

Autonomous Aerial Survey Platform

By

Md. Kowshik

20121020

Sultanul Arefin Nabil

20221030

Mushfeka Anika Bushra

21321070

Al Hossain Shawn

22121073

A Final Year Design Project (FYDP) submitted to the Department of Electrical and Electronic Engineering in partial fulfillment of the requirements for the degree of Bachelor of Science

Department of Electrical and Electronic Engineering
Brac University
January 2026

© 2026. Brac University
All rights reserved.

Autonomous Aerial Survey Platform

By

Md. Kowshik

20121020

Sultanul Arefin Nabil

20221030

Mushfeka Anika Bushra

21321070

Al Hossain Shawn

22121073

A Final Year Design Project (FYDP) submitted to the Department of Electrical and Electronic Engineering in partial fulfillment of the requirements for the degree of Bachelor of Science

Academic Technical Committee (ATC) Panel Member:

Dr. A.S. Nazmul Huda (Chair)

Associate Professor, Department of EEE, Brac University

Nahid Hossain Taz (Member)

Lecturer, Department of EEE, Brac University

Sharif Mohd Shams (Member)

Lecturer, Department of EEE, Brac University

Department of Electrical and Electronic Engineering

Brac University

January 2026

© 2026. Brac University

All rights reserved.

Declaration

It is hereby declared that

1. The Final Year Design Project (FYDP) submitted is my/our own original work while completing degree at Brac University.
2. The Final Year Design Project (FYDP) does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The Final Year Design Project (FYDP) does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. I/We have acknowledged all main sources of help.

Student's Full Name & Signature:

Md. Kowshik
20121020

Sultanul Arefin Nabil
20221030

Mushfeka Anika Bushra
21321070

Al Hossain Shawn
22121073

Approval

The Final Year Design Project (FYDP) titled “Autonomous Aerial Survey Platform” submitted by

1. Md. Kowshik (20121020)
2. Sultanul Arefin Nabil (20221030)
3. Mushfeka Anika Bushra (21321070)
4. Al Hossain Shawn (22121073)

of Fall, 2025 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of Bachelor of Science in Electrical and Electronic Engineering (BSEEE) on 05-01-2026.

Examining Committee:

Academic Technical
Committee (ATC):
(Chair)

Dr. A.S. Nazmul Huda
Associate Professor, Department of EEE
Brac University

Final Year Design Project
Coordination Committee:
(Chair)

Dr. Mirza Rasheduzzaman
Associate Professor, Department of EEE
Brac University

Department Chair:

Dr. Md. Mosaddequr Rahman
Professor and Chairperson, Department of EEE
Brac University

Ethics Statement

This is to state that the Final Year Design Project (FYDP) requirements have been met. As a result of the introduction of the "Autonomous Aerial Survey Platform" project. Our group members wrote this document in collaboration. All supporting writings that have been used have been given the relevant citations. during the analysis, literature review and data collection. According to the plagiarism report, the score for similarity with this report is 11%. The elements of this venture have been approved with the help of our Advisors.

Abstract

The project explores the concept of developing an autonomous aerial survey drone with the ability to conduct an efficient and high-resolution survey using a minimum level of human involvement. The system combines with powerful navigation and mission planning algorithms to provide feasible information gathering across various terrains. The drone can operate according to predefined survey patterns, dynamic environments. It can autonomously control data collection, storage and transmission using onboard real time optimization for post processing. The results of the survey are anonymized imagery and multi sensor data that can be used in surveying and fast evaluation in fields like agriculture, infrastructure inspection, disaster response and environmental monitoring. The project focuses on safety, reliability, and scalability. Whereby the survey operations can be repeated at a low cost and used in decision support and continuous monitoring.

Keywords: Automated data acquisition; deep learning; computer vision; operational security; automation.

Acknowledgement

We wish to express our sincere appreciation to the Chair of our ATC committee, Dr. A.S. Nazmul Huda Associate Professor, for his invaluable guidance and comprehensive evaluation of the project. In addition, we will thank Mr. Sharif Mohd Shams and Mr. Nahid Hossain Taz, Lecturers at the Department of Electrical and Electronic Engineering at Brac University, who have given their valuable suggestions that can help to make the project better.

Table of Contents

List of Figures	12
List of Tables	18
Chapter 1: Introduction [CO1, CO2, CO3, CO10]	19
1.1. Introduction	19
1.1.1. Problem Statement	20
1.1.2. Background Study	21
1.1.3. Literature Gap	27
1.1.4. Relevance to Current and Future Industry	30
1.2. Objectives, Requirements, Specification and Constraints	31
1.2.1. Objectives	31
1.2.2. Functional and Non-Functional Requirements	36
1.2.3. Specifications	36
1.2.4. Technical and Non-Technical Consideration and Constraint in Design Process	38
1.2.5. Applicable Compliance, Standards and Codes	39
1.3. Systematic Overview/Summary of the Proposed Project	40
1.4. Conclusion	41
Chapter 2: Project Design Approach [CO5, CO6]	42
2.1. Introduction	42
2.2. Identify Multiple Design Approach	42
2.3. Describe Multiple Design Approach	43
2.3.1. Design Approach 1	43
2.3.2. Design Approach 2	44
2.4. Analysis of Multiple Design Approach	45
2.5. Conclusion	46
Chapter 3: Use of Modern Engineering and IT Tool [CO9]	47
3.1. Introduction	47
3.2. Select Appropriate Engineering and IT Tools	47
3.3. Use of Modern Engineering and IT Tools	48
3.3.1. Python	48
3.3.2. WSL	48
3.3.3. TenorFlow	49
3.3.4. Keras	49

3.3.5. PowerShell	50
3.3.6. Bash	50
3.3.7. Unix Toolchain	51
3.3.8. Hyper-V	51
3.3.9. Ultralytics	52
3.3.10. MATLAB	52
3.3.11. TileServer GL	53
3.4. Conclusion	54
Chapter 4: Optimization of Multiple Design and Finding the Optimal Solution [CO5, CO6, CO7]	55
4.1. Introduction	55
4.1.1. Methodology	55
4.1.2. System Flow Chart	64
4.2. Optimization of Multiple Design Approach	64
4.2.1. Deep Learning Based Optimization	64
4.2.2. Dataset Pre Processing	68
4.2.3. Design and Development of a Custom Data Visualization Suite	72
4.2.4. Optimizing Design Approach 1	79
4.2.5. Optimizing Design Approach 2	110
4.2.6. General Object Detection Based Accuracy Optimization	125
4.2.7. Drone Simulation	143
4.3. Identify Optimal Design Approach	174
4.3.1. Computational Performance	174
4.4. Performance Evaluation of Developed Solution	186
4.5. Conclusion	187
Chapter 5: Completion of Final Design and Validation [CO8]	188
5.1. Introduction	188
5.2. Completion of Final Design	188
5.2.1. Software Stack	188
5.2.2. Hardware Components	196
5.3. Evaluate the Solution to Meed Desired Need	205
5.3.1. Optical Correction	205
5.3.2. Prototype Flight Test	206
5.3.3. Manual Flight	206

5.3.4. Autonomous Flight	207
5.3.5. Flight Data Preview	208
5.3.6. Detection Capability	209
5.4. Conclusion	213
Chapter 6: Impact Analysis and Project Sustainability [CO3, CO4]	214
6.1. Introduction	214
6.2. Assess the Impact of Solution [CO3]	214
6.3. Evaluate the Sustainability [CO4]	215
6.3.1. Environmental Factors	215
6.3.2. Economic Factors	217
6.3.3. Societal Factors	220
6.4. Conclusion	222
Chapter 7: Engineering Project Management [CO11, CO14]	223
7.1. Introduction	223
7.2. Define, Plan and Manage Engineering Project	223
7.2.1. EEE499P The Planning Phase	223
7.2.2. EEE499D The Design Phase	224
7.2.3. EEE499C The Completion Phase	224
7.2.4. Risk Management Plan	225
7.2.5. Additional Fault Mitigation Scheme:	226
7.3. Evaluate Project Progress	227
7.4. Conclusion	227
Chapter 8: Economical Analysis [CO12]	228
8.1. Introduction	228
8.2. Economic analysis	228
8.3. Cost Benefit Analysis	229
8.4. Evaluate Economic and Financial Aspects	232
8.5. Conclusion	233
Chapter 9: Ethics and Professional Responsibilities [CO2, CO13]	234
9.1. Introduction	234
9.2. Identify Ethical Issues and Professional Responsibility	234
9.3. Apply Ethical Issues and Professional Responsibility	234
9.4. Conclusion	235

Chapter 10: Conclusion and Future Work	236
10.1. Project Summary	236
10.2. Future Work	236
Chapter 11: Identification of Complex Engineering Problems and Activities	237
11.1. Identify the attribute of complex engineering problem (EP)	237
11.2. Provide reasoning how the project address selected attribute (EP)	237
11.3. Identify the attribute of complex engineering activities (EA)	237
11.4. Provide reasoning how the project address selected attribute (EA)	238
References	239
Appendix	247
FYDP P Log Book	329
FYDP D Log Book	337
FYDP C Log Book	343

List of Figures

Figure 1: Systematic Overview of Design Approach 1	43
Figure 2: Systematic Overview of Design Approach 2	44
Figure 3: List of Selected IT Tools	47
Figure 4: Methodology of the Project	55
Figure 5: Data Acquisition and Pre-Processing Pipeline	56
Figure 6: Types of Computer Vision Problems	61
Figure 7: Virtual Environment Architecture	62
Figure 8: Virtual Machine Memory Management	63
Figure 9: System Flow Chart	64
Figure 10: Data Point Mesh Generator	76
Figure 11: Custom Data Verification Tool	77
Figure 12: Deep Exploration Tool	78
Figure 13: Design Approach 1 Overview	79
Figure 14: Training vs Validation Accuracy for ClassicCNN	80
Figure 15: Training vs Validation Accuracy for EfficientNetB0	81
Figure 16: Training vs Validation Accuracy for MobileNet	82
Figure 17: Training vs Validation Accuracy for MobileNetV2	83
Figure 18: Training vs Validation Accuracy for NasNetMobile	84
Figure 19: Training vs Validation Accuracy for ResNet50	85
Figure 20: Training vs Validation Accuracy for ResNet50V2	86
Figure 21: Training vs Validation Accuracy for ResNet101	87
Figure 22: Training vs Validation Accuracy for ResNet101V2	88
Figure 23: Training vs Validation Accuracy for VGG16	89
Figure 24: Training vs Validation Accuracy for VGG19	90
Figure 25: Training vs Validation Accuracy for Xception	90
Figure 26: Average Power Draw During Training for All 12 Models	91
Figure 27: Peak Power Draw During Training for All 12 Models	92
Figure 28: Total Energy Consumption During Training for All 12 Models	93
Figure 29: Time Required Per Iteration for All 12 Models	94
Figure 30: Time Required Per Step for All 12 Models	95
Figure 31: Total Time to Train All 12 Models for 100 Epochs	96
Figure 32: Power Draw Profile for ClassicCNN in x86-64 Hardware	97
Figure 33: Power Draw Profile for EfficientNetB0 in x86-64 Hardware	97
Figure 34: Power Draw Profile for MobileNet in x86-64 Hardware	98
Figure 35: Power Draw Profile for MobileNetV2 in x86-64 Hardware	98
Figure 36: Power Draw Profile for NasNetMobile in x86-64 Hardware	99
Figure 37: Power Draw Profile for ResNet50 in x86-64 Hardware	99
Figure 38: Power Draw Profile for ResNet50V2 in x86-64 Hardware	100
Figure 39: Power Draw Profile for ResNet101 in x86-64 Hardware	100
Figure 40: Power Draw Profile for ResNet101V2 in x86-64 Hardware	101
Figure 41: Power Draw Profile for VGG16 in x86-64 Hardware	101
Figure 42: Power Draw Profile for VGG19 in x86-64 Hardware	102

Figure 43: Power Draw Profile for Xception in x86-64 Hardware	102
Figure 44: Total Energy Consumption During Inference for All 12 Models	103
Figure 45: Peak Power Draw During Inference for All 12 Models	104
Figure 46: Average Power Draw During Inference for All 12 Models	105
Figure 47: Time to Process 1k Frames for All 12 Models	106
Figure 48: Time Required to Process One Frame for All 12 Models	107
Figure 49: Theoretical Max Video Framerate for All 12 Models	108
Figure 50: Inference Efficiency for All 12 Models	109
Figure 51: Design Approach 2 Overview	110
Figure 52: Training and Validation Accuracy for Top 4 Models	111
Figure 53: Average Power Draw on ARM for Top 4 Models	112
Figure 54: Peak Power Draw on ARM for Top 4 Models	113
Figure 55: Total Energy Consumption During Training on ARM for Top 4 Models	114
Figure 56: Training Time Required Per Iteration on ARM for Top 4 Models	115
Figure 57: Total Training Time on ARM for Top 4 Models	116
Figure 58: Power Draw Profile on ARM for EfficientNetB0	117
Figure 59: Power Draw Profile on ARM for MobileNet	117
Figure 60: Power Draw Profile on ARM for ResNet50V2	118
Figure 61: Power Draw Profile on ARM for Xception	118
Figure 62: Total Energy Consumption During Inference on ARM for Top 4 Models	119
Figure 63: Peak Power Draw During Inference on ARM for Top 4 Models	120
Figure 64: Average Power Draw During Inference on ARM for Top 4 Models	121
Figure 65: Time to Process 1k Frames on ARM for Top 4 Models	122
Figure 66: Time Required to Process One Frame on ARM for Top 4 Models	123
Figure 67: Theoretical Max Framerate on ARM for Top 4 Models	124
Figure 68: Inference Efficiency on ARM for Top 4 Models	125
Figure 69: Environment Setup for Object Detection	127
Figure 70: Class Balancing for Object Detection	128
Figure 71: YOLOv8 Training Metrics Without Early Stopping	130
Figure 72: YOLOv8 Training Metrics With Early Stopping	130
Figure 73: YOLO11n Training Metrics Without Early Stopping	131
Figure 74: YOLO11n Training Metrics With Early Stopping	131
Figure 75: YOLO12n Training Metrics Without Early Stopping	132
Figure 76: YOLO12n Training Metrics With Early Stopping	133
Figure 77: YOLO12s Training Metrics Without Early Stopping	133
Figure 78: YOLO12s Training Metrics With Early Stopping	134
Figure 79: Confusion Matrix for YOLOv8	135
Figure 80: Confusion Matrix for YOLO11n	136
Figure 81: Confusion Matrix for YOLO12n	137
Figure 82: Confusion Matrix for YOLO12s	138
Figure 83: Crack Detection	139
Figure 84: Corrosion Spot Detection	140
Figure 85: Pothole Detection	141
Figure 86: Insulator Fault Detection	142

Figure 87: Entire Drone Control Methodology	143
Figure 88: Environmental Factors on the Entire Architecture	143
Figure 89: Referencing Using Sensors	144
Figure 90: Maneuver Control	145
Figure 91: Quadcopter Trajectory for Medium Density Zigzag	146
Figure 92: Trajectory Verification Points During Drone Flight for Medium Density Zigzag	147
Figure 93: Trajectory Error Correction by Speed & Yaw Adjustment for Medium Density Zigzag	147
Figure 94: Aerodynamic Profile and Trajectory Tracking of the Drone for Medium Density Zigzag	148
Figure 95: Battery Discharge and Motor Power Dynamics for Medium Density Zigzag	148
Figure 96: Quadcopter Trajectory for High Density Zigzag	149
Figure 97: Trajectory Verification Points During Drone Flight for High Density Zigzag	149
Figure 98: Trajectory Error Correction by Speed & Yaw Adjustment for High Density Zigzag	150
Figure 99: Aerodynamic Profile and Trajectory Tracking of the Drone for High Density Zigzag	150
Figure 100: Battery Discharge and Motor Power Dynamics for High Density Zigzag	151
Figure 101: Quadcopter Trajectory for Quadrilateral Multi-pass Zigzag	151
Figure 102:: Trajectory Verification Points During Drone Flight for Quadrilateral Multi-pass Zigzag	152
Figure 103: Trajectory Error Correction by Speed & Yaw Adjustment for Quadrilateral Multi-pass Zigzag	152
Figure 104: Aerodynamic Profile and Trajectory Tracking of the Drone for Quadrilateral Multi-pass Zigzag	153
Figure 105: Battery Discharge and Motor Power Dynamics for Quadrilateral Multi-pass Zigzag	153
Figure 106: Quadcopter Trajectory for Staggered Continuous Zigzag	154
Figure 107: Trajectory Verification Points During Drone Flight for Staggered Continuous Zigzag	154
Figure 108: Trajectory Error Correction by Speed & Yaw Adjustment for Staggered Continuous Zigzag	155
Figure 109: Aerodynamic Profile and Trajectory Tracking of the Drone for Staggered Continuous Zigzag	155
Figure 110: Battery Discharge and Motor Power Dynamics for Staggered Continuous Zigzag	156
Figure 111: Quadcopter Trajectory for Maximal Width Staggered Continuous Zigzag	156
Figure 112: Trajectory Verification Points During Drone Flight for Maximal Width Staggered Continuous Zigzag	157
Figure 113: Trajectory Error Correction by Speed & Yaw Adjustment for Maximal Width Staggered Continuous Zigzag	157
Figure 114: Aerodynamic Profile and Trajectory Tracking of the Drone for Maximal Width Staggered Continuous Zigzag	158

Figure 115: Battery Discharge and Motor Power Dynamics for Maximal Width Staggered Continuous Zigzag	158
Figure 116: Quadcopter Trjectory for Pseudo Penrose Pattern with Minimal Scattering	159
Figure 117: Trajectory Verification Points During Drone Flight for Pseudo Penrose Pattern with Minimal Scattering	159
Figure 118: Trajectory Error Correction by Speed & Yaw Adjustment for Pseudo Penrose Pattern with Minimal Scattering	160
Figure 119: Aerodynamic Profile and Trajectory Tracking of the Drone for Pseudo Penrose Pattern with Minimal Scattering	160
Figure 120: Battery Discharge and Motor Power Dynamics for Pseudo Penrose Pattern with Minimal Scattering	161
Figure 121: Quadcopter Trjectory for Pseudo Penrose Pattern with Moderate Scattering	161
Figure 122: Trajectory Verification Points During Drone Flight for Pseudo Penrose Pattern with Moderate Scattering	162
Figure 123: Trajectory Error Correction by Speed & Yaw Adjustment for Pseudo Penrose Pattern with Moderate Scattering	162
Figure 124: Aerodynamic Profile and Trajectory Tracking of the Drone for Pseudo Penrose Pattern with Moderate Scattering	163
Figure 125: Battery Discharge and Motor Power Dynamics for Pseudo Penrose Pattern with Moderate Scattering	163
Figure 126: Quadcopter Trjectory for Weighted Area Pan	164
Figure 127: Trajectory Verification Points During Drone Flight for Weighted Area Pan	164
Figure 128: Trajectory Error Correction by Speed & Yaw Adjustment for Weighted Area Pan	165
Figure 129: Aerodynamic Profile and Trajectory Tracking of the Drone for Weighted Area Pan	165
Figure 130: Battery Discharge and Motor Power Dynamics for Weighted Area Pan	166
Figure 131: Quadcopter Trjectory for Adaptive Weighted Area Pan	166
Figure 132: Trajectory Verification Points During Drone Flight for Adaptive Weighted Area Pan	167
Figure 133: Trajectory Error Correction by Speed & Yaw Adjustment for Adaptive Weighted Area Pan	167
Figure 134: Aerodynamic Profile and Trajectory Tracking of the Drone for Adaptive Weighted Area Pan	168
Figure 135: Battery Discharge and Motor Power Dynamics for Adaptive Adaptive Weighted Area Pan	168
Figure 136: Quadcopter Trjectory for Simple Topographic Path	169
Figure 137: Trajectory Verification Points During Drone Flight for Simple Topographic Path	169
Figure 138: Trajectory Error Correction by Speed & Yaw Adjustment for Simple Topographic Path	170
Figure 139: Aerodynamic Profile and Trajectory Tracking of the Drone for Simple Topographic Path	170
Figure 140: Battery Discharge and Motor Power Dynamics for Simple Topographic Path	171

Figure 141: Quadcopter Trjectory for Dynamic Topographic Path	171
Figure 142: Trajectory Verification Points During Drone Flight for Dynamic Topographic Path	172
Figure 143: Trajectory Error Correction by Speed & Yaw Adjustment for Dynamic Topographic Path	172
Figure 144: Aerodynamic Profile and Trajectory Tracking of the Drone for Dynamic Topographic Path	173
Figure 145: Battery Discharge and Motor Power Dynamics for Dynamic Topographic Path	173
Figure 146: Training vs Validation Accuracy Curves for Top 4 Performing Models	174
Figure 147: Power Draw Profiles for Top 4 Performing Models	175
Figure 148: Time Deference to Process 1k Frames	176
Figure 149: Time Deference to Process One Frame	177
Figure 150: Inference Efficiency Comparision	178
Figure 151: Inference Energy Consumption Comparision	179
Figure 152: Comparision of Maximum Theoretical Framerate	180
Figure 153: Peak Power Draw Comparision During Training	181
Figure 154 Average Power Draw Comparision During Training	182
Figure 155: Comparision of Total Energy Consumption During Training	183
Figure 156: Peak Power Draw Comparision During Inference	184
Figure 157: Power Draw Profile Comparision	185
Figure 158: Total Training Time Comparision	186
Figure 159: High Level System Software Overview	188
Figure 160: Max Raw Usable FPS	190
Figure 161: Actual Useful FPS	191
Figure 162: Time Sync Architecture	192
Figure 163: Location Data Recording System Overview	193
Figure 164: 4x4 ArUco Marker	194
Figure 165: Full Version of DroneLAB	195
Figure 166: Modified Version of MapDVT	195
Figure 167: High Level Overview of Flight System	199
Figure 168: High Level Overview of Imaging Sub-System	200
Figure 169: Camera Modules	201
Figure 170: High Level Overview of Power Delivery System	202
Figure 171: Communication Architecture	203
Figure 172: Telemetry Radio	203
Figure 173: Flight Data Recorder	204
Figure 174: Optical Image Correction Using Checker Board Pattern	205
Figure 175: Initial Flight Test	206
Figure 176: Manual Flight and Telemetry Test	206
Figure 177: Autonomous Flight Test	207
Figure 178: GPS Drift	208
Figure 179: Post Calibration GPS Performance	208
Figure 180: Road Tracking RGB	209

Figure 181: Road Tracking 1 Bit	210
Figure 182: Path Tracing	210
Figure 183: Shallow Angle Detection	211
Figure 184: Detection Performance	211
Figure 185: Manual Override	212
Figure 186: Real Time Measurement Assist	213
Figure 187: EEE499P Gantt Chart	224
Figure 188: EEE499D Gantt Chart	224
Figure 189: EEE499C Gantt Chart	225
Figure 190: Commercial Mapping Software	230
Figure 191: Commercial Flight Control Software	231

List of Tables

TABLE I Non Exhaustive List of Codes Regarding Bangladesh	39
TABLE II Concise List of International Codes	40
TABLE III Direct Comparison of Both Design Approaches	45
TABLE IV Classification Model Parameters	60
TABLE V Optimal Design Comparison Criteria for Choosing the Final Solution	187
TABLE VI Components	196
TABLE VII Project Plan EEE499P	223
TABLE VIII Project Plan EEE499D	224
TABLE IX Project Plan EEE499C	224
TABLE X Risk Response Matrix	225
TABLE XI Design Approach 1 Budget	228
TABLE XII Design Approach 2 Budget	228
TABLE XIII Cost of Existing Market Solution	229
TABLE XIV Budget Breakdown for Autonomous Areal Survey Platform	229
TABLE XV List of Attributed Regarding Complex Engineering Problems	237
TABLE XVI List of Attributed Regarding Complex Engineering Activities	237

Chapter 1: Introduction [CO1, CO2, CO3, CO10]

1.1. Introduction

Routine and systematic checks on infrastructure can show developing faults in an early state. Taking action based on data obtained from surveys is critical to ensure safety of life and assets. The field of surveying has always been an involved and manual task from the beginning. Even with the advent and prevalence of modern tech it has not experienced any ground breaking advances compared to other fields. Especially in developing nations where the expenditure and capital limit often hinder timely survey operations. Surveys help specialists to get a detailed situation report of any area of interest. Performing such operations after regular intervals can provide insight and data which can later be scrutinized. Results obtained from these tests are used to create a road map of tasks that should be carried out. However, there are a few key limiting factors. The budget and manpower required increase exponentially with the scope and duration of a project. These skyrocketing costs often deter authorities who are fighting with strict budget limits. As a result timely surveys are not carried out but most often surveys are omitted as a whole. Causing significant gaps in the understanding and situational awareness of authorities of the state of the structure/installation. Thus preventable accidents cause death and destruction. This project aims to solve this issue by automating this method thus drastically cutting costs, manpower requirements so that frequent detailed surveys can be carried out with minimal financial investment. Surveying has undergone a radical development, moving towards a labor manual process to advanced, technologically driven processes that reinvents efficiency, precision and safety standards. Although traditional surveying is also considered to be fundamental development of infrastructure and environmental evaluation, they have become more and more demonstrably competent in achieving modern needs in terms of fast data capture, high spatial resolution and cost effective operations. Traditional techniques like total station survey and aerial reconnaissance by humans necessitates weeks in the field, massive allocation of manpower, and heavy capital expenditure, and at the same time, exposing staff to dangerous circumstances and less data contentions than modern alternatives. The advent of the Unmanned Aerial Vehicle (UAV) technology has inspired a shift in this paradigm. Surveying practices provide unprecedented abilities to autonomously gather data across diverse applications. UAV based systems that utilize high end sensing capabilities, such as Structure from Motion (SfM) imaging, thermal, LiDAR, and multi sensor spectral cameras show the ability of undertaking comprehensive surveys within hours. At the same time producing millions of data points that are more accurate. These websites deal with urgency, drawbacks of conventional methods such as inaccessibility to danger, topography, poor sampling density, impaired human safety and poor detection and special purpose capabilities (e.g. wildlife monitoring and infrastructure) Assessment. The Autonomous Aerial Survey Platform represents a strategic response to these identified challenges, integrating cutting edge drone technology with sophisticated sensor arrays and computational processing capabilities. The present project is going to create a flexible, self driving system. surveying solution based on the synergistic benefits of aerial mobility, multi sensor, processing automatic data, and high resolution comprehensive spatial integration. Information and reducing resource spending, time constraint, and operational risks. In infrastructure monitoring, environmental evaluation and topographical mapping applications.

1.1.1. Problem Statement

The financial and manpower requirement for any survey depends on the scope, duration and type of installation (high security and hazardous sites require expensive gear and specially trained personnel) where it will be carried out. Aside from this a few contributing factors often hinder the frequent survey operations from being carried out. Traditional surveys are carried out manually, thus the quality is primarily limited by the skill, equipment and environment that the personnel will operate in. Established methods suffer from a few (discussed in detail in the following background study section) critical flaws primarily including but not limited to the following, time and economic limitations, data density and quality, safety and accessibility, deficiencies in sensing and detection capabilities, institutional and training barriers. The origins, details, effects and mitigation strategies of these limiting factors are discussed in later parts of this document. Modern surveying experiences multimodality issues that essentially limit the efficiency of operations, quality of data, safety guidelines, and cost effectiveness in all aspects of infrastructure monitoring, environmental analysis, and topographical mapping. The traditional surveying methodologies, with their historical importance and long standing reliability, exhibit severe shortcomings that keep reducing their applicability in the current environment that requires high resolution data of high turnaround and economic operations. Traditional methods of surveying have grossly limited time efficiency, and need as much as sevenfold the time of modern methods to cover similar survey geographies. Fieldwork often spans over several weeks, which is directly linked to increased labor expenses, loss of project schedule, and decreased the ability of an organization in carrying out numerous surveys at once. The conventional approaches result in data that have huge spatial distance, which restricts resolutions and comprehensiveness. By contrast, the current drone systems produce millions of continuous data layers that disclose minor details that are not visible in traditional methods. Surveys conducted by humans in dangerous localities subject their staff to high safety hazards caused by machine malfunctions, damaged infrastructure, and unstable ground. These hazards are especially severe in remote areas, where the rescue may not be carried out in time. Moreover, such terrain has accessibility constraints which form physical barriers which in many cases lead to partial data or total failure to survey key areas. The conventional surveying methods have significant low quality detection, especially in ecological and infrastructure surveying. As an illustration, species counts based on direct observation have lower detection quotas, particularly with sensitized or sparse populations. On the same note, infrequent hand checks on infrastructure usually do not impact early signs of structural deterioration, making the structures more susceptible to disastrous collapses. The teaching of academic surveying is still restricted to the conventional methodology, which leaves students with a restricted opportunity of practical application because of the time constraints, site availability issue, inadequate instrumentation and adverse surroundings. This methodology inhibits institutional change, as organizational attachment to old fashioned methods and risk averse patterns of decision making hamper the implementation which are better proven as modern methods in place of outperforming ones. Traditional surveying techniques are deeply underprivileged where only a few data is generated which lacks essential information in a complex setting such as a forest. They also put the personnel at great risk of danger in hazardous or distant locations

where rescue is not easy. Also, these technologies cannot be successfully applied to specialized tasks, such as wildlife detection or infrastructure deterioration early warning. In order to eliminate these shortcomings, this project develops an Autonomous Aerial Survey Platform. This system combines the UAV mobility and high capability sensors as well as automated processing to safely obtain high density and accurate data in all landscapes. It eradicates human risk, saves time and money and alters the ways of infrastructure, environmental, and topographical surveying.

1.1.2. Background Study

Moving from labor intensive traditional methods to sophisticated technologically assisted processes, surveying has undergone significant transformation in recent years. The modern methods offer much better efficiency, comprehensive data collection and most importantly accuracy of that collected data. Driven by the need to overcome the inherent limitations of conventional surveying techniques that require substantial resources in the form of time, manpower and hefty financing. The development and implementation of new techniques has revolutionized various applications ranging from infrastructure monitoring, environmental assessment and even wildlife observation/preservation. These methods provide unprecedented insights while minimizing resource expenditure. By examining the progression from traditional surveying techniques to bleeding edge methods, comparative analysis of advantages and applications across diverse sectors one can find and identify key pain points and optimize/adapt each technique for implementation in future surveying practices.

Traditional surveying methodologies have served as the bedrock of environmental assessment, land measurement and infrastructure monitoring. These conventional approaches may have laid the foundation for modern processes but have slowly fallen short of meeting current requirements as all of them require substantial manual labor, intervention and resource allocation in every step in the form of field measurements, total station surveys and human occupied aerial surveys. A primary tool in traditional surveying is the total station, which culminates electronic distance and angular measurement capabilities (most of them have high accuracy positioning systems), with all these features this device can determine its spatial orientation or the environmental deviation with respect to its origin with a superior degree of measurement confidence compared to any other contemporary error prone manual method [1][2]. While effective for certain use cases, these traditional methods leave a lot to be desired as they require periods of extensive fieldwork, oftentimes lasting for weeks at a time where their modern counterparts can complete the task (often with a wider scope, more accurate data gathering and precise logging) in mere hours [2].

Orthodox surveying techniques have stood the test of time despite people being aware of their obvious limitations primarily due to financing constraints and institutional inertia. The persistence of these methods speak volumes to their trust garnered by large scale works in the past which leveraged purely manual and often archaic methods (luddites in the industry still advocating for it, although this is subject to change as people cannot deny the benefits for too long in the face of the leapfrog pace of technological advancement). Continued reliance on

these methods has been facilitated by strong drivers like monetary limitations and time crunches within many organizations, despite being known to yield biased results or samples that do not represent the subject altogether [3]. Especially challenging for traditional surveyors are forested areas, where due to obvious environmental aspects severely decrease their effectiveness and substantially impede measurement accuracy and comprehensive coverage (mainly by limiting movement due to poor visibility, physical obstacles as traversing the terrain is key to gathering data) [3]. These inherent limitations of these approaches have increasingly raised the need and generated interest in finding alternative methods that offer greater efficiency without compromising on data quality. Institutional teaching methods surrounding surveying have also been curbed by traditional approaches, as experimental teaching methods often present student/trainees with lackluster opportunities for real world applications due to temporal limitations, site availability, lack of proper instrumentation and unfavorable environmental conditions [4]. These academic limitations further emphasize the need for innovation to ensure the future are equipped with both theoretical knowledge, practical competency so that they can sharpen themselves and also further enhance surveying practices with their prowess.

There are significant limitations of conventional surveying methods impacting their effectiveness across a wide plethora of applications. One primary constraint is the low sampling density (large steps between data points and these gaps between samples hinder the effectiveness of gathered data as the experts cannot paint a finer picture) due to inherent flaws in these methods. Basically, this flaw substantially limits the resolution and comprehensiveness of the collected data [1]. Studies indicate that conducting surveys using traditional techniques often result in a maximum seven fold increase in time compared to modern alternatives in equivalent areas which further point out that these older methods are exceedingly more time-intensive [2]. A substantial time investment directly correlates to increased labor costs and a delayed project timeline. Accuracy and representativeness of collected data have also caused both speculation and raised questions in systematic comparisons. Research examining the bias in the data obtained through traditional techniques of boreal forest surveys have highlighted significant discrepancies with probabilistic sampling methods applied to the same geographic area [3]. These discrepancies point to fundamental concerns regarding the reliability of gathered data through conventional means, especially for applications that require precise environmental characterization or infrastructure assessment. Considerations in safety draw attention to another significant negative side of conventional surveying approaches. This problem is exacerbated in hazardous or areas with limited access due to terrain or other environmental factors. Remote locations are especially risky as rescue attempts are often futile and cannot be carried out in time often resulting in loss of life. Human-occupied aerial surveys have posed risks to personnel as equipment failure leads to fatalities in regions with challenging topologies [5]. Similarly, data collection efforts in compromised infrastructure or unstable terrain expose surveyors to potentially dangerous conditions (HAZMAT, radiological or plain and simple physical risks). All of these above can be easily avoided with the implementation of remote sensing equipment.

Another considerable limitation is the detection capability, one example being ecological assessment. Investigations comparing conventional wildlife surveying methods and modern alternatives show a rather low detection rate [5][6]. This disparity between detection rates are especially noticeable when surveying obscure or species with low population densities, the fact that the bottleneck is human visual observation becomes clear as day. Advancements in surveying methods can be directly linked to the evolution and accelerated development of various technological processes attempting to address the drawbacks of traditional methods. Drone technology has caused a revolution and has created a paradigm shift in surveying methods. Drone based aerial laser scanning offers unprecedented resolution for topographical mapping and infrastructure assessment. Studies indicate that this method can yield 6.5 times the number of cross sections and almost 9 times more points per cross section when compared to total station surveys maintaining equivalent accuracy for core metrics like channel width and cross sectional area measurements [1]. A drastic increase in data density helps to portray a more comprehensive picture of the terrain or to generate better results when characterizing infrastructure. UAV-SfM (Structure from Motion) platforms present another revolutionary approach which combines aerial imaging with complex computer processing. This method can generate ortho-photos and high resolution Digital Elevation Models (DEM), together creating a comprehensive contextually sound dataset which exceeds the performance standards of regular topographical maps [2]. Integration of thermal imaging sensors has enhanced surveying capabilities to another degree, enabling efficient wildfire detection and environmental observation with accuracy far greater than conventional aerial surveys [5]. Technologies that leverage satellites have transformed large surveying applications to a similar degree. Interferometric Synthetic Aperture Radars (InSAR) has shown promising results for monitoring subtle infrastructure defects over large-scale networks; This can serve as an early warning system for potential structural failures ahead of time before physical manifestation or visible damages [7]. Integration of Global Information System (GIS) infrastructure with InSAR data enables automated detection of potential structural issues across vast regional transportation networks, marking the start of a new era and next level monitoring efficiency [7]. The contributions of ground based technical innovations caused notable surveying process advancements. For localized measurements terrestrial laser scanners have offered a new degree of precision. Whereas automatic monitoring systems like PRIME (PROactive Infrastructure Monitoring and Evaluation) offer continuous, volumetric earthwork assessment conditions by utilizing geometrical measurements [8]. Incorporating complex data processing algorithms, quality assurance systems and analysis functions give practical information to end users with little to no human intervention. For biodiversity surveys Environmental DNA (eDNA) methods have brought about a revolutionary change. The main detection process of species is via water sampling opposed to direct/visual detection [9][6]. There were clear indications that eDNA can detect certain species of frogs that are 20 km (approx) downstream from their population center, attesting to the exceptional sensitivity of this method when seen in contrast to other methods for surveying wildlife [6]. This capability is exceptionally valuable for observing rare species in hard to reach locations.

Comparison between traditional and modern surveying approaches naturally point out the superior performance advantage. In elaborate comparisons of the UAV-SfM method versus

the total station (tested in heterogeneous build up environments) the UAV based method is not only less error prone but also grossly outperformed the latter method [2]. Temporal efficiency was the most noticeable, completing a week's worth of work in a couple of hours [2]. This time cut directly correlates to a faster turnaround time and cost savings for the project. The immense spatial resolution achievable with modern methods is another advantage over traditional techniques. Whereas the total stations are limited to manually measured discrete points, the UAV-SfM method creates comprehensive 3D point clouds that contain millions of data points [2]. This fundamental difference in data density allows for the creation of ultra detailed terrain maps and characterization along with in depth assessment of infrastructure. Only due to this high resolution comprehensive data, surveyors can detect subtle features that might be of significance. Which used to remain undetected. For ecological applications the detection capability also reflects this performance discrepancy. Drone based thermal imaging for a certain species of deer results in accuracy figures nearing 80% compared to established population values (later flights boosting this to over 90%) [5], far exceeding the typical sighting probabilities using traditional thermal imaging surveys conducted from manned aircrafts. This really drives home the effectiveness of drone based surveying for wildlife monitoring [5].

Comparing channel morphology with DLS systems with total system measurements have shown another really interesting correlation between metrics that have been extracted from both of the methods, irrespective of the dramatically higher resolution of the former method [1]. This result points out the comprehensiveness of modern technology with superior accuracy for crucial measurements, ultimately this is a better way of obtaining high density data without making any significant sacrifices.

eDNA sampling methods for biodiversity assessment showed significant promise compared to conventional aquatic surveying techniques, enabling detection of rare species of sparsely populated species over a great distance [6]. This exceptional performance characteristic made it more than just a viable alternative to more manual or hands on approaches. In the application of infrastructure monitoring these advanced surveying techniques have transformed safety/security monitoring by ensuring that any concerns are being addressed in time. Recently extreme weather events caused by global warming have caused the natural aging process of infrastructure to accelerate ultimately requiring more management, observation and preventative measures but to support these efforts the funds allocated are often not enough for small to medium industry let alone large scale and extensive systems [7]. Modern surveying technologies aim to fill this gap by offering an efficient, comprehensive monitoring process that can detect early signs of damage and help prevent catastrophic failure while needing less funding. InSAR tech has demonstrated its capability for large (regional) scale infrastructure monitoring. Surface defects can be detected on various materials (asset types) with exceptional accuracy and resolution [7]. Integration of InSAR data and the GIS infrastructure (with automated post processing tailored for a specific application) can enable the almost real-time assessment of large scale transport routes and shows possible events where human intervention could be needed [7]. This “early warning system” marks a new direction for infrastructure management, preventing failures by performing predictive assessments at critical sites by observing faint warning signs. Composite materials capable of self-sensing brings another

innovative approach to the table for infrastructure monitoring. The incorporation of carbon based fillers into construction materials creates a structure which is capable of monitoring the levels of strain and stress it is subjected to [10]. Smart construction materials try to mitigate the placement and maintenance limitations of conventional sensors. As an added bonus it also improves the durability and reliability of the monitoring systems by the merit of utilizing the bulk fabrication technique [10]. By integrating technology into the structure itself it turns into a distributed network of sensors which can provide continuous health data.

The PRIME infrastructure monitoring system marks the beginning of the move towards a fully automated monitoring solution. Implementation of standardized geo-electrical ground monitoring systems can achieve remote real-time capabilities [8]. End users can receive timely information on regular intervals due to the automated processing and data flow system. It also incorporated filtering, quality assurance and all other bells and whistles one can hope for [8]. This level of automation greatly reduces manpower requirements for effective infrastructure assessment while also increasing the frequency of each assessment so that structures are not left in an unknown condition. After restoration projects the channel morphology study can illustrate the practical advantages of modern surveying approaches. The total station method measures a broad range of critical metrics but it comes at a cost of requiring a lot of hands-on work. Despite the great deal of hands-on work it cannot achieve the data density of modern study techniques like DLS. It enables more comprehensive, high resolution assessment of the outcome post restoration and monitors the long term stability [1]. Applications in environmental assessment have been transformed by a great degree by the implementation of modern technologies. Traditional biodiversity studies typically rely on direct observation, capture and other methods that are labor intensive. It also often yields incomplete results as opposed to the eDNA method, especially for rare species with very low population densities by detecting trace amounts of bio-active molecules and markers [9]. The capabilities of eDNA have been proven again by the detection of very rare amphibian species over a long distance (downstream of a waterway) and in remote areas, where the traditional methods are exceptionally challenging [6]. The fact that it can detect populations of less than 100 individuals from almost 23 kilometers away is astonishing on its own [6]. This new level of sensing capability brought about a new revolution in terms of finding and characterizing critically endangered or species that have been labeled as “missing”. Green infrastructure in urban settings have also benefited significantly in remote sensing processes by utilizing visible and IR spectrum (Thermal Imaging) cameras mounted on mobile platforms [11]. This method enabled automated assessment of plant growth and soil moisture status but with a single plant level resolution [11]. When CV (Computer Vision) algorithms are applied to this data it can generate effective leaf area and water stress indices of trees. Which allows mapping of spatial and temporal patterns that correlate weather trends and urban management practices [11]. Drones equipped with thermal imaging hardware have demonstrated their superior performance compared to traditional survey methods [5]. As mentioned before, there were a few studies that have shown a high degree of accuracy for certain species of deer during evening flights, far exceeding the limits of conventional techniques [5]. Microbiome surveys that compare traditional cultivation methods with cutting edge sequencing techniques have shown a dramatic enhancement of detection capacity with the help of next gen technologies

[12]. Researchers have recovered a significant number of taxonomic units from microbes using pyrosequencing methods which were not detected using older processes. This provided a new degree of insight into ecological relations and the diverse world of microbes [12]. Again the credit goes to the expanded detection capability which helped to gain a comprehensive understanding of complex environmental systems and their response when subjected to change.

Efficient utilization of financial and human resources is the main driver for the implementation of modern surveying techniques. Adoption of these technologies across a wide range of fields will allow for more comprehensive, frequent and accurate understanding of the environment and infrastructure due to the high resolution, data density and frequency of surveys as they will not require as much resource utilization. These will ultimately reduce costs, prevent damages and shrink project timelines [3][2]. Modern alternatives offer a significant advantage in the reduction of financial expenditure and resource utilization, in the meantime improving the quality of obtained data and its comprehensiveness. The most dramatic advantage in the case of modern surveying approaches is time savings, the UAV-SfM method has proved its capability to conduct comprehensive surveys in a fraction of the time that a human team would take [2]. By achieving field efficiencies orders of magnitudes better than previous approaches directly translate to lower costs, reduced project timelines and increases in operational capacity for surveying organizations unlike anything that has ever come before. The reduced field time also lowers the exposure time to harmful conditions that cause bodily harm and damage equipment, both are very difficult to replenish. The need for capital investment increases the ongoing maintenance expenses as traditional surveying requires the use of specialized and expensive tools [3]. The fact that many modern surveying approaches combine consumer grade equipment with advanced processing algorithms to achieve a professional grade results at significantly lower costs. This can be observed by the fact that UAV based approaches use consumer grade cameras mounted on commercial drones and offload the detection demand to sophisticated post processing algorithms from onboard equipment ultimately reducing the cost [2].

Manpower requirements pose another challenge which can be resolved by utilizing modern surveying methods as they do not require trained personnel working in the field for extended periods of time [2]. Modern technologies often allow a trained person to rapidly collect massive amounts of data, which is fed through complex post processing algorithms by automated computational systems [7][8]. This is another way how modern methods save costs and improve data quality while not taking as much time or as many qualified professionals for a given task. Additionally the data density is orders of magnitude better than other methods, this helps to gain a comprehensive understanding of the environment or structure to be monitored/surveyed. Since modern methods combine advanced sensing, faster collection methods, complex computation and post processing algorithms, they achieve a greater result in a fraction of the time. Whereas previous methods, the data quality and density is directly proportional to time spent on the field [1][2]. DLS and UAV-SfM can achieve millions point per unit area over continuous surfaces which required so much time before that no organization took on the task, given it was possible in the first place [1][2]. Surveying technology continues to evolve at an ever-accelerating pace. Several new cutting edge approaches promise to

transform this field in more innovative ways. The integration of AI and ML shows a promising path, where automated feature extraction, abnormality detection and predictive analytics can be performed on massive datasets generated by all the methods mentioned before [7][8]. These computationally intensive approaches effectively scale the value of collected data by extracting details that were unnoticed by humans or traditional algorithms. VR and AR technology offers another degree of modernization by enabling the visualization and inter-activeness of complex spatial data generated through modern means of assessment [13]. Immersive visualization of data could potentially transform how surveying data is analyzed and evaluated at a fundamental level by making essential information easily perceivable so that humans can take action accordingly [13]. The integration mixed reality and accelerated high resolution data collection components could enable real time data visualization with a context first mindset. Deep learning powered auto-focusing methods for microscopy and whole slide imaging prove the appropriateness of this technology to address broader surveying challenges [14]. Through the use of such computational methods the image capturing can be isolated from focusing actions, which enable rapid scanning while continuously keeping razor sharp focus [14]. Similar methods could potentially address focusing challenges in aerial and terrestrial surveying tasks, enhancing the data quality without any efficiency losses. Embedded monitoring methods and self-sensing materials represent a new horizon for infrastructure surveying applications [10]. Continued development of smart construction materials which incorporate sensing capabilities directly into the structure itself, making the beginning of fundamental shift in the direction as this distributed network of sensing elements can enable continuous measurements without requiring retrofitted sensor installations [10]. Eliminating many of the sensor related challenges used in traditional systems.

Integration of multiple sensor modes unifies surveying platforms, this is another trend with significant positive potential. The combination of thermal, visible, multi-spectral and hyper-spectral imaging capabilities along with laser scanning and measurement techniques allows simultaneous primary and complementary datasets that serve the purpose of a more comprehensive environmental and infrastructure characterization compared to any single mode technique [11][5]. This multi-modal approach maximizes the collection of information and boosts efficiency all the while providing a more comprehensive and richer contextual understanding of the surveyed system.

1.1.3. Literature Gap

Based on a comprehensive review of our project details and present academic literature, the following critical research gaps are identified for an autonomous aerial surveying platform built on standard hardware, off the shelf (COTS) components and fully FOSS toolchains [79][80][81][82]. While demonstrating that consumer grade drones can achieve acceptance of the surveying accuracy, significant gaps remain in the cost optimized system design [80][83][84][85]. Comparing COTS component existing performance against specialized surveying hardware. Which is technically nonexistent for developing country contexts [79][86][84]. Supply chain resilience and component availability documentation for limited

resources remain unaddressed [87]. Cost performance tradeoff models for surveying applications in a perfect budget lack empirical validation [81][80]. Long term reliability studies of COTS components under operational surveying conditions are scarce, with most research focusing for little time [88][89]. Individual FOSS tools like OpenDroneMap, ArduPilot and PX4 are well documented but end to end integration challenges remain incomplete [90][91][92][80]. Comprehensive integration frameworks look after the flight control (ArduPilot/PX4), mission planning, data acquisition and photogrammetric processing. (OpenDroneMap/WebODM) are missing from literature [92][93][94][90]. Standardized data exchange protocols between heterogeneous FOSS components lack formal specifications [93][94]. Version compatibility matrices and long term maintenance strategies for FOSS ecosystems are not documented [95][96]. Unified configuration management is the system that handles multiple interdependent FOSS tools that remain unexplored [94][97]. Troubleshooting methodologies for multi component FOSS failure are largely absent [98][99]. Autonomous drone research mainly assumes access to high performance edge analysis by computing, creating a substantial gap for resource constrained systems [100][101][102][103]. Adaptive autonomy frameworks that scale functionally dynamically. Which happens based on computational resources, are not researched [102][104][100]. Energy aware navigation algorithms specifically optimized for survey for COTS hardware lack comprehensive investigation [105][106][107]. Fail safe mechanisms for consumer grade autonomous systems are critical. Surveying scenarios remains underexplored [108][109][88]. Hybrid online/offline learning approaches which help to enable autonomous improvement while respecting power constraints are insufficiently developed [110][104]. Real time path planning on embedded systems having limited processing power (e.g., Raspberry Pi or Consumer flight controllers). Which should be researched more deeply [111][108][100]. There are several unsolved issues with the project's goal of creating unique CV models for data collection. Domain specific training datasets in aerial surveying for diverse geographical and climatic contexts (especially developing countries) are critically lacking [112][113][114][115]. TinyML and model compression techniques tailored specifically for aerial survey objects on resource constrained hardware which are not investigated [116][117][118][111]. Transfer learning protocol is from high end sensor to low end sensor which disturbed the standardization [119][120]. Benchmark datasets tell us that the low cost sensor configuration (cameras and basic IMUs) are less exposed [83][113][112]. Real time inference optimization in the looking of edge devices (Nvidia Jetson Nano or embedded GPUs) for filling up surveying specific tasks needs more decoding [117][121][116]. In the case of some studies showing the consumer drones which show accuracy standards, critical methodological gaps persist [84][122][85][123][80]. Calibration methodologies designed for COTS multiple sensor fusion (consumer GNSS, IMU and cameras) lack holistic development [122][124][123][84]. Uncertainty quantification frameworks are used for low cost sensor chains with error propagation where modeling is not sufficient [125][123][84]. GNSS denied operation strategies for autonomous surveying inside challenging environments remain underexplored [126][108][122]. Field deployable calibration procedures suitable for non expert operators are missing [127][128][84]. Temporal accuracy degradation models for COTS components over extended field use are not documented [89][128]. The project's multi role design objective faces architectural gaps. Standardized modular interfaces for low cost surveying payloads that enable rapid reconfiguration are absent

[129][130][91][93]. Software abstraction layers that offer independence for various surveying tasks need development [97][96][95]. Rapid reconfiguration protocols that reduce the need for recalibration are not well studied [131][132][133]. Payload aware flight planning algorithms that automatically adapt to various sensor configurations are not thoroughly studied [132][133][105]. Design patterns are not documented for scalable multi role architectures in situations with limited resources [93][94][88]. The focus of our project on OPSEC offline operation draws attention to an important research gap. Autonomous mission execution Insufficient research has been done on surveying applications without external communication infrastructure [134][135][108]. Secure offline data logging lacks standardized procedures with tamper evident features [136][135][137]. Offline mission validation and pre deployment verification techniques are inadequate [138][139]. Air gapped operation frameworks especially for surveying missions are yet mostly speculative [136][134]. There is little literature on low cost drone OPSEC in delicate situations or threat models (like Bangladesh) [140][135][136]. Despite the maturity of tools like OpenDroneMap and WebODM, the usability and quality assurance gap remains [91][141][80][92]. Automated quality assurance pipelines There is a lack of development in FOSS photogrammetry that verifies output accuracy [142][80][125]. User centered design research for inexperienced operators in poor nations is deficient [143][92]. Processing efficiency studies on typical laptops and other consumer grade computing gear are inadequate [144][141][145]. Standardized accuracy reporting frameworks for surveying items resulting from FOSS must be established [123][80][125]. Reproducibility protocols and FOSS post processing workflows lack defined version control techniques [146][92]. Most research validates in a controlled manner and creates gaps for operational context [147][108][138]. Tropical/subtropical validation studies there are few survey devices which are affordable and important for Bangladesh [148][149][89][87]. Maintenance protocols lack documentation for COTS systems in hot, humid conditions[150][128][89]. Environmental degradation effects on inexpensive sensors (temperature, dust and humidity) require methodical research [148][89][150]. Long term operational reliability data are not accessible for long term survey initiatives [151][88][89]. Comprehensive failure mode analysis There are no surveying applications that are unique to COTS drone systems [109][147][138][136]. Our project's need for operability by personnel with limited training reveals some gaps in available documentation. Training frameworks are underdeveloped for non specialist operators in environments with low resources [152][143]. Context appropriate documentation particularly in poor nation research on approach is lacking [153][154][143]. Sustainable capacity building models are not researched to perfectly guaranteed long term knowledge [155][87][152]. Standardized competency assessments for cost drones development development is necessary [156][143]. Institutional adoption barrier studies are not familiar for governmental and academics [157][158][87]. The research is muchly dependent on focusing on developed nations which creates a gap for our deployment context [159][156][87]. Regulatory frameworks Not sufficient to study on low cost FOSS based drones in South Asian environment (Bangladesh, Nepal) [154][87][159]. Community centered ethics not sufficient research on data sovereignty procedure to surveying application [87][159][152]. Certification pathways unexplored are FOSS based systems that are different from proprietary platforms [160][156][87]. Privacy preserving protocols applications for community based surveys must be developed [135][140][87]. Insurance and liability frameworks for low cost surveying operations in

developing countries are absent [160][152]. Battery limitations significantly constrain surveying missions, yet research gaps persist. Energy aware surveying algorithms that maximize mission planning for COTS batteries limitation have not received enough attention [106][102][105]. Field deployable renewable charging There is no validation of solutions for remote surveying operations [150][152]. Battery degradation modeling and its long term effects on mission dependability require investigation [161][88][150]. Standardized energy budgeting frameworks there are no planned surveying missions [107][106]. Thermal management strategies are unexplored for tropical conditions (important to Bangladesh) [118][89][150].

1.1.4. Relevance to Current and Future Industry

1.1.4.1. Impact of the Project

Due to the fact that, none of the measures notably alters the distinguishing features of the final product, the effect cannot be precisely attributed to the opted pattern, particularly since the principal distinguisher is performance, thus it has no discernible impact on important outcome metrics. By revolutionizing field data collection, processing, and interpretation, high resolution UAV sensing offers major advantages. It uses GPS, onboard processing and AI/CV enabled real time detection. These technologies support applications like surveillance, infrastructure inspection and wildlife monitoring. The system also generates faster actionable insights for better decision-making. Deploying an Autonomous Aerial Survey Platform (AASP) would provide significant technological, economic, environmental, and societal benefits. It requires fewer personnel, equipment and time. This can economically lower survey costs. While also facilitating multi sector use, including agriculture, planning, infrastructure and disaster response. As a result, it enhances accessibility and increases the return on investment for public agencies and SMEs. Lightweight electric (and possibly solar-assisted) UAV operations can reduce emissions and ecological disturbance during monitoring when used in place of vehicle-based or piloted surveys. Socially, AASP can improve service delivery in hard-to-reach regions (notably in Bangladesh) by supporting emergency logistics and continuity services such as delivering medical supplies and educational materials, while also creating new training and job opportunities in piloting, maintenance, and data analysis.

1.1.4.2. Sustainability of the Project

In the context of their deployment in Bangladesh, social, economic, as well as environmental aspects need to be addressed. Drones would help in cutting down on emissions by replacing the use of transportation in conducting fieldwork, but their production, as well as the processing, has a large emission cost since the power infrastructure in the country largely uses fossil fuels. Rather, solar-powered models with greater efficiency in flights, energy, or algorithms in this area would be highly significant in the solar-rich environment of the country, but sustainability would be impacted by the processing or disposing of drone components, which largely lack green alternatives in the country, or the lack of e-waste recycling in the country, which would

pose risks to hazard wildlife in the country. The presence of high startup costs can act as a barrier to adoption, although partnerships involving cost-effective platforms, subsidies, local production, and technology transfer can reduce dependence on importation and expand adoption. However, costs are susceptible to shedding in terms of loading, charging points, data processing needs (where edge computation can provide benefits), and maintenance related to flood conditions in terms of heat, moisture, dust, floods, and corrosion, which require modularity, durability, and strong parts sourcing channels. Notwithstanding, there are hopeful applications in agricultural productivity enhancement, riverine industry monitoring, flood monitoring, and disaster relief activities. In healthcare supply chain management, disaster response related to floods and cyclones, and even educational functions during disruptive events, drones can be used. Though drone use requires public engagement strategies on issues such as privacy concerns, noise pollution, and cultural considerations, job displacement can be mitigated by job creation in drone operation, maintenance analysis, and software development if Bangladesh provides adequate support infrastructure.

1.2. Objectives, Requirements, Specification and Constraints

1.2.1. Objectives

To overcome the challenges related to the issue of having a complete and effective system must be implemented. It should meet all functional and non-functional requirements. It is key that the solution must be obtained while keeping a “quality first” mindset. Durability and ease of use must be ensured. The final design should be upgradable and future proof as there will be financial and time constraints while designing the system. The following objectives are selected to help navigate the obstacles during development.

