was an enriching experience filled with valuable lessons, challenges, and achievements. Key aspects of this experience included:

- Learning from Failures and Debugging: Overcoming obstacles through detailed debugging and learning from setbacks was essential to achieving even small successes.
- Embedded Programming and Drones: Gained hands-on experience in embedded programming and deepened understanding of drone functionality and programming.
- Ground Control Software: Became proficient in using ground control software, understanding the necessary preparations and procedures before and after drone flights.
- Trial Participation: Acquired practical knowledge on how to conduct effectively to field trials.
- Image Processing and Integration: Enhanced skills in image processing tasks and learned the basics of system integration.

These experiences have been instrumental in advancing my knowledge and skills as a Software Developer. I am grateful for the guidance and support from everyone who contributed to my growth during this internship.



XHERPA DRONE