1.2.1.1. Data Collection for Custom CV Models

Comprehensive data gathering from relevant sites for creating computer vision models represents a fundamental step in the system development process. Machine learning and deep learning have shown its exceptional capability across many fields, but this performance depends directly on the quality of training data. Data collection for CV models face numerous critical issues. According to a 2023 review on data augmentation procedures, the performance improvements of machine learning and deep learning techniques can be directly attributed to the sheer amount of training data. Although it is not always possible to get the required data required to create a dataset that can be as versatile and robust as needed [15]. This challenge is quite apparent in specialized fields. For example livestock farming, the findings from a 2024 survey states that data collection, organization, selection and preservation requires a lot of time, tedious and requires a lot of resources [16]. As it is directly related with model performance, the importance of this criteria is clear. The trained models often generalize real world scenarios very poorly if the dataset is of low quality or lacks sufficient depth. Again for specialized domains, the source data is often seasonal or constrained for environmental reasons. If the agrifood industry is taken as an example, dataset generation is heavily constrained to specific periods. Atmospheric conditions and adverse weather prevent sufficient (of similar quantity and comprehensiveness) data collection in all the necessary classes, causing some

classes to have more data than others resulting in an imbalanced dataset [15]. Researchers face similar issues in the field of medical imaging, as training deep learning models require a vast quantity of labeled data. This is not only time consuming and very costly, especially for medical images [17]. Highlighting the fact that being careful with data collection strategies is paramount for building reliable computer vision systems.

1.2.1.2. Autonomous Control Architecture for Hands-Free Operation

The autonomous control mechanism allows the system to operate independently as a whole with very little human intervention. This is an essential criteria for the design of modern systems. Recent developments in this area have primarily focused on creating and implementing control mechanisms that can adapt to complex environments. The very powerful DRL (Deep Reinforcement Learning) framework is slowly becoming a compelling choice for autonomous control. A recent study showed how DRL has become a versatile learning framework for autonomous driving. It allows self driving systems (known as agents) to learn driving techniques. To be more specific, agents learn behavior patterns and decision making in complex situations like intersections. From taking in environmental data these agents can show great improvements over time [18]. This capability is very important for systems that must navigate changing environments, keeping safety the top priority while being as efficient as possible. A parallel can be drawn from these driving agents and the autonomous drone control architecture. The drone might not be able to take real time data and change its algorithm to the same degree as a self-driving car due to obvious computation constraints (limited by power and communication restrictions) but it will be able to make changes in future trips by taking in data before the drone itself launches. The implementation of autonomous control mechanisms are thoroughly demonstrated in multi-UAV systems. A 2024 review highlights that systems that continuously learn and adapt to their environment (known as online learning) by changing its behavior as it receives new data in real time performs better in dynamic environments, as opposed to their offline learning counterparts (where all the training takes place before deployment). These performance gains can be seen with better collision avoidance, better swarm behavior and adaptive path planning [19]. These capabilities allow systems to adapt to changing conditions without constant human supervision. UACAMS (Universal Autonomous Control and Management System), used in maritime applications demonstrates the different levels of autonomous control implementations. Starting from remote control and all the way to fully autonomous missions, this system configuration enables autonomy in all four levels. Depending on user specific requirements each the implementation and deployment of all four levels can be customized [20]. This level of flexibility is essential for adapting to different user needs and operating conditions. Autonomous control is crucial as it reduces the cognitive load on human supervisors improving efficiency and allowing safe access to dangerous, remote and inaccessible locations without endangering lives.

1.2.1.3. Accurate On-Board Data Acquisition

The primary target of a precision DAQ (Data Acquisition) platform is to gather accurate and reliable data during operation. This is the basis of subsequent decision making processes and analysis via post processing. DAQ operations are important in many fields. Most commonly seen in the biomedical fields as the demand and widespread use of wearable health tech is on

the rise. According to researchers, one of the biggest challenges with ECG smart textiles is the reliability and quality of collected data by the device itself [21]. This is just one example among many, highlighting the importance of developing robust acquisition systems that have a built in quality control system to maintain data integrity. Regardless of environmental factors and physical constraints. Due to the need for high precision and real time data in environmental monitoring systems, the integration of IoT technology has turned into the primary objective [22]. Such capabilities enable the creation of an effective monitoring system. The merger of sensor technology, wireless communication and data processing allow for comprehensive environmental assessment with next to no human intervention. The importance of accurate data acquisition cannot be underestimated as the influence of ingesting data in every consecutive step following the initiation of a process. In the case of environmental monitoring, to make informed decisions based on real time information, precise data collection systems such as acquisition, transmission and finally analysis [22]. Shoddy or inaccurate data acquisition often causes erroneous analysis reports, incorrect conclusions. Ultimately leading to potentially harmful decisions. Even the most sophisticated post processing algorithms are not immune to low quality data.

1.2.1.4. High Quality Imaging for Accurate Record Keeping

Capturing high fidelity images ensures the records have detailed and clear visual data. This is very important for accurate documentation, analysis and decision making in computer vision (even in manual vision based) powered systems. The challenges to achieve high fidelity images differ from case to case. For agricultural applications, researchers who monitor plant health note that unclear edges and blurred motion boundaries are notable challenges. This is especially apparent when analyzing complex motion patterns, a very common subject when studying natural leaf movements [23]. The need for advanced techniques for detailed preservation of data, which in turn is key for accurate representation of physical phenomenon. As another general example, 3D human pose estimation requires precise visual data. Here also researchers emphasize that ambiguous depth perception and the lack of robust datasets are main challenges [24]. The lack of high quality images (in general not having high fidelity imaging capability) makes the already difficult task of extracting text from images (billboards, street signs, store fronts, etc.) harder as the same text looks very different from one scene another, due to differences in lighting, orientation, viewing angle, color, fonts and much more [25]. This capability is also critical as it creates the foundation for accurate visual analysis. As previously discussed, in the field of precision agriculture the detection of early drought stress is possible [23]. While for security and surveillance application the anomaly detection in the majority of recorded video datasets [26]. Even the most advanced analysis and post processing algorithms depend heavily on the raw quality of captured data to produce reliable results, as these algorithms cannot properly retrieve information from poorly obtained data.

1.2.1.5. Multi Role Design

This design methodology refers to any system that is capable of performing multiple functions. It also adapts to different operating conditions, meeting various user needs while not requiring specialized equipment for each different task. The need for multi role implementation is important in many domains. Researchers described that dual UAV assisted edge computing

systems (where one vehicle acts as an MEC server to offload client side computation tasks while the other one can act as either a jammer or a relay, depending on user needs) leverage one another to enhance the communication channel security and also reduce system latency [27]. This flexibility allows one system to carry out multiple technical requirements. Platforms like the HydroDron are prime examples of multi role design, especially for maritime applications. It is designed to carry out hydrographic survey missions with multiple configurations of the payload (multi beam echo sounder, LiDAR, Sonar, Radar, multi spectral photographing systems to name a few) [20]. This efficient resource utilization across different mission profiles can be directly attributed to the versatility. The need for a multi role design is mainly to maximize the systems utility and overall cost effectiveness. With the ever growing system complexities the cost of development is also increasing at an alarming rate. This can be somewhat mitigated by making the system somewhat able to perform multiple functions and carry out different tasks while adapting to changing environments. This ability is becoming increasingly valuable as the capability of having operational flexibility is something specialized systems cannot do, while also having the capacity to change as per requirements and react according to unexpected situations.

1.2.1.6. Comprehensive and Tailored Post Processing Capability

This essentially is the ability to transform raw data into a format upon which either an automated system or a human operator can make swift and accurate decisions. It is primarily realized by specialized algorithms and analysis techniques specifically designed for the task at hand. These post processing techniques vary from one application to another. In optical spectroscopy the data augmentation processes are mainly targeted to add slightly modified versions of data to existing datasets to expand it. This ultimately leads to a dataset that contains laboratory test results alongside a vast collection of synthetic data which itself is derived from real world values [15]. This addresses the data limitation issue while not degrading its analytical validity. Self supervised learning offers a promising solution for medical image analysis by allowing systems to train on useful insights from copious unlabeled medical datasets [17]. This eliminates the need for the expensive and time consuming task of data labeling without reducing analytical power. In security applications AI can enhance security measures by allowing detection, feature selection, classification, obfuscation of sensitive data, model combination, output interpretation, adaptation, optimization, data generation and much more [28]. Wide ranging capabilities like this showcase the expanse and critical need for post processing in today's systems. The need for comprehensive post processing lies in its ability to get the maximum value from any given dataset. It is next to impossible to extract valuable information and generate context for complex problems from raw data. Thus the main purpose of post processing algorithms is to interpret raw data and add context for a given application domain. Tailor made post processing guarantees that the system is capable of producing outputs that are relevant, accurate and can be taken action upon. Finally this is a very efficient method for transforming data into knowledge for end users, ultimately they can use it to take action and decisions for specific tasks.

1.2.1.7. Environmental Data Collection

Collecting data from the environment involves gathering information about the surrounding conditions that concern system performance, user experience or any specific phenomenon under surveillance. Environmental monitoring systems have significantly improved throughout the years, especially in modern times through the integration with IoT technologies. A 2025 review of such technologies revealed that they facilitate pinpoint and efficient data collection and analysis. This holds true across various environmental domains (air, water and soil quality) [22]. These applications demonstrate the utility and broad scope for implementation in numerous applications. Environmental data collection in agricultural settings enables the assessment of critical parameters such as watering conditions to predict whether a plant is well watered or suffering from water stress [23]. This application highlights how gathering environmental data is crucial to catch early indications of critical conditions before they cause serious and often irreversible damage. To provide adequate context for system operation and decision making, the collection of environmental data is crucial. Having a broad understanding of the deployment condition of the system allows the scope of change, reflecting and countering shifts in environmental variables. Identification of potential values, transmission and analysis is enabled by monitoring the environment, forming efficient and comprehensive pipelines for assessment and response [22].

1.2.1.8. Offline Operation Mode to Maintain Proper OPSEC

To maintain operational security (OPSEC) it is paramount to enable an offline operation mode. Most attack vectors require some form of network connectivity, eliminating this attack surface reduces remote attack vulnerability, ensuring continued operation. This feature is also useful if the system is suddenly disconnected from the network, allowing almost full functionality. This prevents the system being idle and not able to be used while it is being brought back online. It is also an added benefit as it allows the system to be operational in remote locations where network connectivity is often very bad or outright it is unreachable. The importance of offline operation is emphasized by research on air gapped systems. These are completely isolated from the network, like remote islands on an ocean with high walls guarding their inner garden. Originally they were thought to be the pinnacle of security and people touted them to be impenetrable [29]. Although recent studies have shown that these air gapped systems are not as invincible as they have been once thought to be. Unconventional methods have been deployed to defeat the physical security of these networks. Information and unauthorized communication channels can be established by using optical, EM, plain magnetic, plain electric, acoustic and even thermal means [29]. Highlighting the need for the deployment of comprehensive security measures, even for offline systems. In the face of ever evolving digital threats, the implementation of artificial intelligence within security mechanisms has become either an obvious decision or something everyone is trying to implement. Recent research shows how ensuring the security of complex computational and embedded systems requires the utmost care and attention of security analysts and professionals. The vulnerability of critical infrastructure is ever more prominent due to the current digital domain of espionage, warfare and terrorism, this is exacerbated by increasing amounts of more sophisticated and complex vulnerabilities, zero-days and back doors from design flaws [28]. Again pointing out the need

for robust security systems, offline operation capabilities are all modern systems. Coming back, the offline operation mode is crucial for OPSEC because it drastically reduces the attack surface to potential adversaries. The elimination of network based attack vectors can help systems to maintain their operational integrity even in hostile environments. This capability is particularly important in defense applications, critical infrastructure (power grids, nuclear and conventional power plants, petrochemical infrastructure like refineries and transport pipelines) and sensitive industrial settings (weapons manufacturing, either conventional or nuclear) just to name a few. As data security and continued operation is the topmost priority even in wartime.

1.2.2. Functional and Non-Functional Requirements

1.2.2.1. Functional Requirements

1. **Real Time Monitoring:** Continuous tracking and analysis of operational parameters
2. **On-Board Data Processing:** Captures environmental data, conducts preliminary analysis before offloading to base station
3. **Post Processing Capability:** Base station shall be able to provide advanced processing of recorded data for in depth analysis
4. **Autonomous Roaming Within Predefined Bounds:** The drone can execute a round trip within a predefined path without human interaction

1.2.2.2. Non-Functional Requirements

1. **Scalability:** Can be swarmed if necessary given there is a central control node
2. **Reliability:** Customizable suite of failsafes which can be tailored per application basis
3. **Competitive Pricing:** Strategic pricing compared to similar solutions available in the market
4. **Regulatory Compliance:** Features can be modified to align with regulatory requirements and industry standards
5. **Data Retention:** Base station can store processed and raw data for archival/research purposes

1.2.3. Specifications

1.2.3.1. General Specifications

1. **On Board Compute Platform:** Capable of performing onboard data processing and general computing tasks. Must have expandable storage and wireless connectivity.
2. **Camera:** Should be able to capture high resolution full color images and record video at reasonable frame rates (min. 30fps).

- 3. Sub-Routine Handler:** Can manage peripheral routines and should support standard bus protocols like I²C and SPI. Must have HID emulation capability.
- 4. GPS Tracking Module:** Should be able to connect to multiple navigation satellites besides the default GPS, including GLONASS, Gallelio and BeiDou. An integrated compass is also crucial.
- 5. Drone Frame:** Lightweight and strong ideally made from carbon fiber or glass fiber reinforced plastics. Standardized mounting screws and motor support is required.
- 6. Motor:** Must use high efficiency brushless motor technology. Should have adequate cooling so that it can provide enough thrust without overheating during the entire flight duration.
- 7. Electronic Speed Controller:** Can precisely control the motors and have high continuous and burst currents. Must be operated via PWM.
- 8. Propeller:** Optimized for efficient thrust generation. It should also be balanced so that it produces minimal vibration.
- 9. Flight Controller:** Must have IMU, Gyroscope and Barometric sensors onboard. Should support multiple radio interfaces, soft remote kill switch and hardware arming circuit for safe operation.
- 10. Battery:** Must have high discharge rates, a near flat discharge voltage curve and high current anti spark connectors.

1.2.3.2. System Level Specification

- 1. Motor Count:** 4
- 2. System Voltage:** 14.8V
- 3. Battery Capacity:** 5.6Ah
- 4. ESC Count:** 4
- 5. Redundancy:** Dual GPS, Dual Compass
- 6. Imaging module power:** 27W
- 7. Base station:** 24V
- 8. Flight time:** 30 min
- 9. Liftoff capacity:** 2.5 kg
- 10. Payload capacity:** 450g

1.2.4. Technical and Non-Technical Consideration and Constraint in Design Process

1.2.4.1. Technical Requirements

While working on this project we encountered a lot of challenges requiring designing the whole system and seamless integration. The following subsection highlights the most demanding of our collective attention.

1. **Weight:** The device must be of low weight so that it can fly efficiently. Carbon-Fiber based frames are considered the best option.
2. **Storage:** Most standard configurations of the image processing uses micro SD cards (Class 10, UHS-I, V30) but that poses a serious issue. As standard NAND based memory cells have a finite lifespan and that is worse. Write endurance of NAND flash is limited (unlike MOS Capacitor based DRAM) but as micro SD cards are limited by aggressive efficiency and power draw limits that severely hampers it. Additionally the controllers on the micro SD cards work over the SD bus (newer SD express cards use the PCIe NVMe bus) which is severely limited in throughput. Thus a PCIe Gen 3 NVMe SSD is quite ideal for this use case.
3. **Imaging:** Dual spectral imaging is available as needed. The RGB camera is the primary camera and is corrected for spherical aberration as it is the default camera for manual precise landing/aruco based measurement.
4. **Flight Time:** The standard flight time is severely diminished by the fact that batteries cannot be sourced easily and at a cost effective price point.

1.2.4.2. Non-Technical Requirements

The following is a list of the key non technical requirements that was the goal of this project.

1. **Ease of Use:** As this drone requires very little manpower to operate, the ease of use so that it can also be deployed by anyone with minimal training is another plus point.
2. **Repairable Nature:** Almost all modern tech is built to maximize profit at the expense of the end user and the environment. Anti right to repair and ultra miniaturized devices mainly exacerbate this. As a requirement we opted to use non specialized parts (off the shelf) so that it can be repaired at a very affordable cost.
3. **Cost:** As a crucial driver of any project it needed to be controlled. The repairable design and the use of off the shelf “jelly-bean” parts attribute to this. This model can also easily leverage economies of scale.

1.2.4.3. Constraints

1. **Battery Life:** As hinted in our previous section finding batteries in the country without specialized supply chains are quite difficult. Especially as we had to deal with sellers who are quite far down in the supply chain as manufacturer quotes often exceeded 30% or more when they come to us so for the realization of this prototype we opted to use low capacity battery to prove the concept.

2. **Adoption Challenge:** Resistance to technological change might pose as a barrier for large-scale adoption.
3. **Security Risk:** Susceptibility of wireless communication channels and interception risks could allow a bad actor to gain unauthorized access via physical means or indirect privilege escalation. While no sensitive data is stored on-board but the probability of this happening may discourage deployment.
4. **Offline Operation:** The whole system must be air-gapped and does not require internet access to function to maintain operational security but integration can be cumbersome for remote applications.

1.2.5. Applicable Compliance, Standards and Codes

A crucial impact is that the adherence to a list of applicable codes also creates a sort of “safety-net” for the project in almost all deployments. Both local (the country of origin or R&D) and international code adherence is required as per our design rules. This also extends for overseas deployment that will require additional testing to ensure compatibility with that jurisdiction.

Local Codes (Bangladesh):

TABLE I Non Exhaustive List of Codes Regarding Bangladesh

Regulation / Code	Clause / Section	Description	Applicability	Reference
CAAB Order No. 30/2020	Section 5.1	Operators must submit a drone flight application at least 45 days in advance.	Applies to all drones (except toy drones).	CAAB
CAAB Order No. 30/2020	Section 5.2	Drone operations prohibited within 10 NM of airports without authorization.	Mandatory near airports.	CAAB
CAAB Order No. 30/2020	Section 5.3	Permission required for private property takeoff/landing.	Applies to private property operations.	CAAB
CAAB Drone Registration Guidelines	Section 2.1	All drones must be registered with CAAB.	Mandatory for all operators.	CAAB
CAAB Drone Operation Manual	Section 4.5	Operators must maintain a logbook of drone flights.	Commercial drone operations.	CAAB
CAAB Drone Policy	Section 6.1	Drone pilots must be 18+ years old.	Applies to all operators.	CAAB
CAAB Drone Policy	Section 6.2	Liability insurance is mandatory.	Applies to all operators.	CAAB
CAAB Drone Policy	Section 6.3	Flights must be during daylight and in Visual Meteorological Conditions.	All operations.	CAAB
CAAB Drone Policy	Section 6.4	Max drone height is 200 feet AGL.	Applies to all operations.	CAAB
CAAB Drone Policy	Section 6.5	Authorization required within 10 NM of the aerodrome.	Mandatory near aerodromes.	CAAB

International Codes (Concise):

TABLE II Concise List of International Codes

Regulation / Code	Clause / Section	Description	Applicability	Reference
ICAO Annex 2	Rule 3.1.2	Must comply with airspace restrictions and obtain clearance.	All international drone operations.	ICAO
ICAO Annex 2	Rule 3.3.1.2	Flight plans required for controlled airspace.	Mandatory in controlled airspace.	ICAO
ICAO Circular 328	Section 3.2	UAS integration into non-segregated airspace.	UAS operations globally.	ICAO Circular 328
ICAO UAS Toolkit	Various Sections	Global best practices for drone regulation.	For regulators and operators.	ICAO UAS Toolkit
EASA Regulation (EU) 2019/947	Article 11	Operators must submit risk assessments.	EU drone operations.	EASA
EASA Regulation (EU) 2019/947	Article 14	Operator registration required.	Mandatory in EU.	EASA
EASA Regulation (EU) 2019/947	Article 15	Drone geo-zones must be respected.	Applies throughout EU.	EASA
EASA Regulation (EU) 2019/947	Article 16	Authorization required for high-risk operations.	For special category drone missions.	EASA
EASA Regulation (EU) 2019/947	Article 17	Drone pilot training and competency required.	All EU drone pilots.	EASA
EASA Regulation (EU) 2019/947	Article 18	Outlines operator responsibilities including safety and maintenance.	EU drone operators.	EASA

1.3. Systematic Overview/Summary of the Proposed Project

Our project aims to deliver a low cost aerial survey platform by leveraging off the shelf hardware. Additionally free/open source toolchains to automate infrastructure surveying, reducing time, cost and required manpower while improving data density and safety. The initiative addresses the high costs and manpower needs of traditional surveys. It is done via automating data collection and analysis for infrastructure monitoring and environmental assessment. This enables frequent, detailed surveys with minimal financial investment. However conventional surveying is labor intensive, slow and expensive. This issue is especially evident in developing regions, leading to gaps in situational awareness and safety risks. The project seeks to lower barriers to timely surveys. Our proposed drone based platform is built from off the shelf components and FOSS (free/open source) toolchains. So to capture, process and deliver survey grade data with reduced dependence on specialized equipment and personnel is easily attainable. Data and capability goals emphasize higher data density and quality. Also improved safety and better accessibility in challenging environments (e.g., forested areas). This is higher compared to traditional methods. All of this is managed while maintaining cost effectiveness. Impact and use cases mainly target infrastructure monitoring, environmental assessment and other survey oriented applications. It enables scalable deployments in resource constrained settings. Educational and capacity building angle highlights institutional and training barriers in traditional surveying and positions. This

platform is a practical, hands on bridge for students and practitioners to gain real world experience.

1.4. Conclusion

This chapter established the core motivation for developing an aerial survey platform by showing conventional surveying. This may be historically reliable. It often becomes impractical when time manpower, safety constraints and financing limitations are considered. The scaling effect with the size and complexity of the area or installation being assessed also adds to this issue. It highlighted how delayed or even omitted surveys can create critical gaps in situational awareness. This lacking increases the likelihood that developing faults remain undetected until they cause severe damage or loss.

Chapter 2: Project Design Approach [CO5, CO6]

2.1. Introduction

Autonomous survey drones are manufactured to carry out the data collection process with little human involvement. Intervention based onboard processing to derive actionable information out of real-time or even near real-time imagery. The designs that are taken into consideration here onboard server-side, an image buffer and local CSV logging processing, or real-time onboard detection. With post flight offload to ground based digest, considering 2 complementary strategies, such as, a self-contained time synchronized system detecting detections with accurate GPS/RTC, timestamps, and directed-offload workflow that exploits processing power and cooling by delegating heavy computation to a ground station after the drone has docked and recharged.

2.2. Identify Multiple Design Approach

It is essential for a low cost, open source autonomous aerial survey drone to evaluate multiple design approaches. The architecture only becomes understandable and best when we compare real performance trade offs. This includes detection latency, durability, power draw, payload calculation expenses, connection dependability and long term maintainability. Instead of investing early in a single pipeline the prototype of two complementary designs can enable objective comparison which will enable under identical missions. This can also provide evidence based selection for deployment. Onboard processing with coordinated times this design puts sensing and computing together on the drone, eliminating the need to rely on a strong connection and allows responding instantly to high risk situations. With an RTC augmented by GPS time, there is also high certainty of timing the tick rate, and the geo tagging of each detection in a CSV with frame timestamps. The limited box detection requirements and the flight path controller generate a deterministic pipeline in which the detector produces logging and possible action without necessarily awaiting post flight transfer. The primary trade offs include increased power consumption, increased system on a chip resource requirements, increased cost but reductions in latency and strong performance in communication hostile environments are gained. Onboard real time detection and offloading of the plane after flight this design aims to balance the processing headroom and thermal margins between the data capture and heavy analytics. The drone performs preliminary detection on the local camera stream and transfers the post processing to the docking station (which is compute intensive) after dropping the captured data through a secure channel. The use of a NB IoT/LPWAN hotspot to the transfer station enables the establishment of an opportunistic connection near the base in order to run more powerful hardware for supporting more advanced recognition algorithms. The resulting working process produces a richer work. Ground station analytics, whether sustainability and service life can be improved with modular upgrades and OSEC has the advantage of wiping buffers after transfer. Nonetheless, it creates the latency between detection and final analysis and relies on stable after flight connection to the digest generation.

2.3. Describe Multiple Design Approach

2.3.1. Design Approach 1

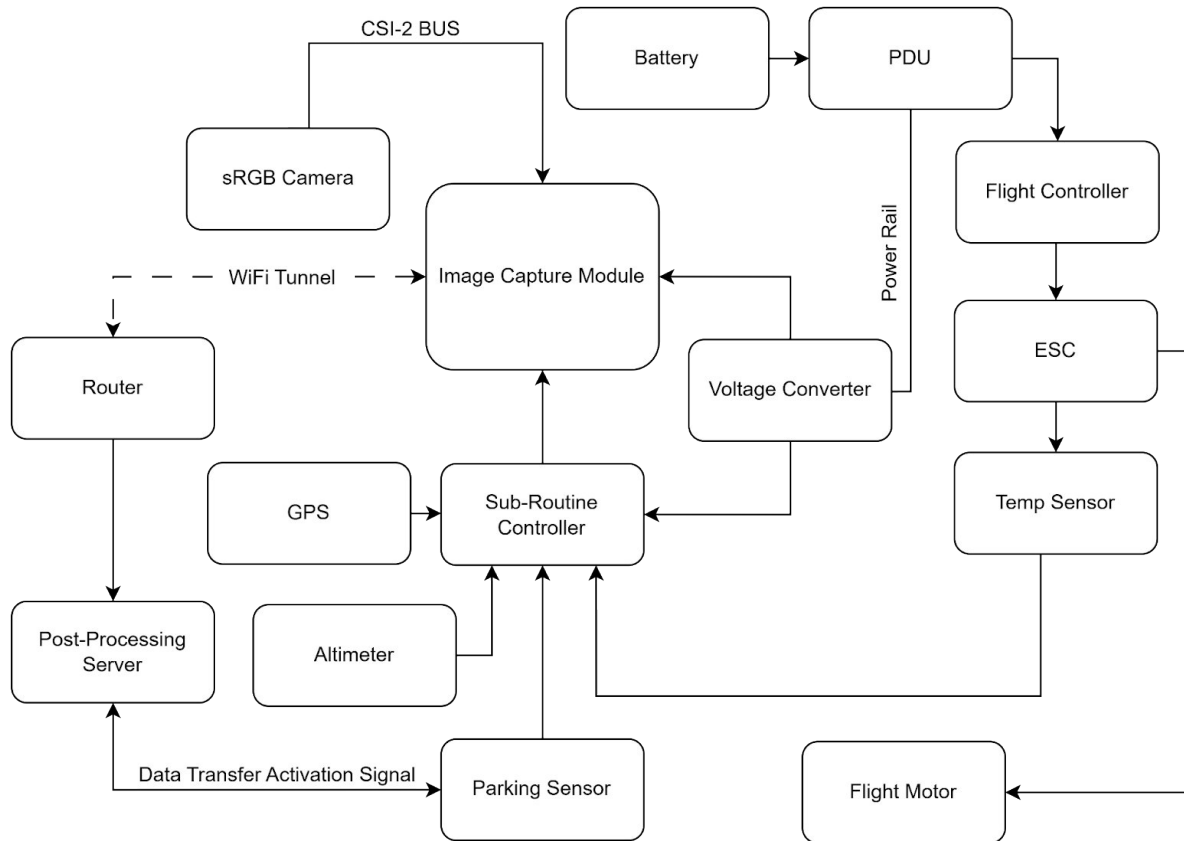


Figure 1: Systematic Overview of Design Approach 1

This design mainly utilizes Server Side Processing and Detection of sRGB/NearIR Image Buffer. The drone shall be completely self contained such that it can capture image data and process it on board, offloading the digest file on the docking station. A predefined flight path will exist on the controller which will trigger three subsystems. A RTC, a high precision GPS and the recording system. The RTC will keep time augmented by high accuracy GPS time and will provide high precision tick rates for on board processing (GPS time provides low precision high accuracy syncing). General detection algorithm consists of timestamps (ticks will be synchronized between subsystems) to append GPS coordinates to a CSV file once a predefined object is detected within the bounded box (target area) of the frame buffer. Each detection event (over a certain confidence percentage) shall generate an entry in the aforementioned CSV file. The few preliminary advantages include real time tracking of immediate/high-risk events that demand swift action and among the obvious downsides are the requirement for more power hungry components and complex SOC's which drive up the deployment cost.

2.3.2. Design Approach 2

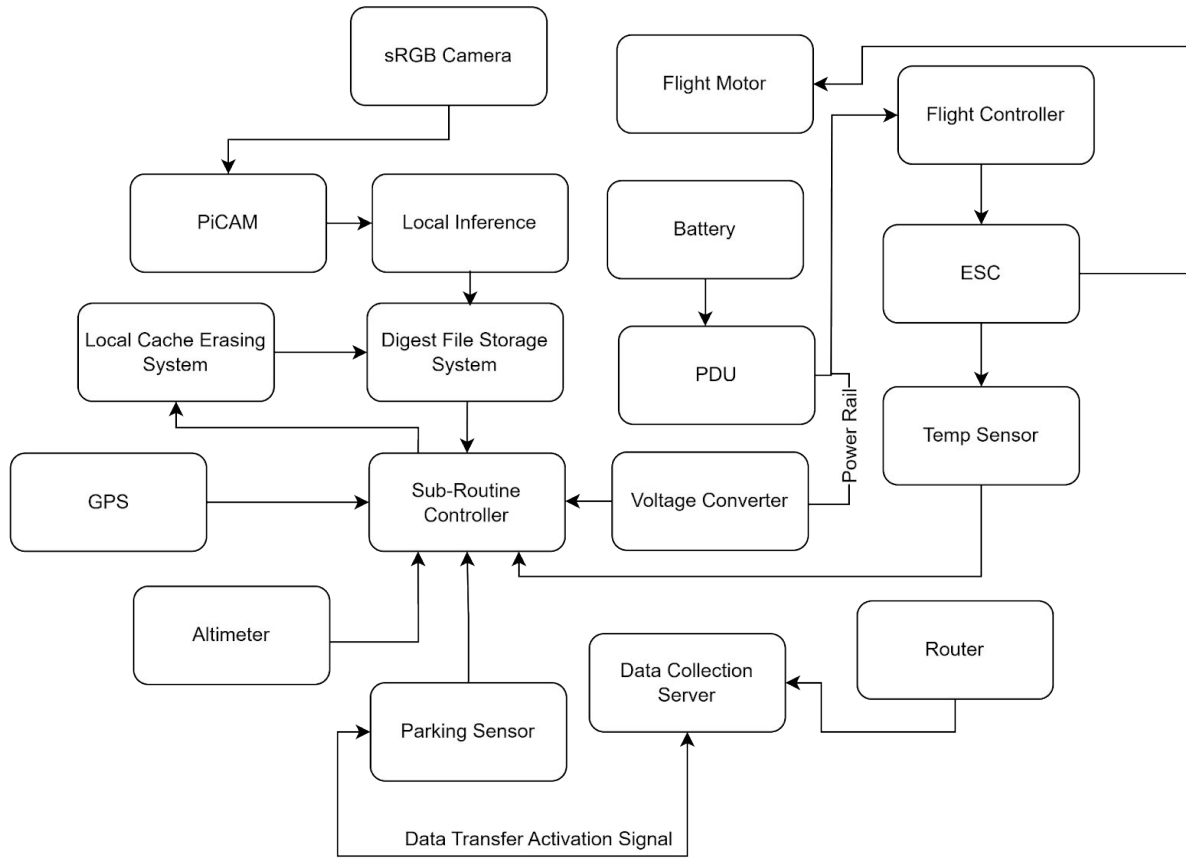


Figure 2: Systematic Overview of Design Approach 2

Alternatively this design utilizes Real-Time On-Board Image Detection from Video Stream/Camera Feed. A main difference between the self-contained system and a server based one is that the drone transfers the footage for the post-processor engine and to generate the digest file. A NB-IoT/LPWAN hotspot will be available close to the base-station (landing platform). Sensors will detect the presence of the drone which will initiate the charging and offloading process. Given a cursory glance, some obvious benefits are noticeable, one of them is the scope of using advanced image recognition and post-processing algorithms can be utilized as there will not be a power constraint. In broader terms better cooling, no space and power constraints allows the system developers to opt for more powerful hardware. The prospect of retrofitting in the future for extending the service-life of the product which also caters to improve sustainability metrics. The main detection algorithm will be identical to on-board processing with a few changes. Once the drone has parked on the landing platform and has established a proper wireless connection the files generated on the imaging module will be transferred via secure tunnel. A CSV file containing timestamps (with corresponding video time), Latitude, Longitude and Altitude in ISO standard formats. Upon the completion of the transfer process the image buffer on the drone will be wiped to maintain OPSEC. The video file and corresponding CSV file will be processed in the aforementioned manner.

2.4. Analysis of Multiple Design Approach

TABLE III Direct Comparison of Both Design Approaches

Differentiating Factors	Design Approach 1	Design Approach 2
Processing Location	Embedded SoC/NPU on on-board AI accelerator (Optional)	Cloud/Remote or local infrastructure
Power Consumption	Low (Limited Performance)	Medium to high (Scalable Performance)
Hardware Complexity	Requires light-weight and ruggedized components that are power efficient also	Contains easily swappable/serviceable and upgradable components for diverse deployment scenarios
Latency	Faster response but slower/imprecise detection due to limited hardware	Slower response but precise/fast detection due to the availability of high power hardware
Algorithm Complexity	Light weight models only	Advanced CNN's or transformers
Data Security	Unencrypted on local storage (wiped upon digest file transfer)	Encrypted on device/transferred via secure channels (wiped for additional security)
Upgradability	Costly hardware upgrades and often cannot be transferred between sites without major renovations (complex software DevOps cycle due to on-device inference mechanisms)	Cheaper hardware costs and streamlined DevOps cycle. Easily deployed as the device utilizes x86 based compute platforms
Cost Factors	Higher cost of deployment and replacement/repair	Comparatively lower deployment/repair cost with additional cloud fees (if remote server infrastructure is used)
Failure Recovery	Limited onboard redundancy and archival is available to digest file only (post processing)	Industry standard server-side redundancy and backups available. Archival is available for both raw and processed data (depending on user preferences)
Typical Use Cases	Emergency response, security monitoring	Surveying, mapping and agricultural analysis

2.5. Conclusion

It is very difficult to draw any conclusions from this comparison even though this is really direct/one-to-one. Thus a plethora of tests have been planned to evaluate each design with a standardized set of test scenarios. Based on the results of those tests we can achieve a detailed view and deep understanding of each design's strengths and weaknesses. This will be our main guideline in choosing the best solution that satisfies the requirements and goals of our problem statement. Tests eliminate subjective biases and give a solid baseline to the ground truths thus it is the only way to make well informed decisions that are based in fact. There is no alternative to extensive testing to find objective truths and to draw definitive conclusions from any direct comparison.

Chapter 3: Use of Modern Engineering and IT Tool [CO9]

3.1. Introduction

A low cost autonomous survey drone can be obtained when the development stack values open ecosystems, rapid iterations, and cross platform compatibility. Python is the main simulation, data development language of the project. processing, and perception, which are machine learned, assisted by mature libraries such as OpenCV, NumPy and newer frameworks such as TensorFlow, Keras and Ultralytics to real time detection and segmentation. A windows based work station is still viable to everyday engineering on, and WSL2 and Hyper V offer flexible Linux and multi VM, GPU accelerated training environments, reproducible experiments and rollback of experiments are available during risky changes. Bash, Unix toolchain (grep, awk, sed, find, xargs), allows lightweight, auditable telemetry parsing automation and pipeline scripting. PowerShell provides strong orchestration among windows services, devices and so on. repeatable build workflows. Lastly, self hosted mapping through TileServer GL empowers independence and saves on repetition with offline capable geospatial mission planning and visualization without third party tile users. Combined, these tools allow building a feature rich, but cost conscious workflow that aids the. complete lifecycle simulation and model training through onboard inference, survey output. generation, and safe mapping delivery.

3.2. Select Appropriate Engineering and IT Tools



Figure 3: List of Selected IT Tools

3.3. Use of Modern Engineering and IT Tools

3.3.1. Python

Python has been the go to choice for researchers working in the realm of drone simulation and required machine learning for surveying operations. As it remains to be a mandatory instrument for the development. It has also become an inevitable favorite among researchers and engineers alike. This is mainly due to its simple and arguably less clunky syntax compared to other giants in the field like C/C++. Another factor to the widespread adoption is the ease of use and availability of massive swaths of libraries. Due to the existence of extensive libraries most functionality is baked into the language itself. Whereas other languages required engineers, developers and researchers to write code from scratch. The transparent structure and sheer clarity engrained in the language itself also aids in the rapid progression of the project at hand. This is partly due to the ease of use of this language and it's hard to catch nature as it is a function of the syntactic simplicity. Other languages (equal and often far exceeding the capabilities and performance standards of python) mainly due to their messy, jargon-ridden and convoluted syntax cause developers to get caught up in complex and often very complicated issues that require a lot of hands on work from the debugging team (who themselves face a lot of issues as it is very hard to debug a massive codebase) to untangle the code and patch the bug. Developers turn to python to find robust solutions for their machine learning needs. Titans like OpenCV, Pytorch, Tensorflow. Raw low level libraries like Pandas and Numpy (also SymPy) and data visualization packages like matplotlib (old school) and Seaborn make it quite obvious why researchers and analysts make it easy to understand why developers, engineers and researchers alike use python as their go to choice for anything machine learning or deep learning related (even exploratory data analysis and visualization). The astonishing feat of computational intelligence achievable by powerful tools like TensorFlow and Pytorch and later object detection models like Ultralytics is the main driving force behind the use of this tool for multiple projects like ours with scopes and budgets far exceeding any conceivable limits.

3.3.2. WSL

The most efficient way to run Linux on a Windows host is to use it's Windows Subsystem for Linux (as a system package, WSL2 is the latest version as of this writing). It utilizes Intel VT-x of AMD SVM to virtualize a part of the linux kernel (through a hypervisor that simultaneously handles both the Linux and the Windows NT kernel) for use. A large limitation of this is the lack of the systemd (openrc/runit in some cases) daemons. The direct and efficient access to numerous hardware features including USB interfaces and the GPU (crucial for our case as the project requires a whole suite of training, data processing, performance via inference and vast amounts of post processing to make the raw data presentable requires a lot of parallelization, not that it will not run on the CPU rather parallelization offers a crucial speed increase on the order of magnitudes which allows for rapid prototyping and testing). With this system hardware prototyping and simulation is seamless without ever leaving the Windows desktop (which we rely on for other software suites as they are not available most of the time). The command line is the most appealing and often considered to be a strong suite of the linux/unix operating systems. Powerful command line utilities make it the ideal choice for

development of custom machine learning algorithms or even modify existing pre-trained neural networks. The libre architecture of linux allows for end users to pick and choose the exact piece of software that they need. The optimization and integration of Linux and Windows filesystems have been pushed to new heights. CUDA is a key technology when it comes to AI and ML tasks as traditional CPUs have a handful of cores designed to perform a complex range of tasks whereas with NVIDIA's CUDA architecture is optimized to run a single simple task multiple thousands of times per second. This high parallelization is the main reason why GPUs are needed for machine and deep learning tasks and the most feasible way to access this hardware from a windows system is windows subsystem for linux.

3.3.3. TensorFlow

TensorFlow is the best drone based machine learning tool. It is because of many things. It supports many kinds of machine learning models. Deep neural networks are one of them. These are very important for visual recognition tasks that drones perform during automated surveying. The TensorFlow software runs on a variety of hardware, ranging from high-performance servers to small devices such as Raspberry Pi. It is popular among researchers because it allows flexibility to rapidly prototype and deploy production-ready designs. It can be optimized for efficient on-device inference or scaled up for end-to-end server-side processing. This flexibility enables teams to pick just the right balance between speed and granularity. TensorFlow includes a lot of ready-made algorithms and packages. These include very comprehensive support for image classification, object detection and segmentation. Built-in CNNs are one of the essential elements of drone image analysis. Transfer learning capabilities: Users can utilize pre-trained models and fine-tune them for specific drone survey tasks. This time saving will result in faster and more accurate deployment. TensorFlow has a huge community. Because it is open source. Developers can benefit from thousands of open-source projects. Regular updates and an enhanced ecosystem/integration is another positive aspect. Documentation is also thorough. Tutorials and Guides are Easily accessible. This simplifies the learning curve and empowers the team to implement solutions efficiently. TensorFlow has integrations for visualizations and data management. For both onboard rapid detection and detailed cloud-based analysis, it is easily deployed in most installations. It has shown its potential in many drone and image processing applications. Raspberry Pi camera and standard parts compatibility are cost-effective. This helps to justify the use of low cost, scalable and easily maintainable drones. With TensorFlow, devices can be surveyed frequently to optimize safety and detect faults before they become a problem. Due to its comprehensive features, scalability, compatibility, and supportive ecosystem, TensorFlow is the most suitable tool for this specialized application in drone automation and image detection.

3.3.4. Keras

Researchers apply Keras for drone based surveys due to its simplicity in learning and usage. The framework provides a simple and clean Application Programming Interface. Different model architectures can be quickly created, tested and modified. As a clear result accelerated research and development processes can be observed. Keras supports a variety of deep learning backends. One of the biggest names is TensorFlow, which facilitates easy deployment to various hardware platforms. Pre-trained models provide users with a head start of already

proven algorithms. Even with limited training data in hand, it does improve accuracy. Keras has a large community behind it. Problems are usually resolved in a short period of time and resources are readily available. Keras models can be run with optimizations on edge devices like Raspberry Pi. This is important for onboard processing in which computational resources are constrained. Keras has good support for image classification and object detection tasks. It is also well suited for drone missions using cameras. Keras models are portable and can be easily converted to be served. This allows for the efficient integration between on-board rapid detection and remote in-depth analysis. Researchers value its scalability, flexibility, and reliability in doing science. Keras also reduces the time and cost of intelligent surveying solutions. Because of the simplicity of its implementation, its open source character and its ecosystem, it is the most used choice in academic and industrial projects for automated sensing.

3.3.5. PowerShell

PowerShell is key to developing automated workflows. It allows us to develop scripts quickly. It has combinations with many technologies. PowerShell allows us to easily manage devices and services. It can be used for cloud-based automation. It can easily and efficiently process large data. PowerShell has powerful parallel execution. It is able to handle long running tasks with reliability. It's a workflow state-checkpointing tool. It continues execution upon interruptions. PowerShell is used for remote control of devices. It minimizes the amount of manual effort. PowerShell adds consistency to operations. It has high error handling capabilities. It allows secure automation using script signing. PowerShell is available free on all Windows computers. Its easy deployability is of great importance for researchers. It does not require any license or special hardware. Easy to maintain and keep updated PowerShell Teams can recycle scripts for other uses. PowerShell comes with a huge set of cmdlets. It automates the process of collecting, transforming, and reporting data. It is compatible with image processing libraries. PowerShell is easily integrated with machine learning workflows. It is used to access the APIs and databases. PowerShell is designed to be modular for flexible design. Teams can expand their workflows to a large number of devices PowerShell is time-saving and cost-saving for development. It allows survey automation to be more productive. It has access control and auditing capabilities. PowerShell is good for engineers with prior knowledge. It is an activity with detailed control over each task. Its output can be reported and analyzed. PowerShell is a Microsoft-supported technology. Training is made easier because of its common usage in IT. Researchers prefer it because of the consistent reliability in critical applications.

3.3.6. Bash

Researchers find Bash to be the best tool for drone simulation workflows because it is so simple, flexible, and efficient. Bash scripting is a way to automate repetitive and complex tasks through short scripts that are run easily on nearly any Linux platform. This light weight is important for simulations that need to be run on commodity machines such as Raspberry Pis, where the resources are constrained. Bash is a great way to manage processes, file operations, and a sequence of commands, which means we don't need to load up on software dependencies to gain control of the simulation environment. It's easily integrated with other simulation tools and can be used to automatically launch, monitor, and cleanup in a fast development cycle. Its

portability means that the same scripts can be used across different deployment environments, which not only allows for reproducibility but also makes it easier for researchers to share their work with others. With Bash, research teams can easily change workflows, experiment with new situations, and schedule automated jobs - all with less development time and operational overhead. Our work in drone simulation projects demonstrates that being able to directly interact with hardware components and integrate with containerization technologies further reinforces the appropriateness of Bash for drone simulation projects that need onboard and server-side processing. Overall, bash scripting is a cost-effective, reliable, and user-friendly tool to automate the entire simulation process and allow teams to concentrate on developing their machine learning and image detection algorithms instead of the environment setup aspects.

3.3.7. Unix Toolchain

Automation researchers preferred the Unix tool chain because it represents the Unix philosophy of small, composable tools that do one thing well and pipe well for scalable pipeline processing. sed is a fast, in-place stream editor that does pattern-based substitutions, line filtering, record/field processing, column extraction and aggregation, if conditionals, and formatted reporting directly from logs, CSVs and telemetry to log triage and anomaly detection pipelines. The tools simply work together, allowing workflows that enumerate targets with find, filter content with grep, summarize and structure fields with awk, normalize lines with used, and apply at scale in a single pass using xargs or -exec. This composability translates to smaller source code, better testability, and faster feedback loops than monolithic scripts and heavy frameworks, which is well suited to the practical needs of reproducible research and operations. Portability: These tools are often shipped with most POSIX systems, have minimal external dependencies, and are predictable in constrained environments such as Raspberry Pi deployments and headless servers. The commands are scriptable and auditable, pipelines are transparent, affordable and easy to peer review for compliance and safety cases in industrial settings, grep isolates error patterns, time windows, signatures directly from streaming logs for proactive alerting, and quick triage, awk enriches with counts, rates, rolling summaries for trend analysis and thresholding, no round trip to DB, bigdata, xargs reliability, chunking, retrying, parallelizing, splitting walltimes in CI, field The ecosystem is well documented and taught so teams can get started quickly, benefiting from decades of proven patterns and optimizations related to text, filesystem and process workflows. The format ranges from rapid one-liners to powerful pipelines that can be versioned and containerized, which is appropriate for iterative cycles of research and product development.

3.3.8. Hyper-V

Hyper-V suits simulation-centric ML labs and drone R&D It ships with Windows Server, Windows 11 Pro and Enterprise. This reduces entry and running cost. It runs Windows and Linux guests reliably. Strong isolation is offered for risky experiments. It allows dense consolidation on commodity hosts. This makes the hardware utilization and power efficiency better. It provides check points, production check points. Fast rollback-iteration in the and model tuning. It supports live migration across hosts. Maintenance occurs with no planned downtime. It has in-built replication features. Business continuity is made easy for labs. It

integrates very well with automation in PowerShell. Provisioning and tearing down is fast and scriptable. It supports nested virtualization on supported CPUs. Complex Multi-VM topologies are emulated on a single box. It is GPU passthrough and GPU partitioning. Accelerators can be shared between training VMs. It aligns with Windows Server 2025 improvements. GPU-P sharing brings flexible fractional GPU use. Live migration of partitioned GPUs reduces outage. Workgroup clustering makes edge deployments easy. It uses known Windows admin tooling. Teams come on quickly and need little retraining. It scales from laptops to datacenter clusters. The way from prototype to production is smooth. It supports secure boot and virtual tpm. Sensitive data sets remain protected in tests. It integrates with back-up and DR ecosystems. Recovery objectives are normally easy to achieve. It works with common CI pipelines. Regression testing becomes a matter of routine and cheap. It handles camera data pipelines at scale. Storage and networking is configurable and efficient. It reduces manual labor via automation. Operating expenses fall over time. It enables frequent survey simulations safely. Early fault detection becomes practical and repeatable. It is well matched to low cost component strategies. Standard Pis and cameras fit the workflow. Hybrid edge to server processing by support. Onboard logic can be mirrored in VMs. Server-side analytics can be stress tested easily. The stack is the same across environments. This reduces integration risk and rework.

3.3.9. Ultralytics

Ultralytics is a unified vision framework. It supports detection, segmentation, classification, pose, and OBB. It is fast for training and inference. It scales from edge to cloud. It has easy to use Python and CLI APIs. It contains powerful docs and examples. It integrates with MLOps tooling. It gives us pre-trained weights and easy fine-tuning. It is memory efficient during training time and inference time. It reduces requirements on the hardware for real-time use. It enables anchor-free decoupled heads, for accuracy. It eliminates the anchor tuning complexity. It improves C2f modules gradient flow. It has a good mAP at low latency. It runs well on Raspberry Pi devices. This supports Pi cameras and libcamera flows. It offers small models for edge speed. It supports exports to the ONNX and NCNN. It allows quantization in ARM targets. It is officially partnered with Raspberry pi. It displays tested pipelines and benchmarks. It has automated tests for Pi deployments. It provides frequent updates and active support. It maintains a large community ecosystem. It offers model variants of various sizes. It balances accuracy and speed well. It fits real-time surveying on-board. It fits comprehensive server side processing. It reduces operational cost on a scale. It speeds up the prototyping and iteration process. It eases deployment across hardware. It is good for longer term use in the field. It is proven in industrial applications. It is well adopted and battle tested. It is a practical choice for drones. It matches up with inexpensive parts. It supports ongoing improvement of the model. It corresponds to the project's dual mode design. It has to be the best fit for the job.

3.3.10. MATLAB

MATLAB provides an end-to-end workflow for drones. It includes modeling, simulation, perception, and deployment. It is able to unify algorithms and physics into one stack. Simulink helps accelerate control and system design. It allows quick iteration with block diagrams. UAV Toolbox attacks drone workflows directly. It provides planners, controllers, and photorealistic scenes. This has connections with PX4 and MAVLink ecosystems. It supports low and high

fidelity scenarios. It reduces risk prior to flight testing. It is used to speed up parameter tuning and trade studies. It scales from quick runs, to intensive tests. It incorporates computer vision toolboxes. It streamlines feature extraction and detection; It has support for deep learning training and inference. It relieves data labeling and augmentation. It makes it a shorter path from model to code. It generates optimized C and C++ automatic It's fast to deploy to ARM and Raspberry Pi. It uses the ARM Compute Library for speed. It supports INT8 quantization for inference. It eliminates hand written glue code. It supports Raspberry Pi camera pipelines. It delivers turnkey acquisition and preprocessing. It allows edge and server split workflows. It validates autonomy with rich 3D environments. This has links to Unreal and Gazebo style setups. It provides hardware in the loop options. It natively analyzes flight logs and telemetry. It reduces development time and defects. Centralizes tools, docs, and examples. It's commonly used in robotics labs. It is ripe and healthy for harvest. It fits budget constrained prototyping well. It matches our dual processing architecture. It supports onboard speed optimized paths. It supports server side comprehensive paths. It leads to reduction in lifecycle cost and manpower requirement. It enhances reliability through frequent surveys It detects faults earlier with the vision and ML. It aligns with Raspberry Pi official cameras It provides the acceleration of mission ready deployment.

3.3.11. TileServer GL

Hosting a tileserver locally can be better than using third-party mapping providers. The core advantages cluster around privacy, security, autonomy, performance, and long-term sustainability. By hosting our map tiles, access logs, and query patterns all in our infrastructure, us can avoid persistent telemetry and profiling risks which are inherent to centralized LBS platforms. It can also eliminate exposure to cross-jurisdictional data requests and exposure to cross-jurisdictional data requests. Hosting our stack enables end-to-end control over security posture, e.g., fine-grained access control, zero-trust network segmentation, per-service authentication, rate-limiting, WAF rules, and custom control interfaces. It also enables stronger cryptographic management of geospatial assets (at rest and in transit), isolated secret management and verifiable supply chain controls for rendering our toolchain. Self-hosting is immune to vendor policy changes, pricing increases, quota limits on use and opaque throttling. It provides us predictable costs, and the ability to tune tile generation (styles, generalization levels, tiling schemes and cache strategies) to our workload. It also includes on-prem GPU or vector tile pipelines that work for low latency, offline or high concurrency scenarios. It also reduces the risk of external attack surface and dependency, which makes it more resilient to outages, censorship or API deprecation. It can also facilitate compliance by keeping sensitive base maps, overlays and derived analytics within governed boundaries with clear data lineage. Finally, local control makes it easier to support bespoke privacy mechanisms (e.g. query blinding, differential privacy on telemetry, or enclave based processing), integrated CI/CD for styles and data updates, and seamless fusion with internal datasets such as floor plans, asset locations, operational layers which us would not trust to external providers to collectively provide stronger privacy guarantees, tighter governance, better performance predictability and strategic independence.

3.4. Conclusion

The chosen tools constitute an effective end to end pipeline to construct an effective autonomous Python centric survey drone. AI development on a tight budget, Python based development accelerates artificial intelligence and perception. TensorFlow/Keras and Ultralytics make deployable computer vision with no reengineering of base models. WSL2 and Hyper V render high performance. Scripting available on commodity Windows platforms, PowerShell, Bash, and the Unix toolchain enables less manual work by automatically repeating tasks and TileServer GL enables offline, predictable and private geospatial services to visualize a survey. This is the primary benefit of this stack. The stack scales well, it can begin with a single laptop and a low cost drone prototype, followed by more powerful training hardware, more elaborate. simulations, and richer mapping outputs by not compelling costly vendor lock in. Maintaining the workflow as modular, scriptable and popular, the project can provide high level features (autonomy, vision based inspection, effective data management, and mapping) and at the same time be keepable and cost effective as demands expand.

Chapter 4: Optimization of Multiple Design and Finding the Optimal Solution [CO5, CO6, CO7]

4.1. Introduction

The most important part of any design project is the testing and validation of the simulations of the finalized designs. Results obtained from the simulations help us to identify which of the designs are preferable and have better characteristics and performance. Methodical testing is critical. It requires well defined verification plans, clear and definitive pass/fail criteria and extensive indexing of metrics that ensure all functional requirements are met both for routine operating conditions and edge cases (the extent of ruling out and considering edge cases must also be in a systematic manner as they can cause scope widening, grossly inflate R&D cost, unit cost and deadline extensions to the point that the project itself might fail altogether). By applying consistent test conditions, parametric sweeps and corner case analyses for multiple designs, engineers can objectively compare the performance, robustness and adherence with specifications. Iterative refinement is mainly fueled by discrepancies and inconsistency during testing and in results. Detailed debugging schemes and tightening constraints can also aid in mitigating hard to resolve issues. Iterative development requires a lot of temporary fixes and patches, these should be validated thoroughly as they can often cause problems later down the road. Once the system enters advanced stages of development or nears completion these issues can have far reaching consequences. The whole system might require extensive disassembly, rebuilding and even redesign if the need arises. Functional simulation combined with random testing, objective analysis and post simulation reviews can help to identify and resolve many issues before they reach a critical stage. In depth records will add traceability for future research and aid in investigation of critical faults.

4.1.1. Methodology

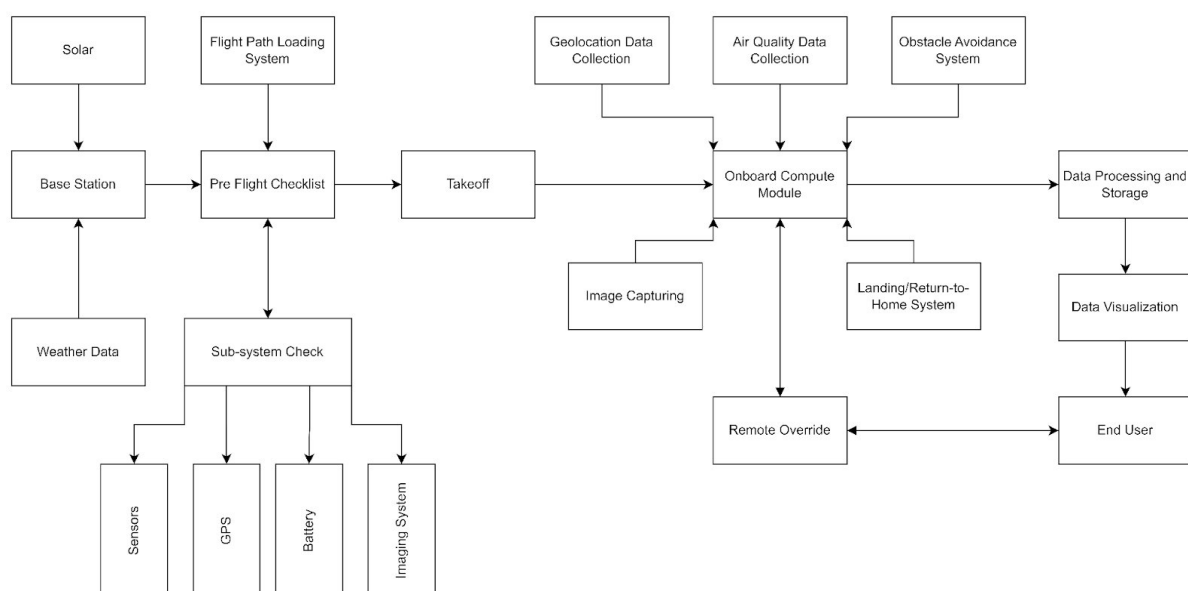


Figure 4: Methodology of the Project

The basic operating principle can be seen in the diagram above. Assuming the system is active and idle with no pending tasks. When an operation is assigned by the designated operator (with proper access control and clearance) a few things start happening. The base station has solar panels and a mini weather station connected to it via the appropriate bus. There will be underlying subsystems that handle pre-flight checks. These start once the system has received a start task flag from the operator's computer. First the drone will check if the weather is optimal and the drone can safely take off (based on wind-speed and precipitation status). A battery scan will soon follow, checking whether the battery is sufficiently charged for the path assigned (based on a pre-defined dataset). If not then the drone will draw power from the solar panel to charge the battery (given it is daytime). After a certain time (dependent on ambient light, efficiency and power delivery capacity of the solar panel and battery DOD) has passed further subsystem checks will start as the system will enter the pre-flight check stage. Subsystems will call up sensors and assess each one by checking its response against a standard profile specific to that sensor. The core system check will include Power, GPS, Imaging Modules along with a few small onboard sensors. A flightpath status check will concurrently happen while everything is being assessed. Once all the pre-flight checks are passed the drone shall take off. Once airborne it will collect various data points. Collection of geolocation data from the main and secondary GPS modules will take place, air quality data collection and the image capturing will occur. Image capturing will persist until the drone has completed the assigned/preloaded path. Then this data will be processed and displayed to the user. Upon completion of the flight route the drone will trigger the landing subsystem and return to its home destination. Another important feature is the constant connectivity to a remote kill switch, that can disarm the drone at any point due to any immediate reason that the operator deems important. This is an additional safety net of this system that can be used in emergency situations.

4.1.1.1. Data Acquisition and Pre-Processing Pipeline

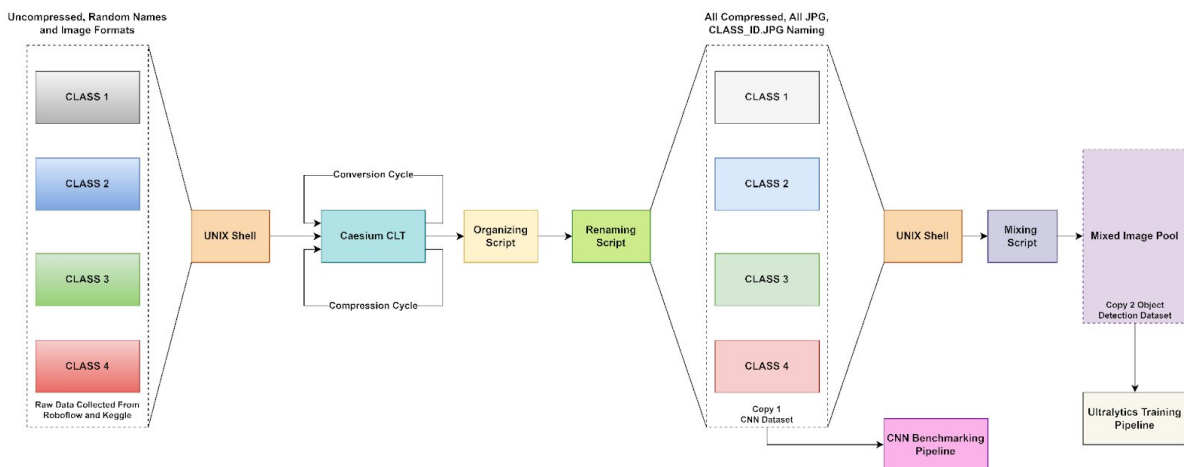


Figure 5: Data Acquisition and Pre-Processing Pipeline

The raw dataset was collected from different sources (mainly Kaggle and RoboFlow). There were multiple issues with the collected datasets. As our machine learning model needs to detect four types of faults thus there must be four classes to the dataset but the images from each class

were individually annotated and had random names. Making matters worse the images were in random formats for some reason. These would have caused different issues down the line and most definitely have delayed the development process of the system. The dataset had just shy of 13,000 images spread unevenly across four datasets. We decided to manually annotate and create the dataset for object detection training. The manual method was chosen as this is a surefire way to mitigate this issue. Trying different methods would have caused issues for us and may have caused us to redo the entire preprocessing step during training or testing. As the dataset is quite large and we had modest hardware to test on. Thus to verify each change would have taken so much time that we could not complete our project in time. We have theorized a method to automate this task for future use. Basically if each class is separately annotated every label file would have the same class (as it is the only class in that dataset). Thus if a script could go through each image, find its corresponding label file and change the first digit to whatever label we desired (as the first number is the class id according to the YOLO annotation format). After that an organizing script or function would rename and move each image and its label file to a different folder. The use of shell tools is preferred rather than python as it is better suited for large scale operations that require extensive use of system resources and shell is the best option without installing any additional abstraction layer thus increasing dependencies and latency (all modern linux kernels have these tools built in so there is no need to install python to perform reasonably complex operations). In order to proceed we made a few scripts and used a few UNIX tools (except the one third party FOSS image processing utility called CaesiumCLT, FFmpeg is another viable option for tasks like this). The initial raw dataset was ingested using the UNIX shell and passed twice through the image processing tool. Once all of the images were converted to PNG (as this resolved some errors when we converted directly from the source format to JPEG) and in the second cycle we converted all PNG files to JPEG (yes, this is a lossy image format and is mainly another conversion cycle, this pass was called a compression cycle as the main goal of this pass to convert the files to a lossy format, JPEG at 70% of its original quality). After this two different scripts were used. One organized the files, another renamed them to ensure consistency across all classes (if we had used the theorized preprocessing pipeline the labels would have been also renamed, paired with an image corresponding to its name to avoid losing images and to prevent corruption). This resulted in a dataset that had images in four directories one for each class (pot holes, metal corrosion and images of high voltage ceramic insulators with damaged sections). This dataset is optimized for classification (for reasons explained in later sections). Now we need another dataset that is suited for object detection. Thus a copy of this sorted dataset is re-ingested into system memory using the UNIX shell fed into a mixing script that dumps all of the images into a folder which can be used for annotation (we would like to further improve this method to exclude the need for manual annotation and to combine smaller annotated datasets into a massive ones required to develop robust models) and ultimately to train an object detection model.

4.1.1.2. Onboard Computational Power Simulation

In order to simulate a Raspberry Pi on traditional x86_64 hardware we need to take a few of the limitations into consideration. A raspberry pi is based on the ARM architecture which itself is a proprietary version of RISC as ARM stands for Advanced RISC Machine. Its performance

and power consumption is unlike the AMD x86_64 architecture (an extension of the 32-bit Intel x86 architecture made to handle 64-bit instructions and memory spaces) rather focussing on raw performance at the cost of power efficiency. Utilizing the Reduced Instruction Set Computing architecture ARM offers higher efficiency when it comes to performance (this is the reason why in embedded and mobile devices manufacturers are using ARM chips as they do not consume as much power nor produce as much heat as their more powerful x86_64 cousins, thus they can be powered off of batteries and can be passively cooled while having impressive performance benchmarks). A Raspberry Pi 4B (8GB version) has a ARM Cortex-A7 CPU (quad core @ 1.8Ghz), memory bandwidth of its LPDDR4-3200MT/s is 25.6GB/s. Where the CPU on our test server is the AMD Ryzen 5 2400G (Zen architecture, codename Raven Bridge, quad core @ 3.6GHz base) has a max supported memory bandwidth of 46.9GB/s (using its max supported 2933MT/s) although we are getting 17GB/s (single channel bandwidth but the CPU can support two) as the server is equipped with a one module of 16GB 2133MT/s DDR4 UDIMM (Un-buffered Dual In-line Memory Module).

```
sudo mkdir /sys/fs/cgroup/rpi-sim-{1,2}
for i in {1..2}; do
    echo "50000 100000" | sudo tee /sys/fs/cgroup/rpi-sim-{$i}/cpu.max
done
```

In our case we used 50% CPU throttling using Cgroups (control groups) to achieve a clock speed of 1.8GHz per machine. As our system is memory limited (even with memory ballooning drivers) our system had to automatically switch between two VM's after a certain period of time. With some memory overhead left for the host.

```
for i in {1..4}; do
    virt-install \
        --name rpi-sim-{$i} \
        --ram 8192 \
        --vcpus 1,maxvcpus=1 \
        --cpu host-passthrough \
        --disk size=20 \
        --network bridge=virbr0 \
        --os-type linux \
        --memory-backing hugepages=yes \
        --memorytune hard_limit=8388608,soft_limit=8388608
    virsh vcpupin rpi-sim-{$i} 0 ${i-1}
    virsh schedinfo rpi-sim-{$i} --set vcpu_quota=50000
    virsh schedinfo rpi-sim-{$i} --set vcpu_period=100000
done
```

After creating four virtual machines with 8GB memory limits, two of the four VMs can be started among the four. After some time these two are suspended and the other two are turned on.

```
virsh start rpi-sim-1
virsh start rpi-sim-2
```

After a certain interval of time.


```
virsh start rpi-sim-3
virsh start rpi-sim-4
virsh suspend rpi-sim-3
virsh suspend rpi-sim-4
```

To automate this process we needed to create two files, an automation script that starts and stops each VM after 30 minutes and a systemd service file so that this script runs upon boot (the network dependency of this does not mean that it requires the internet rather this means after the network daemon is up and running the service will run this ensures that any user can ssh into the machine while it is running thus preventing a lockout condition).

```
#!/bin/bash
while true; do
    echo "$(date): Running VMs 1-2, Pausing VMs 3-4"
    virsh resume rpi-sim-1 2>/dev/null || virsh start rpi-sim-1
    virsh resume rpi-sim-2 2>/dev/null || virsh start rpi-sim-2
    virsh suspend rpi-sim-3
    virsh suspend rpi-sim-4
    sleep 1800
    echo "$(date): Running VMs 3-4, Pausing VMs 1-2"
    virsh resume rpi-sim-3
    virsh resume rpi-sim-4
    virsh suspend rpi-sim-1
    virsh suspend rpi-sim-2
    sleep 1800
done
```

The service file is as follows.

```
[Unit]
Description=Raspberry Pi VM Rotation Service
After=network-online.target libvirtd.service
Wants=network-online.target

[Service]
Type=simple
ExecStart=/usr/local/bin/vm-rotation-script.sh
Restart=always

[Install]
WantedBy=multi-user.target
```

The Unix shell needs the following prompt in order to enable and start this service.

```
sudo systemctl daemon-reload && sudo systemctl enable --now vm-rotation.service
```

CPU Scaling:

$$\text{VM CPU Frequency} = F_H \times \frac{F_P}{F_H} = F_P$$

$$\text{CPU Quota \%} = \frac{F_P}{F_H} \times 100\% = 50\%$$

Memory Bandwidth Scaling:

$$\text{VM Memory BW} = \frac{\text{Host Memory BW}}{\text{Active VMs}} = 172 = 8.5\text{GB/s per VM}$$

$$\text{Memory BW \%} = \frac{1}{\text{Active VMs}} \times 100\% = \frac{1}{2} \times 100\% = 50\%$$

Memory Allocation Formula:

$$\text{VM Memory} = C_P = 8\text{GB}$$

$$\text{Memory Allocation \%} = \frac{C_P}{C_H} \times 100\% = \frac{8}{16} \times 100\% = 50\%$$

Overall Performance Normalization Factor:

$$\text{Performance Factor \%} = \frac{F_P}{F_H} \times \frac{\text{VM Memory BW}}{\text{Host Memory BW}} \times 100\% = 0.5 \times 0.5 \times 100\% = 25\%$$

This indicates that each VM should operate close to a quarter of the hosts peak performance to match the characteristics of a Raspberry Pi 4B (using the two ON, two OFF scheme).

4.1.1.3. Machine Learning Model Classification

4.1.1.3.1. Classification

Through the TensorFlow library one can access the Keras API. Keras offers a multitude of pre-trained models (known as applications). As classification was chosen. A classical CNN model (custom made) was also chosen as a reference. The following table shows a few key parameters of the selected models including their size, number of parameters and depth.

TABLE IV Classification Model Parameters

Model	Size	Parameters	Depth
Custom CNN	434 kB	111.2k	15
Xception	88 MB	22.9M	81
VGG16	528 MB	138.4M	16
VGG19	549 MB	143.7M	19
ResNet50	98 MB	25.6M	107
ResNet50V2	98 MB	25.6M	103
ResNet101	171 MB	44.7M	209
ResNet101V2	171 MB	44.7M	205
MobileNet	16 MB	4.3M	55
MobileNetV2	14 MB	3.5M	105
NASNetMobile	23 MB	5.3M	389

EfficientNetB0	29 MB	5.3M	132
----------------	-------	------	-----

These metrics (size, # of parameters and depth) in the table above significantly influences its performance and resource requirements during training and inference (deployment). On powerful hardware (Design Approach 1, server side processing), large models with millions of parameters and tens of layers (often hundreds) can be trained efficiently. This helps to leverage the extensive computational resources.x86_64 based hardware makes it easy as it is much more cost effective and can far exceed the performance of any onboard solution at a similar price point. The price to performance ratio for these types of hardware is often the leading factor for most of the modern algorithms being trained as developers have it at their disposal. Although this is changing, as edge devices are getting much more powerful, it is still lagging far behind for the previously stated reasons. Often the best performing edge devices have an uplink to servers with much more powerful hardware which can run complex models, this offloads the computational work required to achieve reasonable accuracy and speed. Even though this seems to be an advantage, this adds network dependencies. In the case of a network disruption the whole system becomes inoperable and the real-time performance suffers a lot depending on network latency (which is exacerbated for wireless installations). Deploying models on limited (power and compute performance limits) onboard hardware often shows its downsides quite quickly as the four way tradeoff between accuracy, speed, cost and efficiency is very tricky to manage. It is often impossible to mitigate due to four different mutually exclusive and conflicting requirements.

4.1.1.3.2. Object Detection

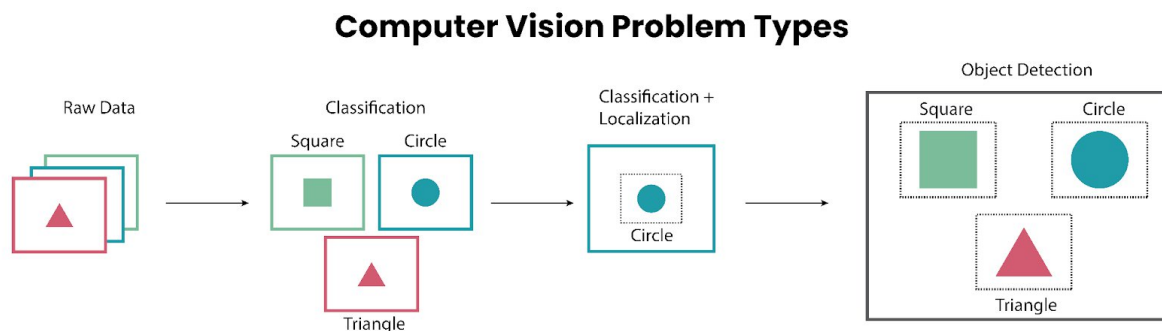


Figure 6: Types of Computer Vision Problems

Performing visual surveys of infrastructures using drones poses a unique challenge. To make computer vision algorithms effective and robust it often has to contain multiple classes and in the case of surveys the output is often needed to be reviewed, studied and stored. This sets a requirement for any potential algorithm, that the output must be human readable. Classification can be modified to fulfil this requirement but classification fails when there are multiple objects in one frame. Additionally classification requires specialized datasets (uniform background

with the object clearly identifiable). Nothing comes close to the sheer performance of object detection in the analysis of features like cracks, defects or corrosion (which is common for survey drones to encounter when monitoring infrastructure). The YOLO system developed by Ultralytics stands out as an optimal solution for such problems. Researchers and developers need to rapidly test and deploy prototypes multiple times before certain reliability and performance benchmarks are met. Unlike generic classification models Ultralytics YOLO is a purpose built end to end object detection framework especially trained for tasks like this. They offer pre-trained models that enable rapid prototyping as well as the capability of users to develop/train their models on dedicated datasets. The high accuracy combined with the ease of use makes it an attractive choice for hyper specialized tasks that are common in infrastructure inspection. This flexibility allows for swift iterations in the design-test cycle. After a robust system has been developed, custom models can be designed and deployed for commercial use.

4.1.1.4. Multi Node Training

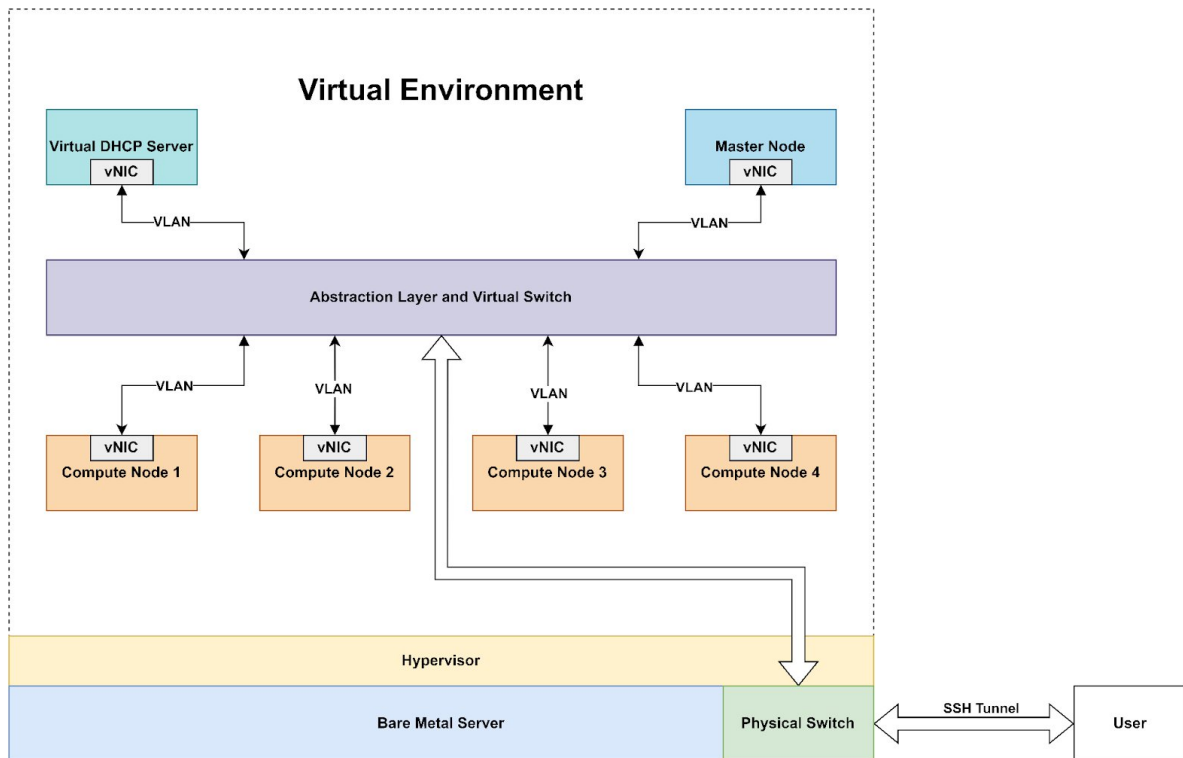


Figure 7: Virtual Environment Architecture

Due to limitations in memory and time multiple models had to be trained at once, as the onboard computational power calculations showed us that it would take a massive amount of time to finish training the model. Inference was way faster compared to the training so it did not require such advanced preparation and environment setup to get up and running. Multi node training mainly leverages multiple virtual machines to train or to perform inference on said models (similar procedure for pre-trained models). Other significant benefits include better environmental isolation and security. Distribution of training workloads across several

dedicated virtual machines can accelerate the training process. This also allows for sandboxing in the case of a model trained on sensitive data or malicious files (for advanced threat protection, reduction of risks associated with data leaks and a lower probability of attack vectors being created or exploited). This setup is highly configurable as it allows for precise allocation of system resources and robust (with the use of high availability technologies), snapshots and images can prevent the system from being rendered inoperable due to any misconfigurations or bugs as when such event occurs it is very easy to restore the machine to a predefined state. When compared to bare metal training this offers many advantages but has one challenge. If the hardware is somehow limited (as in our case memory limitation was a major issue as we only had 16GB of total working memory, in the real world it was closer to 12GB as there is system overhead and other issues that consume this limited resource).

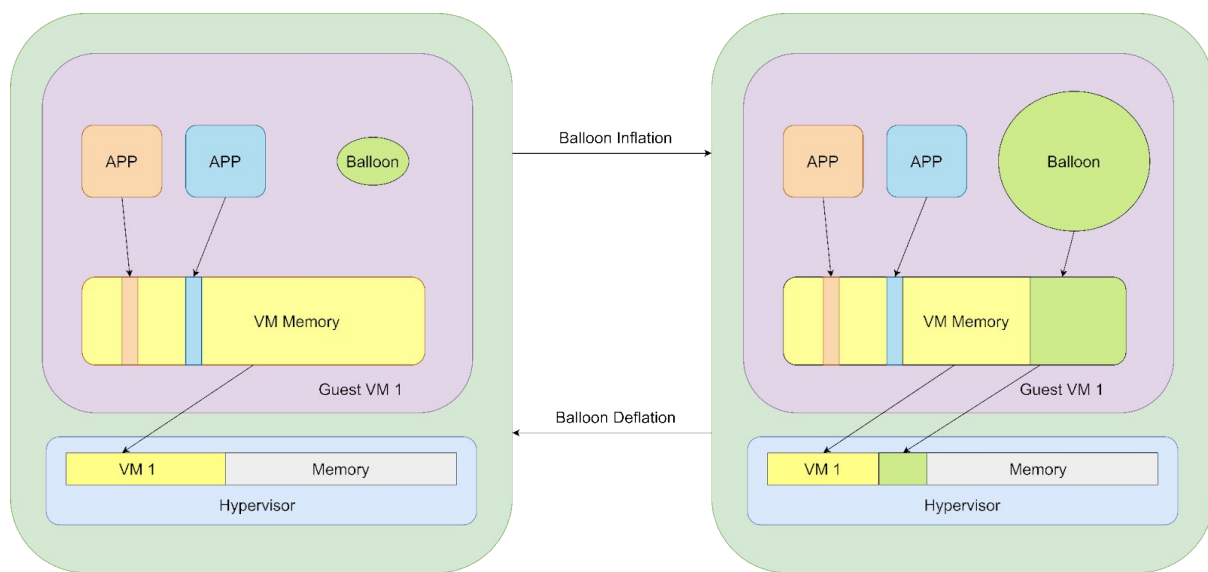


Figure 8: Virtual Machine Memory Management

This diagram shows a technique called memory ballooning. It is used by hypervisors to manage and allocate memory among virtual machines during high demand or when the system (bare metal) is limited. A special driver called a balloon driver (located within the guest virtual machine) "inflates" in order to consume the un-used memory space, essentially reserving the space in working memory for the hypervisor. The hypervisor then flags this memory space and redistributes this within other virtual machines (or to a specific one that has requested it). During deflation the hypervisor releases its "share" and the balloon driver re-allocates the memory space for the guest machines to use. This helps as multiple creates sudden spikes in available memory at different times, this dynamically distributes the limited resource so that one worker node does not consume a majority of the available space. By allowing other machines to utilize the free memory spaces in other virtual machines, it adds dynamic workload handling capabilities to the hypervisor and allows us to perform machine learning tasks without requiring expensive hardware upgrades.

4.1.2. System Flow Chart

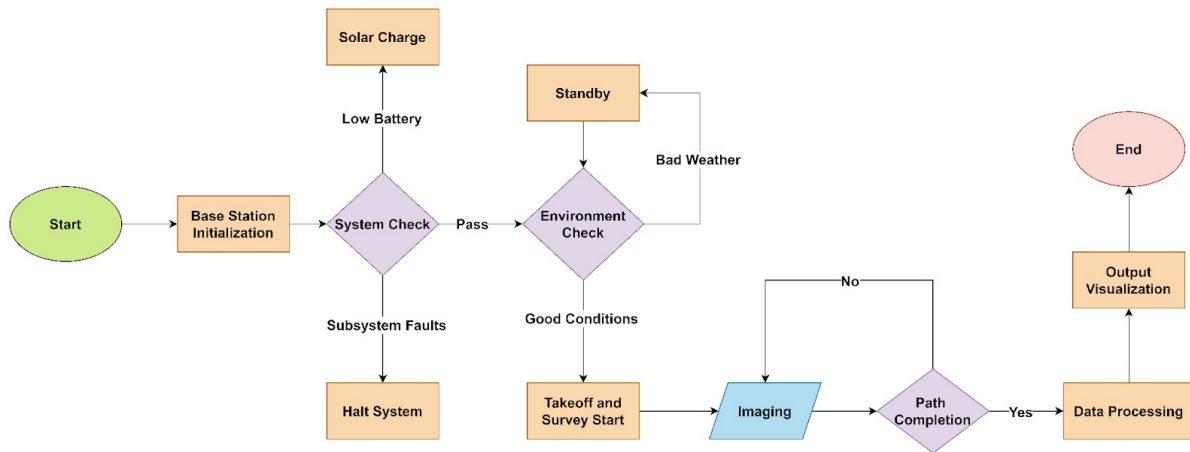


Figure 9: System Flow Chart

The whole system will consist of an automated parking platform and drone data offloading center. The parking platform will be capable of charging, protecting the drone when it is not in use and host a data collection/processing station. When idle, the drone battery charge will be monitored constantly. If this feature goes down below a certain percentage the charge controller will use the available power from the solar panel to charge the drone. The base station will not use any power from the grid under normal circumstances. If the primary power source (solar panel) is damaged or is offline for some reason then the whole station will shift to grid consumption mode. All the micro-controllers and smart sensing peripherals will operate in deep sleep. Only the master station controller will remain active when the system is online and idle. The background flight scheduler triggers the master microcontroller to initiate the flight routine. Going back to the charge controller, if the battery charge is above a certain percentage then the drone will be ready for flight. The base station will also have the capability to monitor the weather to prevent a launch if the conditions are unfavorable. This is mainly to avoid crashes and prevent loss of critical equipment. Once airborne, the drone will continue to take photos until the predefined flight path has been completed. Upon completion of the flight path it will return to base. There will be dedicated sensors that will detect the drones presence. If the sensor reports that the drone has landed then it will start an automatic data transfer process. Before the transfer process a few hidden steps will take place in rapid succession. The landing zone will have a wireless LAN. This network will only allow for registered devices to connect. All of the digested (or raw files, as it depends on the final design approach where what data ultimately goes) data will be transferred over a secure tunnel to prevent MITM attacks.

4.2. Optimization of Multiple Design Approach

4.2.1. Deep Learning Based Optimization

Using machine learning workloads themselves to assess hardware configurations is the most defensible way to choose platform because measure only the outcomes that matter: time-to-accuracy, energy/accuracy, cost/accuracy point, throughput/watt, and tail latency under the

specific models and dataset and software stack we intend to use. Synthetic benchmarks and micro level behavior, but modern ML training and inference exhibit complex, non-linear interactions among compute, memory bandwidth, cache hierarchy, interconnects, storage, and frameworks, that appear in the case of end-to-end runs only. By training and deployment of representative models on actual hardware and hardware that is deliberately constrained (e.g. limiting cores, memory, clocks, power, bandwidth), we expose whether our workloads are compute-bound, memory-bound or I/O/communication-bound and where bottlenecks change as models, batch sizes, precisions (FP32/FP16/INT8), and parallelization strategies change. This approach captures hardware--software co-optimization effects such as kernel fusion, mixed-precision, activation checkpointing, quantization, compiler graph rewrites that materially changes utilization and efficiency but are invisible to generic tests. It also enables apples-to-apples comparisons across realistic KPIs: time-to-first-accurate-model (including data loading and augmentation), energy-to-target-metric (e.g., top-1, or mAP), stability under thermal constraints and throttling, variability across random seeds, and degradations under background load. Artificially constraining hardware is particularly useful in that it simulates edge and embedded deployments, validates graceful degradation strategies and exposes the marginal utility of each resource (extra memory, better clocks, more bandwidth), helping us find the Pareto frontier rather than overprovisioning. Moreover, ML workloads are diverse: CNNs, transformers, GNNs, RL - Stress different subsystems, multi-GPU or distributed training. introduces communication patterns that can only run on real surfaces. Evaluating with our exact pipelines ensures conclusions are workload specific and future proof, while still enabling normalized comparisons based on fixed recipes and budgets of tuned. From an economic and sustainability perspective, energy-to-accuracy and cost-per-epoch measurement over realistic training horizons directly informs total cost of ownership, datacenter capacity planning and carbon targets in a way peak FLOPs cannot. Finally, a measurement campaign based on ML runs supports statistical rigor: multiple trials, confidence intervals, to reduce chance of chasing noise; telemetry correlates utilization with progression of accuracy to differentiate between hardware limits and software inefficiencies; and A/B tests between constrained and unconstrained environments determines the minimal configuration that meets service-level objectives. In short, utilizing ML to test hardware is not just benchmarking. Decision analytics that matches selection of platform with product goals, exposes true bottlenecks, quantifies trade-offs between speed, efficiency and cost, and produces actionable guidance for both training and scale and edge deployment.

4.2.1.1. Synthetic Load Testing Using Classification

To make a hardware decision based on the realities of our workload and not based on inferences from synthetic metrics, we first compared a diverse slate of 12 image classification architectures and selected the best four in accuracy on our target data set, recognising that model rankings are not universal and can be flipped across datasets by data distribution, scale and augmentation sensitivity. This two-stage process constrained hyperparameter tuning to a fixed budget for fairness, ensured identical data pipelines, and normalized batch sizes within memory limits so that accuracy selection reflected true model-data fit and not hardware

artifacts. We then reran those four finalists end-to-end on two environments: our x86 platform and a Raspberry Pi-- class constraint simulated by a limitation of cores, frequency, memory and I/O according to the Pi's envelope, applying appropriate kernel choices, math libraries to avoid bias. For each run we recorded decisions- making KPIs rather than peak FLOPs: time-to-train to a fixed target metric, energy-to-train (by integrating power over the full training horizon), inference latency on single images and small batches, sustained throughput in frames per second in steady state thermals, and efficiency in frames per watt under inference. This design helps us to see if performance is compute-bound or memory/IO-bound, how thermal throttling changes steady state behavior, and how accurate progression AND utilization over time The results were unambiguous for our use case. On x86, larger caches, higher memory bandwidths, wider SIMD and mature kernel libraries delivered much shorter training times which, in turn, reduced total energy-to-train despite higher instantaneous power = energy/power*time, and the orders of magnitude runtime delta favored x86. On the Raspberry Pi -class constraint, limited memory forced smaller batches and more frequent parameter and activation movement, inflating stalls and lowering arithmetic intensity; under sustained load the platform showed down-clocking thermally, further depressing throughput and increasing joules per image. In the time of inference, x86 achieved significantly higher raw FPS and, crucially, better frames-per-watt because the FPS gains outpaced the power increase; the Pi's attractive idle draw could not make up for its lower sustained throughput once pre-processing and framework overheads were included We also measured tail latency and we saw tighter appropriate kernel choices, math libraries to avoid bias. For each run we recorded decisions-making KPIs rather than peak FLOPs: time-to-train to a fixed target metric, energy-to-train (by integrating power over the full training horizon), inference latency on single images and small batches, sustained throughput in frames per second in steady state thermals, and efficiency in frames per watt under inference. This design helps us to see if performance is compute-bound or memory/IO-bound, how thermal throttling changes steady state behavior, and how accurate progression AND utilization over time The results were unambiguous for our use case. On x86, larger caches, higher memory bandwidths, wider SIMD and mature kernel libraries delivered much shorter training times which, in turn, reduced total energy-to-train despite higher instantaneous power = energy/power*time, and the orders of magnitude runtime delta favored x86. On the Raspberry Pi -class constraint, limited memory forced smaller batches and more frequent parameter and activation movement, inflating stalls and lowering arithmetic intensity; under sustained load the platform showed down-clocking thermally, further depressing throughput and increasing joules per image. In the time of inference, x86 achieved significantly higher raw FPS and, crucially, better frames-per-watt because the FPS gains outpaced the power increase; the Pi's attractive idle draw could not make up for its lower sustained throughput once pre- processing and framework overheads were included We also measured tail latency and we saw tighter distributions on x86, what matters to service level objectives To strengthen internal validity, we did runs with multiple seeds, monitored the utilization and memory pressure and checked that algorithmic optimizations (e.g. mixed precision, operator fusion) were consistently enabled across both the environments; we also tested INT8/FP16 variants where supported and still observed the same ordering. Economically, the x86 platform's faster time-to-accuracy reduced opportunity cost in experimentation and reduced the energy and the calendar time needed to reach deployable

models; operationally, it processed background system tasks without performance collapses seen on the constrained environment. While a Raspberry Pi still holds appeal for super low power or low cost. First deployments, our measurements -- time-to-train, energy-to-train, inference latency, FPS and frames per watt-demonstrate that for our data set, model mix, and throughput targets, x86 is both more powerful and more efficient practically. Because of the end-to-end ML runs that the conclusion is based on rather than synthetic proxies, it is generalized across realistic variations in batch size, augmentations, and runtime conditions and provide actionable guidance: We will train and serve in x86 to meet accuracy and throughput goals within energy budgets revisiting edge pathway only if constraints move towards extreme power/ bill of materials limits.

6.3.2 Accuracy Testing Using Object Detection

Having established through end-to-end ML benchmarking that x86 is the more efficient and capable platform for our workload, we pivoted from classification to object detection - which better adds value to the project's operational goals (localizing targets instead of just labeling frames) We chose the Ultralytics YOLO family as our main baseline as this includes a well-engineered spectrum of architectures with a proven accuracy/latency trade-off, mature training tooling & simple export paths to deploy but all important for field systems where we have to tradeoff between balance accuracy and runtime and power budgets. Within that family, the "n" variants (v8n, 11n, 12n) are intentionally lightweight and hence attractive for edge scenarios, while "s" variants (e.g. 12s) add capacity, architectural and recipe refinements expected to deliver increased accuracy when provided with enough optimization time. We trained and tested v8n, 11n, 12n, and 12s, on the same curated data, using the same train/val/test splits, same consistent image sizes and augmentations, and controlled tuning budget (to keep comparisons fair) Our metrics covered end- to-end decision KPIs: mAP at several IoU thresholds to measure localization quality, precision/recall trade-offs to evaluate error modes, time-to-mAP to characterize iteration velocity, and deployment-focused measures-single-image and small-batch latency, sustained FPS under steady thermals, and frames/second during inference. The results were consistent with expectations but were instructive. YOLOv8n was a very good speed baseline with low compute and the fastest training and inference times, but lagged in mAP compared with the newer "n" models, confirming that its efficiency comes with a measurable accuracy trade-off on our dataset YOLOv11n and YOLOv12n were both able to improve accuracy at a small compute increase, delivering improved precision/recall balance without violating our latency or power envelopes; notably, their inference throughput was enough to help us reach our target of FPS on x86 without dropping frames-per-second.watt competitive, which is important for continuous field operation YOLOv12s, by design, showed the most promise: More representational capacity and newer recipe refinements, its early the training curves showed the trajectory to be steeper and the ceiling to be higher. However, because 12s is bigger, it benefits from longer optimization horizons - more epochs, more robust LR schedules, and potentially richer augmentations-to achieve its full potential Due to time constraints , we could not train 12s for as many epochs as the "n" models, which left it under-optimized and, in our measured runs, less than the accuracy of 11n and 12n despite its superior design This outcome is consistent with learning dynamics: increased-capacity detectors usually require increased steps to converge and regularize effectively, and when cut-short may trail smaller models that reach their performance plateau faster Looking at this from a product perspective these results strengthen our deployment strategy. Heavily training a light detector

and serving it on x86 gives the best user experience and economics: It can meet our accuracy targets while keeping things tight by stabilizing inference latency, tail behavior, and doesn't force customers to obtain expensive GPU servers. The “n” variants, especially 11n and 12n, hit the right balance with our distribution of data and throughput requirements today, while 12s is an interesting candidate for a future iteration if we allocate an increased training window or use compression techniques (quantization, pruning, or distillation) to capture its benefits without losing edge efficiency. In short, our process, which is evidence-driven consistent datasets, controlled training budgets and deployment-relevant metrics-shows $v8n < 11n/12n$ in accuracy on our data, 12s promising but undertrained; so we focus 11n and 12n for immediate field deployment and have 12.

4.2.1.2. Accuracy Testing Using Object Detection

Object detection is predominantly a better suited option for ML/DL based tasks where end users need high accuracy and dependable results. This method provides exceptional performance standards for a vast and versatile range of deployment options. The gain in accuracy does not come at the cost of being immensely computationally intensive; rather the accuracy boost is not proportional to the compute resource consumed. As object detection selects the whole image and extracts relevant areas from the whole scene to minimize label noise (a common issue with CNN based classifiers as they compress/combine all the labels into a homogenous layer, this causes multiple objects to be detected under one label thus increasing errors). Whereas object detection uses labels and annotations to narrow down each image and has a much higher degree of granularity. This is evident with an example, if there are two dogs in the same frame and it has been labeled as a dog a classifier will identify the whole image as a dog. It cannot show where the dog is in the frame, if there is anything else in the frame or if there are multiple dogs in the same frame. The problem gets worse if a cat decides to enter the frame. A classifier needs training images with clear background/foreground separation. Setting the accuracy limitations aside, if somehow we need to localize or segment the image with a confidence score it becomes a difficult task due to the inherent nature of classification models and problems. Object detection can solve all of these but it introduces a new problem into the mix. Unlike classification problems the training algorithm needs to be told beforehand where what is and with what level of confidence it detected the object. There is a tradeoff with accuracy and training. Object detection models need a lot of work beforehand to become accurate and classification models need it afterwards (in the form of additional layers and added functionality that further processes the output into meaningful data).

4.2.2. Dataset Pre Processing

The process of collecting and processing data for training is a crucial step for any machine learning task. Our research has led to us finding four datasets, one from Keggles, Roboflow and Open Access sources. Each of these databases were annotated for PASCAL VOC and YOLO. As each class is annotated separately, thus it is not possible to train any object detection model that has all those classes together. The first issue our team faced is that the files were named

randomly (all four classes had a total of 12,662 images). We have downloaded each dataset (as a compressed file zip, rar or 7z files just to name a few). Since it was downloaded in windows PowerShell was an easy option to do some preliminary processing, taking all compressed archives, dumping all the files into a single file and compressing it at last. The first task is to prepare the dataset for multi-class object detection training. Before that a classifier that will perform tests on the dataset to see the performance. Every class is renamed to a consistent naming pattern, each class is a directory (as most classifiers). The following is the task flow to prepare the dataset for a multi-class object detection training session. Local Map Server Setup.

```
$dataDumpPath = ".\DATA_DUMP"
$finalZipPath = ".\DATA_DUMP.zip"
$logPath = ".\extraction_log.txt"

if (!(Test-Path $dataDumpPath)) { New-Item -ItemType Directory -Path
$dataDumpPath -Force | Out-Null }
if (Test-Path $finalZipPath) { Remove-Item $finalZipPath -Force }
if (Test-Path $logPath) { Remove-Item $logPath -Force }

$zipFiles = Get-ChildItem -Path "." -Include "*.zip", "*.rar", "*.7z" -Recurse
| Where-Object {
    $_.Name -ne "DATA_DUMP.zip" -and $_.Length -gt 0
}

$totalFiles = $zipFiles.Count
$processedFiles = 0
$extractedCount = 0
$failedExtractions = @()

foreach ($archive in $zipFiles) {
    $processedFiles++
    $progressPercent = [math]::Round(($processedFiles / $totalFiles) * 100,
2)
    Write-Progress -Activity "Processing Archives" -Status "File
$processedFiles of $totalFiles ($progressPercent%)" -PercentComplete
$progressPercent

    $extractPath = Join-Path $dataDumpPath $archive.BaseName
    if (!(Test-Path $extractPath)) { New-Item -ItemType Directory -Path
$extractPath -Force | Out-Null }

    $timestamp = Get-Date -Format "yyyy-MM-dd HH:mm:ss"

    try {
        switch ($archive.Extension.ToLower()) {
            ".zip" {
                Expand-Archive -Path $archive.FullName -DestinationPath
$extractPath -Force
                $extractedCount++
                "$timestamp - SUCCESS: Extracted $($archive.Name) to
$extractPath" | Add-Content $logPath
            }
            ".rar" {
                if (Get-Command "winrar" -ErrorAction SilentlyContinue) {
                    & winrar x -y "$($archive.FullName)" "$extractPath\"
                    if ($LASTEXITCODE -eq 0) {
                        $extractedCount++
                    }
                }
            }
        }
    } catch {
        $failedExtractions += $archive.FullName
    }
}
```

```

        "$timestamp - SUCCESS: Extracted $($archive.Name) to
$extractPath" | Add-Content $logPath
    } else { throw "WinRAR extraction failed" }
    } else { throw "WinRAR not found" }
}
".7z" {
    if (Get-Command "7z" -ErrorAction SilentlyContinue) {
        & 7z x "$($archive.FullName)" -o"$extractPath" -y
        if ($LASTEXITCODE -eq 0) {
            $extractedCount++
            "$timestamp - SUCCESS: Extracted $($archive.Name) to
$extractPath" | Add-Content $logPath
        } else { throw "7-Zip extraction failed" }
        } else { throw "7-Zip not found" }
    }
}
}
catch {
    $failedExtractions += $archive.Name
    "$timestamp - ERROR: Failed to extract $($archive.Name) -
$(($_.Exception.Message))" | Add-Content $logPath
}
}

Write-Progress -Activity "Processing Archives" -Completed

$allExtractedFiles = Get-ChildItem -Path $dataDumpPath -Recurse -File
$duplicateFiles = $allExtractedFiles | Group-Object Name | Where-Object
{ $_.Count -gt 1 }

foreach ($duplicateGroup in $duplicateFiles) {
    $counter = 1
    foreach ($file in $duplicateGroup.Group | Select-Object -Skip 1) {
        $newName = "$($file.BaseName)_duplicate$counter$($file.Extension)"
        $newPath = Join-Path $file.Directory $newName
        Rename-Item $file.FullName $newPath
        $counter++
    }
}

$emptyDirs = Get-ChildItem -Path $dataDumpPath -Recurse -Directory | Where-
Object {
    (Get-ChildItem $_.FullName -Recurse -File).Count -eq 0
}
$emptyDirs | Remove-Item -Recurse -Force

if ((Get-ChildItem -Path $dataDumpPath -Recurse -File).Count -gt 0) {
    $compressionLevel = "Optimal"
    Compress-Archive -Path "$dataDumpPath\*" -DestinationPath $finalZipPath -
CompressionLevel $compressionLevel -Force

    $originalSize = ($allExtractedFiles | Measure-Object -Property Length -
Sum).Sum
    $compressedSize = (Get-Item $finalZipPath).Length
    $compressionRatio = [math]::Round((1 - ($compressedSize / $originalSize))
* 100, 2)

    $summary = @"
Extraction Summary:
- Total archives found: $totalFiles

```

```

- Successfully extracted: $extractedCount
- Failed extractions: $($failedExtractions.Count)
- Total files extracted: $($allExtractedFiles.Count)
- Duplicate files renamed: $($duplicateFiles.Count)
- Empty directories removed: $($emptyDirs.Count)
- Original size: $([math]::Round($originalSize / 1MB, 2)) MB
- Compressed size: $([math]::Round($compressedSize / 1MB, 2)) MB
- Compression ratio: $compressionRatio%
- Failed files: $($failedExtractions -join ', ')
"@

    $summary | Add-Content $logPath
    Write-Host $summary
} else {
    "No files were extracted successfully" | Add-Content $logPath
}

```

Once in a WSL2 instance (chosen for ease of use to share hardware between the host and guest linux system) and used a few tools like find, AWK, SED and xargs to first isolate all the images (only the ones with the JPG extension). The first step was to find and list what types of files were in the archive (using the command-line utility called unzip).

```
find . -type f -name "*.*)" | awk -F . '{print $NF}' | sort | uniq
```

After that all the image files were isolated as stated before.

```
find . -type f -name "*.jpg" -exec mv -v {} /FYDP/D/ML_data-dump/ \;
```

To count how many files we had the following utilities were used.

```
ls -l /FYDP/D/ML_data-dump/ | wc -l
```

We had to rename all the files consistently with a format similar to img_(number).jpg but we made a mistake and the files were renamed as image<space>(number).jpg format. As it is not recommended to have spaces in the names of files we had to correct for it. The first Bash script was used to rename (corrected for the mistake but after the fact).

```
#!/bin/bash
count=1
for file in *.jpg; do
    if [ -f "$file" ]; then
        mv "$file" "img_{$count}.jpg"
        count=$((count + 1))
    fi
done

```

The correction was made with the following command that utilizes regular expressions.

```
find . -type f -name "img (*.jpg)" -exec bash -c 'for f; do mv -v "$f" "${echo "$f" | sed -E "s/ \([^\)]+\)\(_\1/\1/"}"; done' _ {} +
```

We found out (mentioned earlier) 12,662 files. To split the files between the team members (approx. 4220, divided among three people) to start the annotation process.

```
find . -type f -iname "*.jpg" | sort | head -n 4220 | xargs -I {} mv -v "{}" /FYDP/D/distribution/name
```

4.2.3. Design and Development of a Custom Data Visualization Suite

The map server is a core feature of the system as we aim to remove internet/web-service dependencies for deployment of our project in sensitive installments. Eliminating internet dependency reduces the attack surface drastically to such a point that in order to infiltrate the system one would need to be in the premises and manually alter system parameters which would result in the swift apprehension of the perpetrator. Thus we devised a method to add hosting capabilities to multiple nodes without ever connecting to the internet, effectively making an air-gapped environment with limited permissions (equivalent to a chroot jail in UNIX), given a few files are downloaded, verified and scanned beforehand, then stored accordingly, designated for future use in similar deployments. The following steps are done completely offline. Once a capable machine (most modern machines either feature Intel VT-x or AMD SVM required for virtualization) is selected then a fresh copy of Microsoft Windows is installed on the device. Modern versions of Microsoft Windows force users to create an online account. Previous versions of the OS had a bypass script built into the out of the box experience (OOBE), called *BypassNRO.cmd*. As of the time of writing this document modern versions (greater than 24H2) can be bypassed via the following command in a admin command prompt (launched by pressing Shift+F10 hotkey) after the installation just as the OS message appears to create/sign up to the account (or some form of a network requirement if the system is kept offline as described earlier). All of the following shell interactions are performed with elevated privileges unless specified otherwise.

```
C:\> start ms-cxh:localonly
```

Once a local account has been created, the WSL2 (Windows Subsystem for Linux 2) can be installed. The MSI (Microsoft Software Installer not to be confused with Micro Star International, the hardware manufacturer) file for WSL2 can be collected from GitHub. It is installed after enabling WSL and Microsoft Hyper-V features via DISM.

```
PS C:\> DISM /Online /Enable-Feature /FeatureName:Microsoft-Windows-Subsystem-Linux /All /NoRestart
```

```
PS C:\> DISM /Online /Enable-Feature /FeatureName:VirtualMachinePlatform /All /NoRestart
```

```
PS C:\> DISM /Online /Enable-Feature /All /FeatureName:Microsoft-Hyper-V /All /NoRestart
```

After enabling the physical install of the WSL2 MSI package can begin in the command prompt.

```
C:\> msixexec /i "Drive_Letter:\Path\to\WSL2\Package\version_architecture.msi"
```

Microsoft provides update files (with the extension .MSU, these are called Microsoft Update Standalone Packages). After collection of the package from Microsoft Update Catalog.

```
PS C:\> DISM /Online /Add-Package  
/PackagePath:"Drive_Letter:\Path\to\Update\Package\os_build_version_number.ms  
u"
```

After the update, Hyper-V OS setup can be done with the following steps.

```
PS C:\> Get-NetAdapter
```

```
PS C:\> New-VMSwitch -Name "External-Map" -NetAdapterName "Ethernet" -  
AllowManagementOS $true
```

```
PS C:\> New-Item -ItemType Directory -Path "C:\Hyper-V\Virtual Hard Disks" -  
Force
```

```
PS C:\> New-VM -Name "map-machine" -Generation 2 -MemoryStartupBytes 4GB -  
SwitchName "External-Map" -NewVHDPATH "C:\Hyper-V\Virtual Hard Disks\map-  
machine.vhdx" -NewVHDSIZE 120GB
```

```
PS C:\> Set-VMemory -VMName "map-machine" -DynamicMemoryEnabled $true -  
MinimumBytes 512MB -StartupBytes 4GB -MaximumBytes 8GB -Buffer 20 -Priority  
75
```

```
PS C:\> Set-VMProcessor -VMName "map-machine" -Count 4 -Maximum 80 -Reserve 5  
-RelativeWeight 75
```

```
PS C:\> Set-VMdvdDrive -VMName "map-machine" -Path  
"C:\Path\to\Debian\ISO\filename.iso"
```

```
PS C:\> $dvd = Get-VMdvdDrive -VMName "map-machine"
```

```
PS C:\> Set-VMFirmware -VMName "map-machine" -EnableSecureBoot On -  
SecureBootTemplate "MicrosoftUEFICertificateAuthority" -FirstBootDevice $dvd
```

```
PS C:\> Connect-VMNetworkAdapter -VMName "map-machine" -SwitchName "External-  
Map"
```

After Debian installation it needs to be modified to use a static IP.

```
$ sudo micro /etc/network/interfaces
```

Static IP configuration (with placeholder IP's)

```
auto eth0
iface eth0 inet static
    address [REDACTED]
    netmask 255.255.255.0
    gateway [REDACTED]
    dns-nameservers 8.8.8.8 8.8.4.4
```

Finally enable and restart the networking service to append changes.

```
$ sudo ip link set dev eth0 up
$ sudo systemctl restart networking
```

To enable openssh on a server and copy the map data over via secure copy the following commands are utilized.

```
$ sudo apt update
$ sudo apt install openssh-server -y

$ sudo systemctl enable ssh
$ sudo systemctl start ssh
$ sudo systemctl status ssh
$ ssh user@[REDACTED] 'mkdir -p ~/map_data'
```

Copying the map data from the local machine to the Hyper-V server.

```
PS C:\> scp map_data.mbtiles user@[REDACTED]:~/map_data/
```

To create and run the tileserver these steps were followed.

```
$ sudo apt install apt-transport-https ca-certificates curl gnupg lsb-release
```

```
$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --
dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg
```

```
$ echo "deb [arch=$(dpkg --print-architecture) signed-
by=/usr/share/keyrings/docker-archive-keyring.gpg]
https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable" | sudo
tee /etc/apt/sources.list.d/docker.list > /dev/null
```



```
$ sudo apt update
$ sudo apt install docker-ce docker-ce-cli containerd.io docker-buildx-plugin
docker-compose-plugin

$ sudo systemctl enable docker
$ sudo systemctl start docker
$ sudo usermod -aG docker $USER
$ docker run hello-world
```

Creating a service for TileserversGL

```
$ sudo micro /etc/systemd/system/tileservers.service
```

Tileservers Service Unit

```
[Unit]
Description=Tileservers GL (Docker)
Wants=docker.service
After=docker.service network-online.target
Requires=docker.service

[Service]
Restart=always
RestartSec=5
Environment=DOCKER_IMAGE=maptiler/tileservers-gl:latest
Environment=HOST_PORT=8080
Environment=DATA_DIR=/home/map_data/mbtiles
ExecStartPre=/usr/bin/docker pull ${DOCKER_IMAGE}
ExecStartPre=/usr/bin/docker rm -f tileservers || true
ExecStart=/usr/bin/docker run \
  --name tileservers \
  --rm \
  -p ${HOST_PORT}:8080 \
  -v ${DATA_DIR}:/data \
  ${DOCKER_IMAGE}
ExecStop=/usr/bin/docker stop tileservers
StandardOutput=journal
StandardError=journal

[Install]
WantedBy=multi-user.target
```

To activate the service.

```
$ sudo docker stop tileservers
$ sudo docker start tileservers

$ sudo systemctl enable docker

$ ls -la /home/map_data/mbtiles
```

```
$ sudo netstat -tlnp | grep :8080
```

A machine checkpoint is created to create a restore point/failsafe in the case of a catastrophic misconfiguration, ensuring near zero downtime (the checkpoint will effectively double disk usage to 240GB).

```
PS C:\> Checkpoint-VM -Name "map-machine" -SnapshotName "Checkpoint_$(Get-Date -Format 'yyyyMMdd_HH:mm:ss')"
```

4.2.3.1. Data Point Mesh Generator

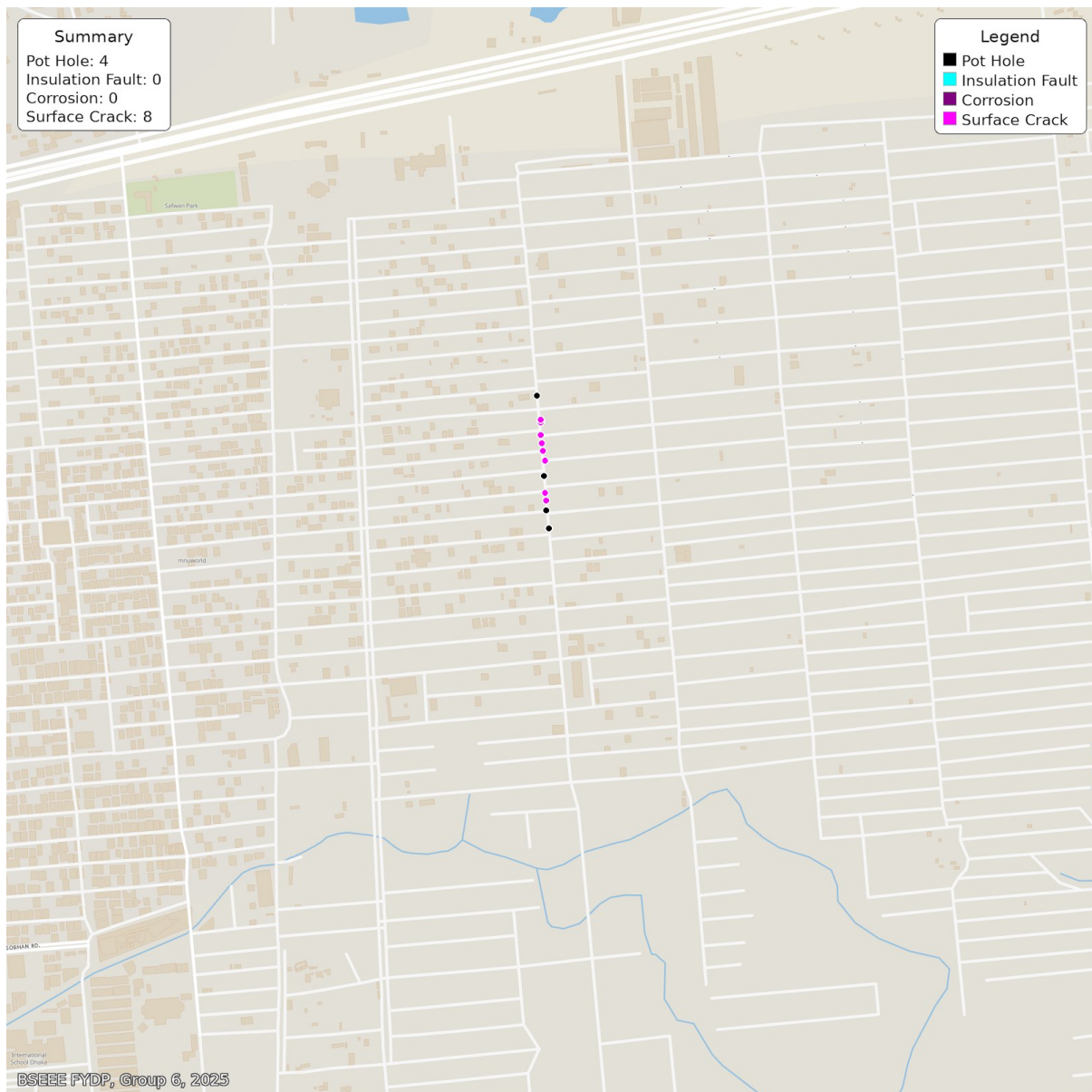


Figure 10: Data Point Mesh Generator

The main goal of the codebase is to create a visual map that shows different types of infrastructure faults and problems commonly encountered during a routine survey in a specific

area. The program initialization sequence starts by downloading a vector map (mutli-layer). As this is a beta system and due to operational constraints in the simulation phase we have locked to a specific geo-location in Bangladesh. After that the program reads through the standardized CSV file that is generated by our custom built object detection algorithm. All the faults according to their geo-location are displayed with different color coded dots according to the standardized format (available across all end products). These colors are kept constant across the next iterations of this mapping tool and it has been set in the beta version of our code. The program finally enhances the final image by adding a color coded legend that explains what each dots color means so that anyone who has access to the images afterwards can easily interpret what each image means. A summary box is also placed to indicate how many of each fault class have been detected. Finally to commemorate our work we added our department, group number and university name along with the year at the bottom corner of the image. As this is just a beta version we did not do the multi layer landmark augmentation. Just to prove the point that our trigonometric projections generate accurate points according to the original fault CSV file.

4.2.3.2. Data Verification Tool

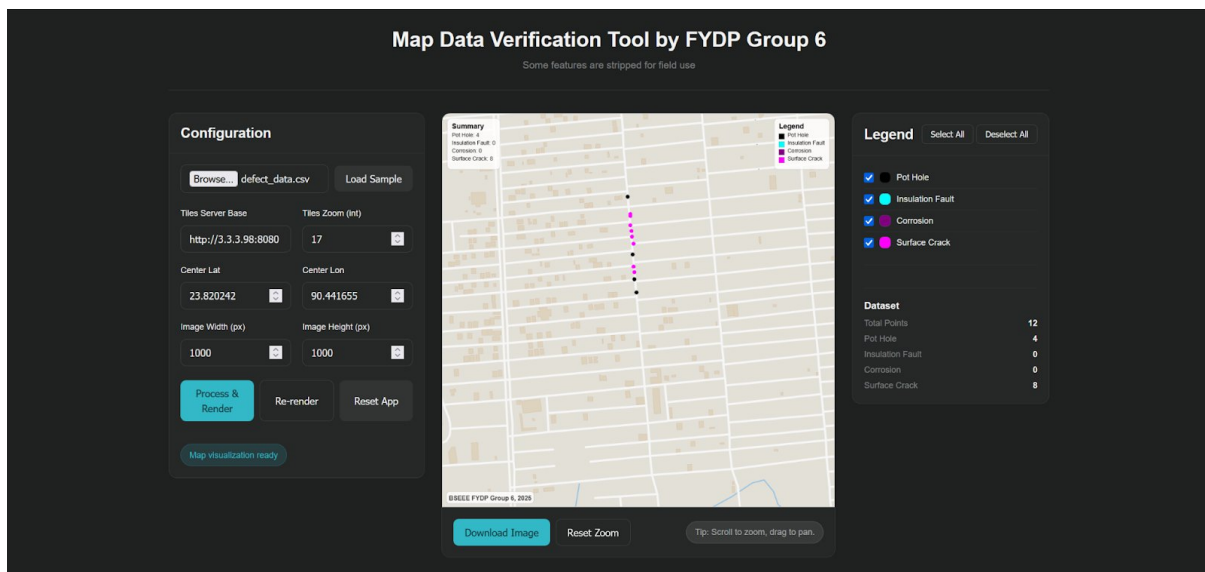


Figure 11: Custom Data Verification Tool

There were multiple scripts used in the creation of this tool connected to the map server created by the detailed process described in the previous sections. It is mainly a browser based map data verification tool that ingests a CSV file (which is generated by our object detection pipeline and is portable across our projects as it uses confined scope standardization) and shows a map of detailed defects. The front end consists of a HTML5 web page (modified by CSS) it has a configuration panel for the view settings and a few of the toggle menus showing important and required control interfaces. After the front end host loads and configures the CSV and server connection is configured (the defaults are changeable in the source code) a few additional pieces of information is required to be entered by the user. The front end canvas also calls for center latitude and longitude of the total map zone. A helper builds a static map of the

URL based on the predefined IP, View Style (defined as server/styles), base coordinates and zoom level. This static URL is then forwarded via web-request to the server which responds with a proper image with appropriate zoom and bounds. The scaling factor depends on this also. All of these are handled by a JavaScript applet. Inside the JavaScript application a MapVisualization class initializes the context shown across the main canvas. Driving the end-to-end flow. The base server style ID, center lat, center lon and the zoom factor is required by TileservrGL, the JavaScript file also has the capacity to perform trigonometric operations to draw DOM objects corresponding to each color attributed to the fault classes. An issue we faced during training is the

4.2.3.3. Deep Exploration Tool



Figure 12: Deep Exploration Tool

Connected to the TileservrGL server set up previously it is a Streamlit based viewer that is designed to provide granular control. The data exploration tool titled "Map-Lab" provides a deeper and more exploratory view of collected data post mission. The careful review of data is of utmost importance to get a proper analysis and give an accurate verdict on decisions to mitigate found issues. As discussed previously the explorations tools give a quick understanding that the tool did not malfunction in the field and provided proper datapoints for further analysis. It basically eliminates the risk of realizing that the survey failed after leaving the site, saving precious resources like time and money. The exploration tool provides a more granular look compared to the quick field verification tools by letting users analyze and validate class faults, multi layered views and advanced filtering options. Careful review of metadata before finalizing deliverables is a key goal for this tool. We took aspects from the JavaScript implementation and added functionality to the original beta version on which this whole system was built upon. It is basically a bootstrap of an Folium map alongside a static export of the server based pipeline so both other the exploratory analysis. Resulting in both exploration and publication-ready views ready at the users fingertips. This applet is launched by Streamlit

which is basically a launch application through a local IP (accessible via a web app). The Streamlit pipeline imports the Folium backend that has an interactive map pane and a pseudo-engine to interpret the fault CSV. CSV is standardized with LAT, LON and Class ID (based on the detection result and it is predefined). Core classes include Pot Holes, Insulation Faults, Corrosion Spots on Metal and Cracks on Concrete Surfaces. These classes are reflected in a human readable (easily readable by the end user) format, thus the labelling is consistent across different tools in our custom suite.

4.2.4. Optimizing Design Approach 1

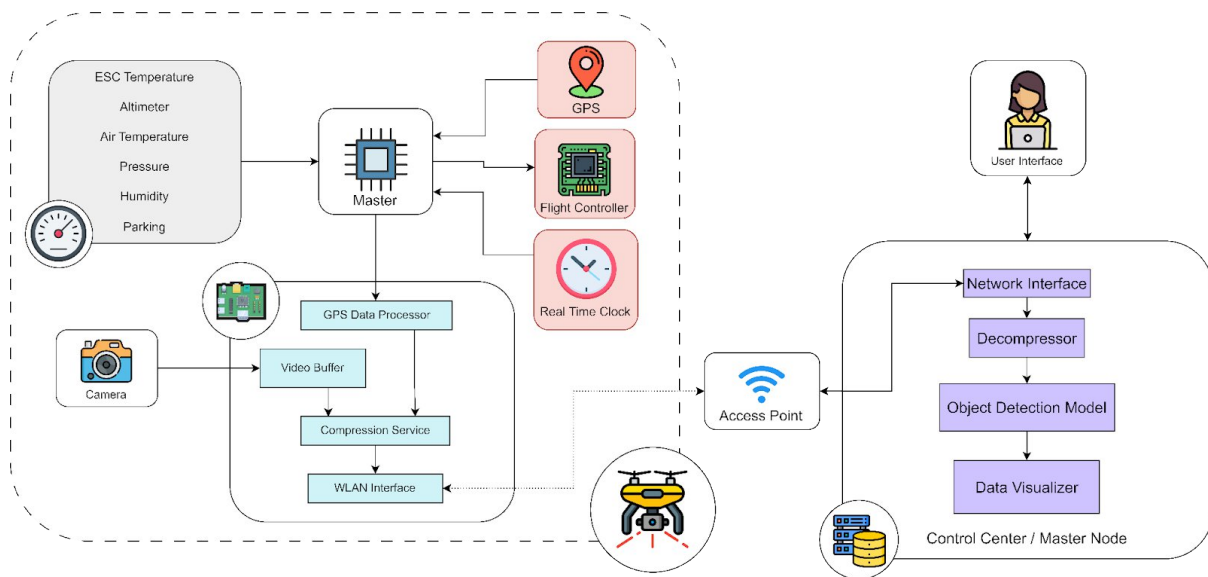


Figure 13: Design Approach 1 Overview

Here, the main difference from the other design is that it is really not altered by a pre-processing algorithm beforehand. Although this increases the load time for the exit point, it has a better processing subsystem for advanced workflows. This approach has the advantage of having large amounts of computational power at the user's disposal. The price to performance ratio is one of the key differences between the two.

4.2.4.1. Training

4.2.4.1.1. Accuracy

Classic CNN: The image represents the proper training and confirmed metrics for the Classic CNN model, which includes proper accuracy and loss over period. Moreover the data presented is not properly presented and is not clear due to the lack of labelling axes, specific value. Thus plots show training and validation accuracy which will eventually increase overtime while training loss is decreasing. In case loss could increase due to overfitting. The values are like 0, 20, 40, 60, 80, 100 period number or metric percentage but there will be no proper context.

There is no proper conclusion. The validation accuracy diverges from training accuracy and it could overfit. If the model is properly trained it could do well.

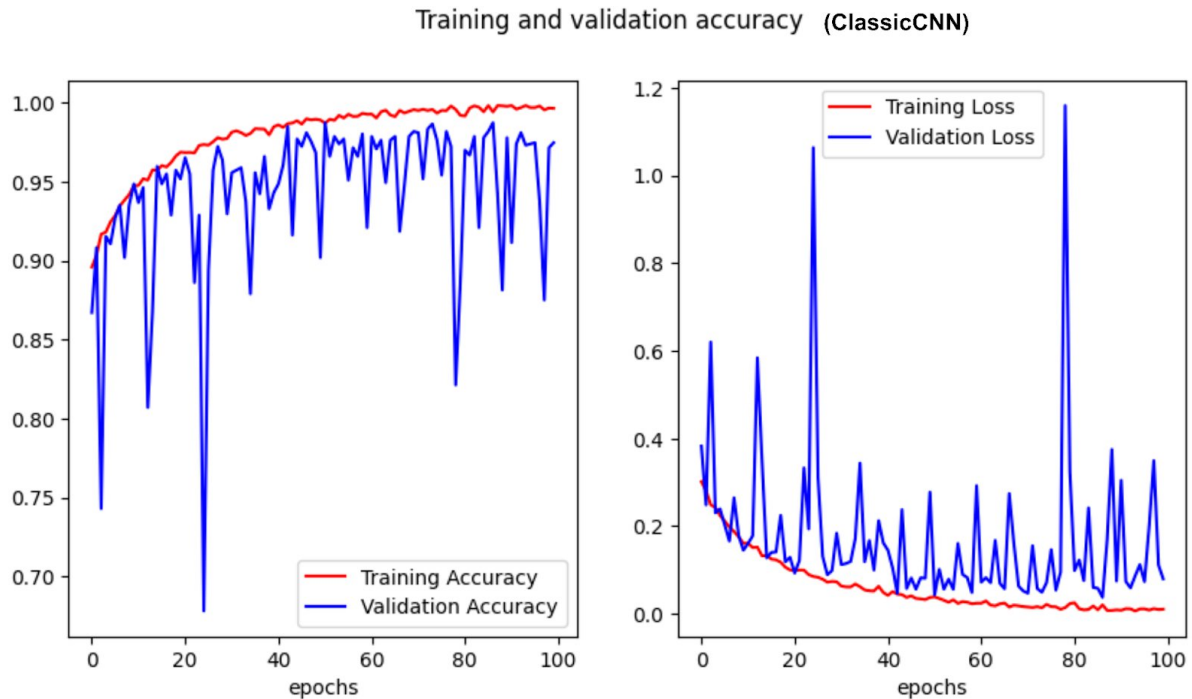


Figure 14: Training vs Validation Accuracy for ClassicCNN

EfficientNetB0: This plot shows a good EfficientNet-B0 training run: training accuracy clumps from ~0.6 to ~0.995, and validation accuracy is very close, saturating around 0.96-0.97 by ~70 to 1000 epochs. Moreover, this training loss falls smoothly toward zero, and validation loss falls very fast from 0.15 to 0.2, having minor noise. But they make up for it with a good generation. There is no divergence or instability. Because validation has proper metrics, and a well-performed early stopping approach cuts training time without losing performance, and light regulation or data augmentation might narrow the residual gap. The learning rate looks appropriate. Moreover, epochs 15-30 have the convergence to the plateau. Overall the model has capacity for this data, which will achieve high accuracy and manage outfits. Further gain will likely come from more data and stronger argumentation, class rebalancing, or larger variants rather than simply training longer.

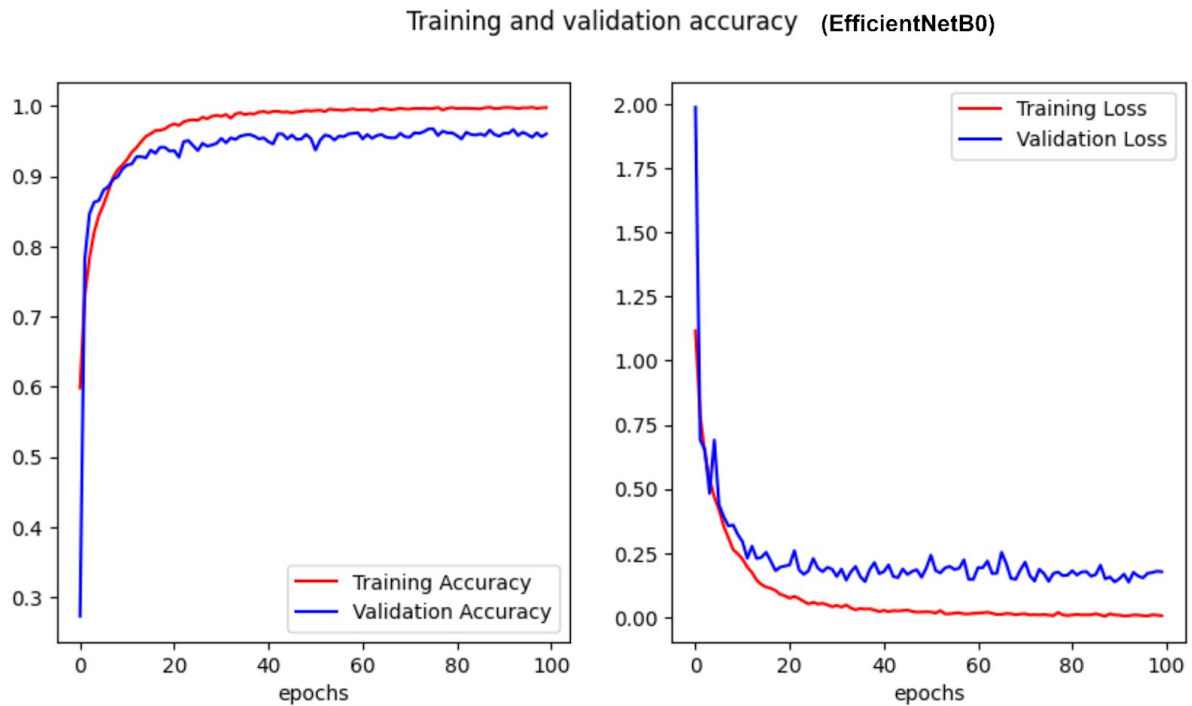


Figure 15: Training vs Validation Accuracy for EfficientNetB0

MobileNet: In this graph we can see that the MobileNet model trained for 100% accuracy achieved fast convergence and strong generalization with good controlled overfitting. Accuracy training rises from ~0.63 to ~0.99 by ~40-60 accuracy, where validation accuracy follows closely, climbing from ~0.17 to ~0.96-0.97 with small fluctuations, indicating stable learning and a smaller gap between training and validation performance. Loss curves corroborate this training loss decays smoothly from ~1.0 to near zero and the initial spike settles from 0.1 to 0.2 with occasional noise with early optimization and minima without divergence implication. The dataset and augmentation transfer settings are appropriate, and extra accuracy is likely a diminishing return; early stopping around the plateau can save further improvement, which would likely require better data, better data turning, a better learning rate schedule, and stronger augmentation rather than training longer.

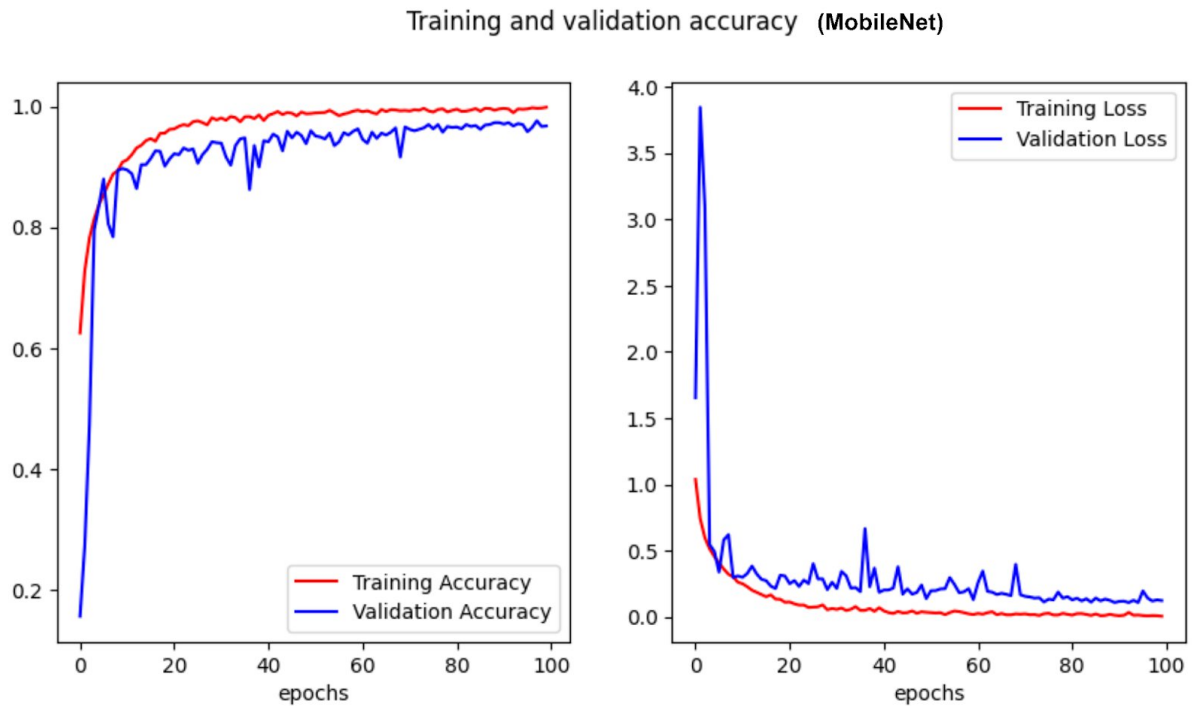


Figure 16: Training vs Validation Accuracy for MobileNet

MobileNetV2: The training accuracy rises fast. In this case, from ~ 0.65 to ~ 0.99 with training loss, which comes to zero steadily, and the model fits the training data well. The validation accuracy starts at low (~ 0.27) but climbs to ~ 0.9 by epoch 18, and it starts to fluctuate between 0.90 - 0.97 while the validation loss spikes around epoch 12. The pattern shows temporary instability, particularly from the learning rate, followed by recovery. The small validation gap ($\approx 0.02 \sim 0.07$) and noisy validation curve point to where it tries to overfit and sensitivity to data variation. For the improvement of stability, we can lower the learning rate using early stopping, add regulation, and check issues during layer unfreezing or LR shift. A larger or balanced validation set or cross-validation also updates the performance.

Training and validation accuracy (MobileNetV2)

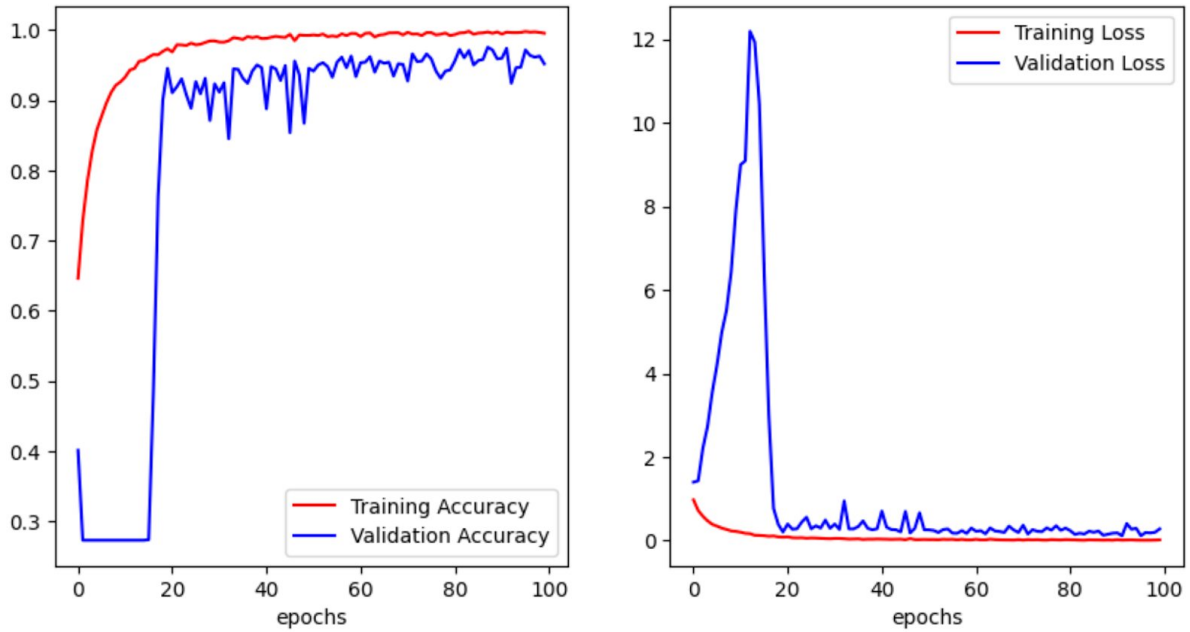


Figure 17: Training vs Validation Accuracy for MobileNetV2

NASNetMobile: The plot shows an outstanding case serving early overfitting which follows an eventual recovery after main training regime change. Moreover, training accuracy climbs smoothly to $\sim 100\%$ with training loss approaching zero which shows the model easily memorizes the training set. On the other hand validation accuracy plunges from ~ 0.55 at epoch 1 to ~ 0.27 and remains flat while validation loss explodes to 15-19 roughly epochs 5-25, meaning model's representations are collapsing from unseen data - due to high learning rate, insufficient regularization, or an abruptly unfrozen backbone causing unstable updates. Around epoch 25-30, validation loss rapidly drops to < 0.5 and validation jumps to 0.9 - 0.97 range and it remains high with small oscillation through epoch 100, suggesting a later schedule change (e.g. reducing learning rate, unfreezing with smaller LR, stronger augmentation, or early stopping patience) stabilized optimization and restored generalization. Primarily we should stop first 20 -25 epochs starting with a stable settings that produced the post -30 behaviour lower initial LR, gradual unfreezing and stronger regularization and apply LR cosine decay and also monitor validation metrics to trigger early rollback or early stopping when loss spikes to preventing damaging overtraining.

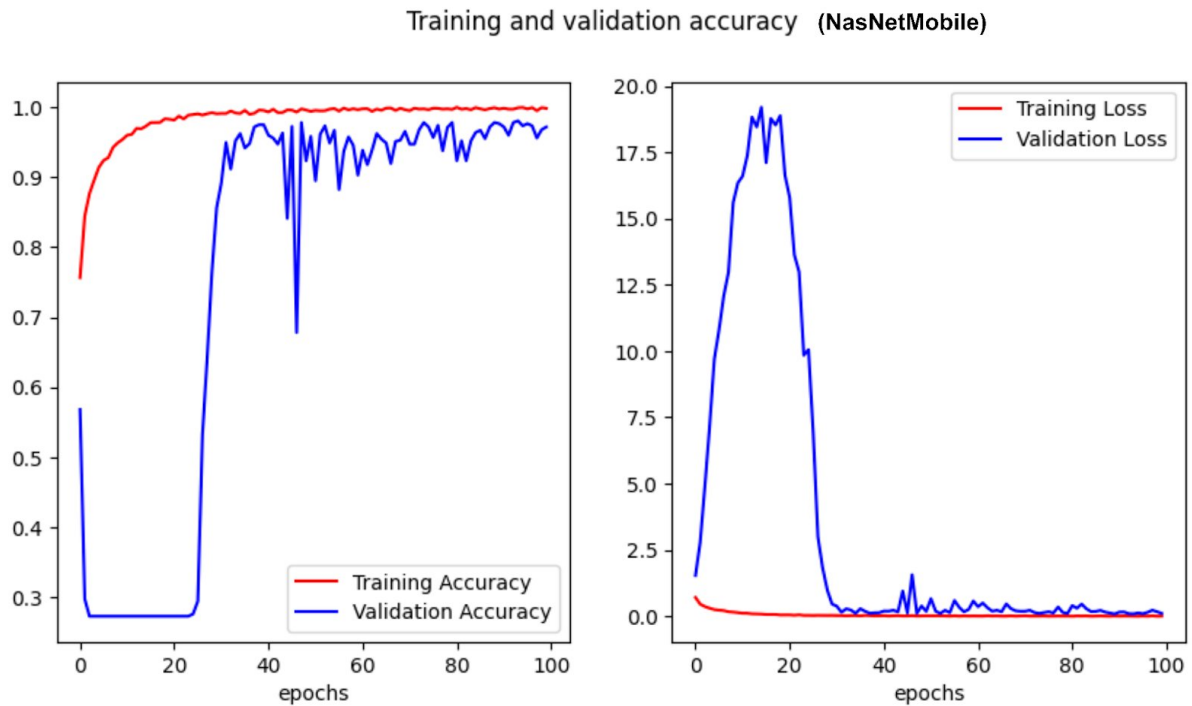


Figure 18: Training vs Validation Accuracy for NasNetMobile

ResNet50: In this plot we see a mild case of overfitting with a good performance overall. The accuracy of training quickly reaches nearly 100% and training loss shattered to zero. Validation remains high which is around 96% - 98%. But validation loss is higher making it unstable and sometimes having spikes. Which states that models sometimes make mistakes even having average accuracy. After the early epochs the gap between training and validation results grows which shows the model is learning more about the given training data but not improving it much. So we could stop training earlier when the validation loss first stopped improving. We have to add stronger regulation for example, augmentation, weight decay, dropout or mixup. Adjusting the learning rate schedule, having models resembling, and applying test time augmentation could help smooth out validation performance.

Training and validation accuracy (ResNet50)

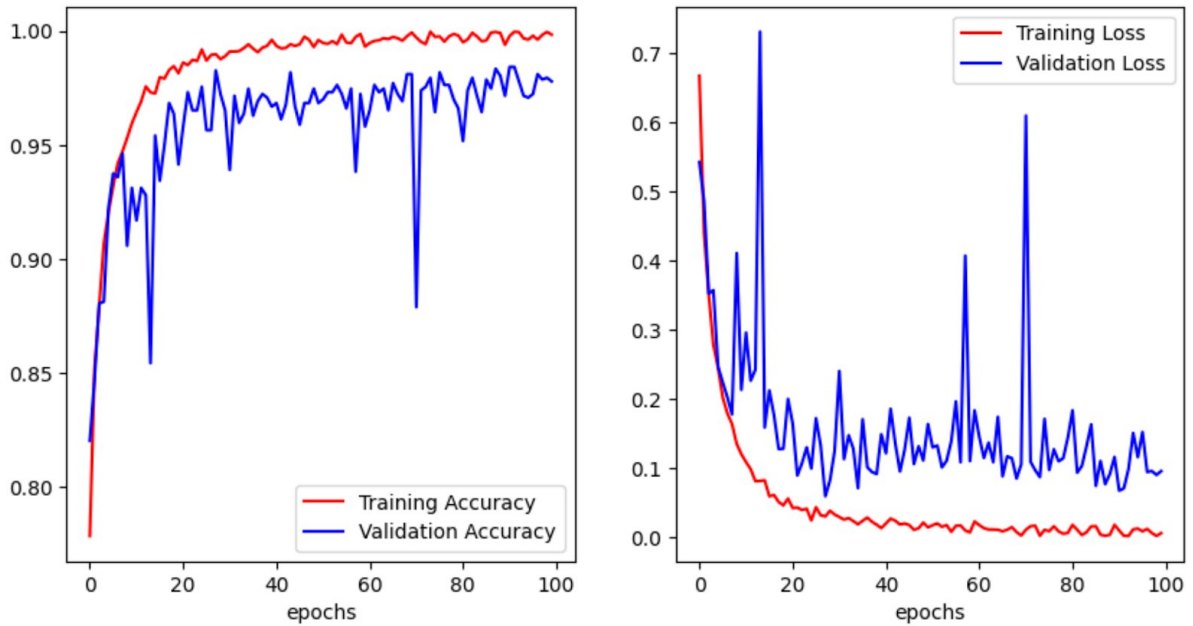


Figure 19: Training vs Validation Accuracy for ResNet50

ResNet50V2: In this model the findings are almost perfectly accurate, having an accuracy of 99-100% with a loss of almost zero. The validation accuracy stays at 95%, and it fluctuates when validation loss stays above, which shows spikes sometimes. The patterns show mild overfitting and instability, mainly from learning rate or batch effects and limited generalization capacity. One of the best approaches is to stop training earlier and add regularization. Also adjust the learning rate or batch size or collect more varied data. Moreover, extra epochs will increase overfitting without improving validation performance.

Training and validation accuracy (ResNet50V2)

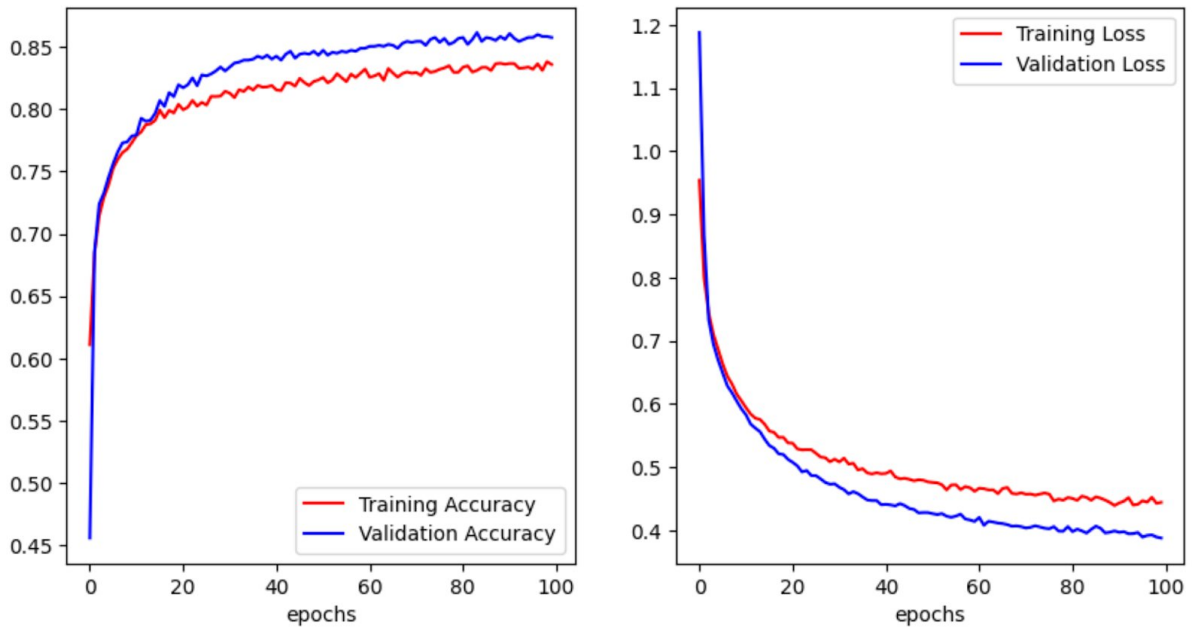


Figure 20: Training vs Validation Accuracy for ResNet50V2

ResNet101: This model has a good learning capacity, having perfect training capacity accuracy (~99%) with low training loss. The improvement of validation accuracy improves steadily up to 98% and follows training accuracy closely, by which the model shows the data is quite overfitted. But the validation shows loss is quite unstable and slightly over spikes. This happens because of an imbalance in the validation set. The learning rate is high at some points and has mismatched data augmentation. Even with these spikes, the overall validation is going down. This model is high-performing and nearly converged, which can be stopped when the validation accuracy stops improving.

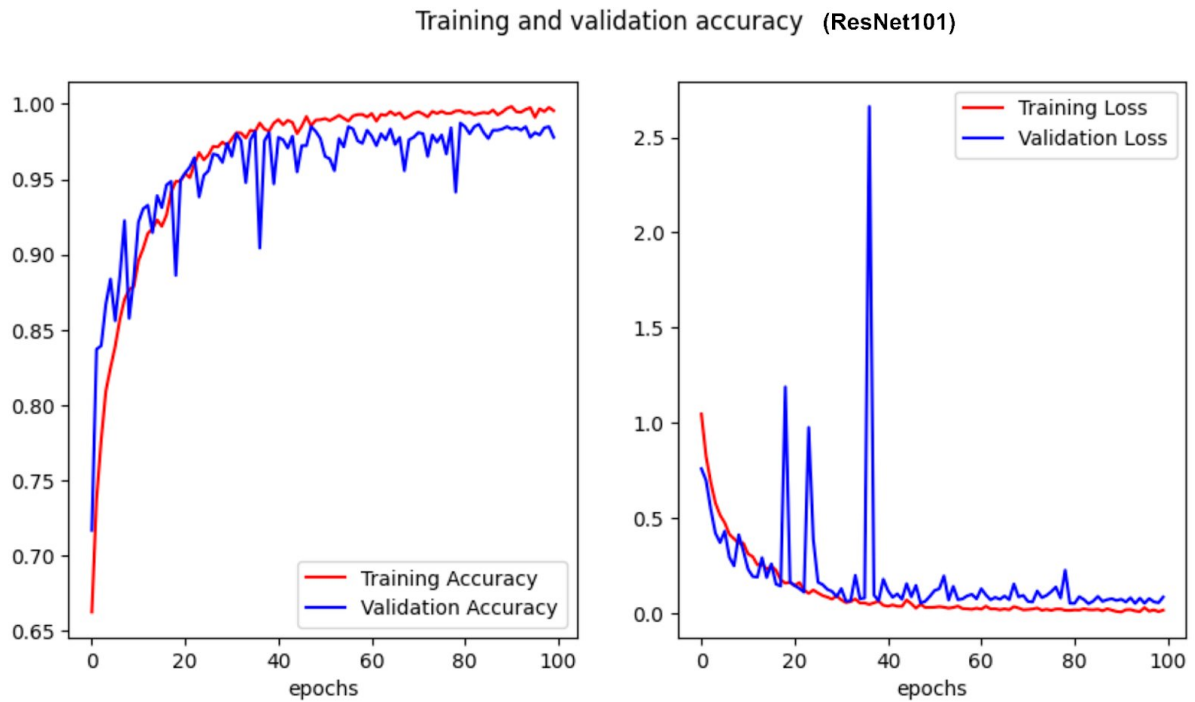


Figure 21: Training vs Validation Accuracy for ResNet101

ResNet101V2: This model training curve shows that it gives the data perfectly, reaching almost ~99-100%. But the validation accuracy is 95% and has fluctuations. Moreover, the validation loss is higher than the training loss, showing sharp spikes. This shows the model is overfitting slightly and not improving much with getting extra time . It gets better to stop training earlier when validation loss first reaches its minimum, adding extra regulation to adjust the learning rate to reduce instability and arranging more data that is diverse to improve generalization.

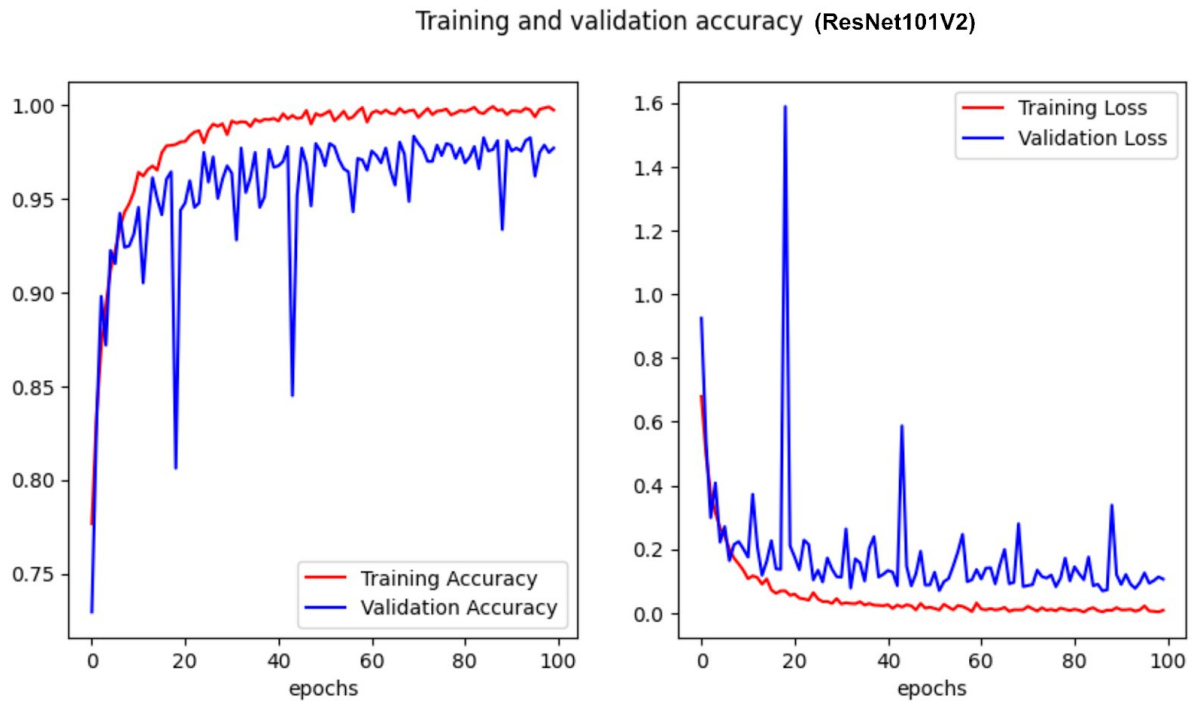


Figure 22: Training vs Validation Accuracy for ResNet101V2

VGG16: The training curve shows the learning of the training is almost perfect, with the accuracy reaching ~100% and loss near zero. It increased the validation within the 90s, but validation loss is unstable with many spikes even as its trend is downward. The model is overfitting, performing very well on training data but showing instability, which shows unseen data. The noisy validation loss could be caused by a small or imbalanced validation set, poor shuffling, noisy data, or a learning rate that is too high. By using stronger regulation, we can improve the stability and generalization. And also can add more data augmentation and refined data splitting and shuffling. We can lower the learning rate and also use early stopping and fine-tune fewer layers of VGG16 instead of the whole training.

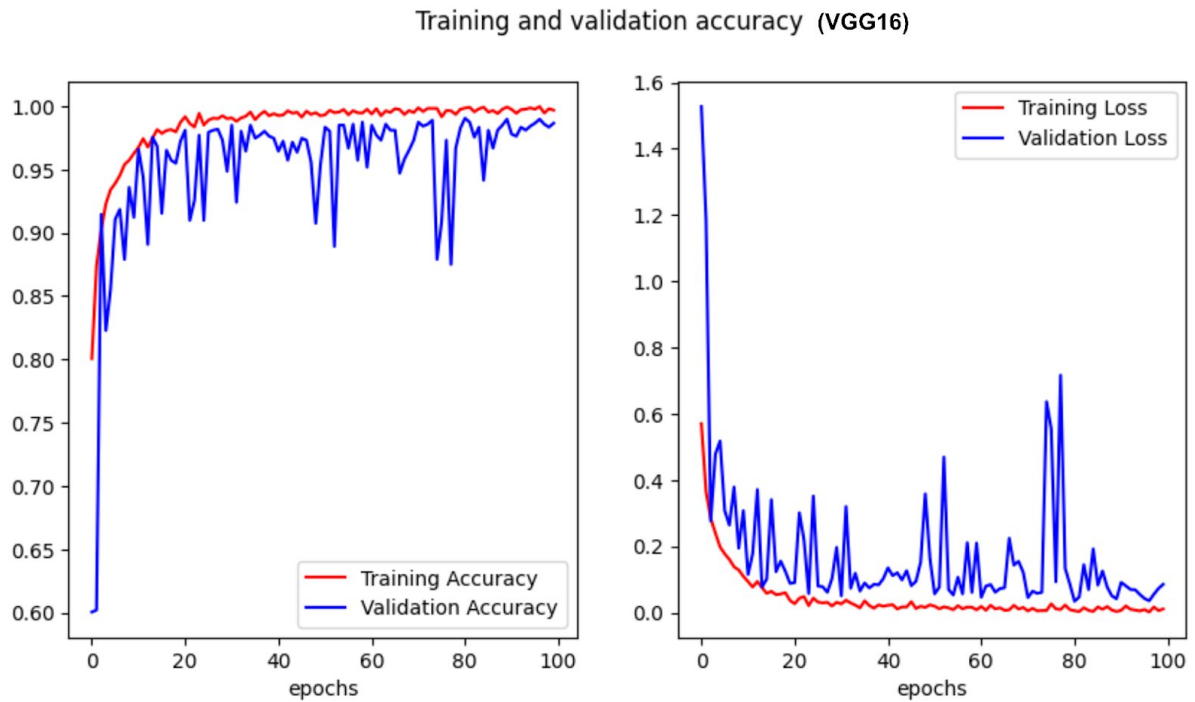


Figure 23: Training vs Validation Accuracy for VGG16

VGG19: The model shows how quickly it reaches very high training accuracy (>99%)? Having almost zero training loss, but the validation accuracy after some early fluctuations settles in the mid to high 90s, while the validation loss is higher and has spikes. The model shows the training data is very tight but has become very overfit with time due to its large size and many parameters. The validation and training gaps show a loss of reflection, which has overfitting and sensitivity in mini-batches. In order to improve generalization, we had to increase regularization. Use strong data augmentation and stop training once validation loses its improvement. Simplifying the phone-connected head and loading the learning rate in the training or applying test-time augmentation can also help reduce variance. VGG19, while stronger in transfer learning, has heavy memory and compilation compared to newer, more efficient models.

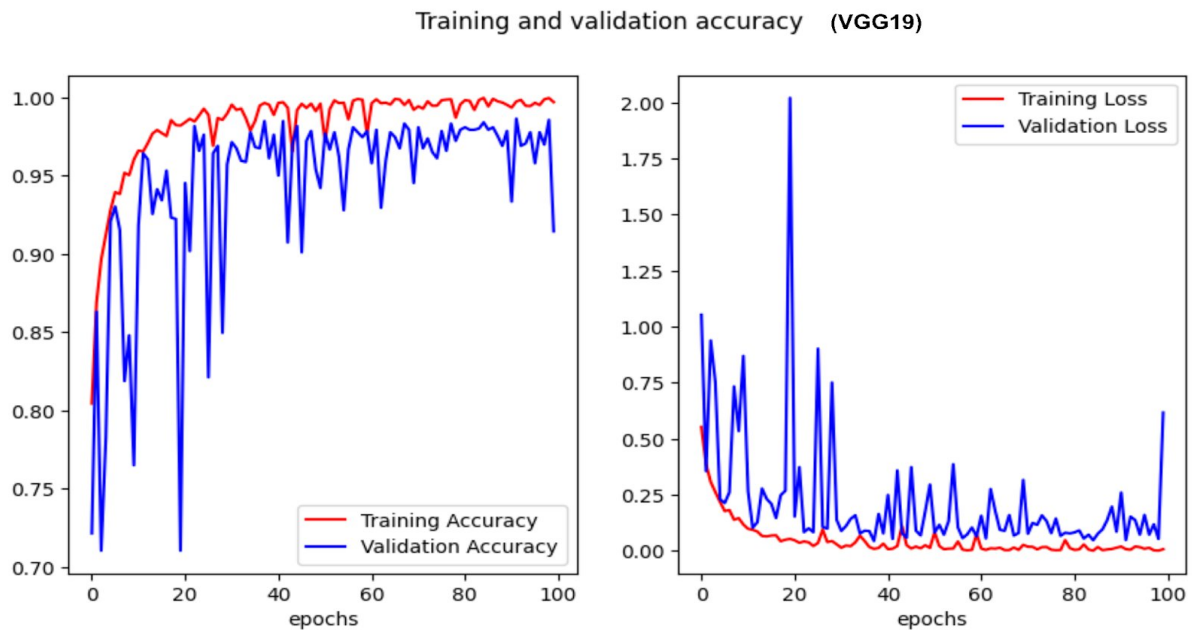


Figure 24: Training vs Validation Accuracy for VGG19

Xception: For this model, showing good training effectively. Training accuracy quickly reaches ~100%, and validation stability stabilizes, which takes 96-99%, having strong generalization and having some small fluctuations. The training loss is almost zero, and the validation loss, which is a bit noisy and has a few bumps, stays low. Spikes have batch effects or a small validation set. To make it more robust, we can add dropout or data augmentation, use a larger validation set, or level off validation loss. But in one word, the validation performed well after deployment.

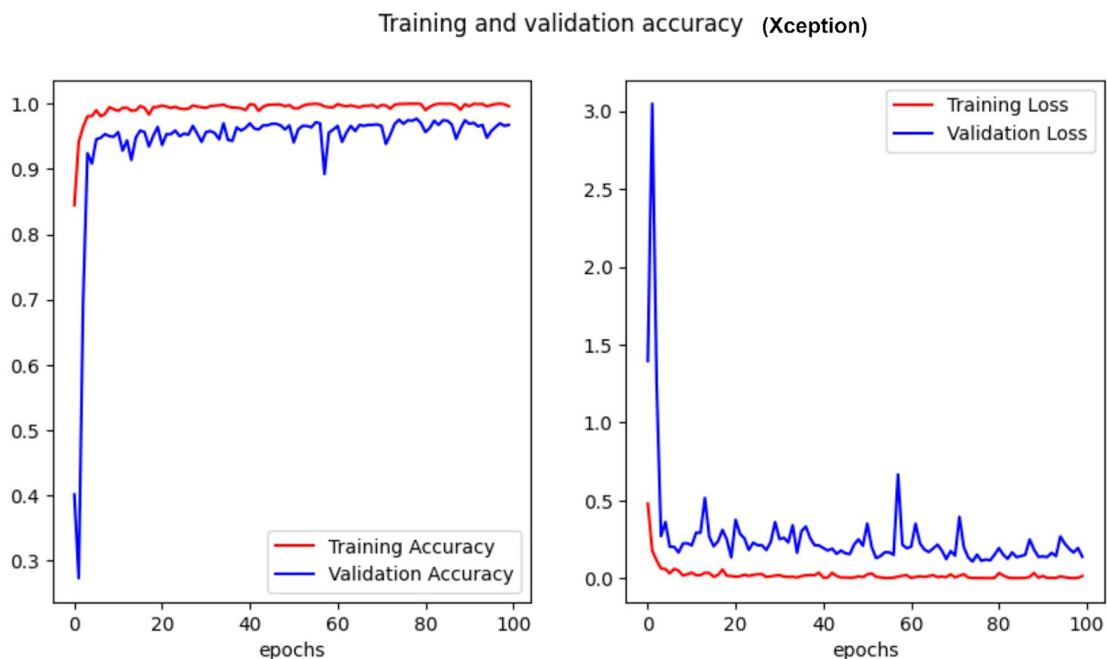


Figure 25: Training vs Validation Accuracy for Xception

With this our extensive synthetic load testing concludes. We observed that training CNN based classifiers is a unique workload by that we mean it is so different from any other synthetic benchmark that it cannot be replaced by just any other method. We also found out that the accuracy and time complexity does not have a direct correlation. Rather the model takes more time to train based only on the amounts of trainable parameters and number of layers. The sheer mathematical size of each model and the number of trainable parameters contribute to train and inference time regardless of accuracy. Whereas object detection requires time based on the size also but it can achieve higher levels of accuracy in less time (iterations or training cycles). But real time inference is solely dependent on hardware. The effect of architectural difference will be clear as day once we move into object detection training and testing in later sections of this report. To conclude the accuracy of a model does not linearly scale with the amount of resources allocated during training rather due to fundamental differences object detection can achieve the same or greater levels of accuracy in the same time (or less depending on careful selection of training preconditions) on the same hardware.

4.2.4.1.2. Average Power Draw

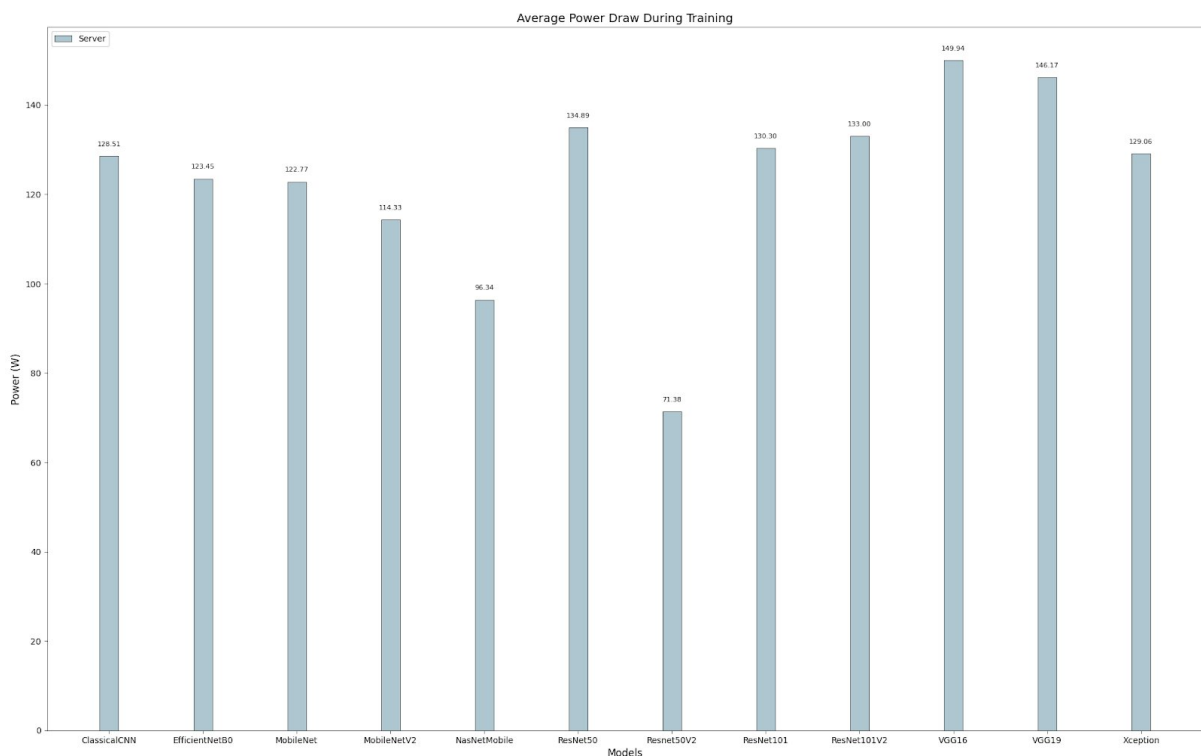


Figure 26: Average Power Draw During Training for All 12 Models

In the above Bar Chart, we have tried to train with different types of models starting from Classic CNN, EfficientNetB0, MobileNet and many others. In total, there are 12 models that we have trained; among those models, different models draw different types of power, if we

carefully visualize the graph we would be able to see that in the Y axis of the graph highlights the power bar where the VGG16 took the highest power about 149.94 W after that VGG19 took 146.17 W later on the ResNet50 took 134.89 W. These three took the highest power from twelve models and also if we compare about from lesser the ResNet50V2 took the least power of 71.38W. Overall, this graph gives us an idea about the average power draw of all twelve models during training. Thus average power draw, the VGG16 has the highest power, which is 149.94 W, which is not acceptable. But if we look at ResNet50V2, it has the lowest power, which is 71.38 W, which is more accurate than VGG16.

4.2.4.1.3. Peak Power Draw

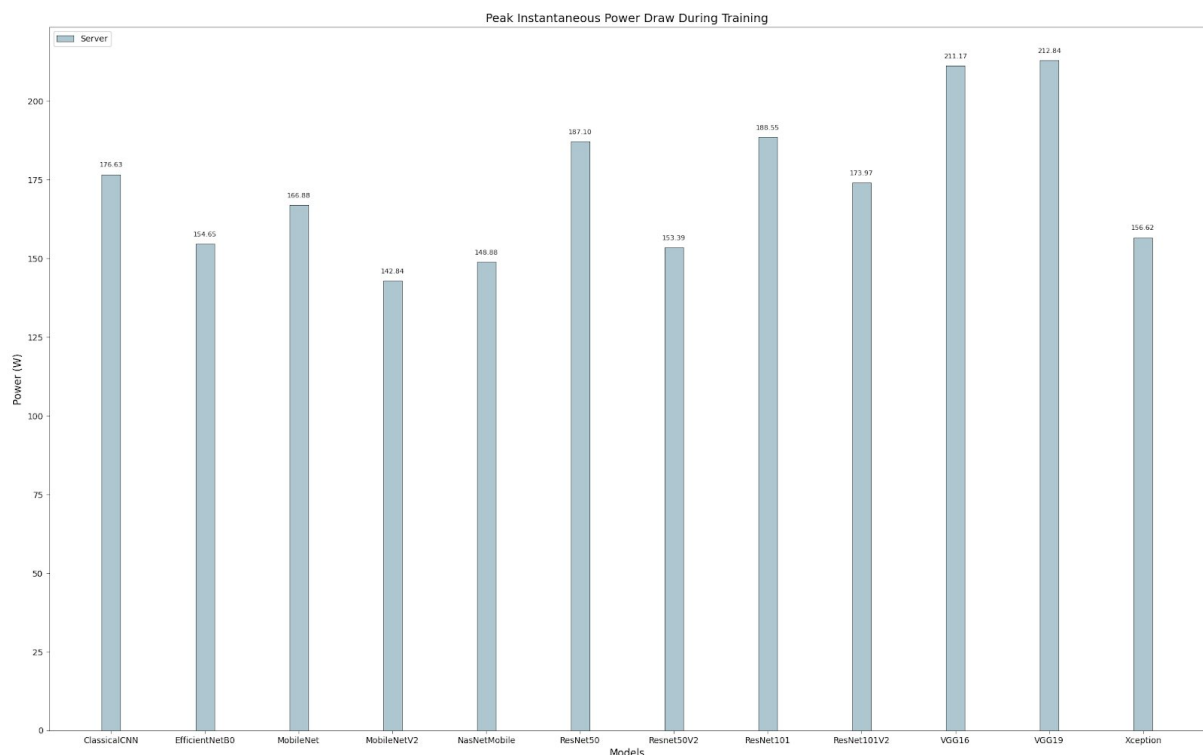


Figure 27: Peak Power Draw During Training for All 12 Models

After that this bar chart shows about the peak instantaneous power drawing during the training of the model where highlighting the X axis with the names of the 12 models and on the Y axis highlighting the corresponding power draw values during the training and if we watch a bit closely we will see that VGG19 drawing the highest amount of peak instantaneous power draw with a rate of 212.84 W during the training and after that the VGG16 creating a closer one with a rate of 211.17 W. Comparing the least value among these 12 models we will be able to find that MobileNetV2 is taking the lesser value among the all, which is 142.84W. All the value differences are not showing very high differences between them. Overall, with this bar chart, we can know about the peak instantaneous power drawn during the training. As a result of the power draw, the maximum is VGG19, which is 212.84W, and the minimum is MobileNetV2, which is 142.84 W. So the MobileNetV2 is performing better.

4.2.4.1.4. Energy Consumption During Training

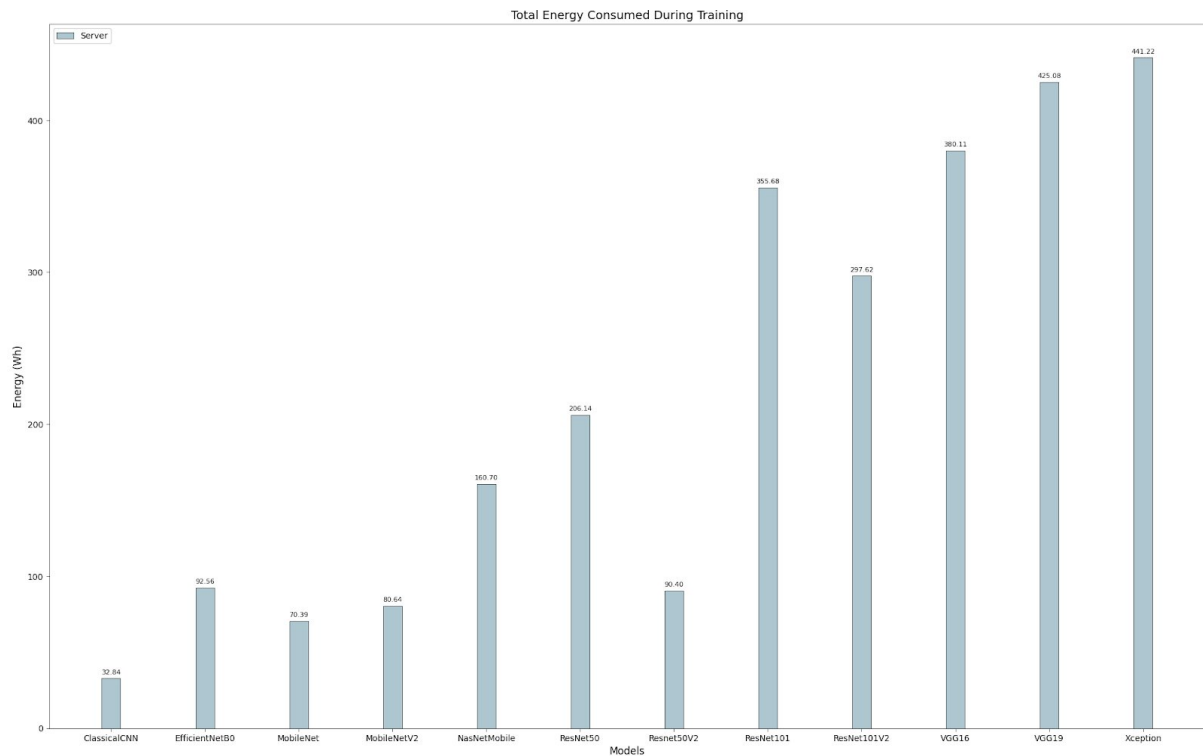


Figure 28: Total Energy Consumption During Training for All 12 Models

Now, if we look at this bar chart, we will be able to find the total energy consumed for those 12 models during the training. For training the epochs, along with the bar chart, we can see which model took the least energy and which one took the most. In the bar chart, in the X axis, we can find the model name, and in the Y axis, we can see the corresponding energy consumption value for all those models. From the start, the ClassicalCNN took the least amount of time among all the models, with 32.84 Wh, and later, the MobileNet took 70.39 Wh during training. However, if we examine the chart, we can also see that the Xception model took the longest to train, approximately 441.22Wh. In conclusion, this chart provides an overview of the power consumption of the different models during the training. The lowest amount of energy required to achieve a goal is the idea.

4.2.4.1.5. Time Required Per Iteration

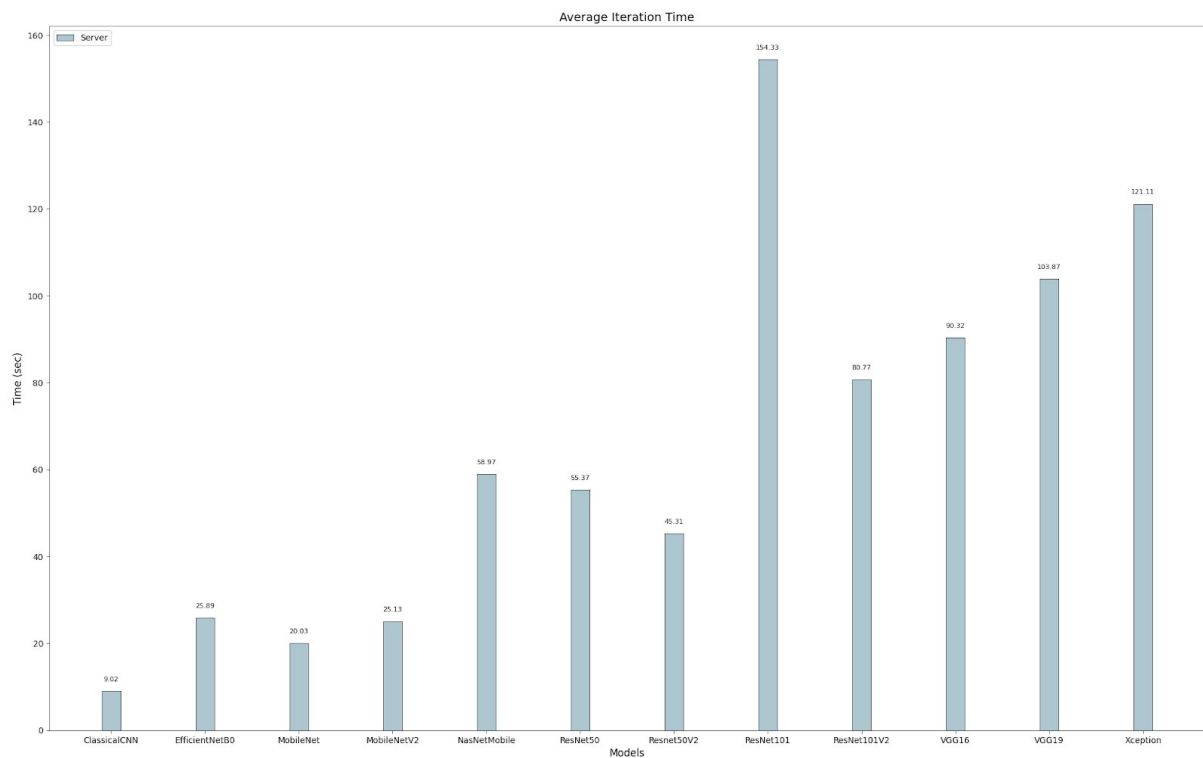


Figure 29: Time Required Per Iteration for All 12 Models

Here is the chart of the average iteration time needed for the training of the twelve different models, where the X axis highlights the model names and the Y axis highlights the corresponding time (seconds) required for the iteration. Among the twelve models, the ResNet101 took the highest amount of time, 154.33 seconds. After that, Xception took 121.11 seconds. If we look at the chart from the lesser values, the Classic CNN will be focused as a less time-taker, with 9.02 seconds. Overall, this chart gives us an idea of the average time taken for the iteration of all twelve models. So in the case of time required per iteration, the maximum is ResNet101, having 154.33 seconds, and the minimum is ClassicalCNN having 9.02 seconds. Here the ClassicalCNN is performing better.

4.2.4.1.6. Time Required Per Step

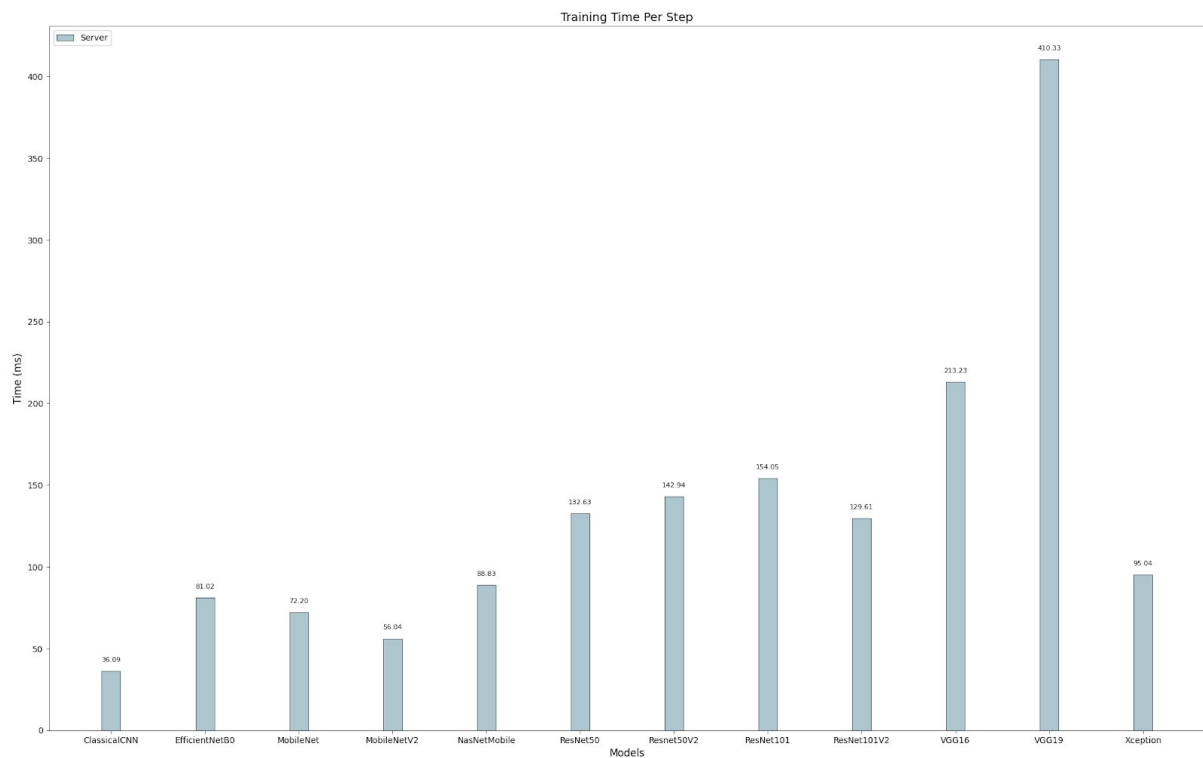


Figure 30: Time Required Per Step for All 12 Models

Now, if we look at this chart, it shows us a visual representation of the training time for each step for the twelve models, where on the x-axis, all twelve models are highlighted, and on the y-axis, the required time for the training is showcased. In the chart, the highest amount of time is taken by VGG19, about 410.33 milliseconds; this one has taken the most time compared to the others. After that, 213.23 milliseconds were taken by the VGG16. Later on, comparing the least, the Classic CNN model took about 36.09 milliseconds, which is the lowest amount among all twelve models. Overall, this bar chart provides a clear view of the required training time for each step across all twelve models. Ultimately when we look at the time required per step, the VGG19 takes the most time, 410.33 seconds. The minimum is the ClassicalCNN which is 36.09 seconds. Here ClassicalCNN is doing better than others.

4.2.4.1.7. Total Training Time

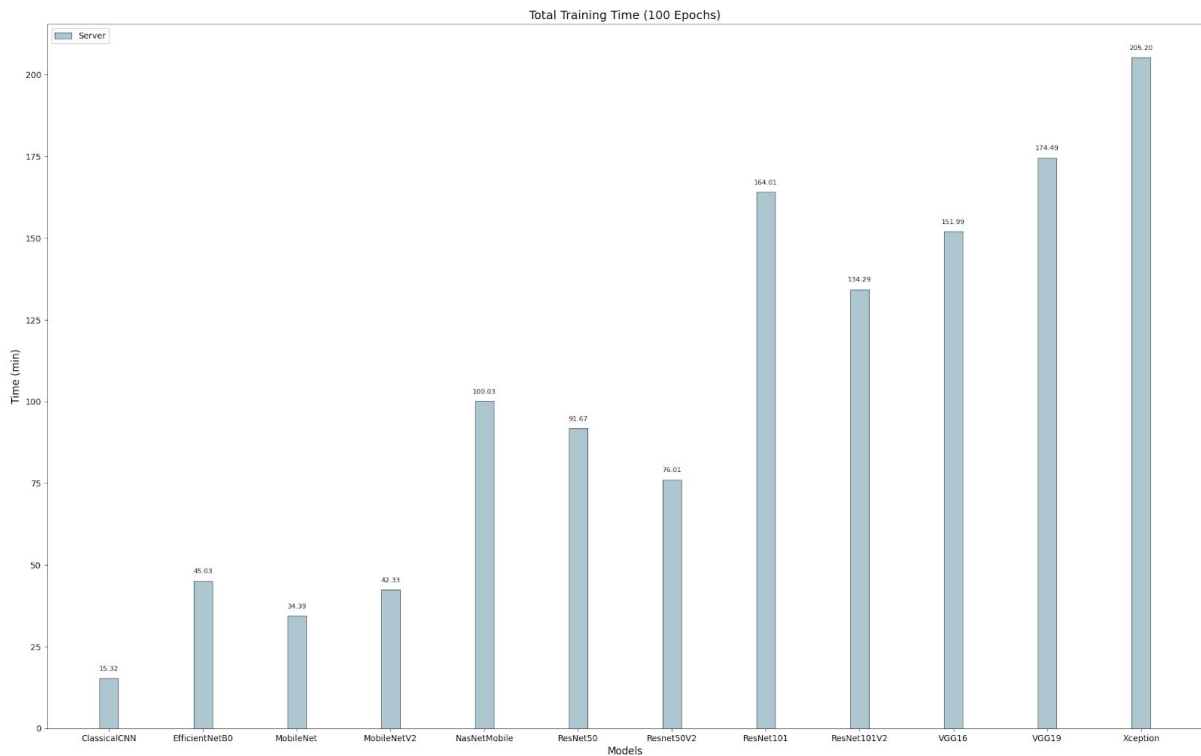


Figure 31: Total Time to Train All 12 Models for 100 Epochs

Now, if we look at this bar chart, we will be able to find the total training time for those 12 models on 100 epochs. For 100 epochs, along with the bar chart, we can see which model took the least and which one took the most. From the start, the ClassicalCNN took the least amount of time among all the models, with 15.32 minutes, and later, the MobileNet took 34.39 Minutes to train 100 epochs. However, if we examine the chart, we can also see that the Xception model took the longest to train, approximately 200.20 minutes, which is equivalent to about 4 hours. In conclusion, this chart gives us the whole idea of the training times of the twelve models for 100 epochs. Thus in case of total training time the maximum is the Xception having 205.20 seconds and the minimum is ClassicalCNN which is 15.32 seconds. Here the Classical CNN has taken the lead.

4.2.4.1.8. Power Draw Profile

ClassicalCNN: During training the ClassicalCNN power draw profile peaked at 176.63W while consuming 32.389Wh of energy. The peaks and valleys of the power curve can be explained by each epoch loading the hardware thus drawing more power. Since this model has the lowest number of trainable parameters (just over 111k) this resulted in a very short training run. Thus there is significantly less horizontal resolution in the power draw curve. The model was less than half a megabyte in size so each of the epochs were very pronounced as each time the images were loaded into VRAM thus spooling the GPU up and causing the sharp transitions. Now for the large peak at the beginning, our running theory is that the GPU heatsink was not thermally saturated thus during the first or so spikes it could pull a lot more power as

the heat was dissipated rapidly across the heatsink and exhausted from the device. Ultimately taking just 15 minutes to train this model might not be optimal for this application but with proper selection of training parameters and careful deployment strategies it can provide fast and accurate results at a fraction of the computational cost (compared to other models)

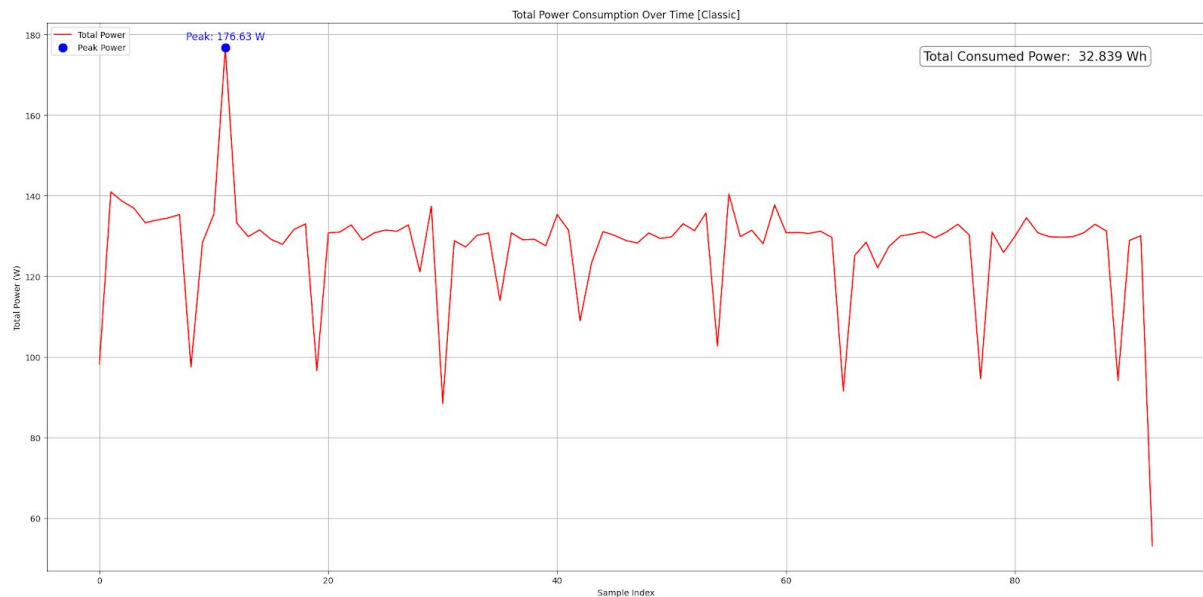


Figure 32: Power Draw Profile for ClassicCNN in x86-64 Hardware

EfficientNetB0: While training EfficientNetB0 the peak power draw was 154.65W with a total total energy consumption of 92.559Wh. Compared to the ClassicalCNN model it is massive, even though it is called efficient (well it is when we look at other models). With nearly 48 times more trainable parameters and 9 times the number of layers. Each layer adds to its capability of performing CV operations. It took four times longer to train. It is a testament of its efficiency.

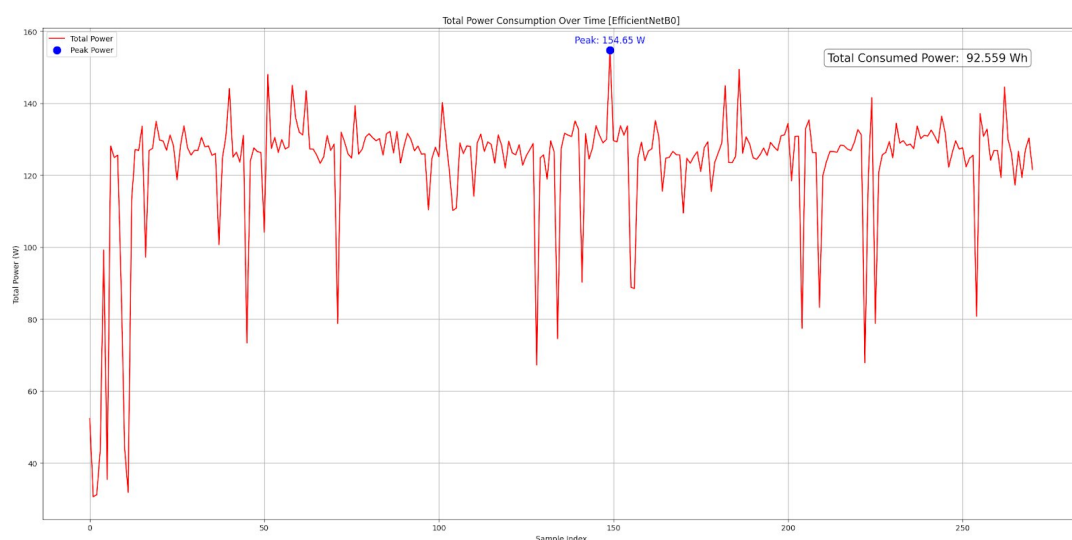


Figure 33: Power Draw Profile for EfficientNetB0 in x86-64 Hardware

MobileNet: This model was an oddity because it completed the whole 100 epoch training cycle while consuming 70.398Wh. With a peak power draw of 166.88W. The reason we called it the odd one out of all the models is because even after having only 42% of the number of layers compared to the previous model it has more than 81% of the number of trainable parameters. The fact that it is somehow 55% of the size of EfficientNetB0 is also quite odd.

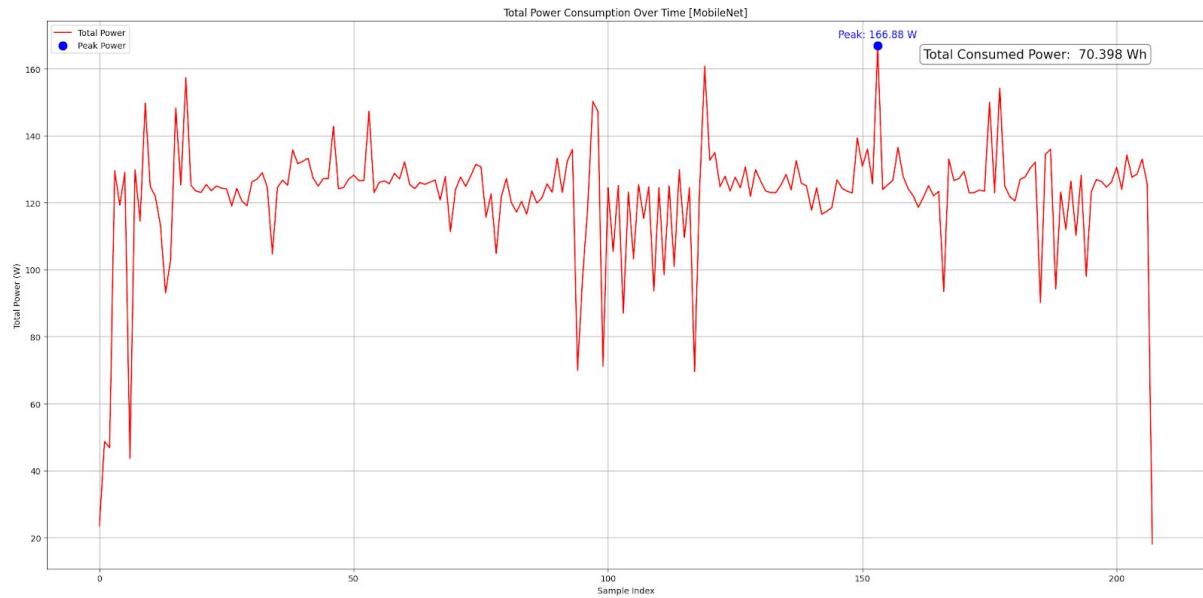


Figure 34: Power Draw Profile for MobileNet in x86-64 Hardware

MobileNetV2: Even though this is the second version of MobileNet, it is smaller in size, has fewer parameters and almost two times the amount of layers. During the 100 cycle training run it caused a peak power draw of 142.84W. The total power consumption is 80.643Wh.

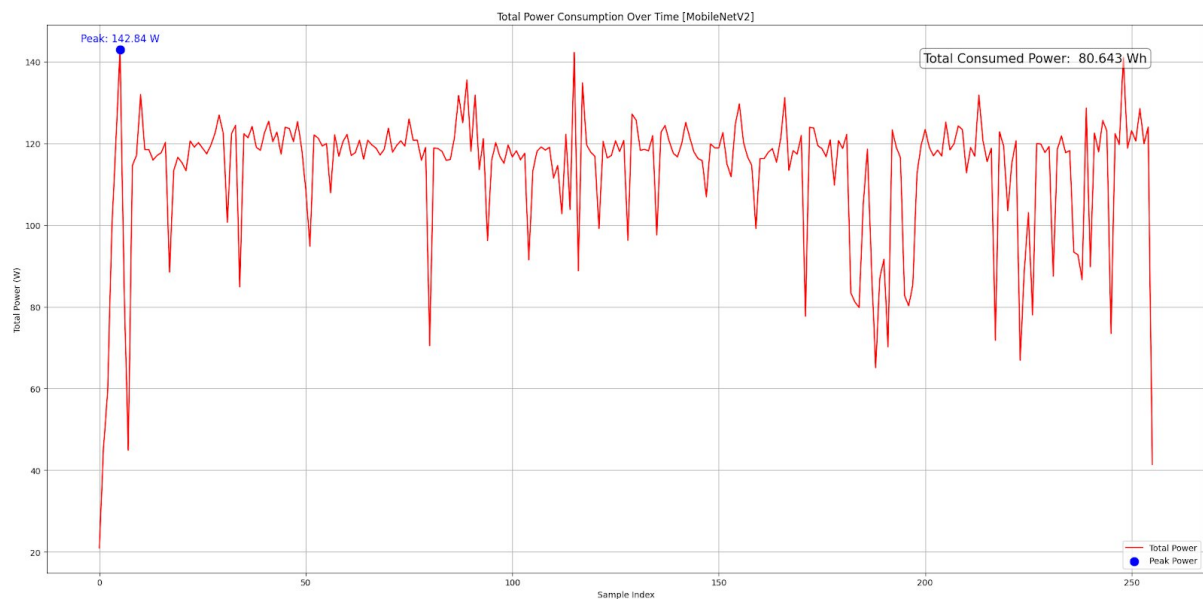


Figure 35: Power Draw Profile for MobileNetV2 in x86-64 Hardware

NasNetMobile: While training those models, this model has the highest number of layers among the 12 tests. But its size does not reflect where it is about 23 MB which is comparatively low regarding the other models size with a depth of 389 layers, though the distinct changes of the power consumption are visible in the graph below where for this model the peak is 148.88W. The total power consumption is 160.702Wh

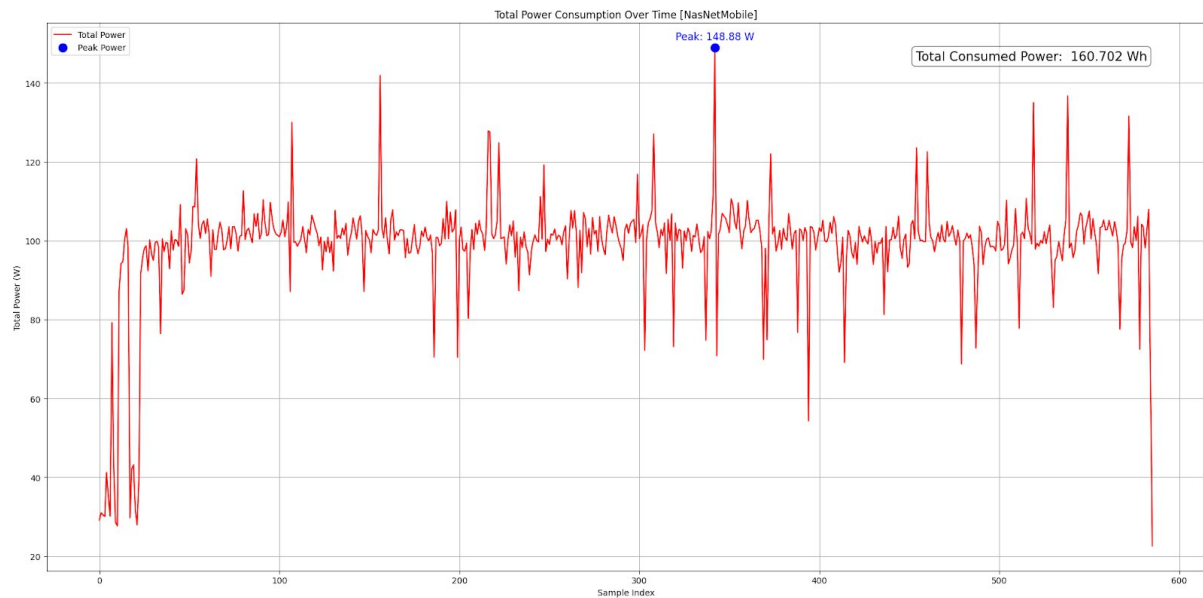


Figure 36: Power Draw Profile for NasNetMobile in x86-64 Hardware

ResNet50: After that, this model has not the highest number of layers among the 12 tests still its size does not reflect where it is about 25.6 MB which is comparatively low regarding the other models size with a depth of 107 layers, though the distinct changes of the power consumption are visible in the graph below where for this model the peak is 187.10W. The total power consumption is 206.142Wh.

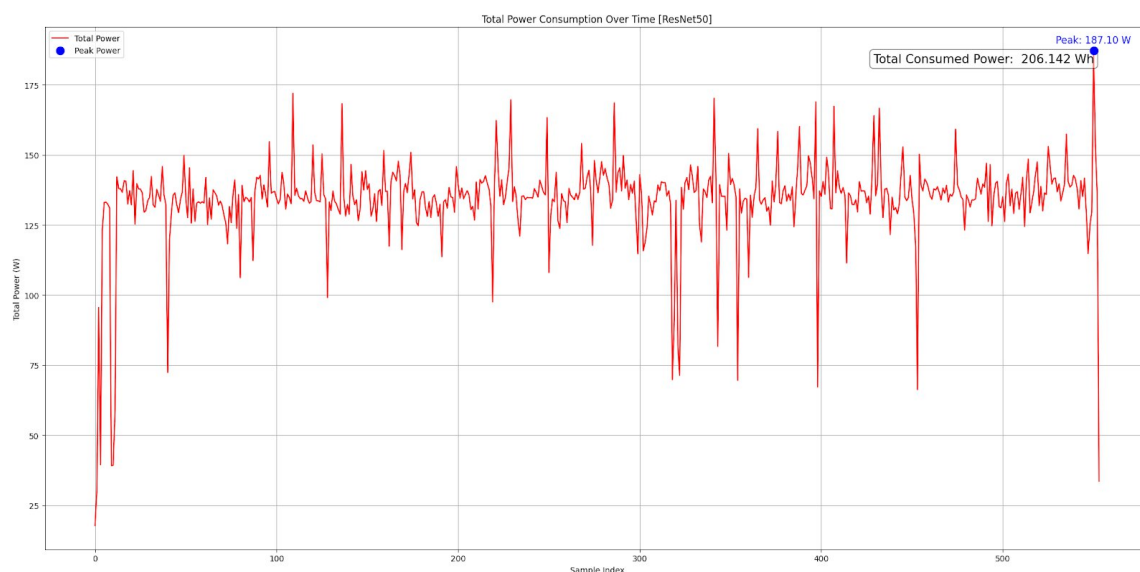


Figure 37: Power Draw Profile for ResNet50 in x86-64 Hardware

ResNet50V2: In training, peak power consumed is seen as 154.65W. Again, the total power consumption is 92.559Wh.

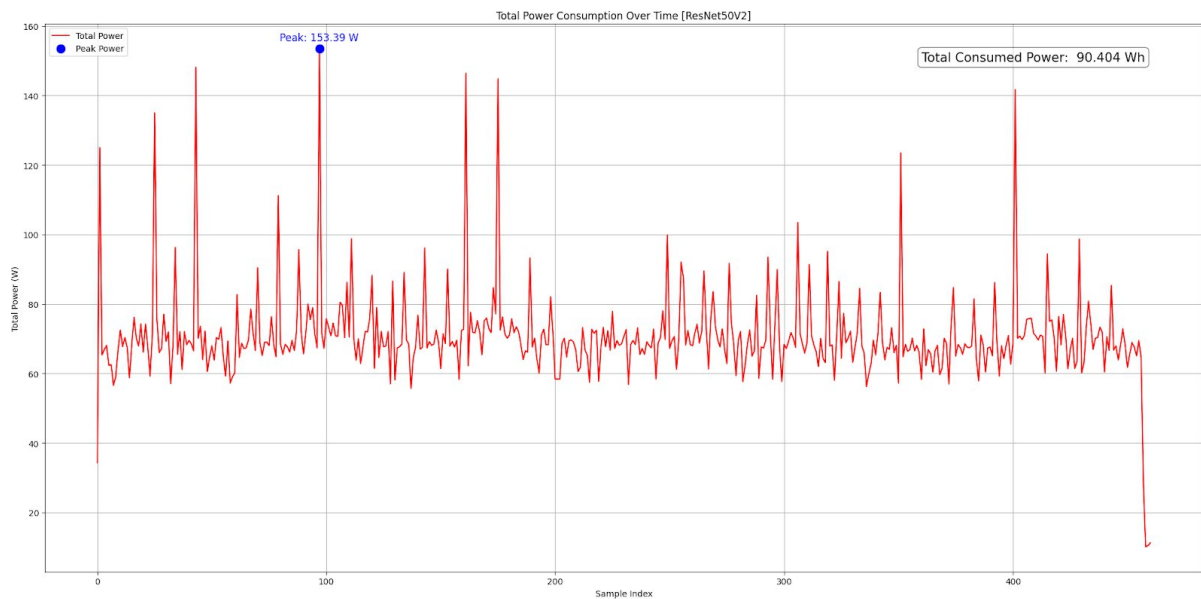


Figure 38: Power Draw Profile for ResNet50V2 in x86-64 Hardware

ResNet101: In this model the peak power consumption is 188.55W. During training we have seen that the total power consumption is 355.681Wh.

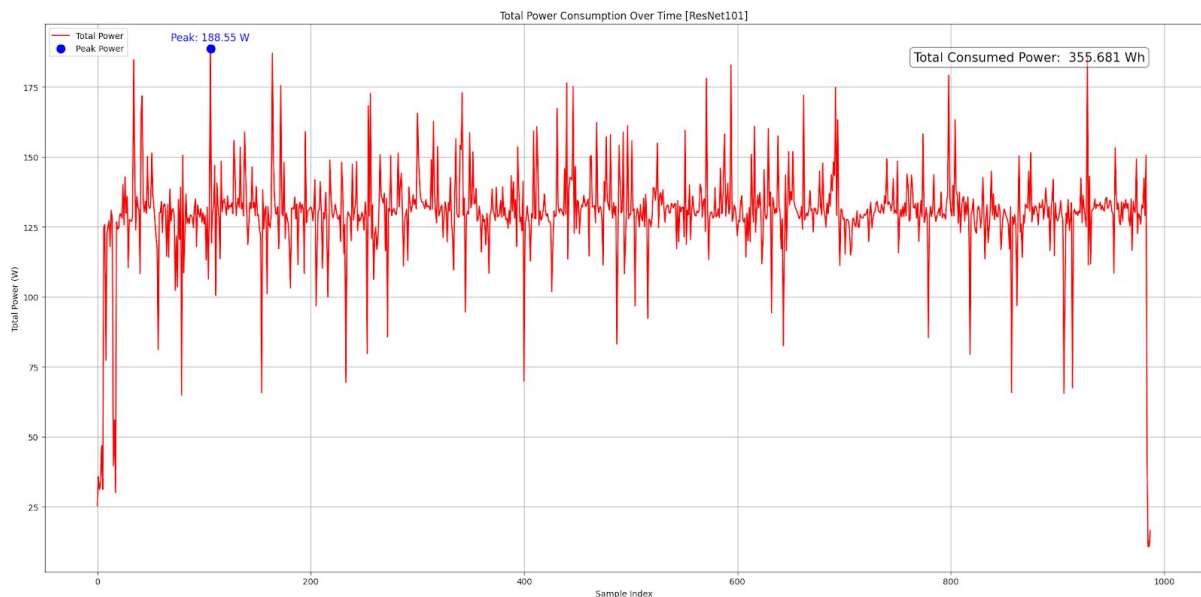


Figure 39: Power Draw Profile for ResNet101 in x86-64 Hardware

ResNet101V2: In this model the peak is 173.97W. The total power consumption is 297.623Wh.

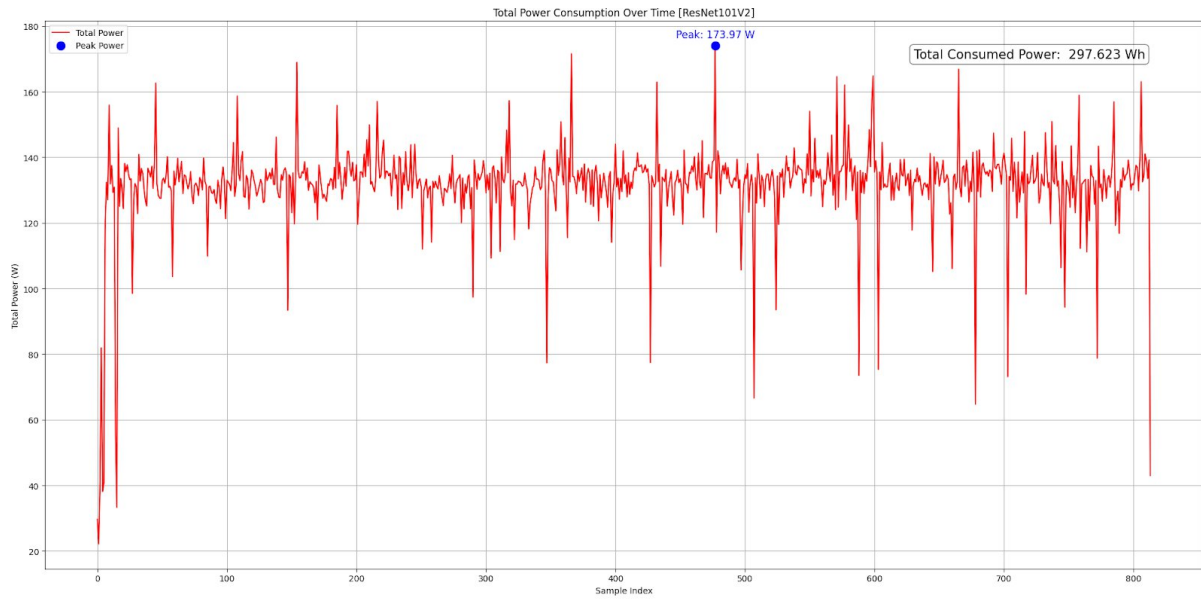


Figure 40: Power Draw Profile for ResNet101V2 in x86-64 Hardware

VGG16: In this model the peak is 211.17W. The total power consumption is 380.111Wh. The Custom CNN is the smallest and the simplest while VGG16/19 are very large and parameters heavy despite shallow depths. The ResNet family is a good balance, where Resnet50 is moderate and Resnet101 are the deeper and more complex ones.

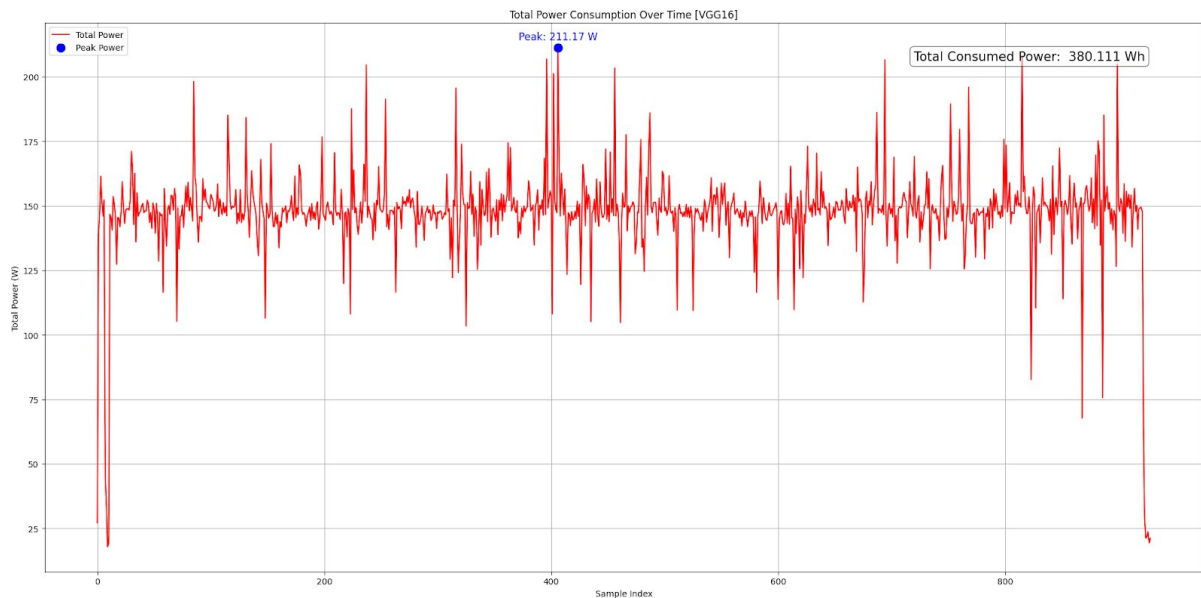


Figure 41: Power Draw Profile for VGG16 in x86-64 Hardware

VGG19: In this model the peak is 212.84W. The total power consumption is 425.082Wh.

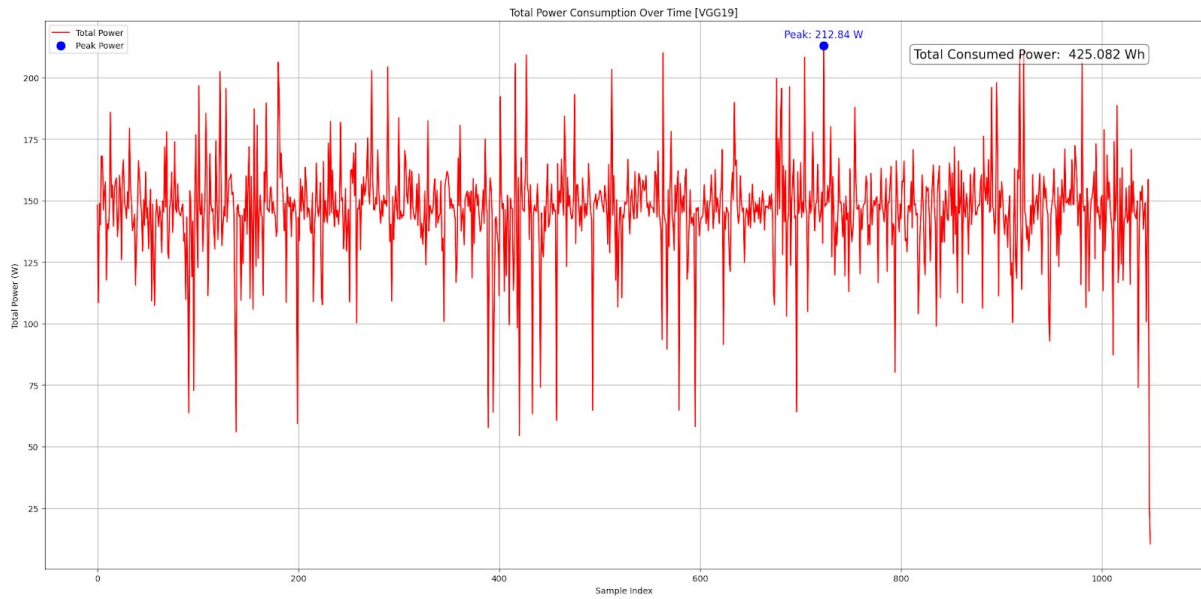


Figure 42: Power Draw Profile for VGG19 in x86-64 Hardware

Xception: In this model the peak is 156.62W. The total power consumption is 441.224Wh. Xception is more compact than VGG but still relatively heavy compared to lightweight models.

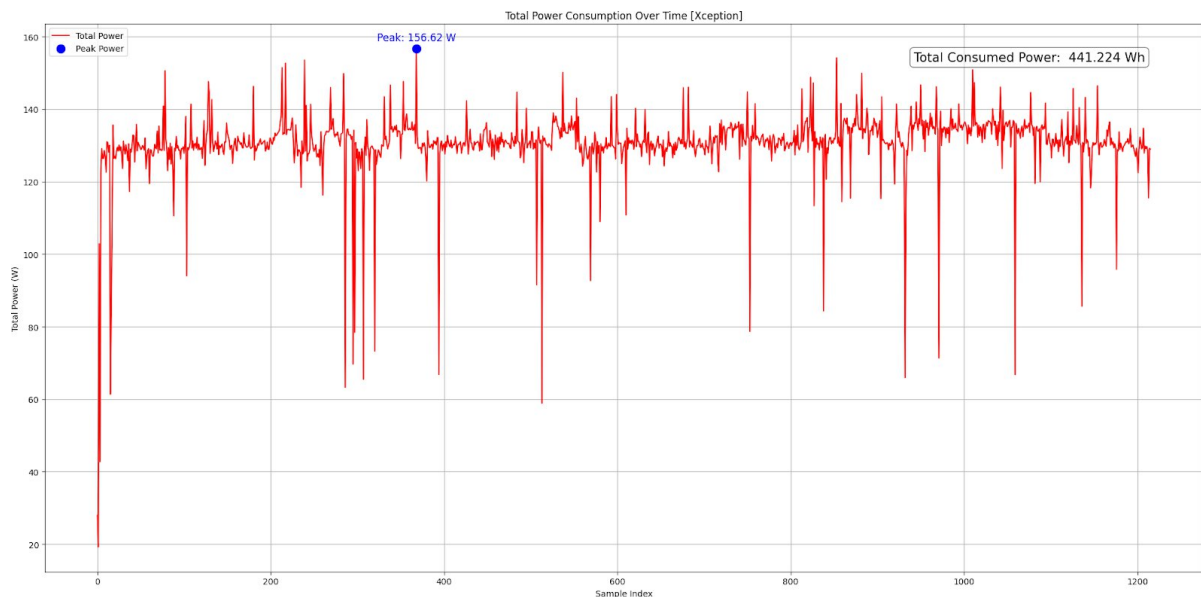


Figure 43: Power Draw Profile for Xception in x86-64 Hardware

We know the Custom CNN is the smallest and the simplest while VGG16/19 are very large and parameters heavy despite shallow depths. The ResNet family is a good balance, where Resnet50 is moderate and Resnet101 are the deeper and more complex ones. Xception is the middle, not as large as VGG, not as light as light models. MobileNet and MobileNetV2 are very compact and efficient. These are appropriate for resource-constrained devices. But

NASNetMobile and EfficientNetB0 represent modern designs that balance accuracy with efficiency.

4.2.4.2. Inference

4.2.4.2.1. Energy Consumption During Inference

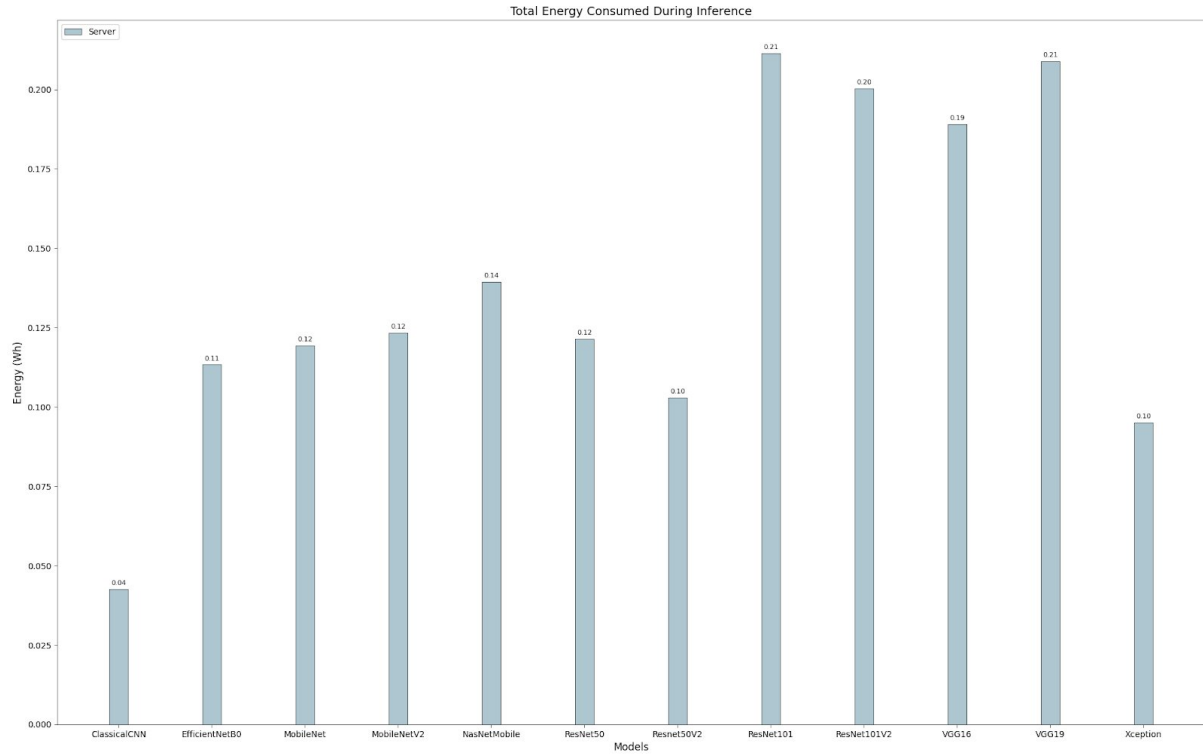


Figure 44: Total Energy Consumption During Inference for All 12 Models

In the case of energy consumption during the interface, the maximum consumption has been seen with ResNet101 and VGG19, having 0.21Wh, and the minimum consumption can be seen with ClassicalCNN, 0.04Wh. So we can say that ClassicalCNN has the best consumption rate.

4.2.4.2.2. Peak Power Draw

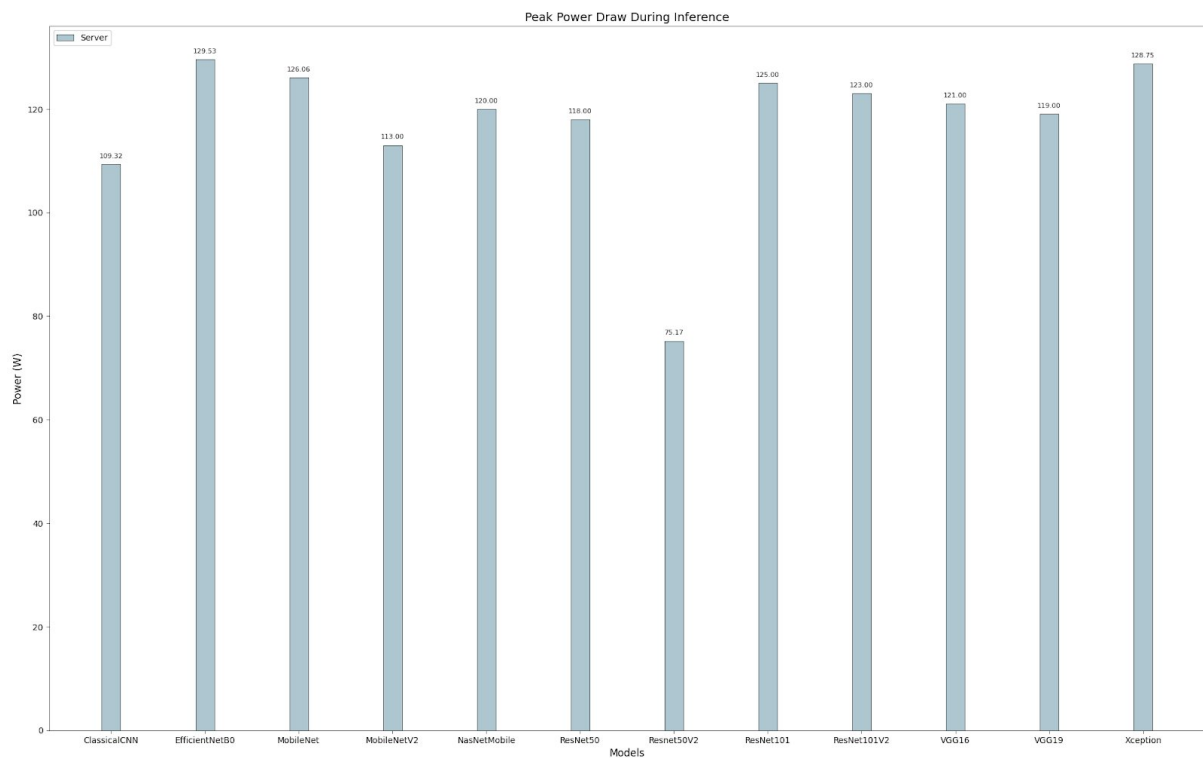


Figure 45: Peak Power Draw During Inference for All 12 Models

In the case of peak power draw we see that EfficientNetB0 has the max consumption (129.53W) and ResNet50V2 has the min consumption (75.17W). This clearly shows that ResNet50V2 is more energy-efficient at peak load time, since it requires substantially less power when compared to EfficientNetB0. Therefore, from the power efficiency standpoint, ResNet50V2 is better than EfficientNetB0 in this case.

4.2.4.2.3. Average Power Draw

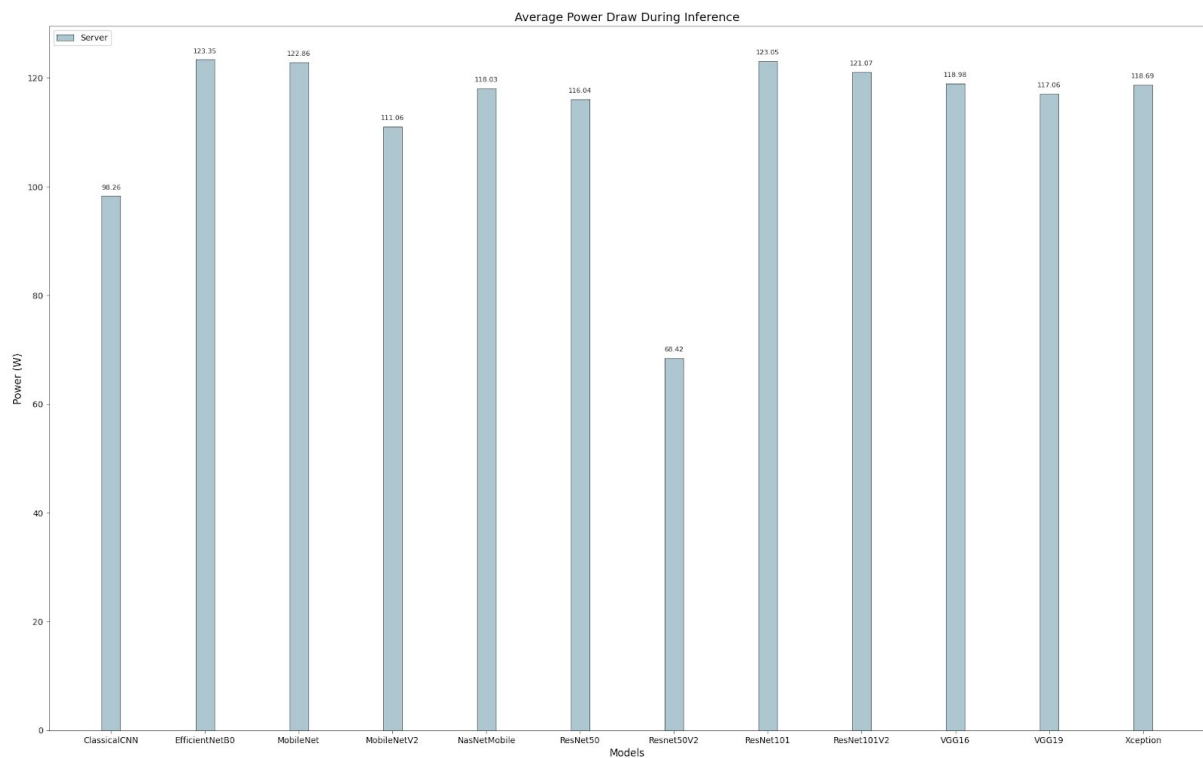


Figure 46: Average Power Draw During Inference for All 12 Models

ResNet50V2 is the most efficient, using only 68.42 W, while Classical CNN (98.26W) and MobileNetV2 (111.06W) also use less power compared to others. Most of the remaining models, like EfficientNetB0 (123.35W), MobileNet (122.86W), ResNet101 (123.05W), VGG16 (118.98W), and Xception (118.69W), consume much more around 117 to 123 W. In simple terms, smaller models are more power-friendly and better for low-energy or edge devices, while larger models use more power but may give higher accuracy.

4.2.4.2.4. Time to Process 1,000 Frames

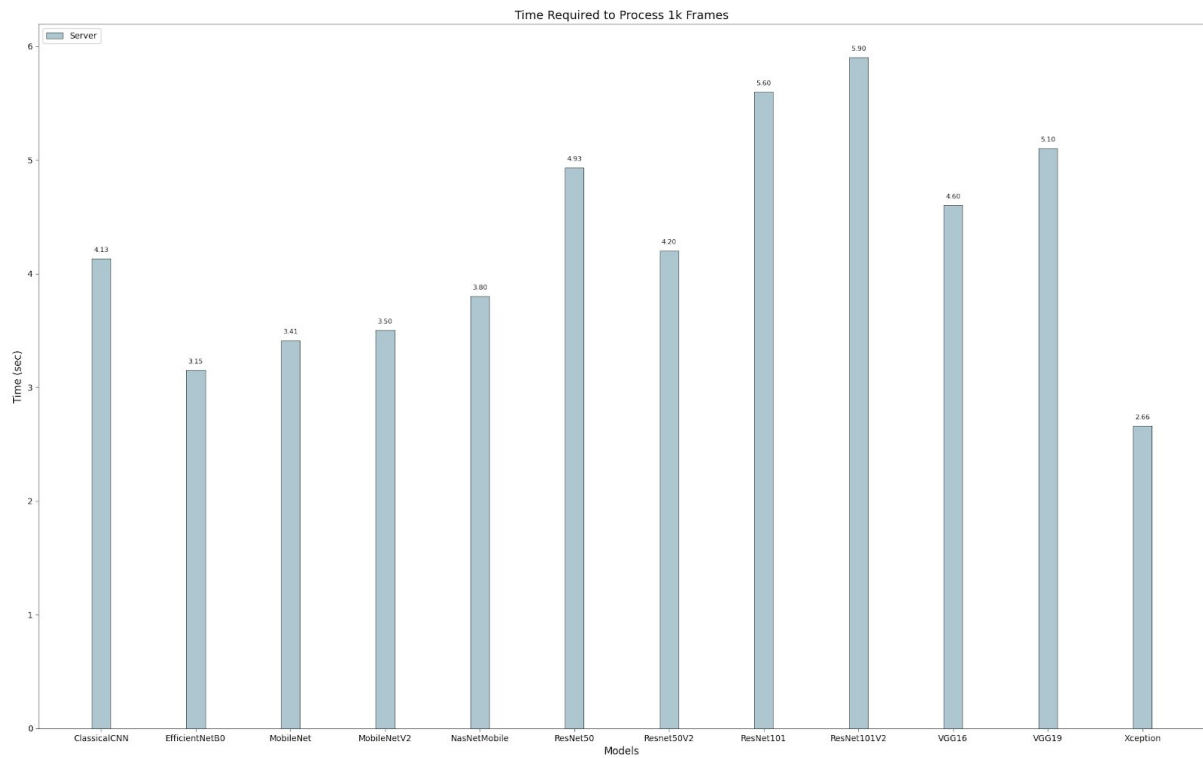


Figure 47: Time to Process 1k Frames for All 12 Models

In the case of the time required to process 1,000 frames, ResNet101V2 is the most time-consuming, because it takes about 5.90 seconds. The least time-consuming model is Xception, taking 2.66 seconds. So we can say that here Xception is doing better.

4.2.4.2.5. Time Required Per Frame

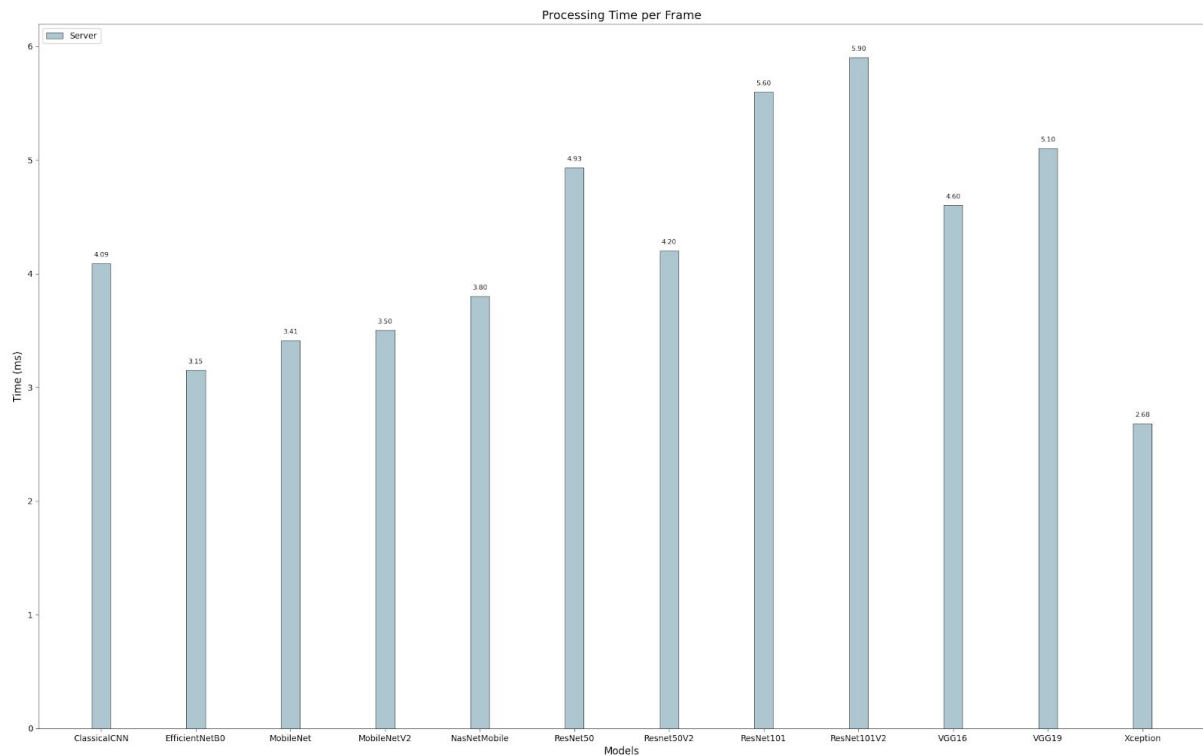


Figure 48: Time Required to Process One Frame for All 12 Models

In the case of time per frame, ResNet101V2 is the most time-consuming, taking about 5.90 seconds, and the model that is the least time-consuming is Xception, taking 2.68 seconds. Xception is doing better.

4.2.4.2.6. Video Stream Handling Capability (Theoretical Maximum)

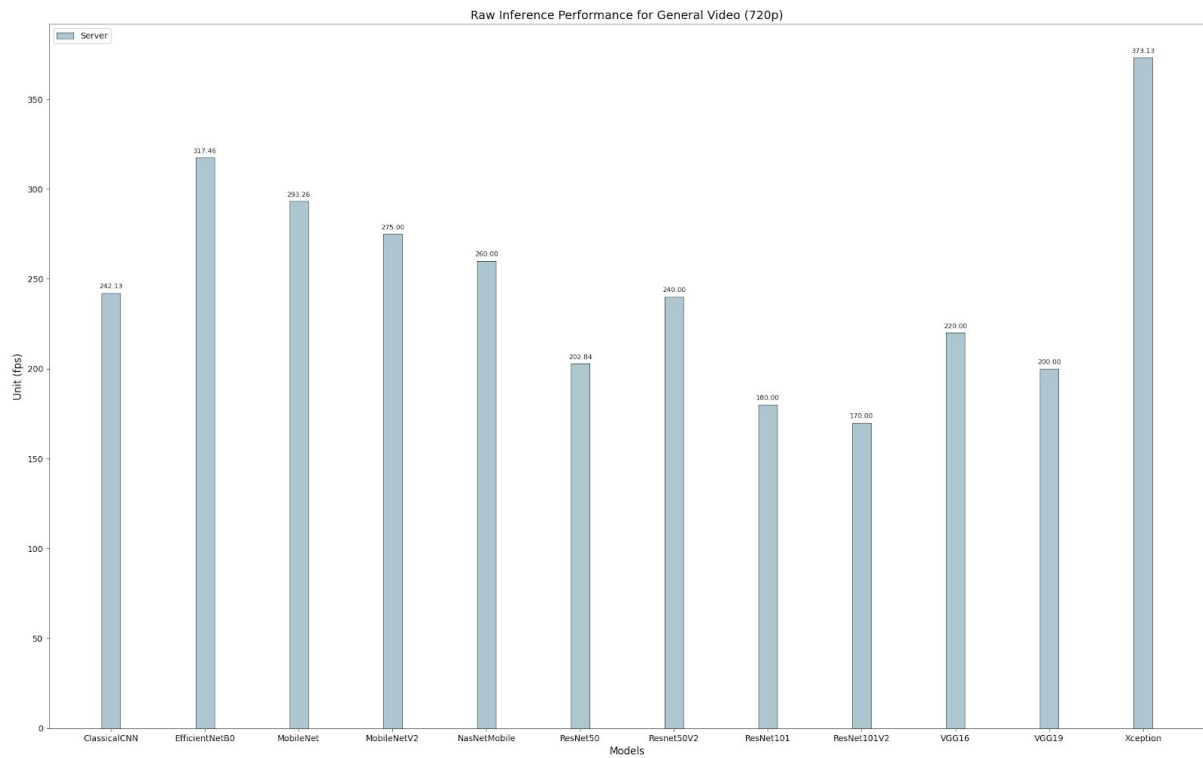


Figure 49: Theoretical Max Video Framerate for All 12 Models

In the case of raw inference performance for general video (720p), Xception is the most time-consuming, taking about 373.13 seconds, and the model that is the least time-consuming is ResNet101V2, taking 170.00 seconds. ResNet101V2 is doing better.

4.2.4.2.7. Inference Efficiency (Frames Per Watt)

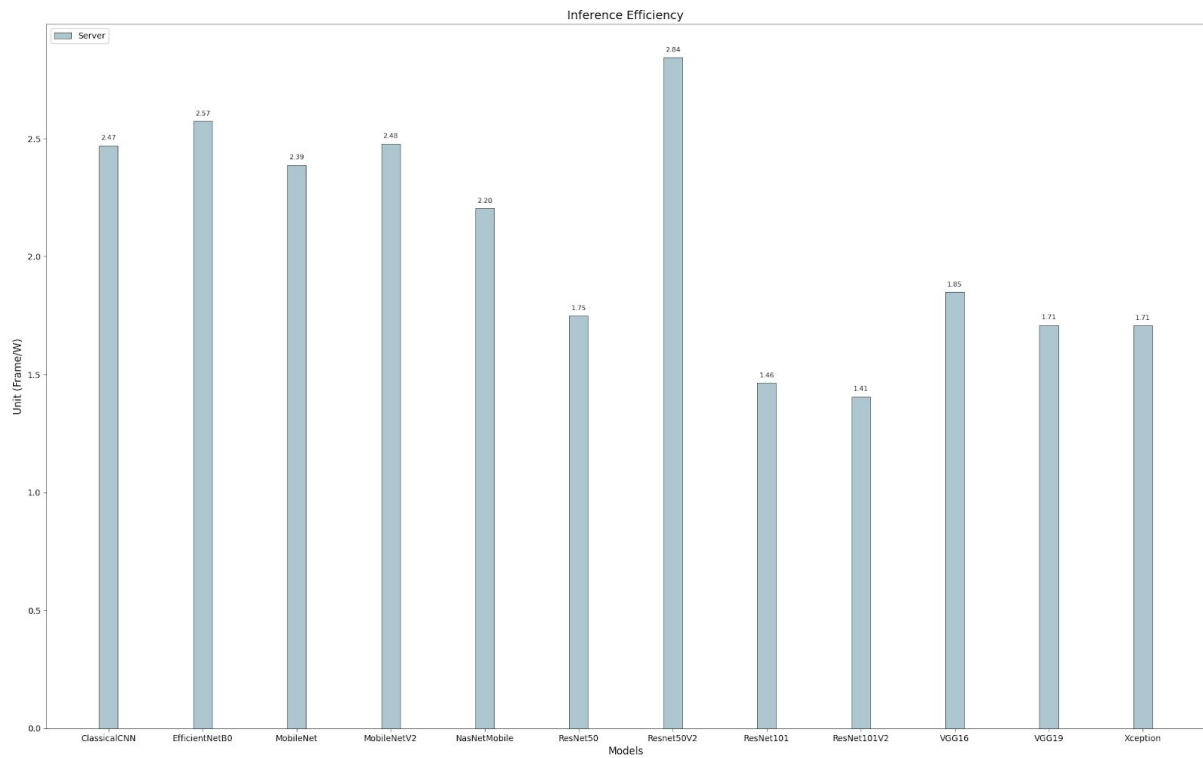


Figure 50: Inference Efficiency for All 12 Models

In the case of inference efficiency, ResNet50V2 is the most time-consuming, taking about 2.84 seconds, and the model that is the least time-consuming is ResNet101V2, taking 1.41 seconds. ResNet101V2 is doing better.

4.2.5. Optimizing Design Approach 2

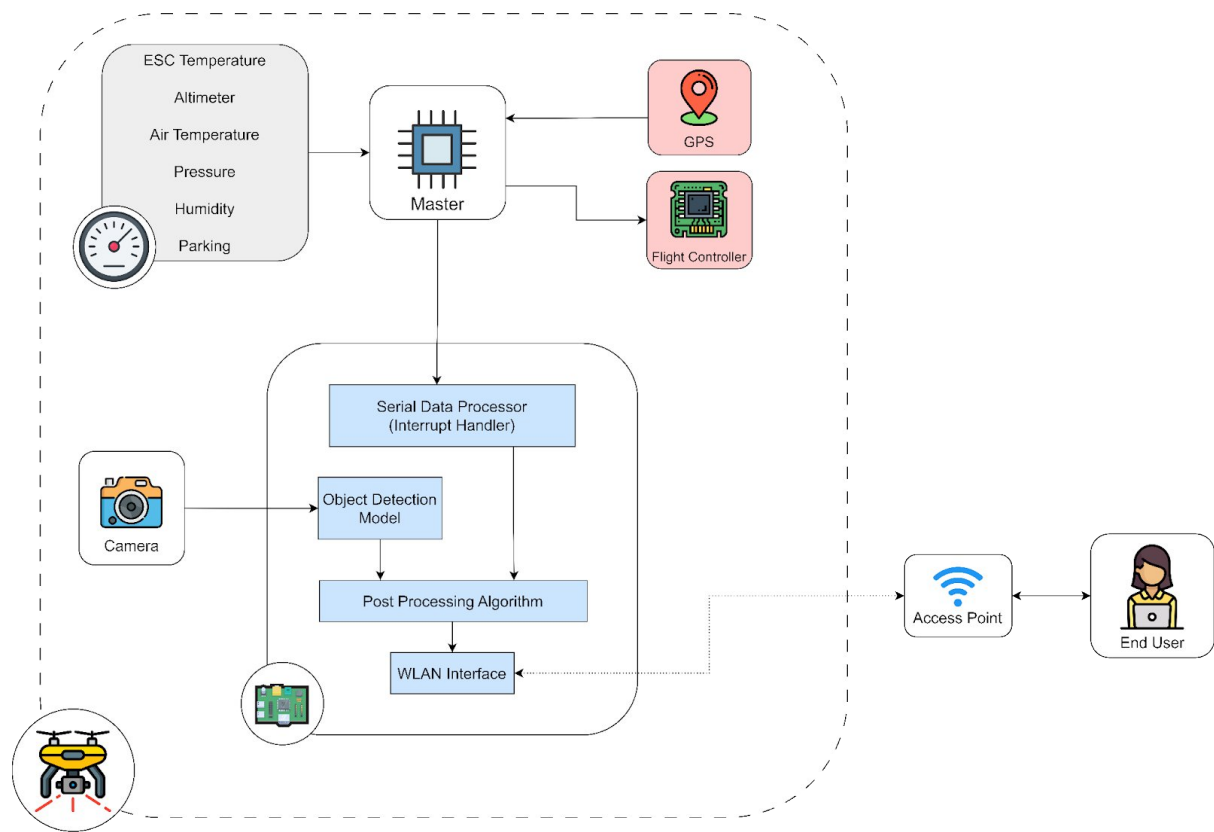


Figure 51: Design Approach 2 Overview

In the simplified flow diagram the system the data is flowing in a manner where most of the processing is done on the drone computer. This reduces time for the transfer as it does not require the transmission of large image formats. Also since the pre-processing is done on the drone the end user can see the result as soon as the drone lands, albeit the processing algorithms are cut down drastically in order to be processable on board keeping the hardware constraints in mind.

4.2.5.1. Training

4.2.5.1.1. Accuracy (Top Four Models)

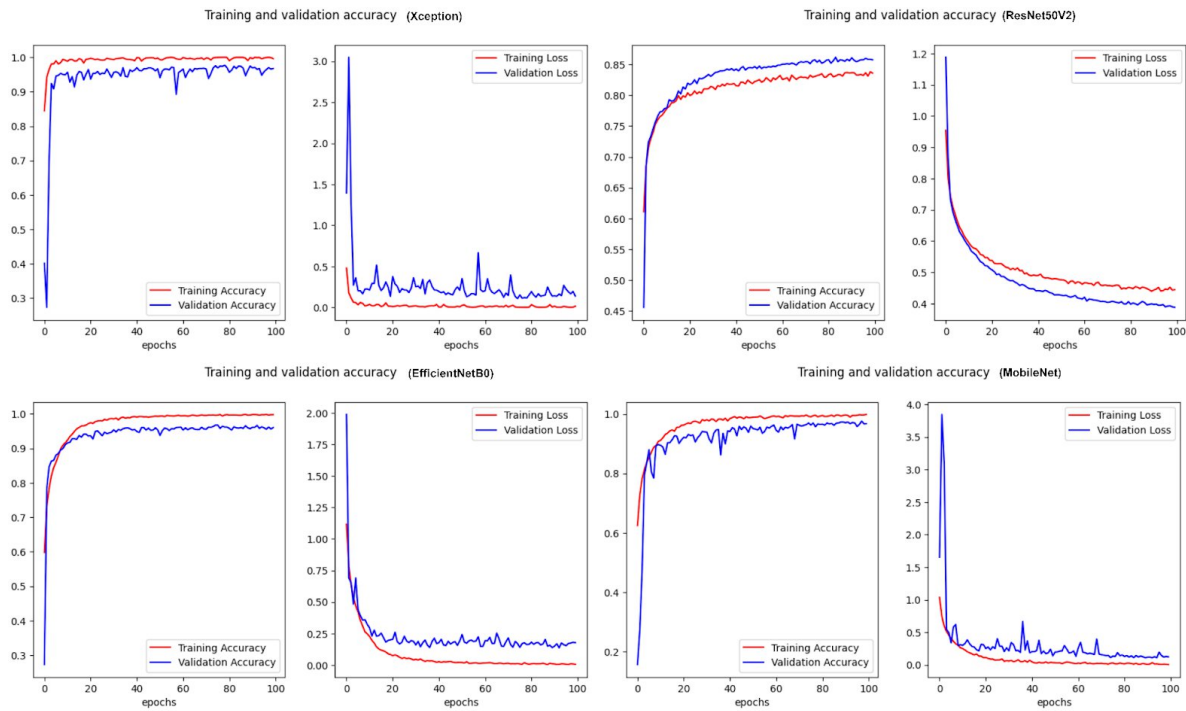


Figure 52: Training and Validation Accuracy for Top 4 Models

The overall top four performing models have the most accuracy in the case of training accuracy and validation accuracy. The models are EfficientNetB0, MobileNet, Xception and ResNet50V2. This has been done in order to save energy as we knew that training models on ARM would take exceptional amounts of time.

4.2.5.1.2. Average Power Draw

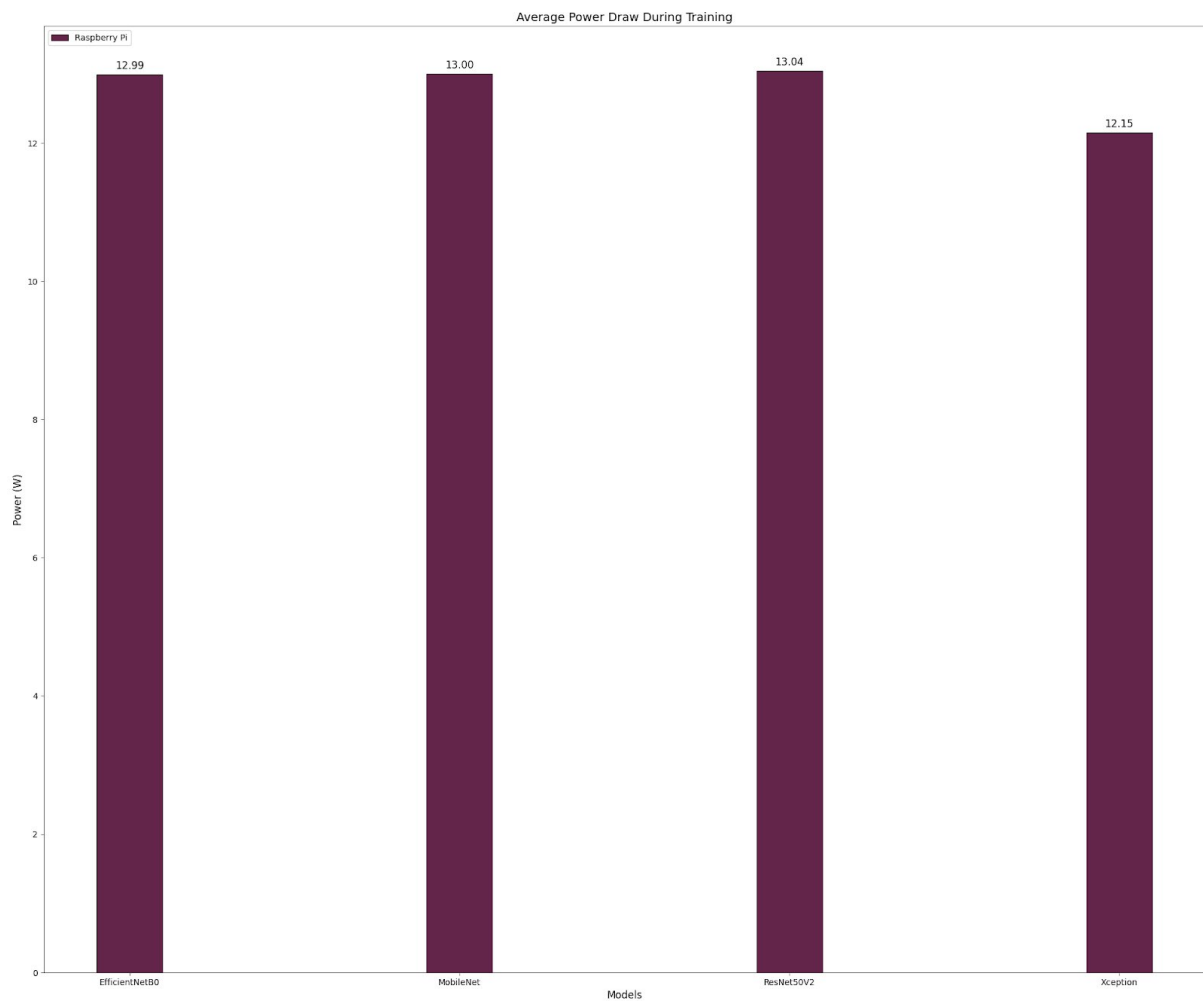


Figure 53: Average Power Draw on ARM for Top 4 Models

Out of these four models the average power draw in Raspberry Pi the same for the first three models are almost the same, having timing from 12.99 watts to 13.04 watts. In case Xception is quite less, having 12.15 watts.

4.2.5.1.3. Peak Power Draw

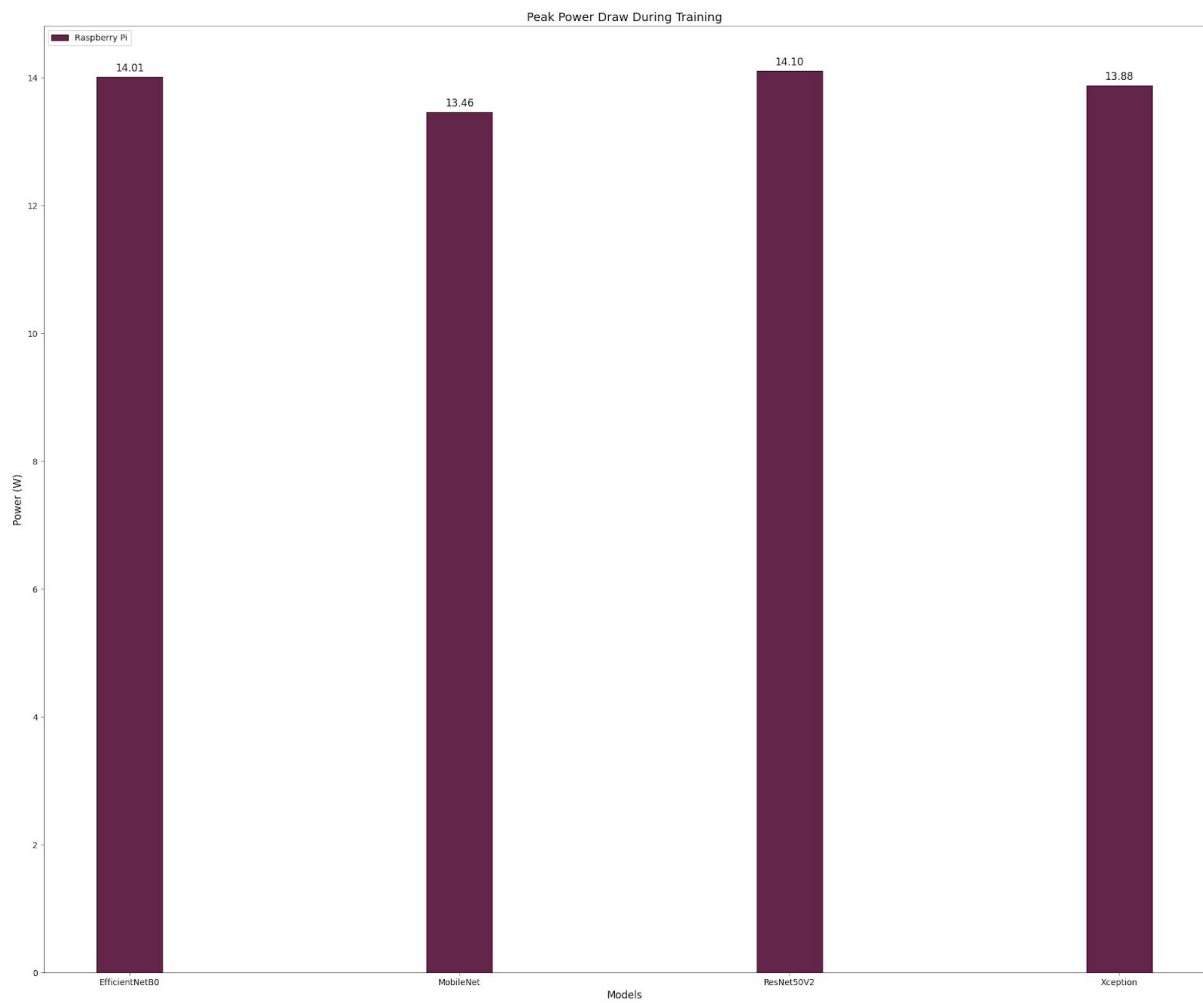


Figure 54: Peak Power Draw on ARM for Top 4 Models

From the models the peak power draw of the highest power drawn is ResNet50V2, having 14.10 W, and the lowest power consumed is MobileNet, which is 13.46 watts.

4.2.5.1.4. Total Energy Consumption During Training

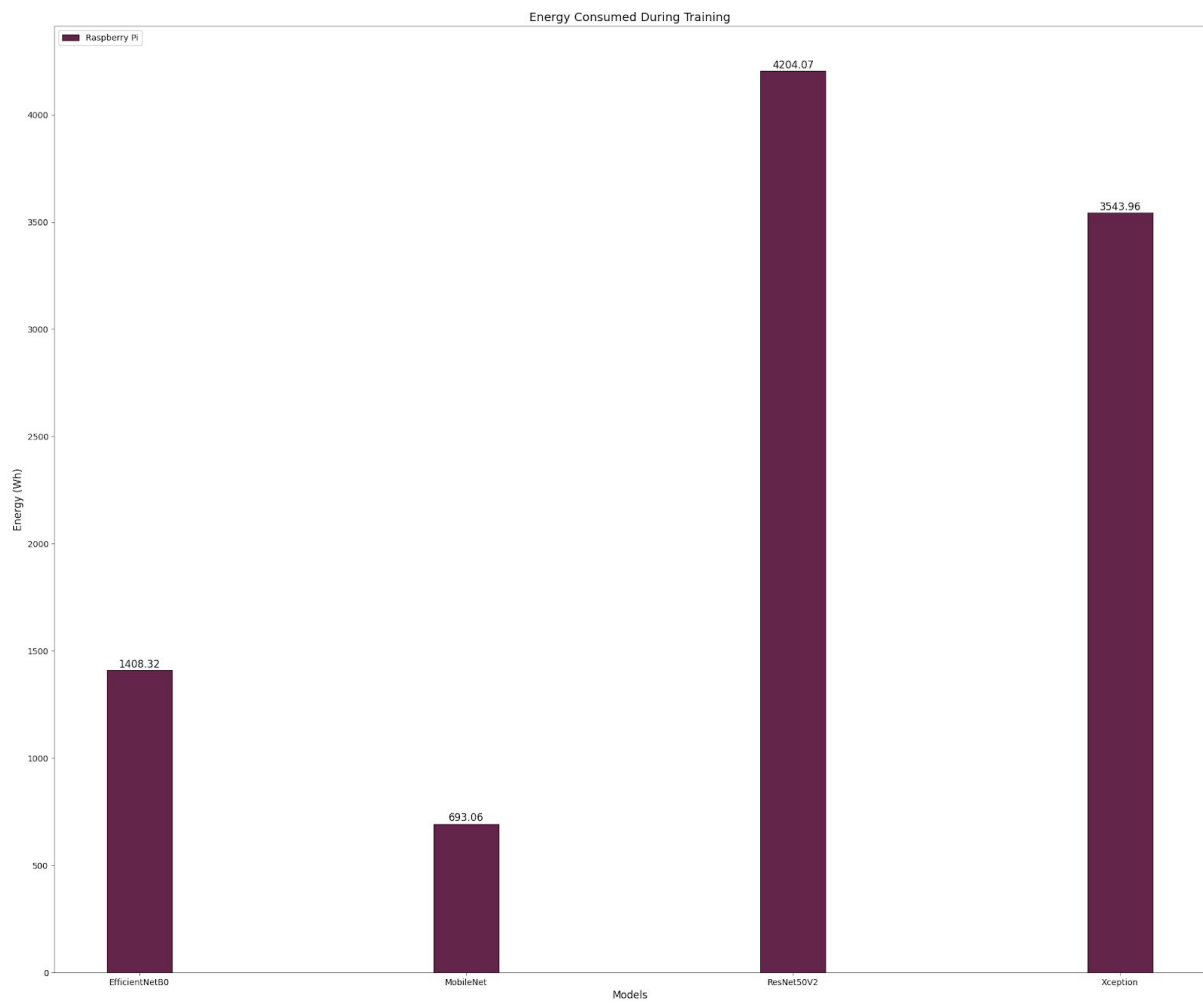


Figure 55: Total Energy Consumption During Training on ARM for Top 4 Models

From the models, the highest energy consumed in training is ResNet50V2, having 4,204.07Wh, and the lowest power consumed is MobileNet, which is 693.06Wh.

4.2.5.1.5. Time Required Per Iteration

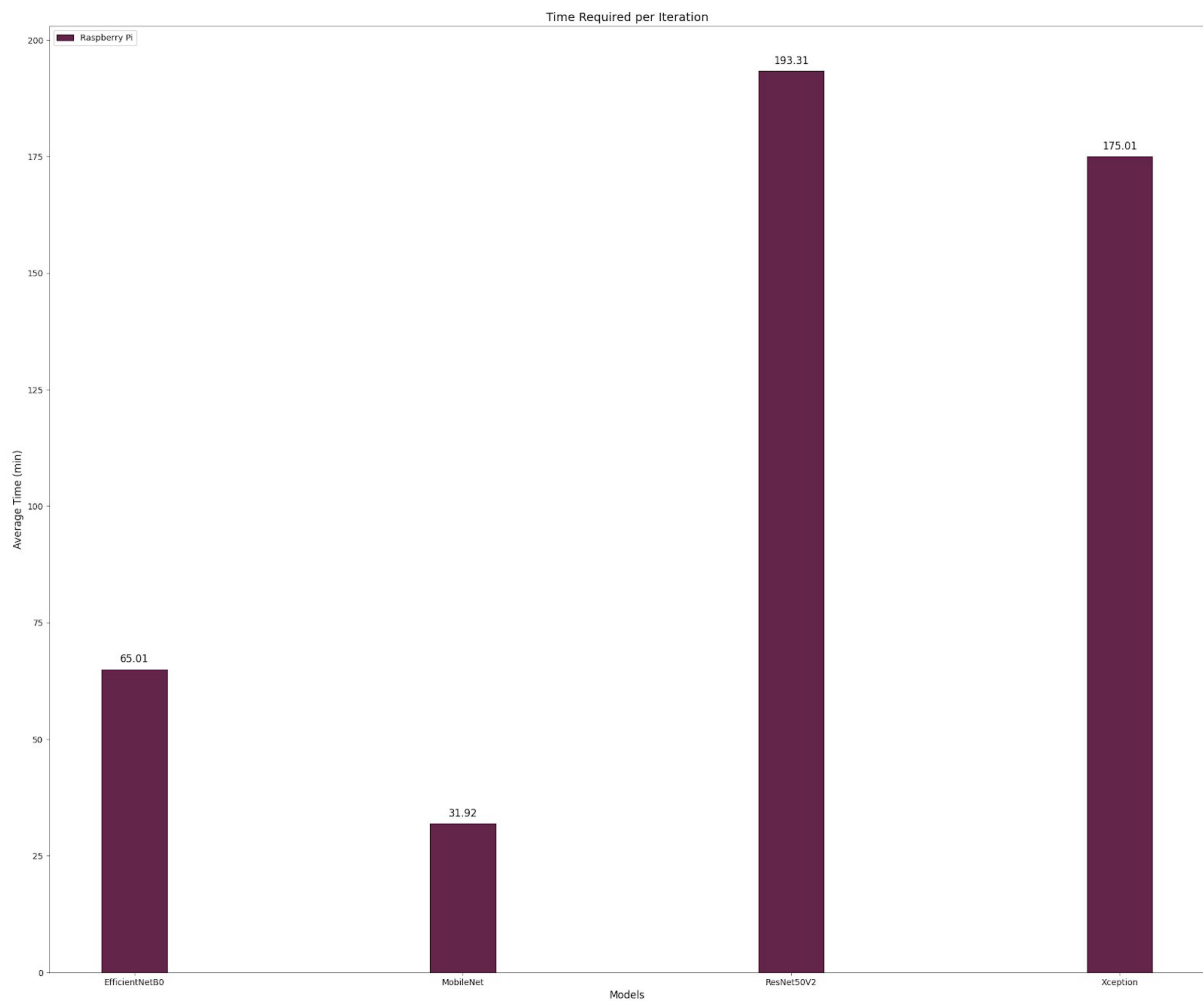


Figure 56: Training Time Required Per Iteration on ARM for Top 4 Models

From the models, the time required per iteration, the highest time consumed is ResNet50V2, having 193.31 seconds, and the lowest time consumed is MobileNet, which is 31.92 seconds. MobileNet is doing better.

4.2.5.1.6. Total Training Time

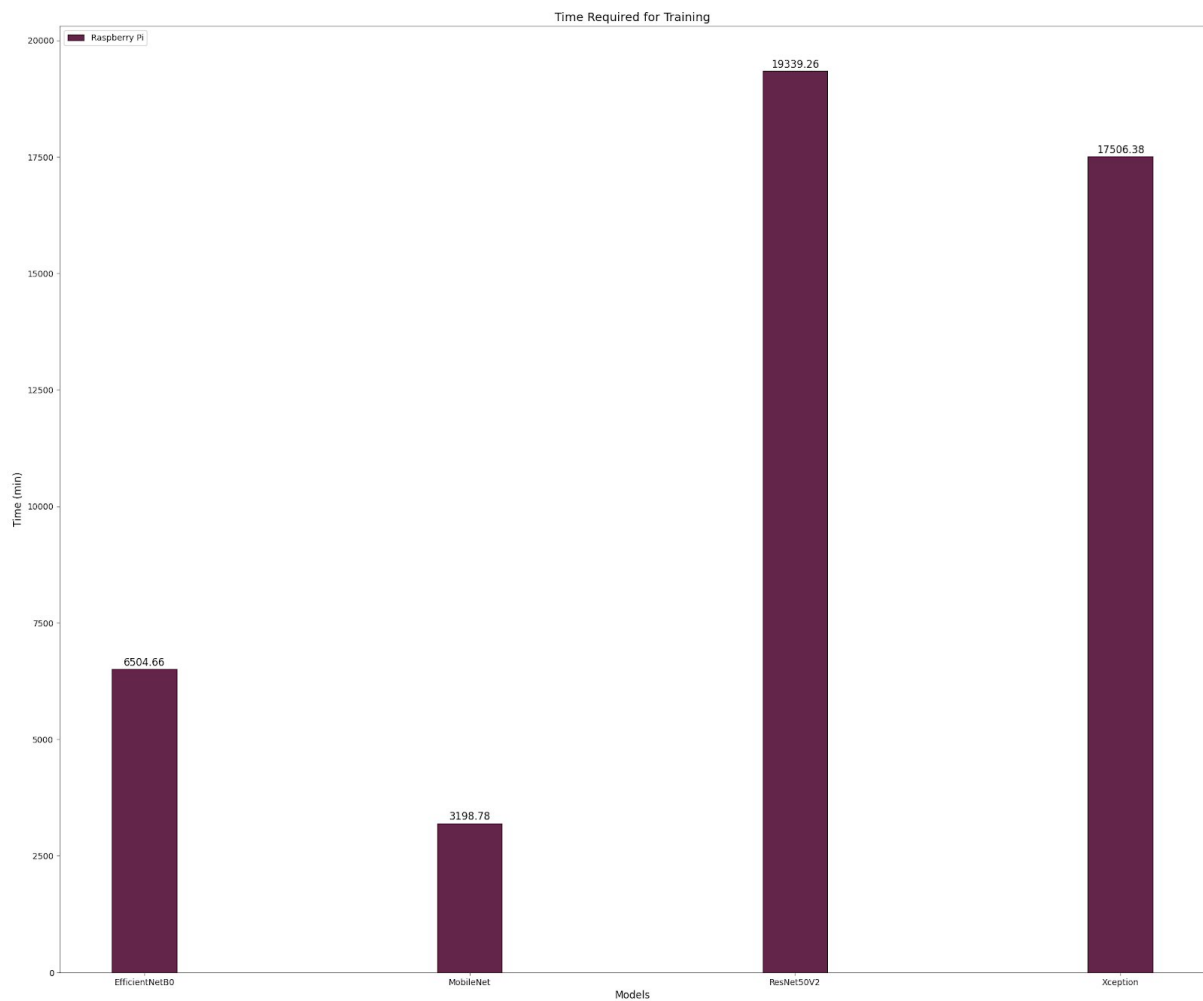


Figure 57: Total Training Time on ARM for Top 4 Models

From the models, the total training time, the highest time consumed is ResNet50V2, having 19,339.26 seconds, and the lowest time consumed is MobileNet, which is 3,198.78 seconds. MobileNet is doing better.

4.2.5.1.7. Power Draw Profile

EfficientNetB0: In this model, in the power draw curve, the peak is at 14.01W, and the total power consumed is 1408.317Wh.

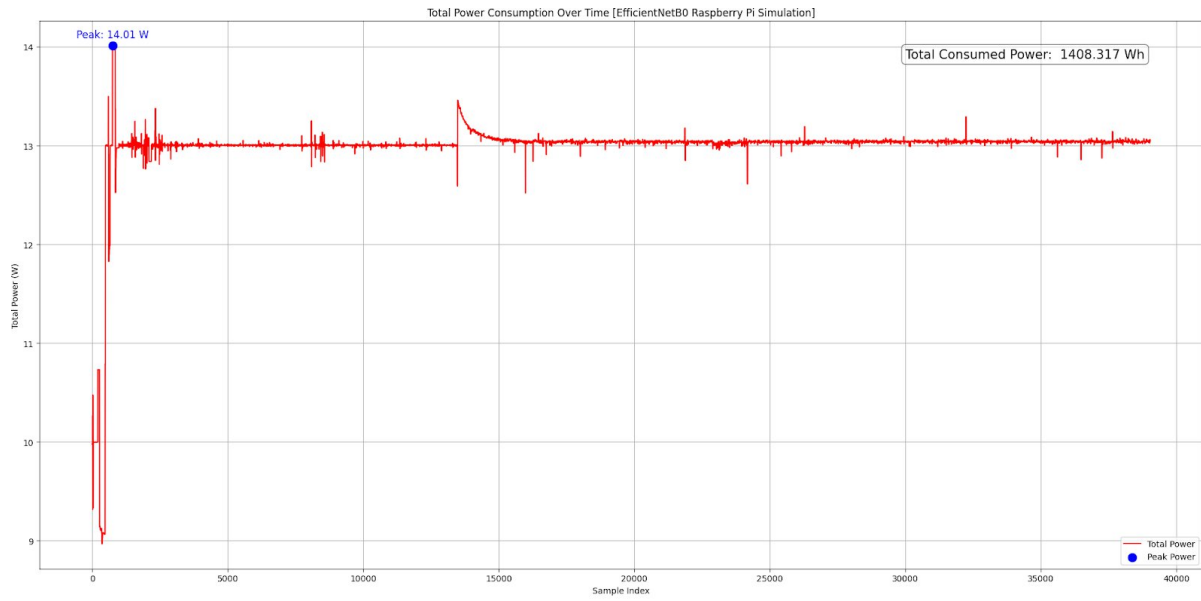


Figure 58: Power Draw Profile on ARM for EfficientNetB0

MobileNet: In this model, in the power draw curve, the peak is at 13.46W. Total power consumed is 693.061 Wh.

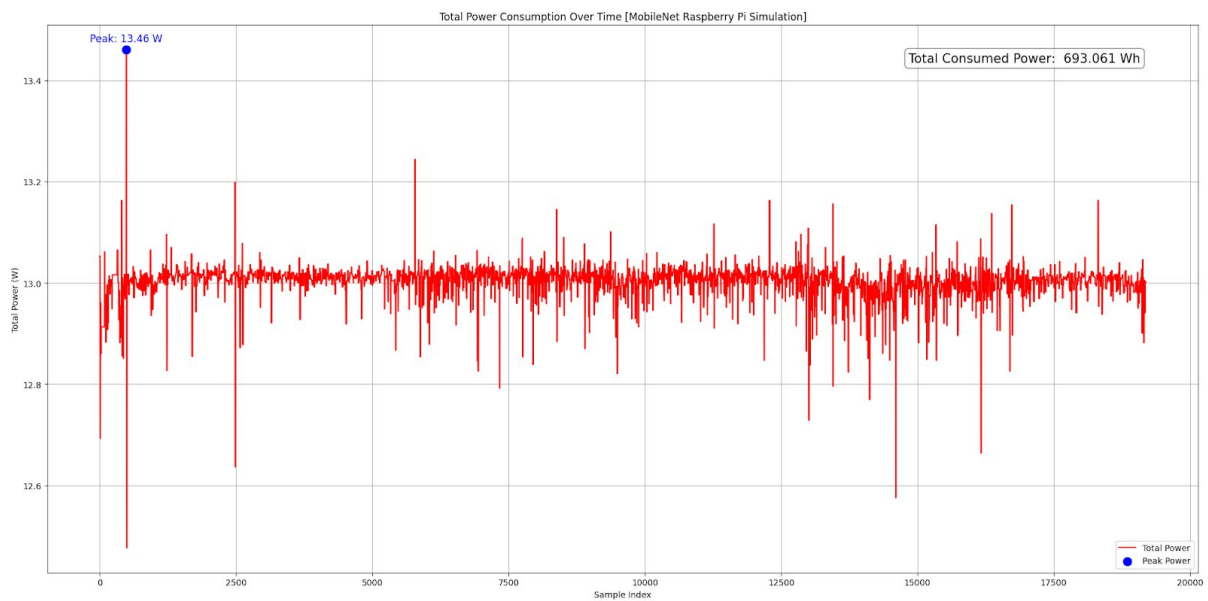


Figure 59: Power Draw Profile on ARM for MobileNet

ResNet50V2: In this model, in the power draw curve, the peak is at 14.10W. Total power consumed is 4,204.071 Wh.

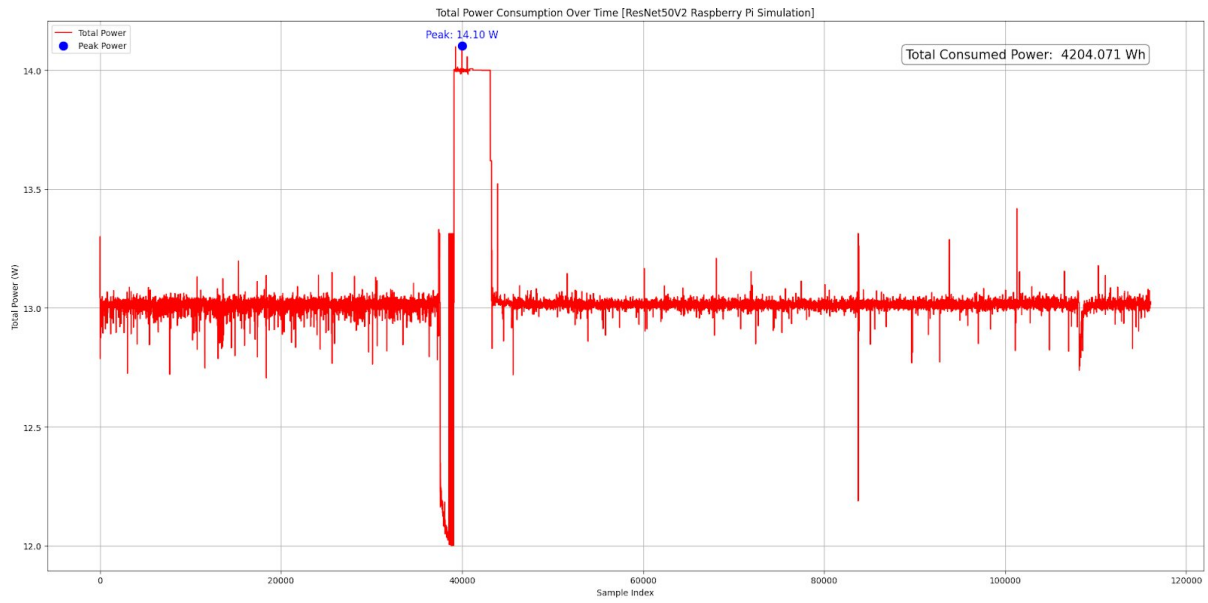


Figure 60: Power Draw Profile on ARM for ResNet50V2

Xception: In this model, in the power draw curve, the peak is at 13.88 W, The total power consumed is 3543.962 Wh.

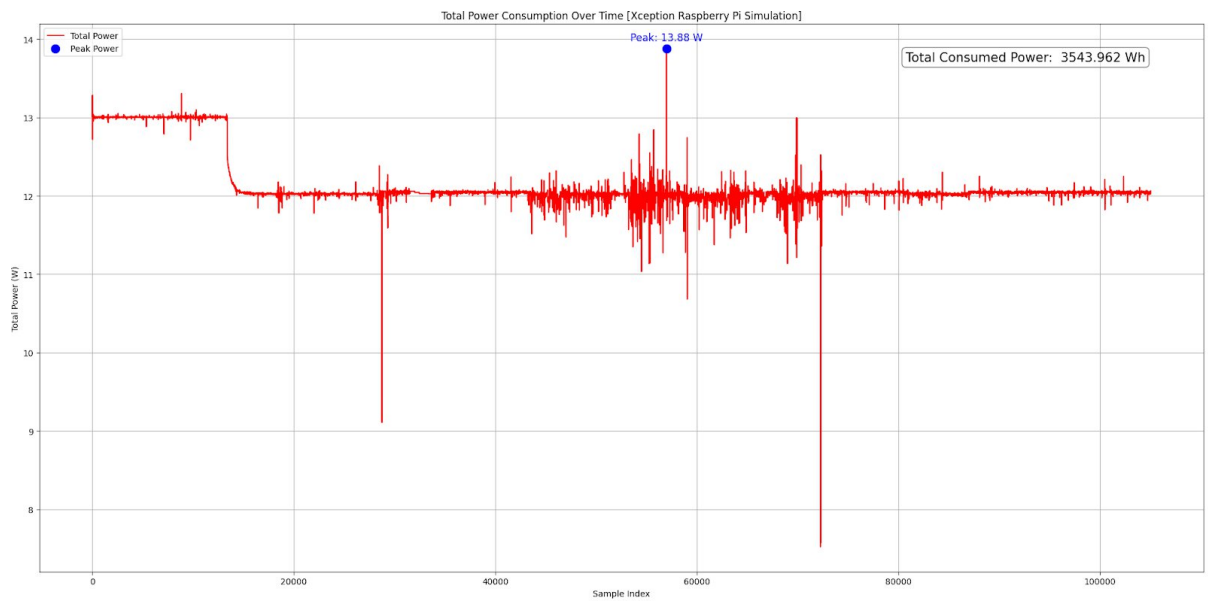


Figure 61: Power Draw Profile on ARM for Xception

4.2.5.2. Inference

4.2.5.2.1. Total Energy Consumption

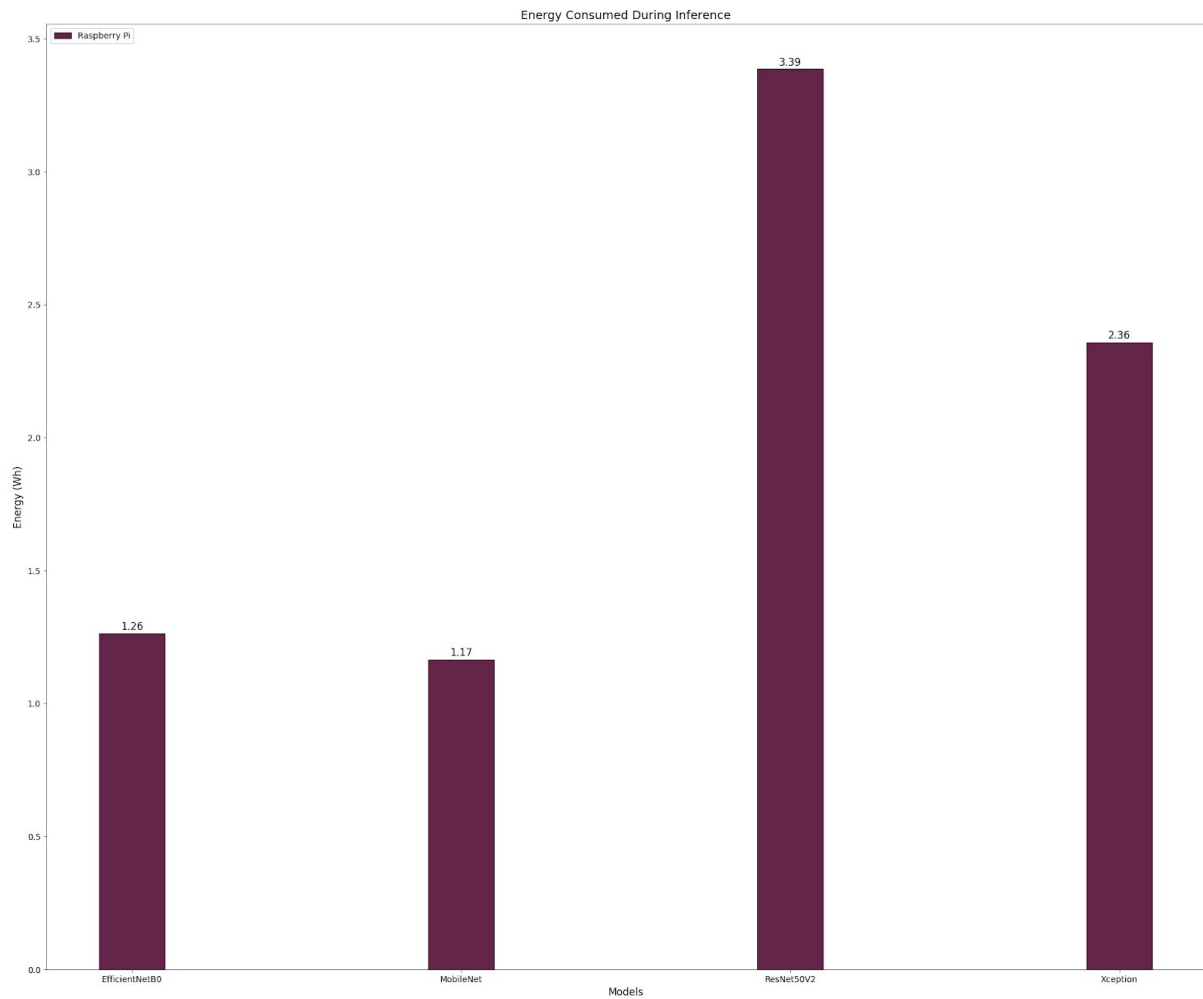


Figure 62: Total Energy Consumption During Inference on ARM for Top 4 Models

Here MobileNet takes the least energy consumption, which is 1.17 Wh, and the highest is ResNet50V2, which is 3.39 Wh. MobileNet is performing well in this case.

4.2.5.2.2. Peak Power Draw

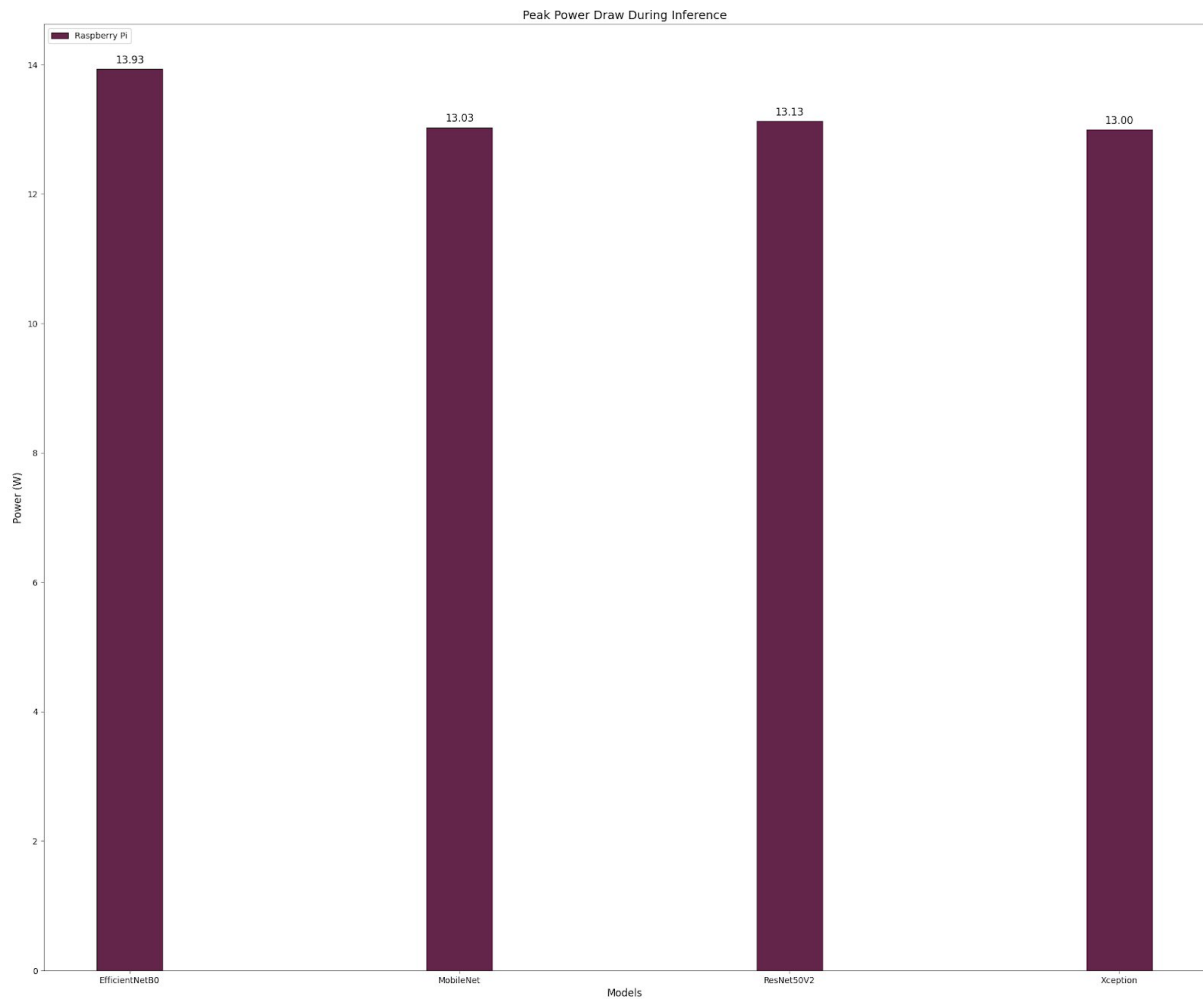


Figure 63: Peak Power Draw During Inference on ARM for Top 4 Models

Xception takes the least in peak power drawn, which is 13.00 W, and the highest is EfficientNetB0, which is 13.93 W. Xception is performing well in this case.

4.2.5.2.3. Average Power Draw

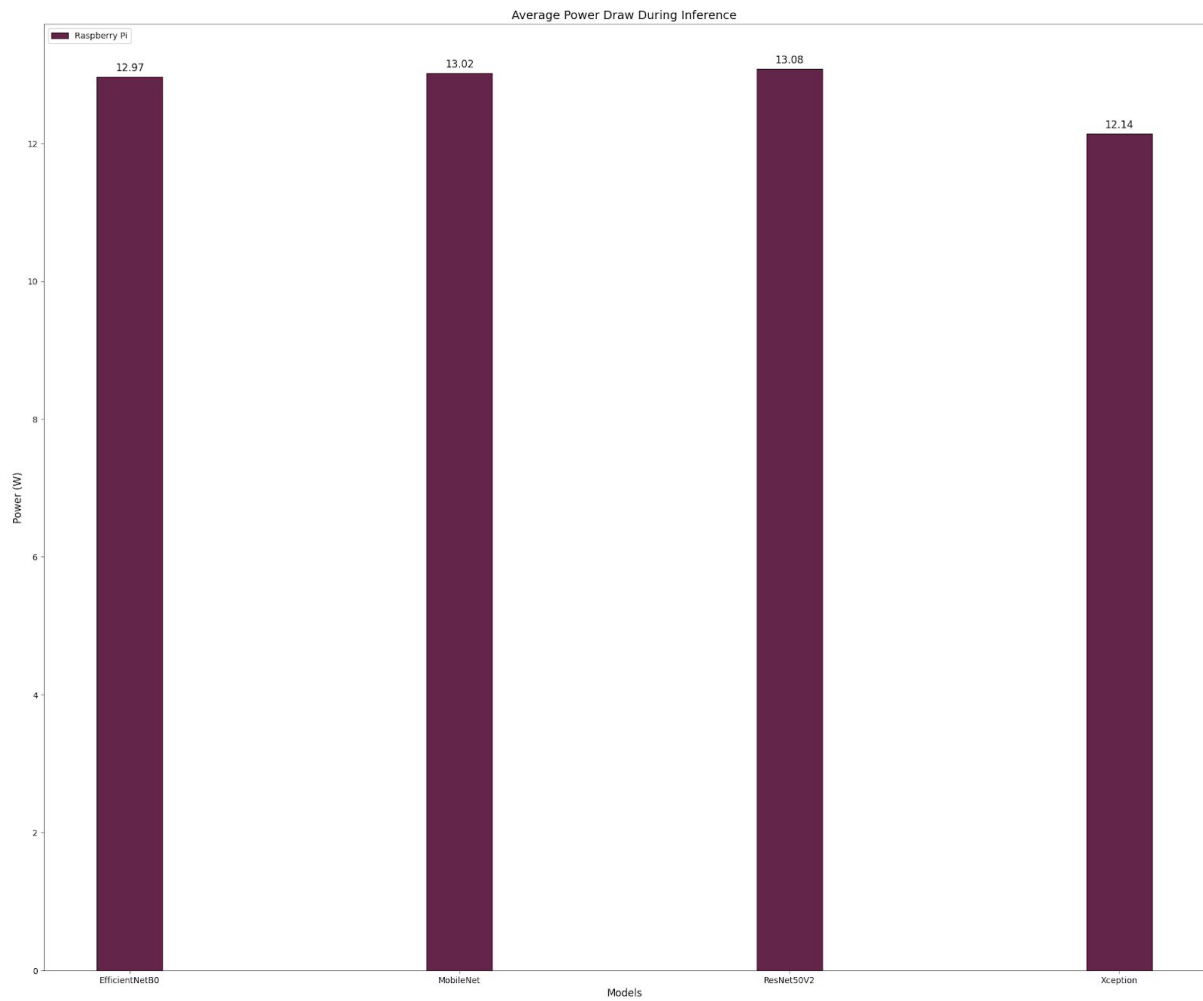


Figure 64: Average Power Draw During Inference on ARM for Top 4 Models

Xception takes the least for average power draw, which is 12.14W, and the highest is ResNet50V2, which is 13.08W. Xception is performing well in this case.

4.2.5.2.4. Time to Process 1,000 Frames

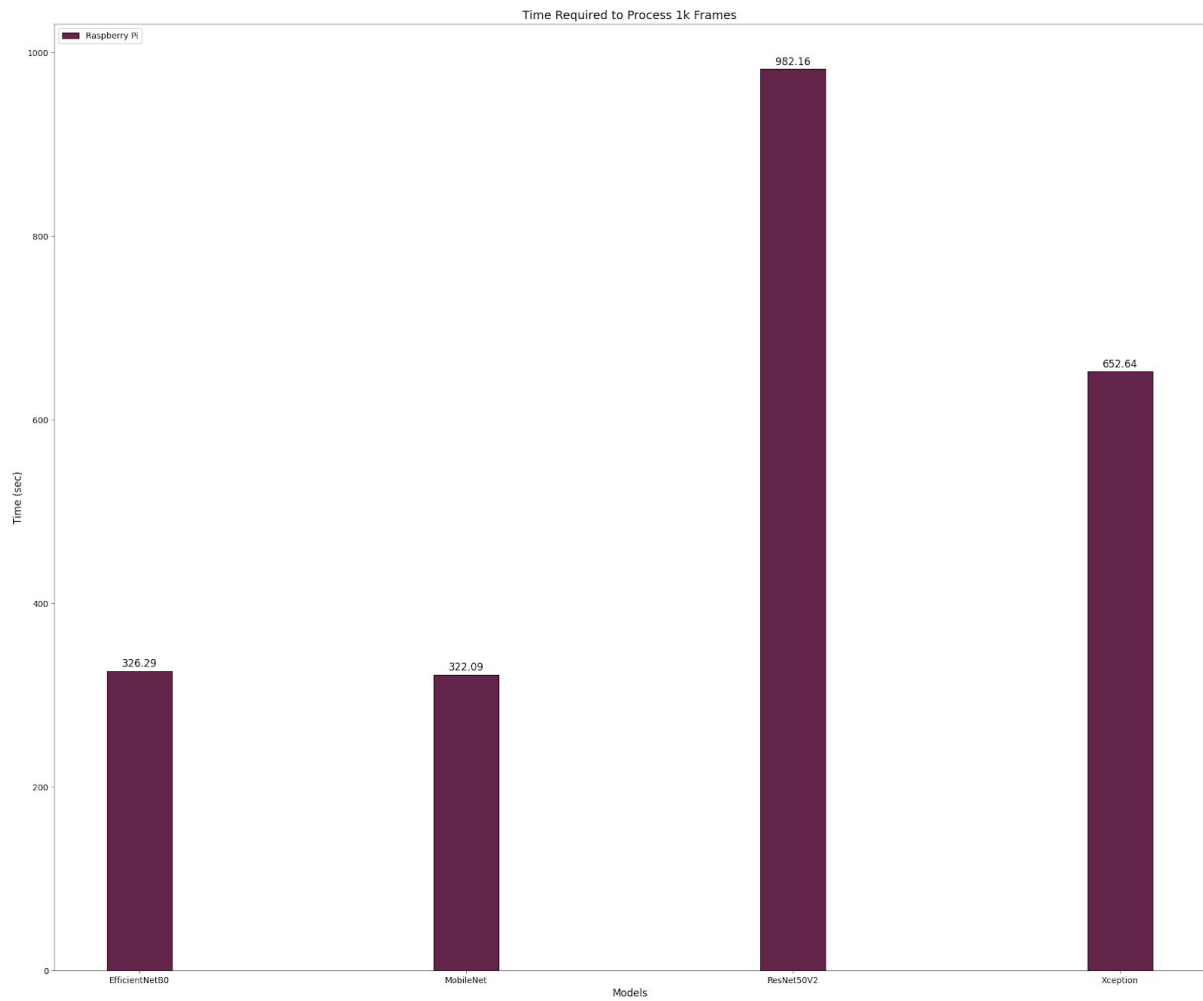


Figure 65: Time to Process 1k Frames on ARM for Top 4 Models

MobileNet takes the least in processing 1000 frames, which is 322.09 seconds, and the highest is ResNet50V2, which is 982.16 seconds. MobileNet is performing well in this case.

4.2.5.2.5. Time Required Per Frame

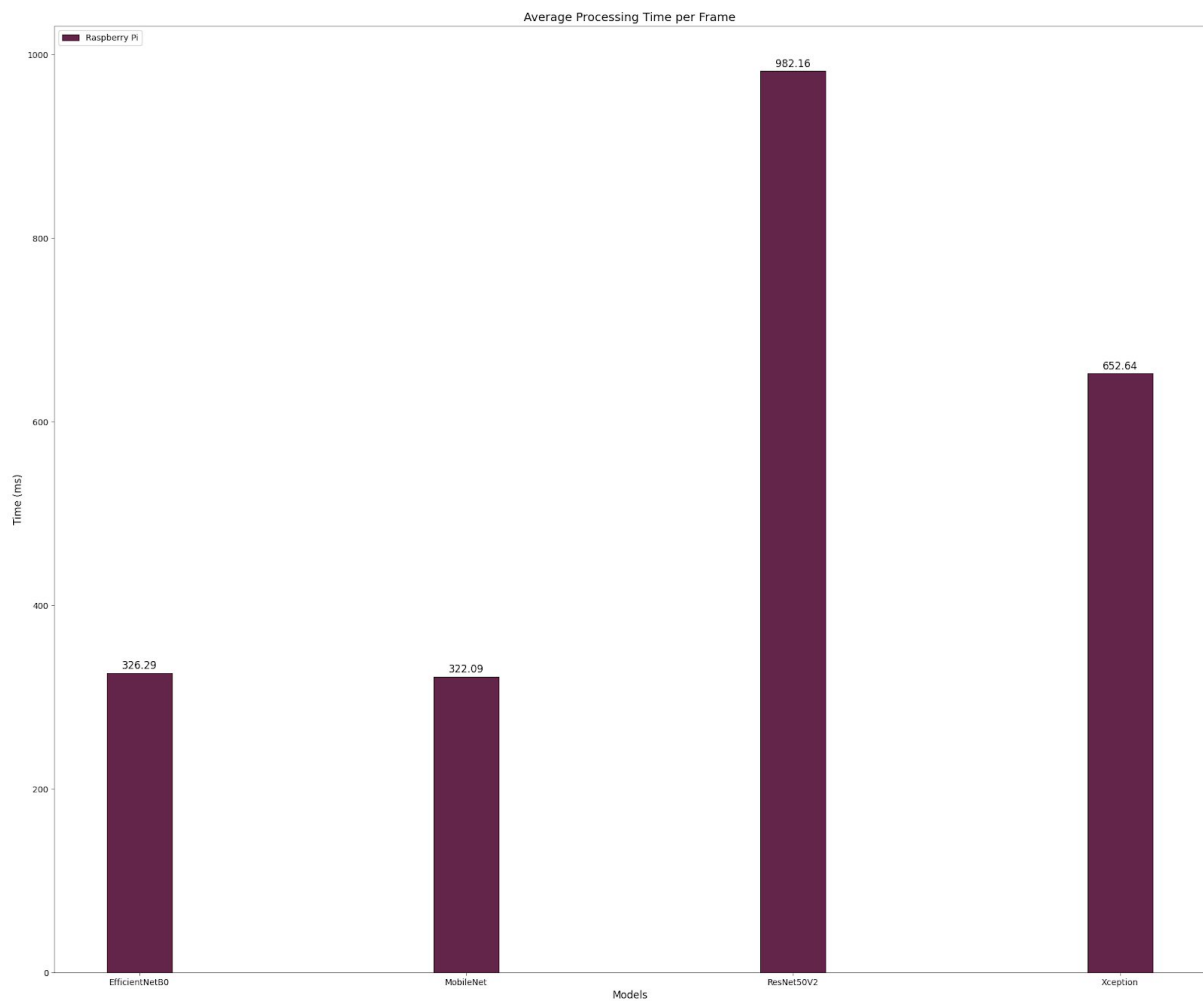


Figure 66: Time Required to Process One Frame on ARM for Top 4 Models

MobileNet takes the least time per frame, which is 322.09 seconds, and the highest is ResNet50V2, which is 982.16 seconds. MobileNet is performing well in this case.

4.2.5.2.6. Video Stream Handling Capability (Theoretical Maximum)

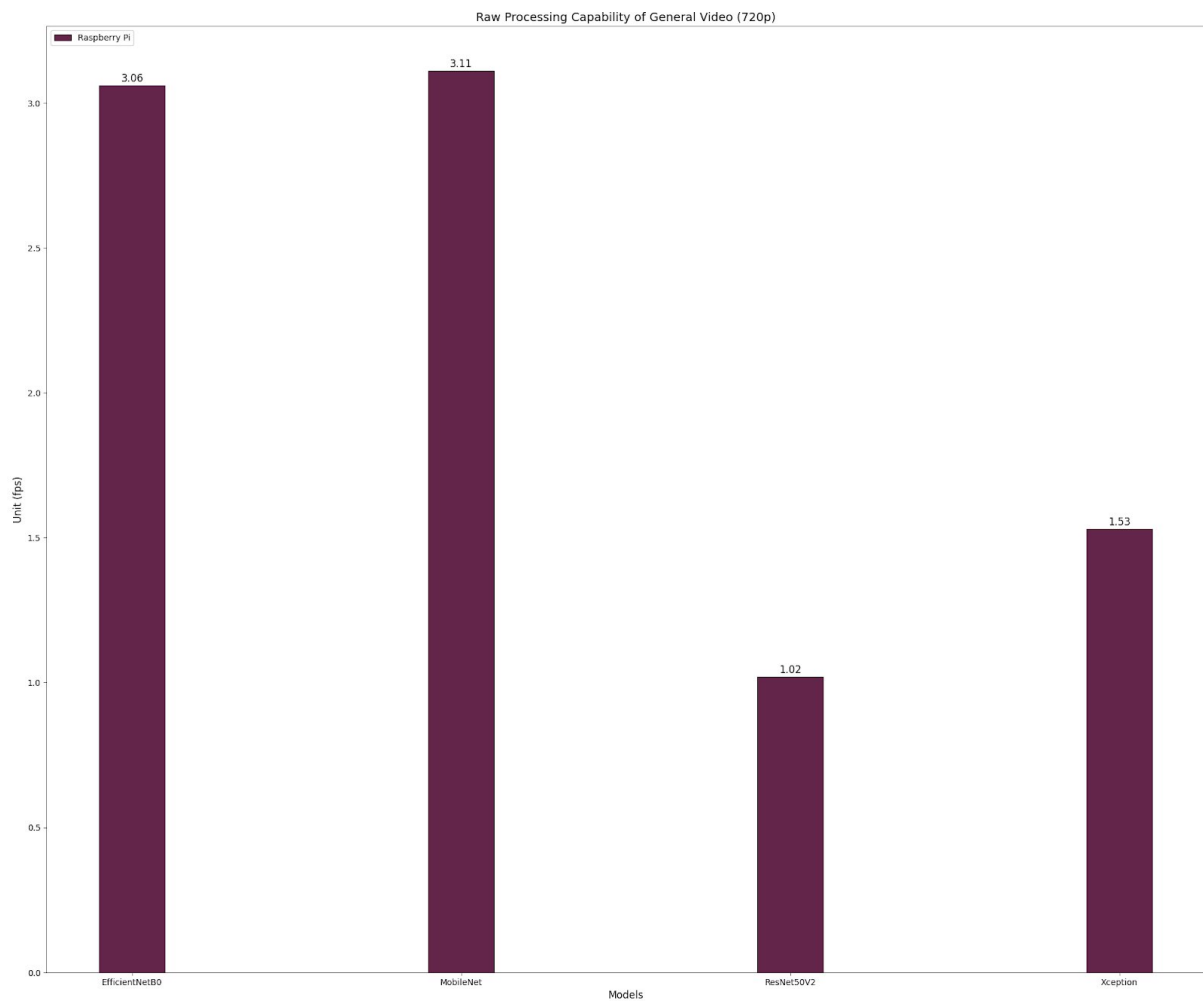


Figure 67: Theoretical Max Framerate on ARM for Top 4 Models

ResNet50V2 takes the least time for video streaming handling, which is 1.02 frames per second, and the highest is MobileNet, which is 3.11 frames per second. ResNet50V2 is performing well in this case.

4.2.5.2.7. Inference Efficiency (Frames Per Watt)

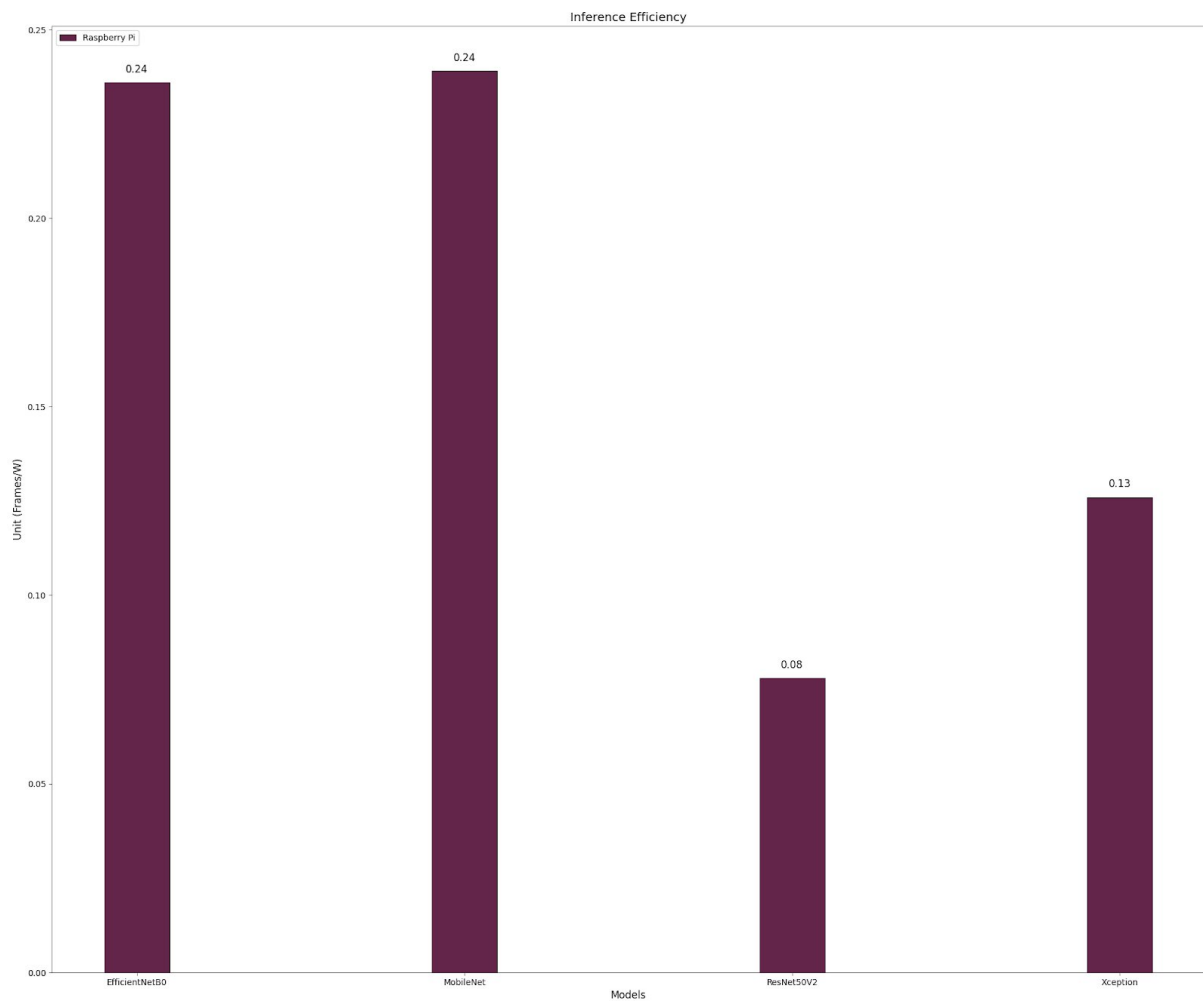


Figure 68: Inference Efficiency on ARM for Top 4 Models

ResNet50V2 takes the least for frames per watt, which is 0.08 Frames/Watt, and the highest is MobileNet and EfficientNetB0, which is 0.24 Frames/Watt. ResNet50V2 is performing well in this case.

4.2.6. General Object Detection Based Accuracy Optimization

Object detection can achieve higher accuracy compared to a CNN based poly-class classifier. This along with the fact that they are easier to train (given the dataset is properly annotated, although this is not a very difficult task but it requires a lot of repetitive low skill work). Low skilled high volume work is a crucial tradeoff for these types of tasks where classifiers require specialized datasets (where the object is clearly isolated from the background). The time consuming factor is creating datasets for classifiers is a whole different workload when compared to object detection models as some images are not simply usable and if those images are over a certain percentage of the total pool of images the whole dataset might be completely unusable whereas with object detection it may be salvageable. Due to its inherent architecture a classifier must compress the entire scene into one plain label. This combines all the different objects in that single image. Lack of background isolation and combination of multiple objects

per image results in significant label noise and low accuracy. Considering two factors that our dataset was not optimal, added to this the fact that the multi-class deployment that we are aiming for automatically makes object detection a far more attractive solution compared to classification. Data obtained from carefully designed deep learning algorithms (specially tailored to benchmarking hardware) especially micro-benchmarks that focus on throughput only can give critical insight to hardware constraints and bottlenecks early in the development process. Synthetic deep learning benchmarks are critical synthetic (often standardized tests like ML Perf are used) are specially curated to represent the specific load that will be generated due to deployment characteristics. Thus it is evident that synthetic benchmarks can reliably show the optimal hardware or the “best/better suited” installation for any task given the previous tests are reflective of their demands. Utilizing a light weight classifier based on CNN to generate ML/DL load in order to simulate working conditions was our choice to find out which of the designs and hardware combinations were optimal. This gave us critical insight into whether a fully fledged object detection can be deployed on any hardware. This can predict the hardware combinations capability of handling such workloads. This is especially important when the need for dataset optimization consumes a large amount of time from the main projects timeline. In order to combat losing progress we decided to use the raw dataset to train and deploy a classifier as it does not require labels (standardized annotations). CNN classifiers can be used as a source of ML/DL load to check the hardware's capability to perform object detection before investing time and effort to prepare and optimize for the problem. As detection requires both classification and localization (calculating bounding box regression and assignment). Thus using CNN classification as an early state capacity checking method to check if the backbone hardware is up to the task. This is based on the simple observation that if a hardware stack cannot reach an acceptable level of accuracy on a clean and single label item from a target class, there is a high probability of that backbone not being able to handle object detection. Basically it is a hardware sanity check or a backbone probe based on the principle that if it is not cut out to handle a simple classification problem it is most likely ill suited for object detection.

4.2.6.1. Environment Setup

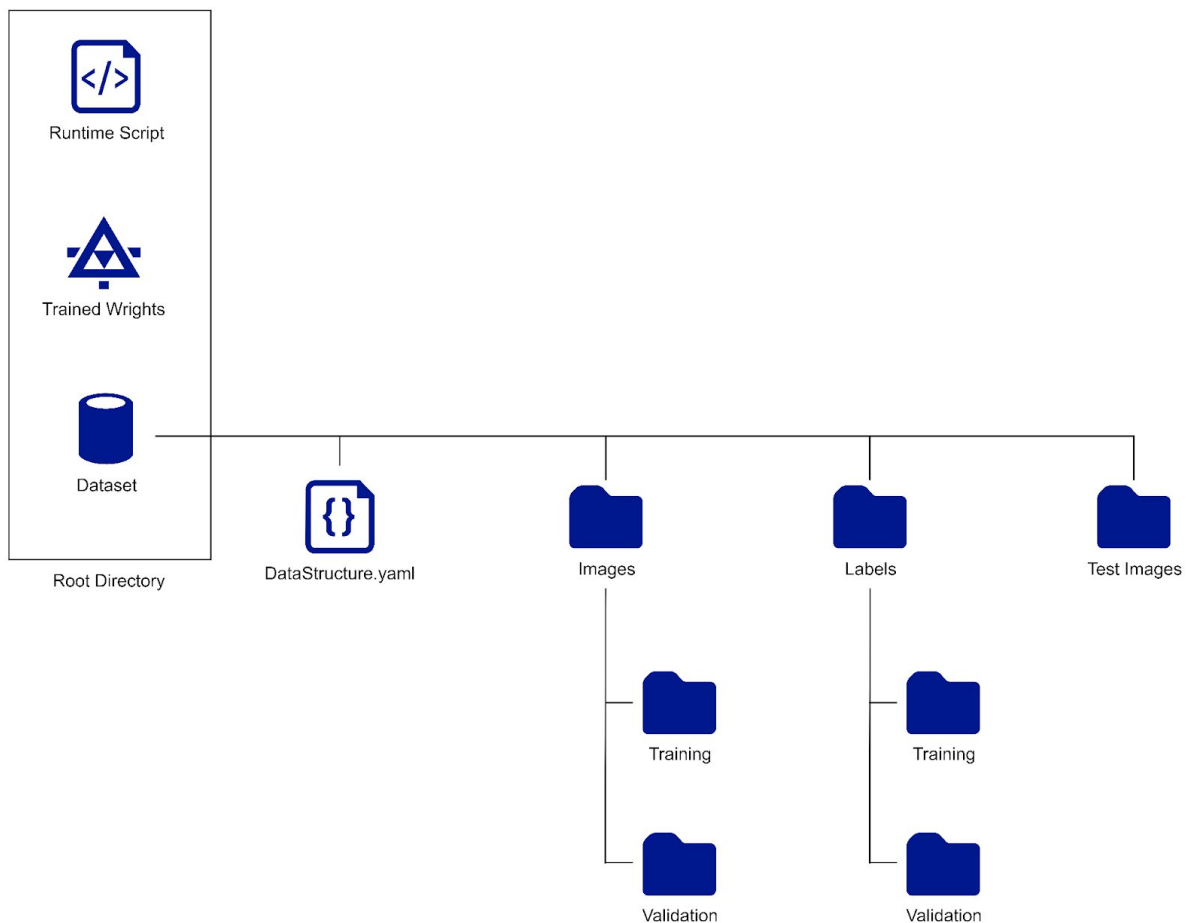


Figure 69: Environment Setup for Object Detection

This layout represents a minimal and compact dataset root for Ultralytics YOLO training and evaluation, where the top level contains runtime script for launching training/inference, any pretrained or previously trained weights, and a Dataset directory containing the actual data and configuration required to operate end to end. Within the Dataset directory is a single configuration file named `DataStructure.yaml` (often called `data.yaml`) that specifies paths to the splits as well as the class list, following the standard convention of Ultralytics where the YAML defines keys such as `path`, `train`, `val` and if necessary `test` and names for the categories. The pairing of `Images/` and `Labels/` directories follows the canonical YOLO organization in which each split has a corresponding images and labels folder, making sure we have one `.txt` annotation file per image, stem identical names and normalized coordinates per label line, which is the expected text format for YOLO detection tasks. Concretely, `Images/` has `Training/` and `Validation/` subfolders containing the visual data for each split, and `Labels/` mirrors that with `Training/` and `Validation/` subfolders storing annotation files, keeping the one-to-one relationship between image and label files across the same split boundaries. The presence of a `Test Images` directory makes it easy, and optional, to use unseen examples for rapid qualitative checks or for an explicit test split mentioned in the YAML, in keeping with Ultralytics guidance that a test section may be provided in the dataset configuration if needed.

The data.yaml is the single source of truth, usually declaring relative paths such as images/train, images/val, and possibly images/test (or corresponding text lists), along with class names that are indexed from 0, which the training routine consumes to discover data and class semantics. In this structure, the labels follow the YOLO text schema wherein each line contains class_id followed by center-x, center-y, width and height, all normalized to the image dimensions, guaranteeing compatibility with the Ultralytics dataloader and training pipeline. The duplication of Training and Validation under both Images and Labels allows the split boundaries to be explicit and self-contained, making it easy to transport and making sure there is no cross-contamination of samples across stages, which is a recommended modus operandi in Ultralytics dataset overview. At the project root, keeping a dedicated runtime script insures repeatable launching of training, validation, and inference while keeping weight artifacts to go along with it for quick recovery, fine-tuning, or not changing the dataset subtree bench marking cycles. On the whole, this succinct hierarchy captures just train/val essentials plus an optional test area plus a single YAML contract, providing the minimum required to start training reliably whilst staying fully aligned with Ultralytics expectations of directory structure and configuration semantics.

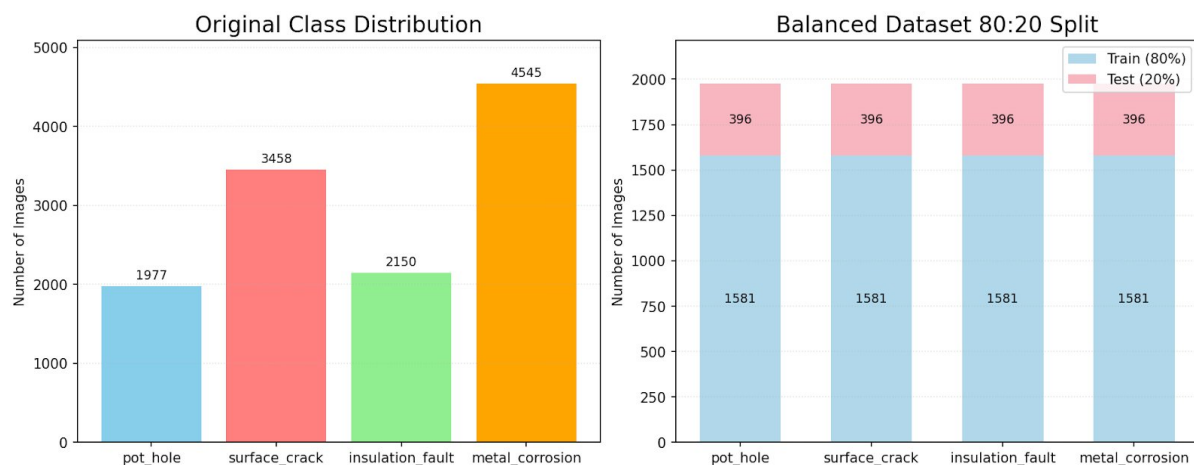


Figure 70: Class Balancing for Object Detection

The workflow begins by diagnosing a multiple class imbalance for the original data set, with potholes represented by the minimum category surface cracks and metal corrosion predominating the distribution, in the left panel making potholes are as visualised by the counts the minority point of reference for once again balancing. To correct this skew, the process takes on undersampling, a standard technique for imbalanced data in which examples are randomly removed from majority classes to match the minority class size, therefore equalizing class frequencies without synthesizing new samples. Concretely, all of the images from potholes are retained, and images from surface_crack, insulation_fault, and metal_corrosion are randomly sampled down until each class matches the pothole tally, making sure that per class is the same number, so that the model isn't going to over love some over represented categories during training. Parity is verified in the right panel by showing the same per-class totals before split, signifying successful rebalance without augmentation and providing a clean base for partitioning. With balance achieved, the dataset is split with an 80/20

train-test split to enable unbiased generalization assessment, based on randomised selection maintaining class balance across subsets in line with best practices for model validation/evaluation. The bars annotated from 1,581 training and 396 testing samples per class reflect this split precisely, reflecting an effectively stratified allocation that reflects the unified class sizes and avoids accidental drift in class proportions between train and test. For reliable model selection and hyper parameter tuning without contaminating the held-out test set, a further 90/10 split is applied to the training portion to create a separate validation subset, corresponding to the standard three-way partitioning strategy used to overfit and guide iteration. In practise, these ratios give roughly global shares of 72% training, 8% validation and 20% testing on a per class basis, which balances learning capacity with robust checkpointing and final evaluation buffers, which aren't touched until the finish. While random undersampling can sacrifice some majority class diversity, it provides a transparent and reproducible baseline if overall sample counts are ample, and can be complemented later by alternatives such as class-weighting or synthetic oversampling if performance plateaus. Overall, this pipeline enforces class parity, maintains representativeness across splits, and creates clear separation of training, validating and testing, thus generating interpretable metrics and fair comparisons between future experiments based on the same data curation principles.

4.2.6.2. Training

During training we have seen that after the full 10,000 (2,000 for the comparatively heavier yolo12s model) cycle run there were some issues visible with a few of the parameters starting showing adverse effects. This negative effect on training factors is most probably because this data set is not optimised for object detection, even though all of the images have been specially curated to include all over classes. We have searched a lot for different data sets, we found four that were fitting. A big challenge was to include all of the classes in one combined data set. As it was not optimised for object detection as we were limited by time. We most probably made some errors. There is a factor of us not being very experienced with image detection DL/ML tasks so might be a factor of not being very optimised with our decisions when it came to and annotating the images.

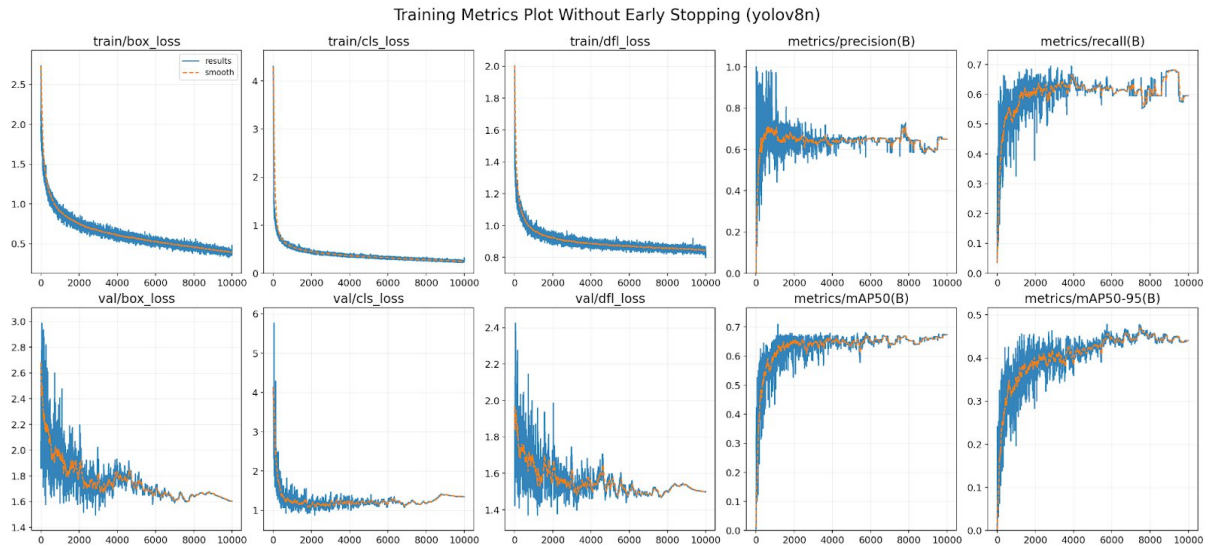


Figure 71: YOLOv8 Training Metrics Without Early Stopping

Here in this image we can see that about 4,000 iterations later there is an increase in error that has negatively impacted our training. In the recall box we can see there is a large jump in the middle of 8,000 to 10,000 iterations and down in the validation and box loss plot there is a slight and gradual decline and around about 4,000 iterations. Just as mentioned before there is a slight yet notable jump in We mean the box loss. This prompted us to use a release of function when we trained the model again as we had access to reasonably powerful hardware. We did the training again with the early stop functions. This results in a moderate to Nero super boost in the end result of our training session as this is an older and light model from yolo, a product of Ultralytics. It hasn't been such a remarkable upgrade when we compare it to the next results in the following test.

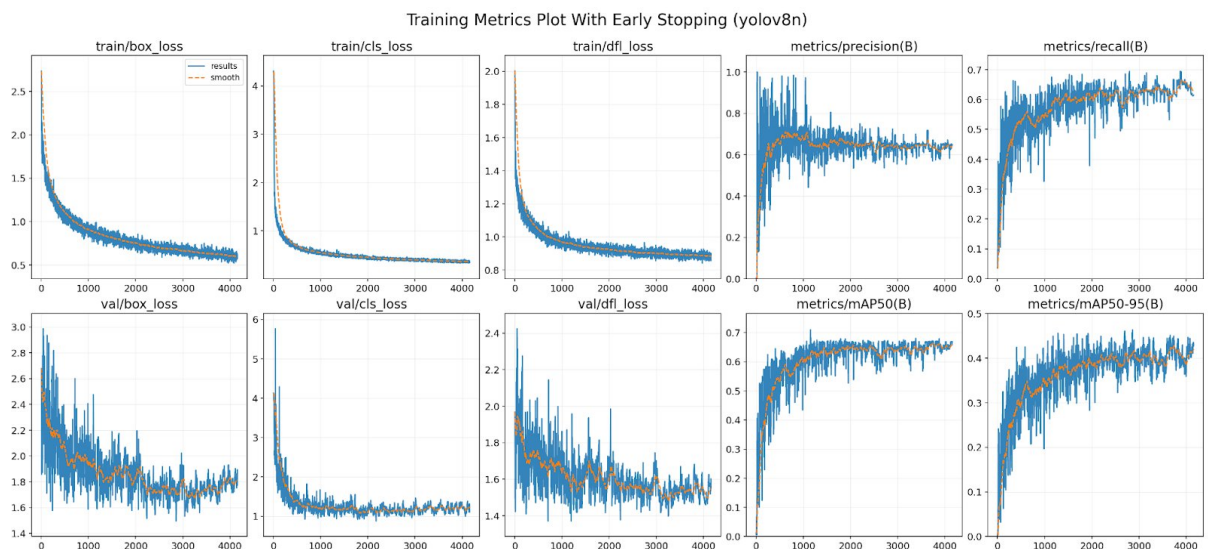


Figure 72: YOLOv8 Training Metrics With Early Stopping

The effect of this early stopping function is clearly seen in the training box and plots in the above image as we can see our training has stopped after just a tad bit 4000 iterations. Thus our end product has a better overall accuracy after the whole training session but it is still comparatively low when we see the next models that are more advanced.

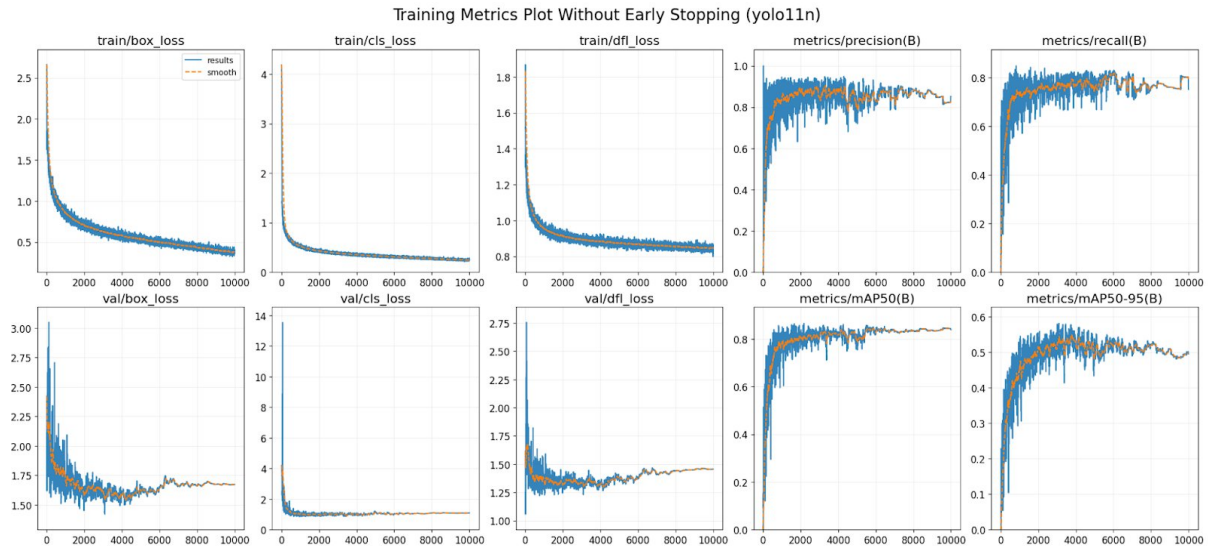


Figure 73: YOLO11n Training Metrics Without Early Stopping

Here in the 11th version of the object detection model we can see that there is a gradual decrease in training and box loss training and class and training dfl loss as well as validation and box loss validation/class and dfl loss. On the hand we can see there is a gradual there is a sudden increase in precision record in the similar data they are charts in the beginning but if we zoom in we can see there is a sudden drop in precision after 4000 iterations. This also caused us to implement a releasing function so that we can see the improvements of our modifications in the following part.

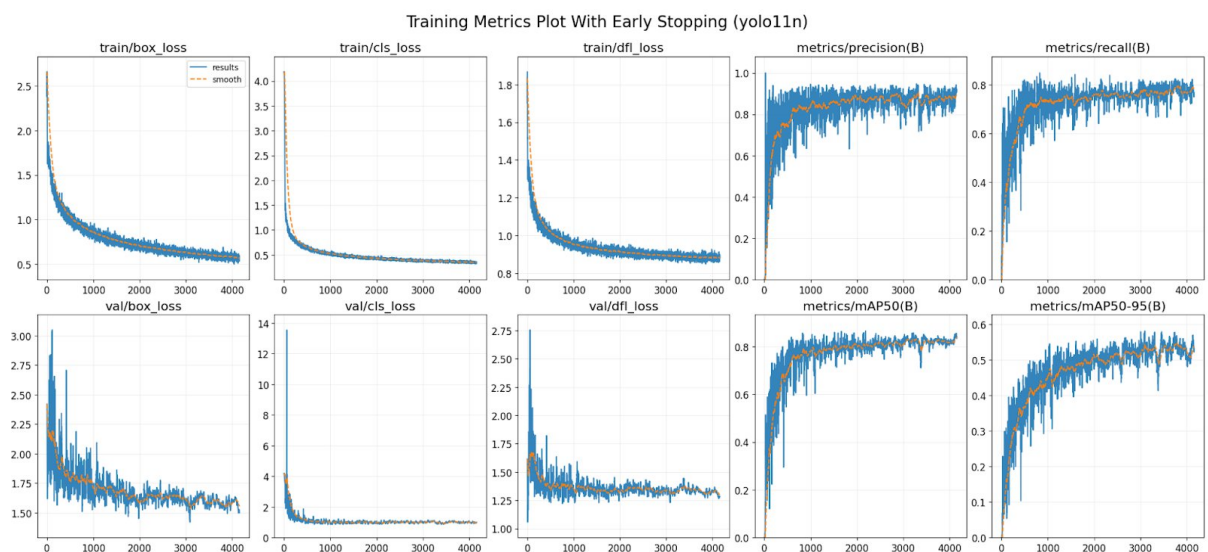


Figure 74: YOLO11n Training Metrics With Early Stopping

The effects of the early stopping function is clearly visible in this plot. Due to this custom function hall in training after a certain amount of error has increased or accumulated IT results in a better product compared to the full 10,000 iteration training cycle. The following graph shows the training characteristics of yolo12n.

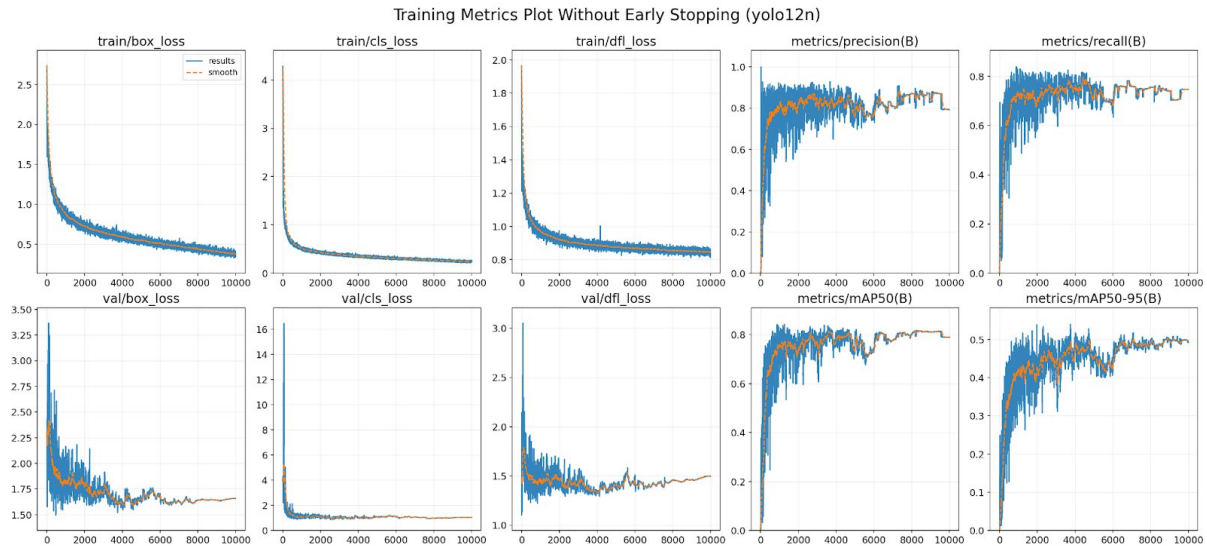


Figure 75: YOLO12n Training Metrics Without Early Stopping

After testing the 11th version we decided to upgrade our model to the 12th version but we decided to use the first like model or the n version. This allowed us to perform the full 10,000 iterations quite swiftly but we again see the same error as we have trained three models before and the number of error suddenly increasing over appearing around the same time indicates that there might be a problem with the data set itself if we zoom into the training versus dfl loss plot we can see there is a massive spike mass of comparative all of the others as this is clearly visible just over 4000 cycles this indicates that we need to apply in other early stopping function to prevent the sudden's bike in errors and increase in loss as a result this will also prevent the drop in precision which appears around the same point.

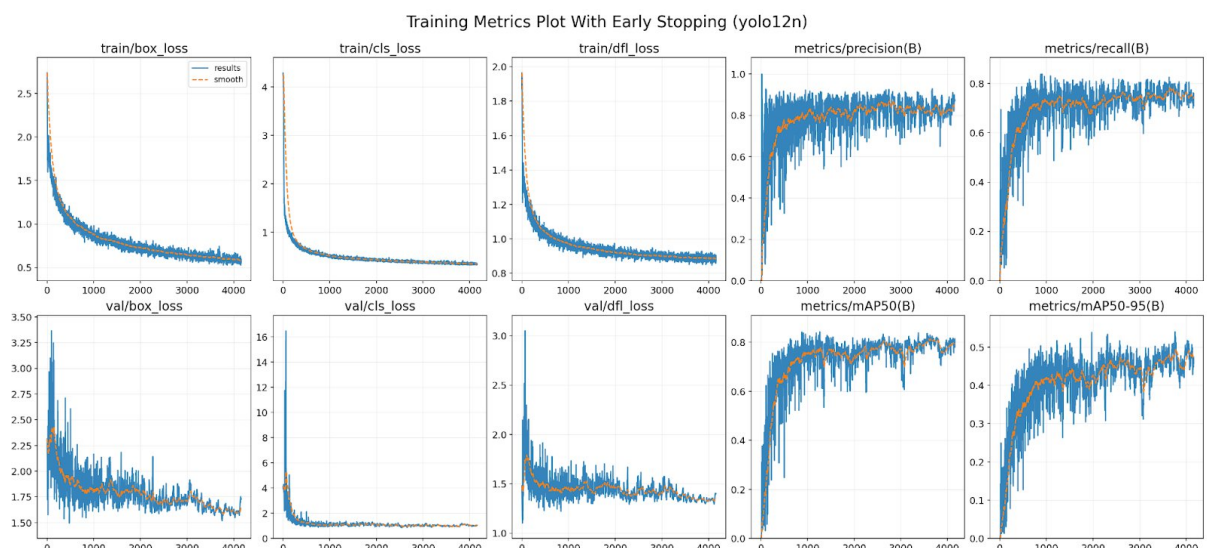


Figure 76: YOLO12n Training Metrics With Early Stopping

As we can clearly see from the plot above the model has stopped training just a bit above 4,000 cycles. This caused a noticeable bump in performance and ultimately resulted in a better performing model in the end run. This large bump in quality is the main reason why people implemented early stopping. In the early days before the ease of using early stopping functions AI researchers of yesteryear did this manually by repeatedly checking where they were in the versus plots vs where they are right now. This method gives the modern early stopping functionality that is so easily usable. People before us had to manually write functions to see if after a certain period of time there was no improvement in training parameters or an ongoing/drastic downward trend (or a spike in error) then to cease training immediately.

We have used the largest model that Ultralytics provide to date (excluding beta/testing builds as using canary builds might introduce such a large amount of uncertainty in the training process. We decided to train the yolo12s model, which is a large object detection model in the stable branch of the Ultralytics product lineup. Since this model is much much larger than other ones we have tested previously it takes more time per iteration compared to the light ('n' variants) models. As a result we could only perform only 2,000 training cycles properly. During this cycle there was an expected error curve in the models.

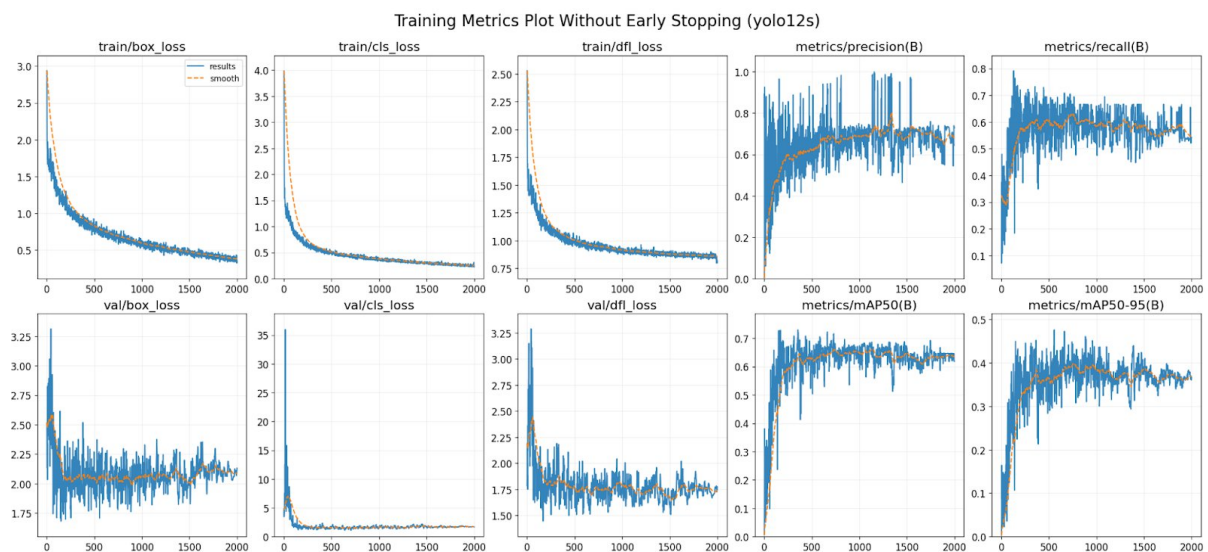


Figure 77: YOLO12s Training Metrics Without Early Stopping

Finally implementing an early stopping function here also netted us the same result as before. Models performance increased compared to the end of the full training cycle. Since this model is larger than the other light versions above and due to time constraints it did not reach the accuracy levels that we hoped for but it still showed promising results.

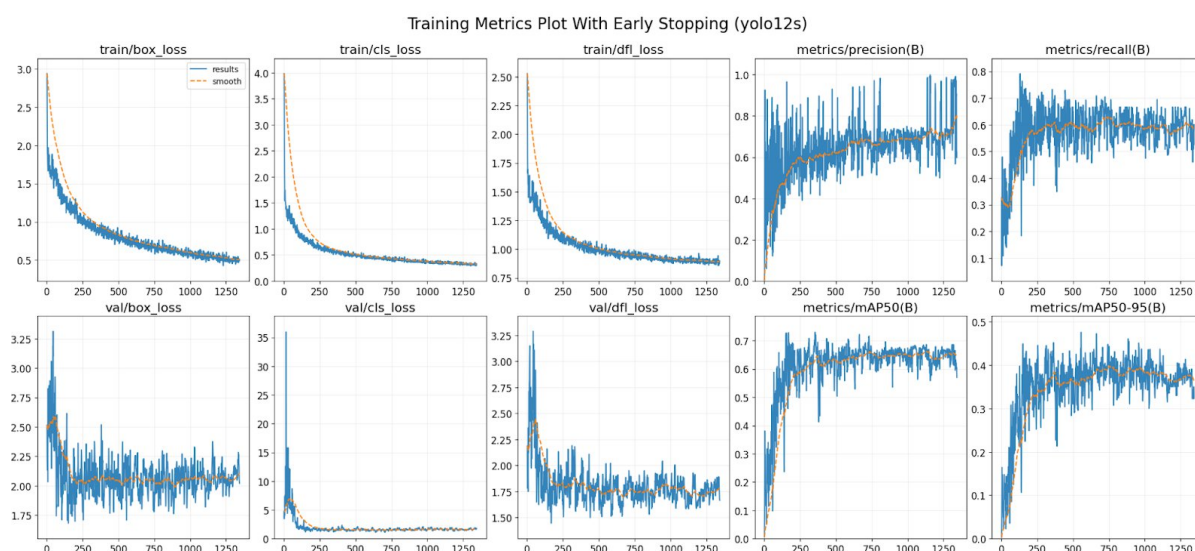


Figure 78: YOLO12s Training Metrics With Early Stopping

4.2.6.3. Performance

The following image shows the confusion matrix of yolov8n. There is quite a bit of noise caused by wrong detections. The following image is yolo11n. This noise is further reduced and so forth for yolo12n. But due to training limitations for the heavier yolo12s model (even though it is a small model, it is still massive compared to the nano variants) there is significant noise due to training limitations.

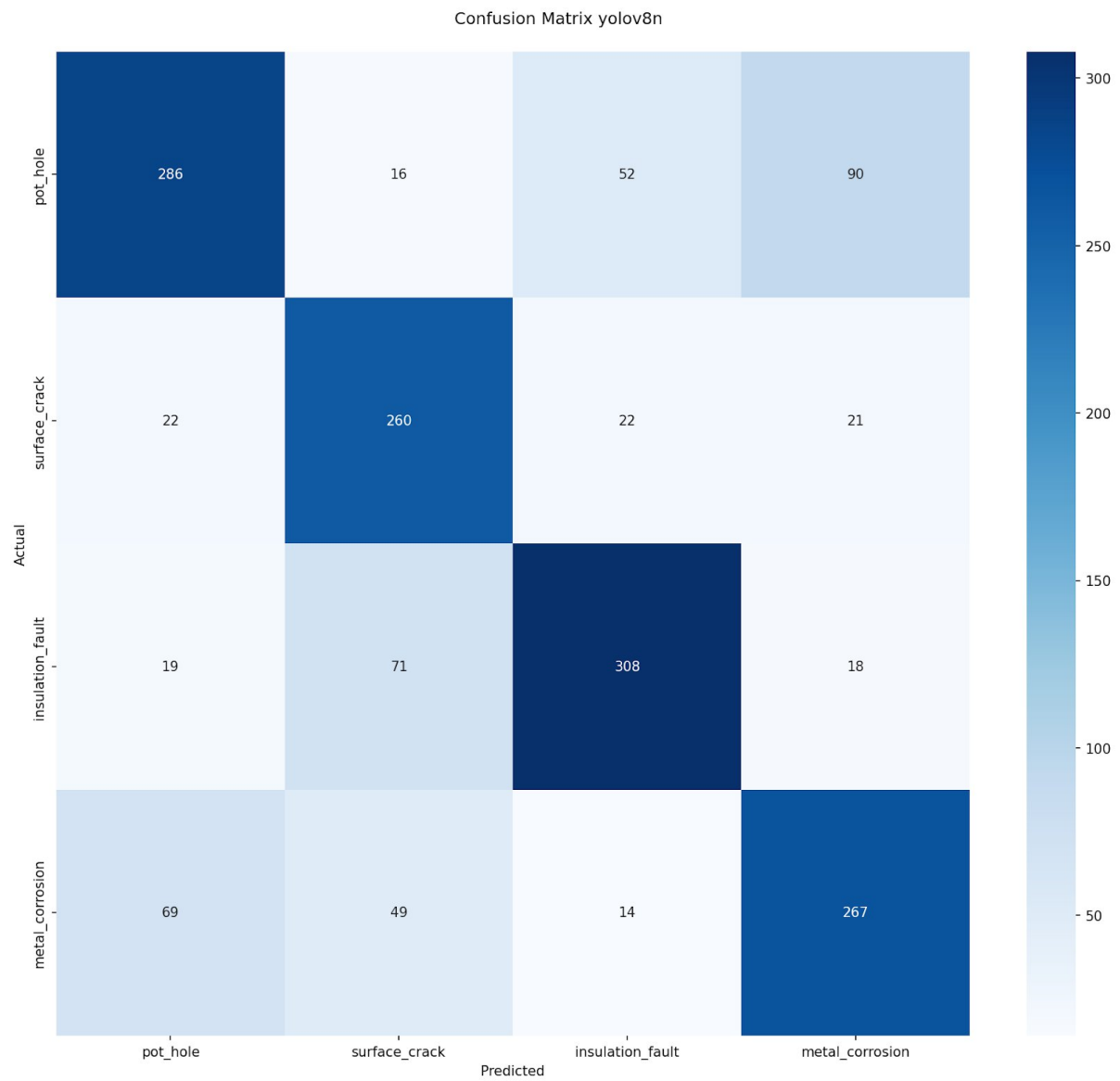


Figure 79: Confusion Matrix for YOLOv8

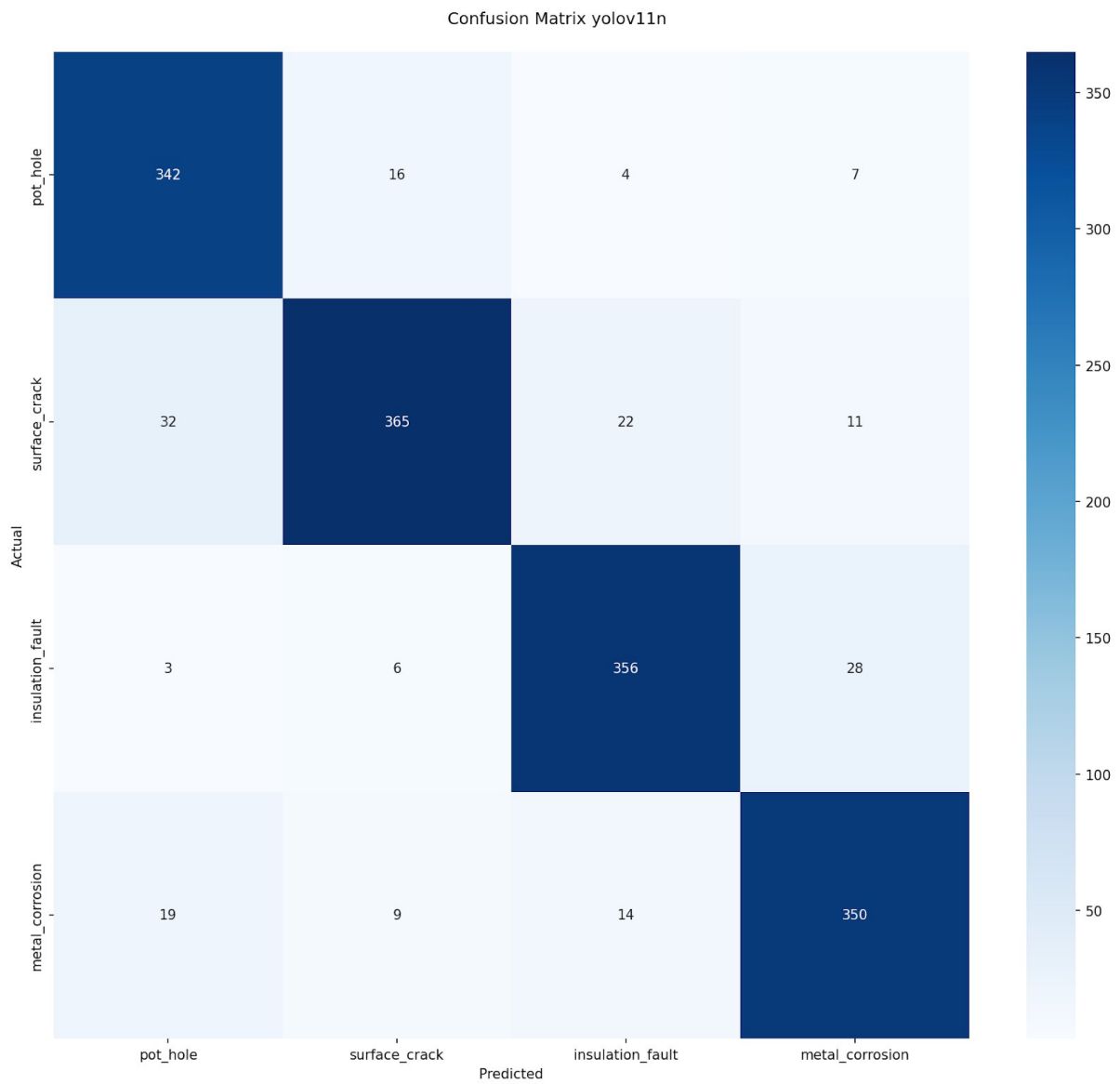


Figure 80: Confusion Matrix for YOLO11n

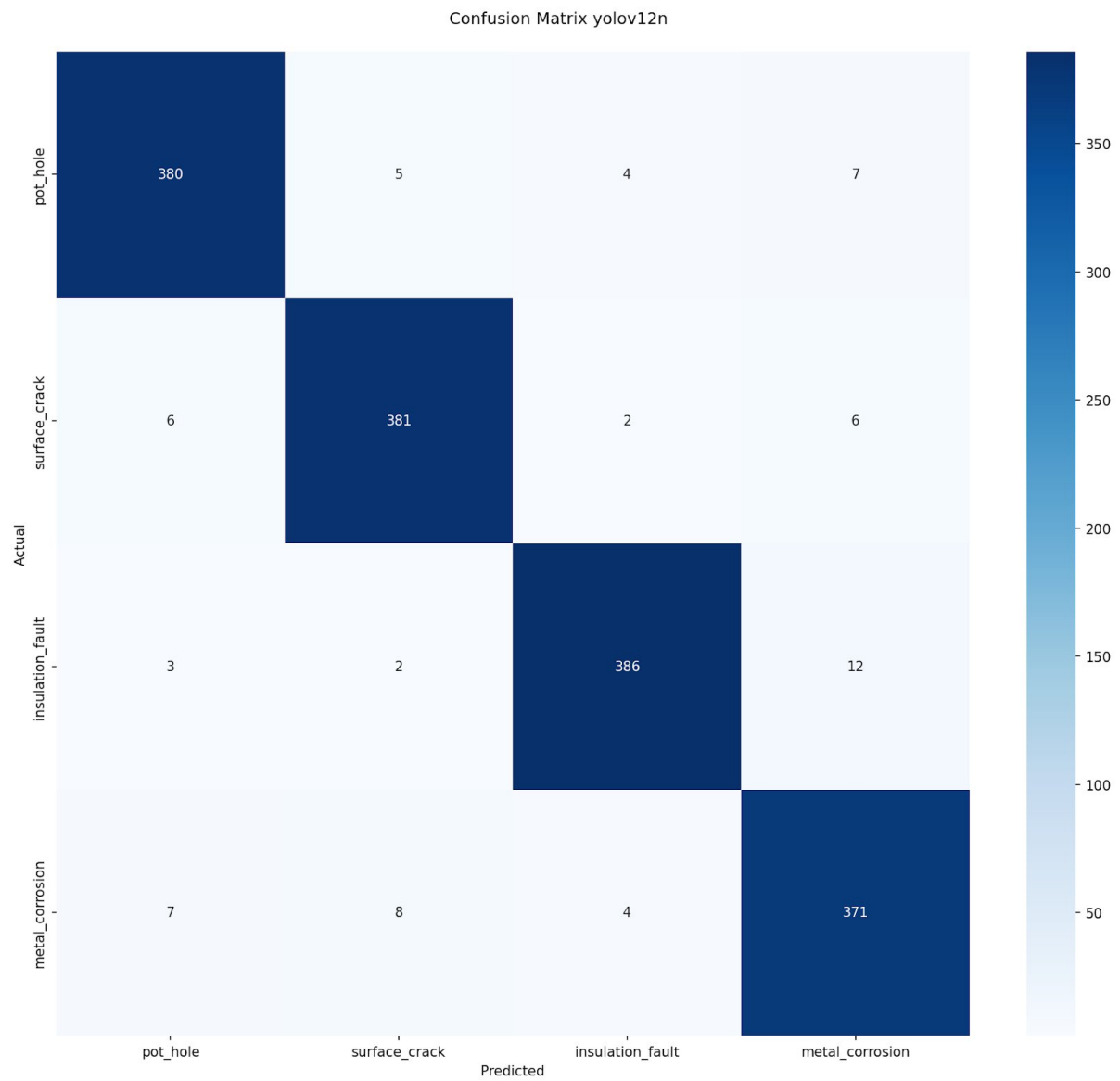


Figure 81: Confusion Matrix for YOLO12n

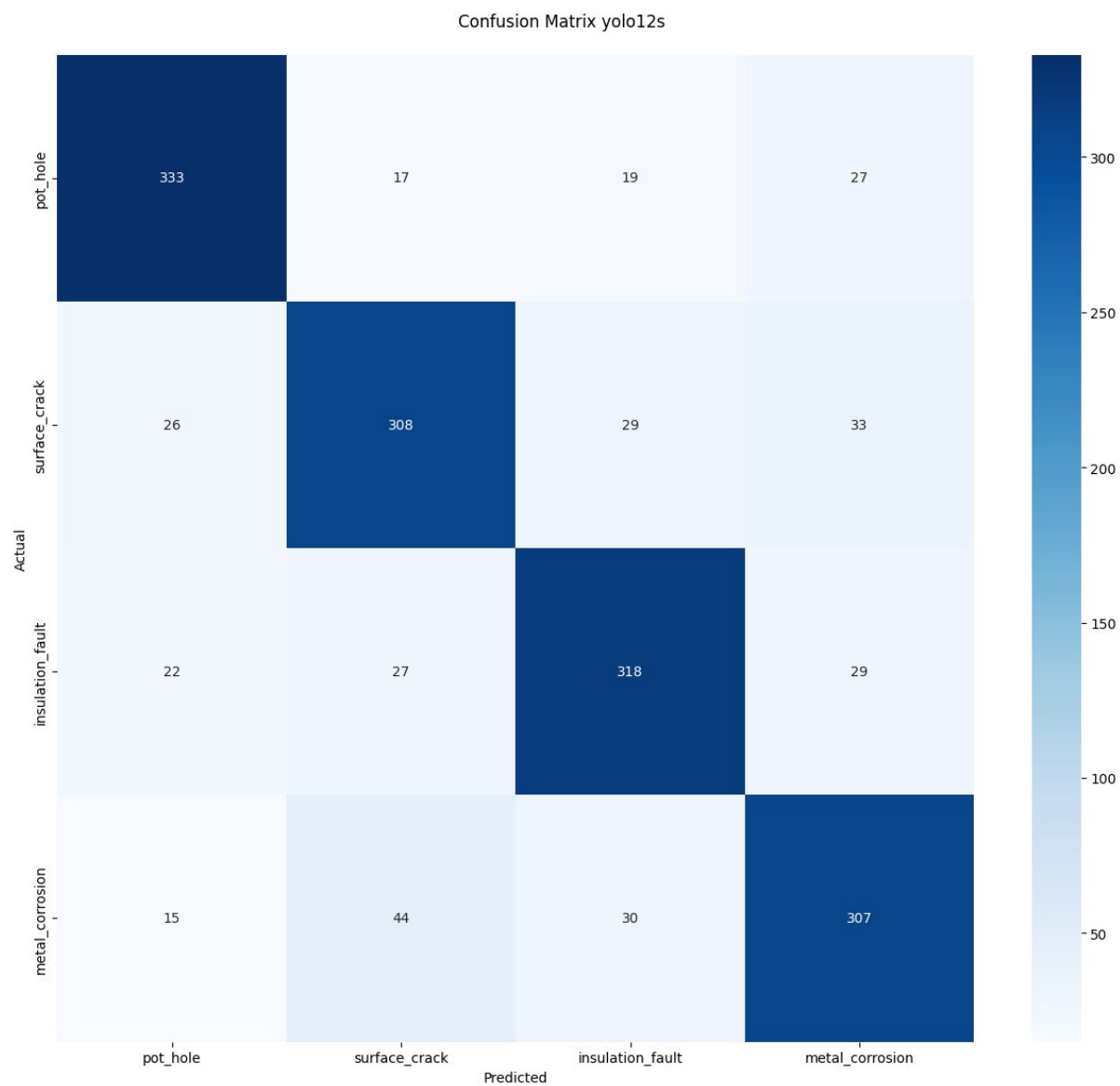


Figure 82: Confusion Matrix for YOLO12s



Figure 83: Crack Detection

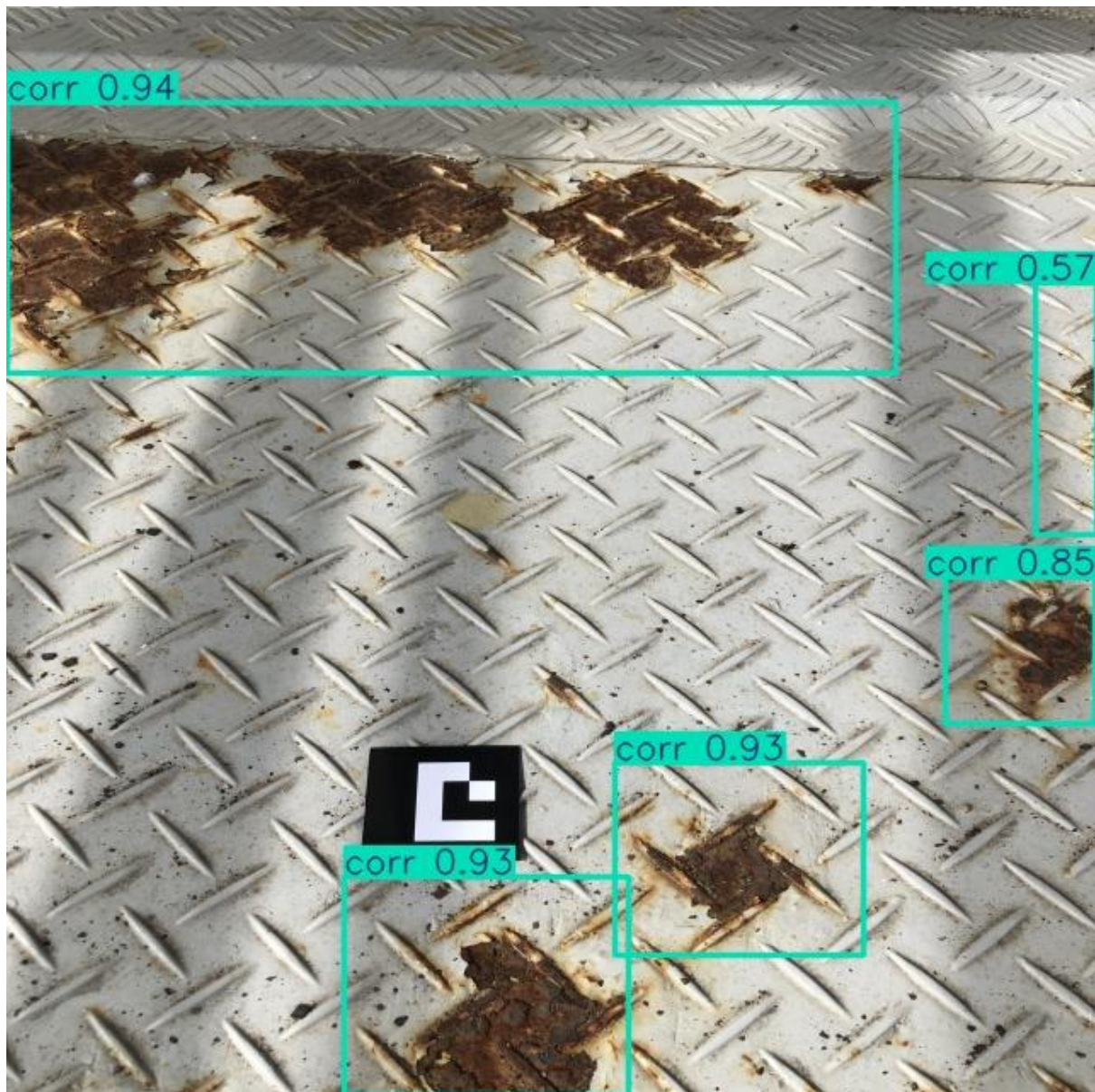


Figure 84: Corrosion Spot Detection

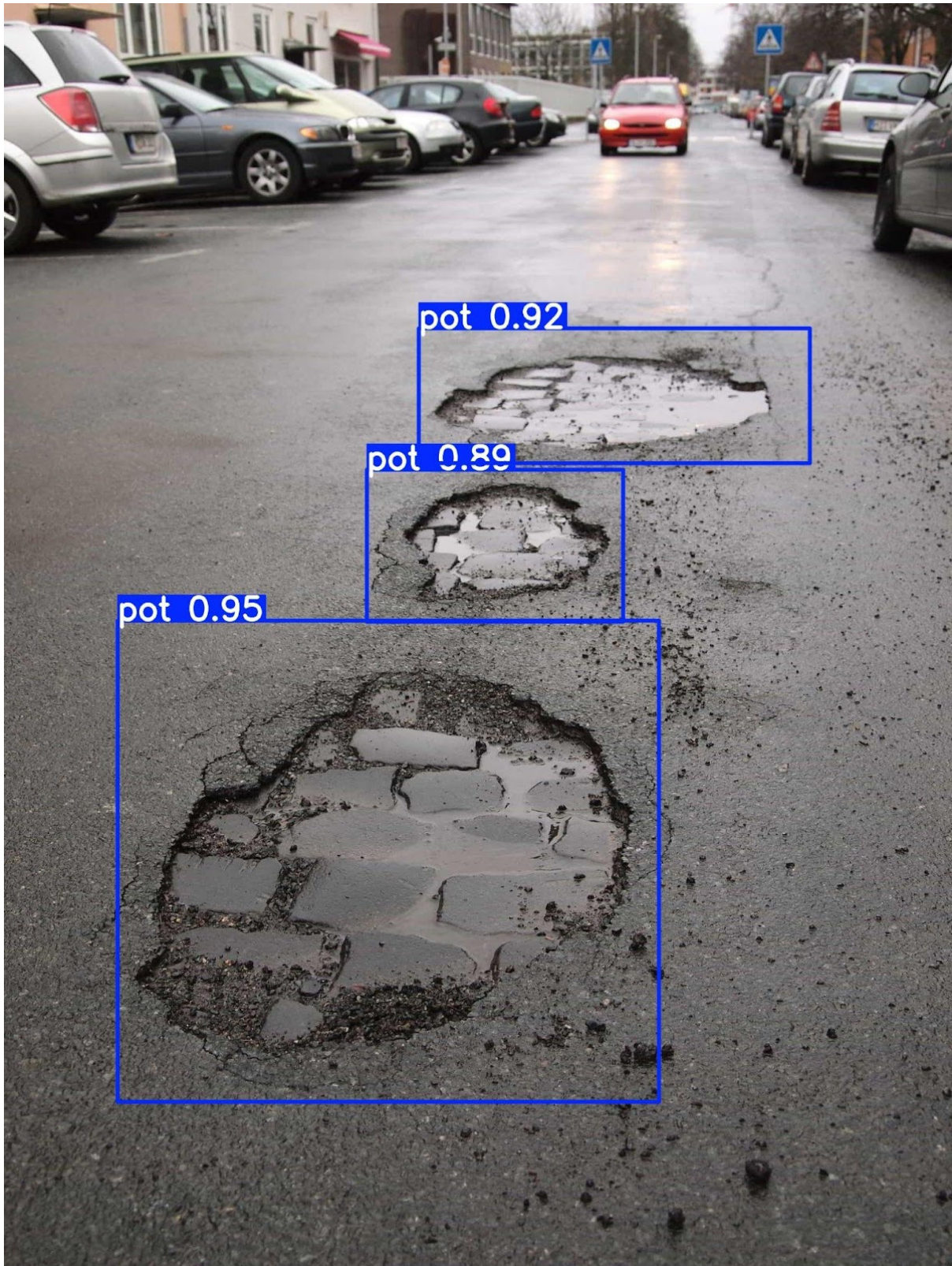


Figure 85: Pothole Detection

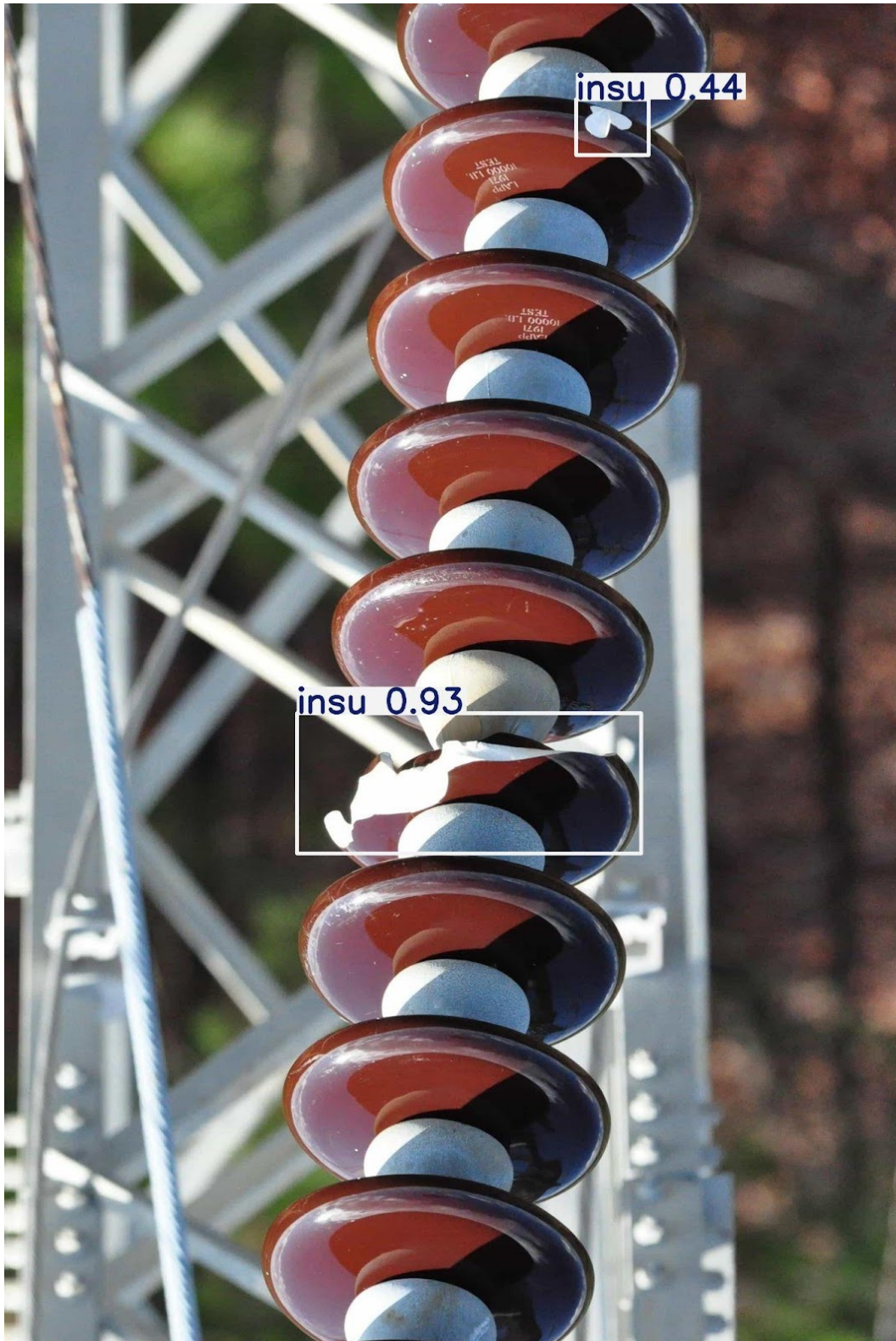


Figure 86: Insulator Fault Detection

4.2.7. Drone Simulation

4.2.7.1. Control Methodology

The control framework shown in the following figure, for the drone follows a cascaded, block to block approach. Each control mechanism flows step by step from high level position references down to low level motor commands. This design is layered which makes the system modular, easy to tune and robust in real time flight.

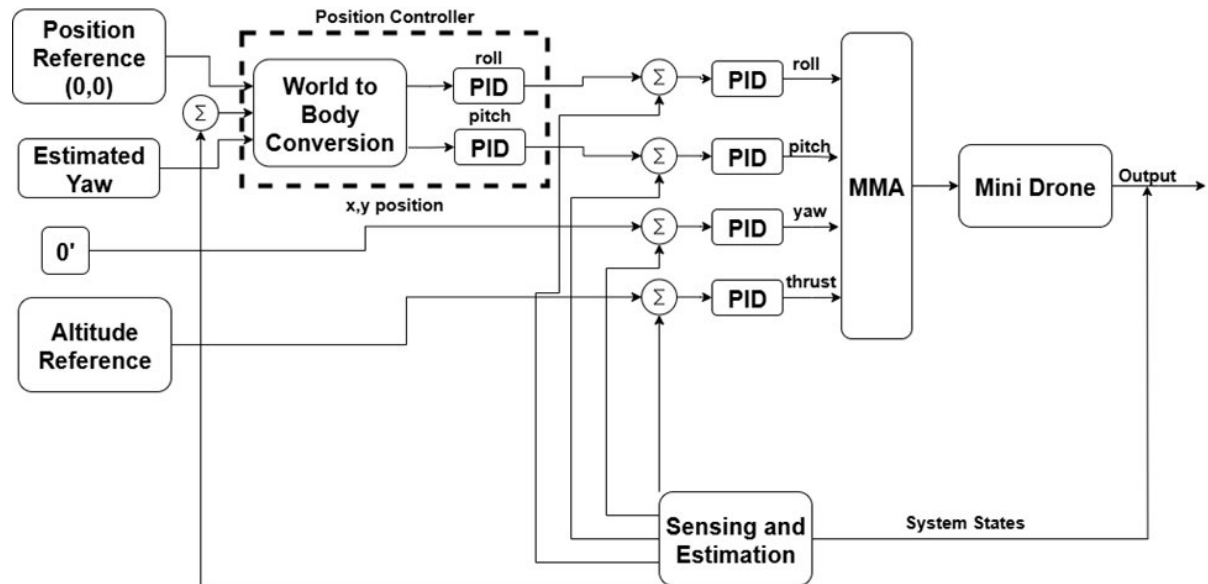


Figure 87: Entire Drone Control Methodology

4.2.7.2. Drone Architecture

External environmental factors like wind, humidity affect the control scheme. That's why it is directly connected to the maneuvering scheme and the quadcopter.

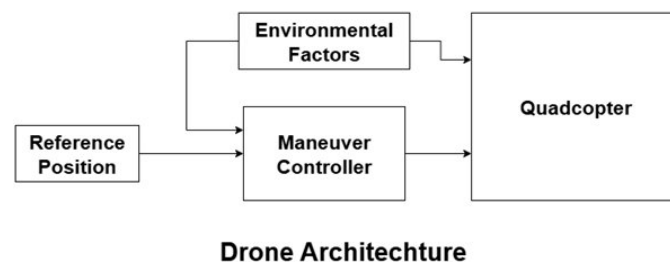


Figure 88: Environmental Factors on the Entire Architecture

4.2.7.3. Reference Position

The process starts with the position reference (left upper corner of the high level diagram of control architecture) which are x/y coordination, yaw, altitude in the world frame (control architecture diagram). This defines where the drone should go or which waypoint it should

follow. This reference position is constantly compared with the estimated position, which comes from the sensing and estimation block (sensor referencing diagram) that combines GPS, IMU and barometer data.

The world to body frame conversion block (sensor referencing diagram) turns the global position error into the drone's local body frame. Since the drone moves based on its own roll, pitch and yaw axis. This part is one of the most crucial parts.

Using the estimated yaw (heading), the block rotates the global error into body coordinates with a coordinate transformation matrix. Now the control signals match the drone's actual actuation axes. This ensures roll and pitch commands make sense for the drone's orientation.

Sensors capture real-time flight data. The sensing and estimation block (control architecture diagram) fuses multiple sources:

- **IMU:** Measures acceleration and angular velocity.
- **Barometer:** Measures altitude.
- **GPS or vision-based localization:** Provides global position.
- **Sensor fusion (e.g., EKF):** Combines noisy data into accurate state estimates.

This block outputs position, velocity, orientation, and altitude estimates, which feed back to earlier blocks to maintain control.

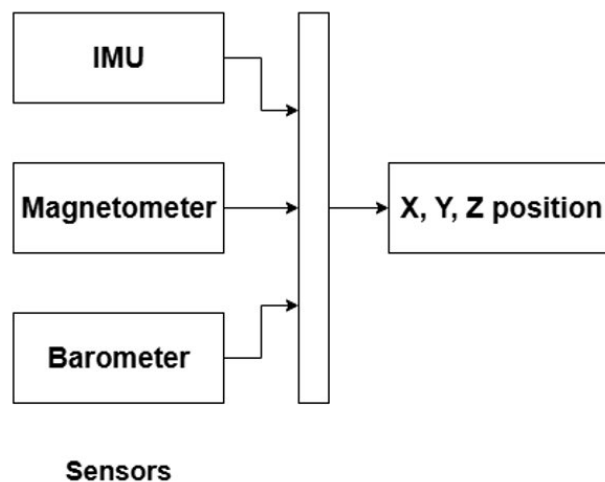


Figure 89: Referencing Using Sensors

4.2.7.4. Maneuver Control

Now the converted position errors go through PID controllers (control architecture diagram) for roll and pitch. Each PID has three parts:

- **Proportional (P):** Corrects immediately based on the current error.
- **Integral (I):** Fixes accumulated errors caused by wind or sensor bias.
- **Derivative (D):** Predicts future trends to reduce overshoot and improve damping.

These controllers (maneuvering system diagram) generate smooth roll and pitch commands to keep the drone on track, even with disturbances.

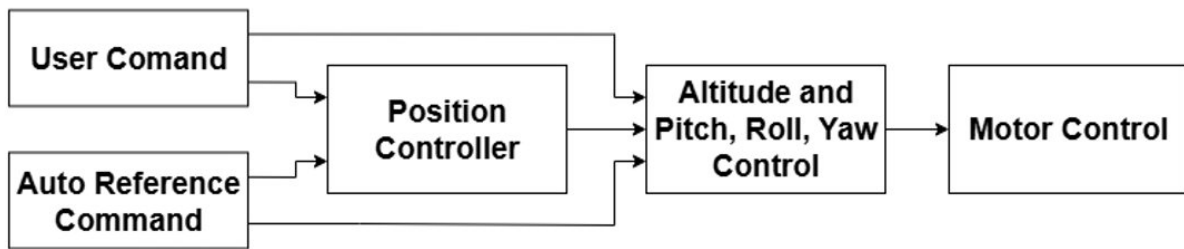


Figure 90: Maneuver Control

The roll and pitch commands enter the attitude control loop, which also uses real-time angular feedback. Dedicated PID controllers (control architecture diagram) for roll, pitch, and yaw manage the drone's rotation.

- **Roll and pitch controllers:** Make sure commanded angles are followed accurately, stabilizing lateral and forward motion.
- **Yaw controller:** Keeps heading stable and corrects drift.
- **Altitude PID controller:** Compares desired and estimated altitude, adjusting thrust to maintain vertical motion.

This cascaded setup (maneuvering system diagram) outer-loop position control feeding inner-loop attitude control ensures overall stability. The inner loop works fast to stabilize orientation, supporting the slower position loop.

The motor mixing algorithm (control architecture diagram) (MMA) converts roll, pitch, yaw, and thrust commands into signals for each motor. Each propeller affects these motions differently. The MMA uses a predefined mixing matrix to balance the inputs.

- Increasing thrust on all motors raises altitude.
- More thrust on one side rolls the drone.
- More thrust on front or rear tilts the drone forward or backward (pitch).
- Opposite diagonal motors adjust yaw through counter-torque.

This ensures motors work together to achieve the desired motion.

The drone executes motor commands, creating forces and torques that move it. The dynamics determine its actual motion.

- **Translational dynamics:** Move the drone in x, y, z directions via thrust.
- **Rotational dynamics:** Rotate the drone based on torque differences.
- **Coupled effects:** Roll, pitch, and yaw affect each other, requiring continuous correction.

The result is the drone's updated physical state.

The final result is stable flight and accurate trajectory tracking. The drone continuously adjusts its position, orientation, and altitude to match the reference. This method handles disturbances like wind, sensor noise, or payload changes, ensuring robust and reliable operation.

4.2.7.5. Aerodynamic Effect Analysis of Different Flight Patterns

4.2.7.5.1. Medium Density Zigzag

Waypoint Map

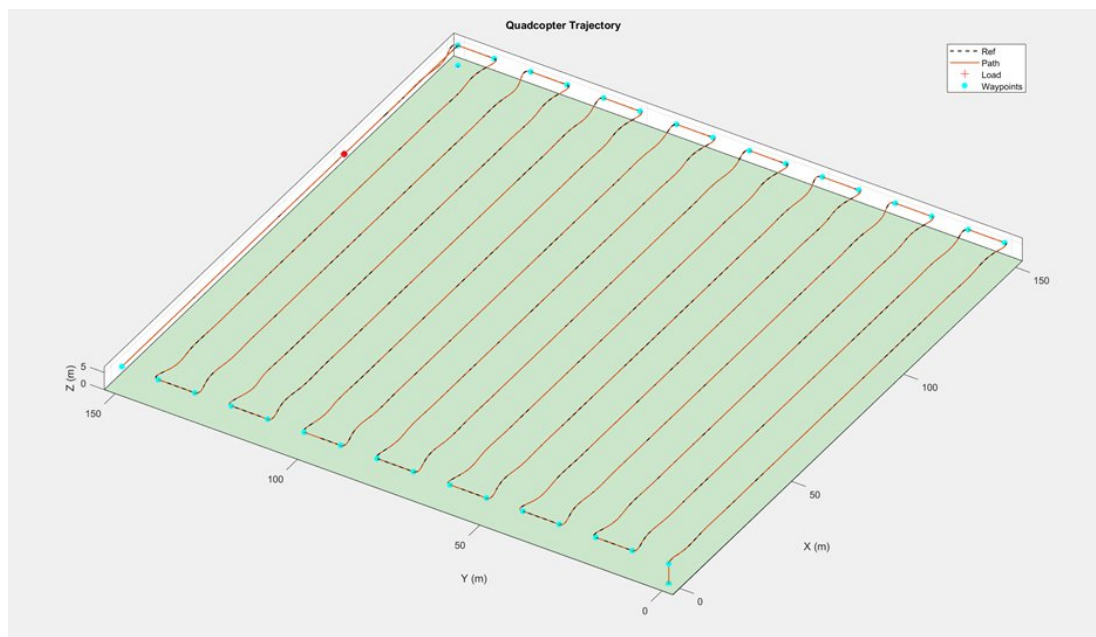


Figure 91: Quadcopter Trajectory for Medium Density Zigzag

Spatial Position Sampling:

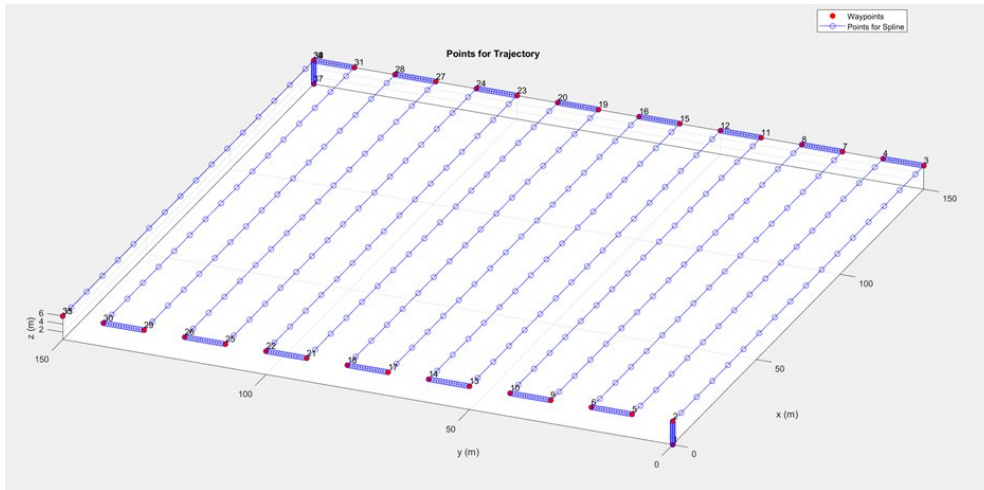


Figure 92: Trajectory Verification Points During Drone Flight for Medium Density Zigzag

Spatial Error Correction:

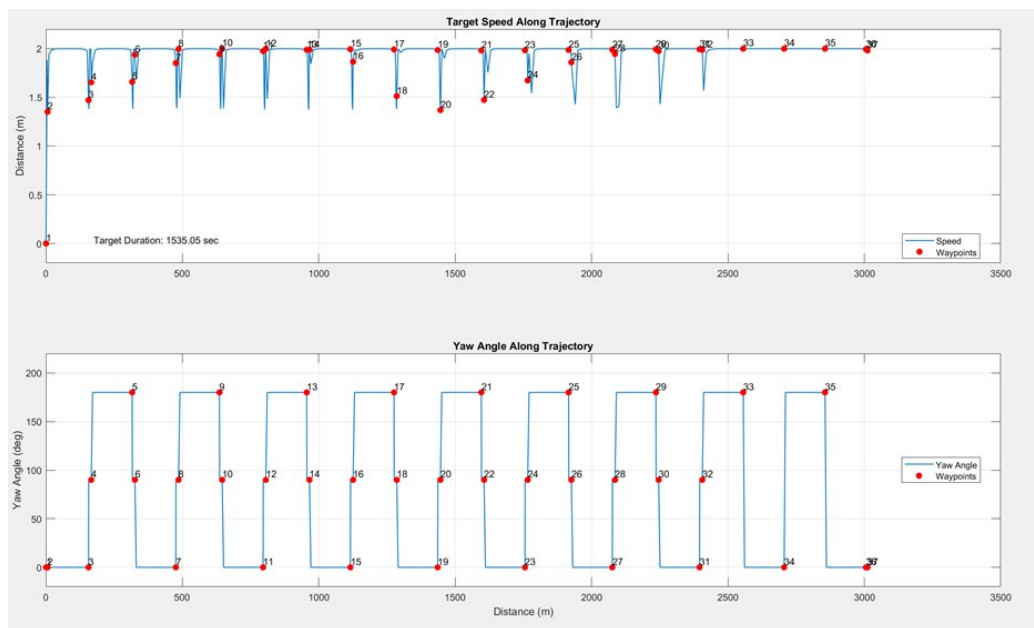


Figure 93: Trajectory Error Correction by Speed & Yaw Adjustment for Medium Density Zigzag

Aerodynamic Profile Analysis:

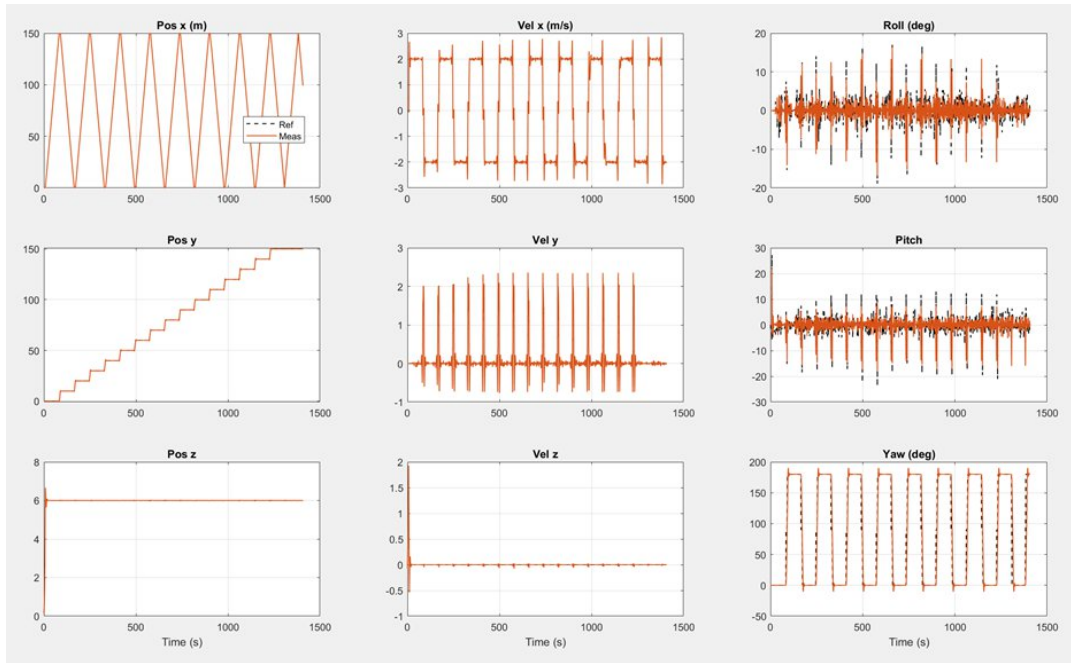


Figure 94: Aerodynamic Profile and Trajectory Tracking of the Drone for Medium Density Zigzag

Battery Behavior Over Flight Duration:

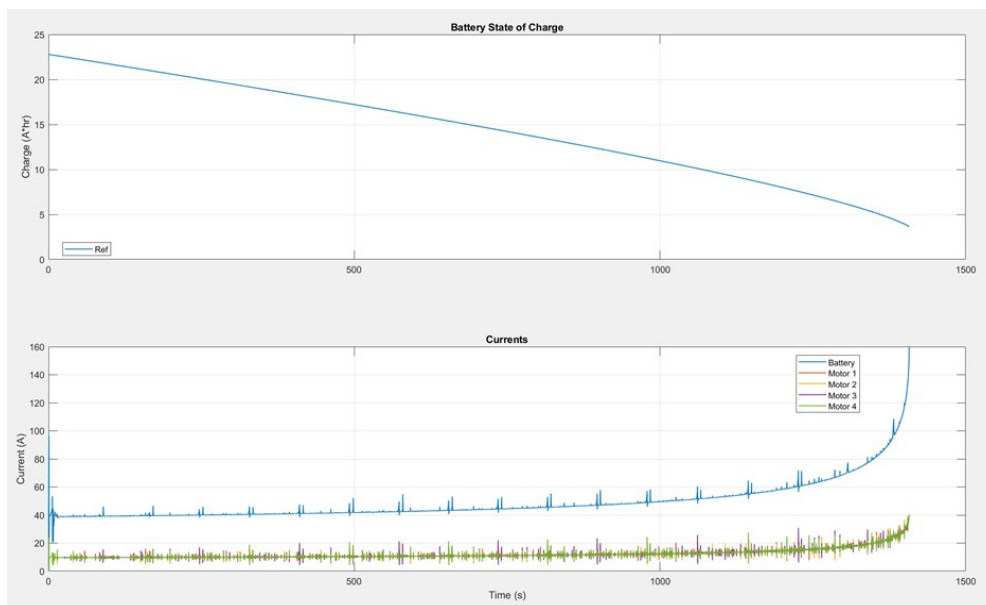


Figure 95: Battery Discharge and Motor Power Dynamics for Medium Density Zigzag

4.2.7.5.2. High Density Zigzag

Waypoint Map:

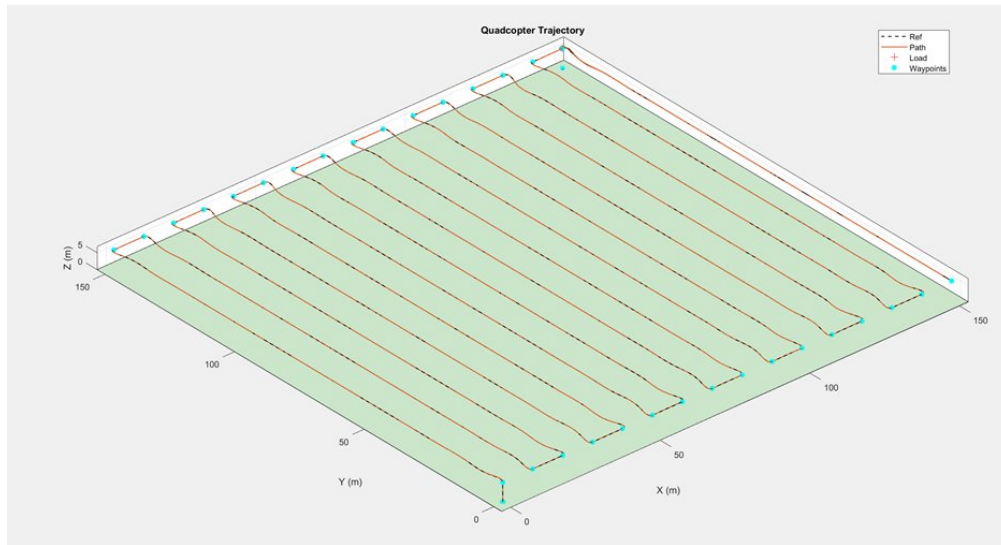


Figure 96: Quadcopter Trajectory for High Density Zigzag

Spatial Position Sampling:

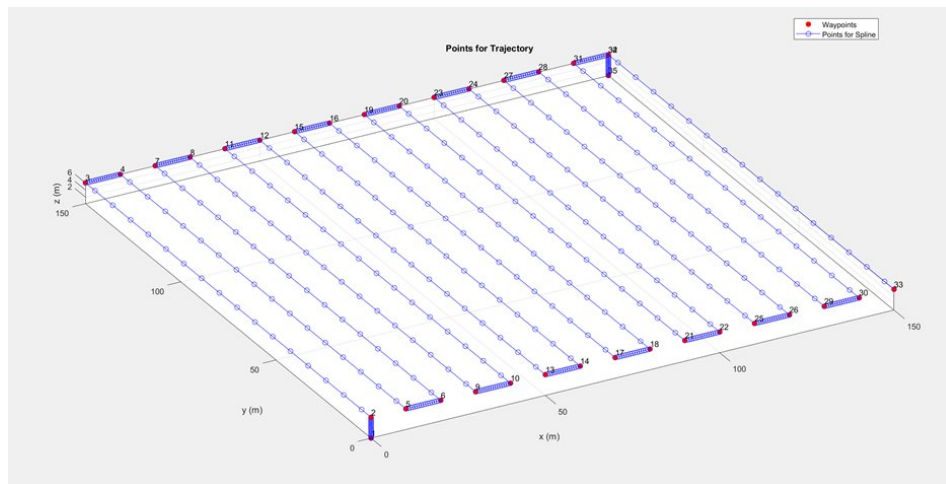


Figure 97: Trajectory Verification Points During Drone Flight for High Density Zigzag

Spatial Error Correction:

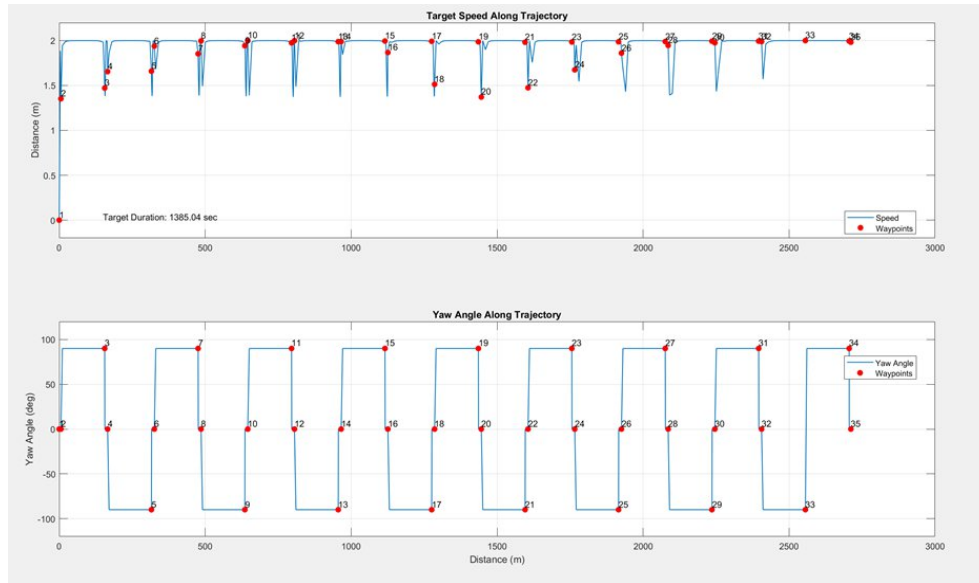


Figure 98: Trajectory Error Correction by Speed & Yaw Adjustment for High Density Zigzag

Aerodynamic Profile Analysis:

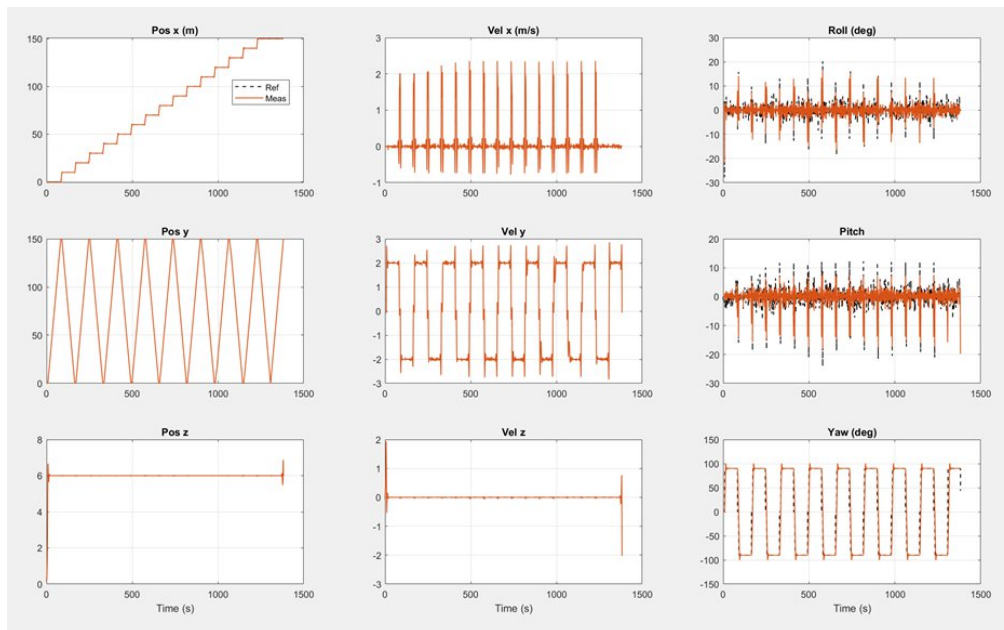


Figure 99: Aerodynamic Profile and Trajectory Tracking of the Drone for High Density Zigzag

Battery Behavior Over Flight Duration:

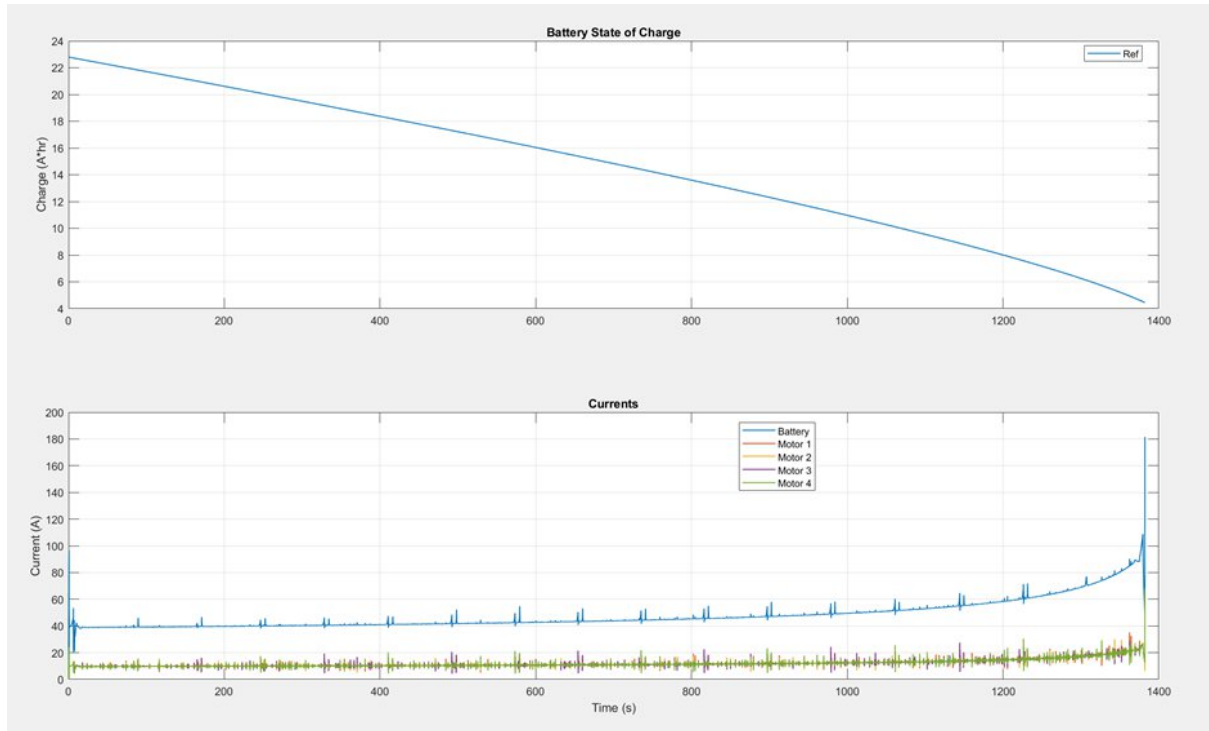


Figure 100: Battery Discharge and Motor Power Dynamics for High Density Zigzag

4.2.7.5.3. Quadrilateral Multi-pass Zigzag

Waypoint Map:

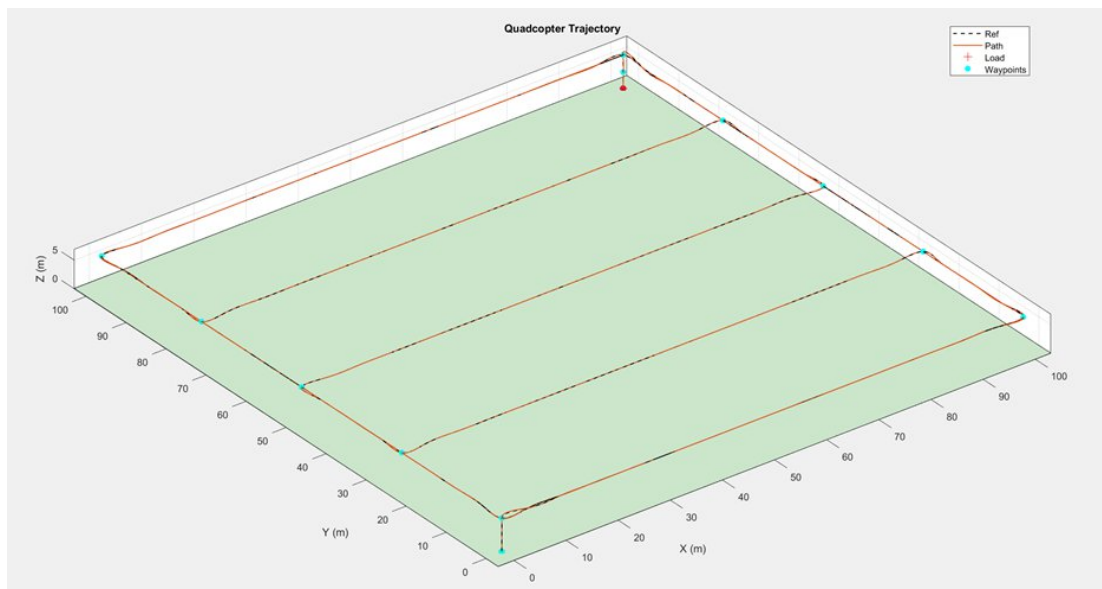


Figure 101: Quadcopter Trajectory for Quadrilateral Multi-pass Zigzag

Spatial Position Sampling:

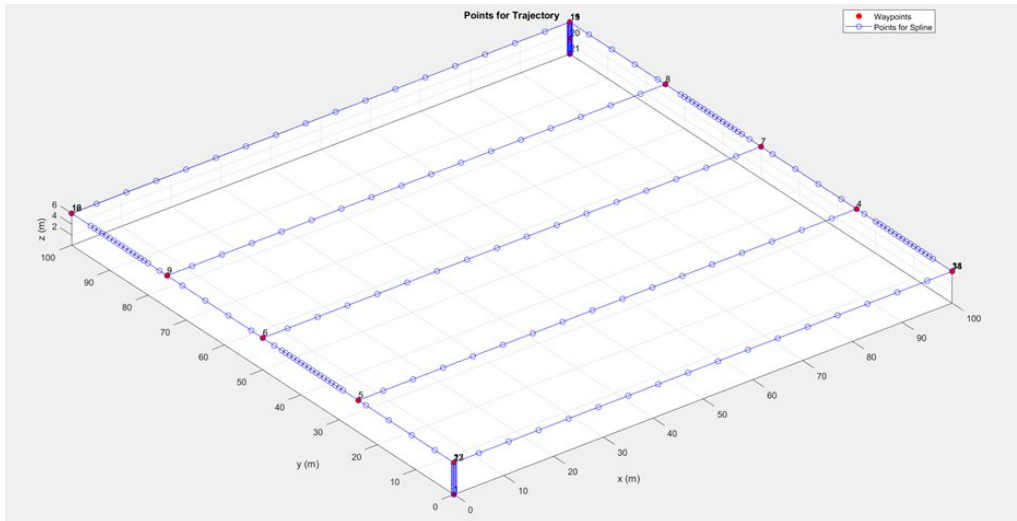


Figure 102:: Trajectory Verification Points During Drone Flight for Quadrilateral Multi-pass Zigzag

Spatial Error Correction:

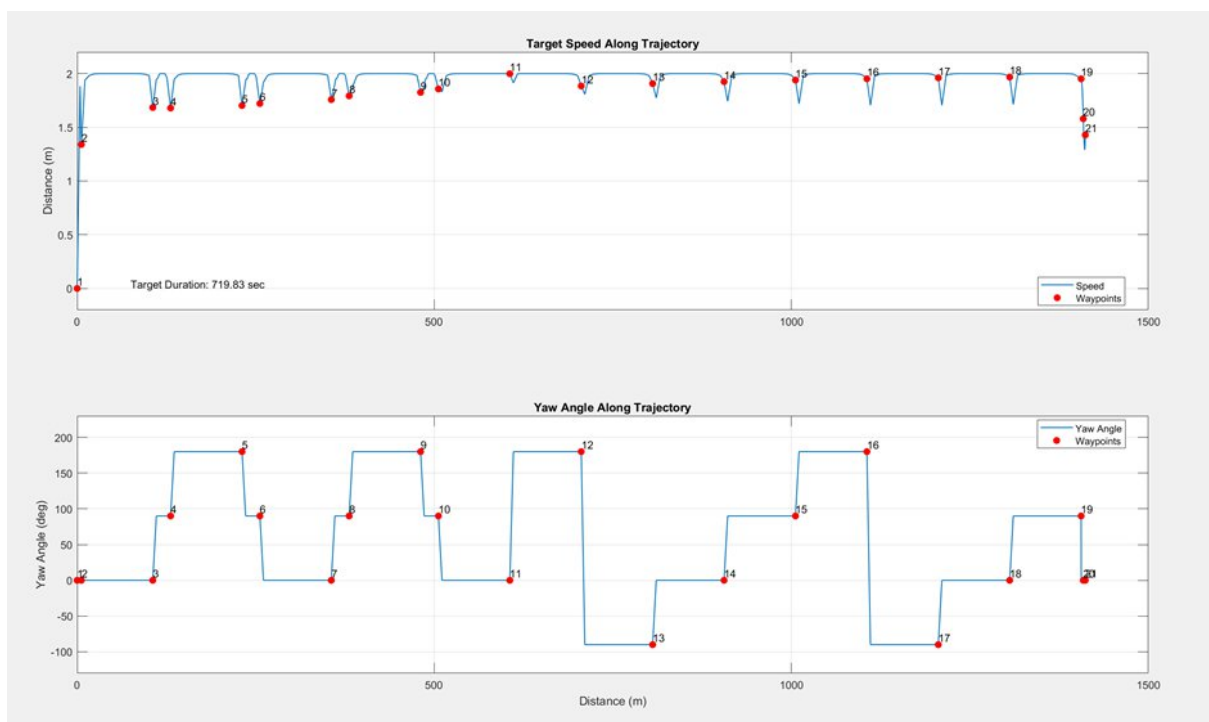


Figure 103: Trajectory Error Correction by Speed & Yaw Adjustment for Quadrilateral Multi-pass Zigzag

Aerodynamic Profile Analysis:

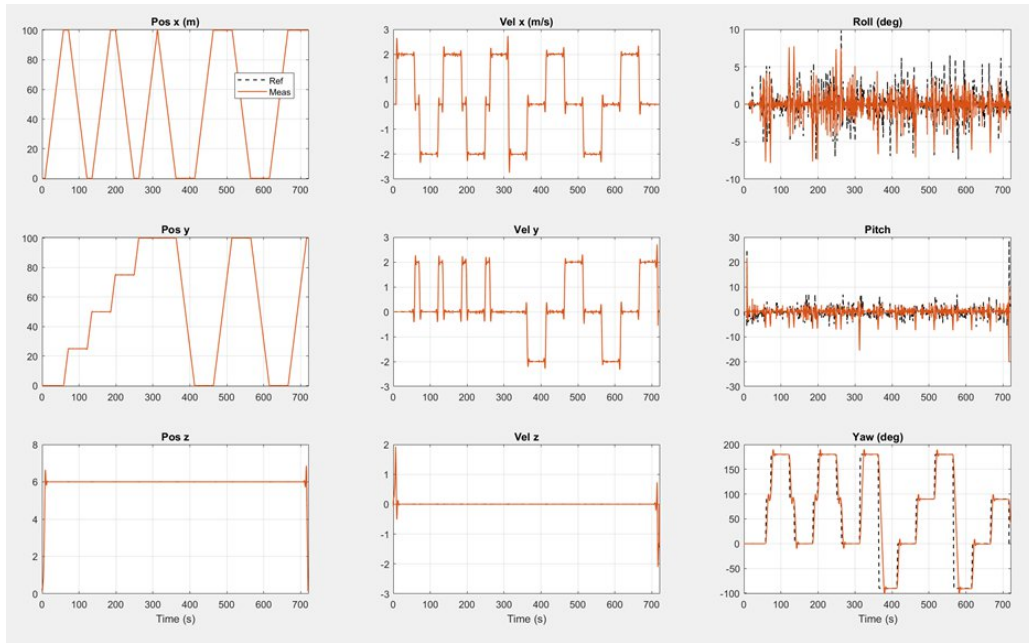


Figure 104: Aerodynamic Profile and Trajectory Tracking of the Drone for Quadrilateral Multi-pass Zigzag

Battery Behavior Over Flight Duration:

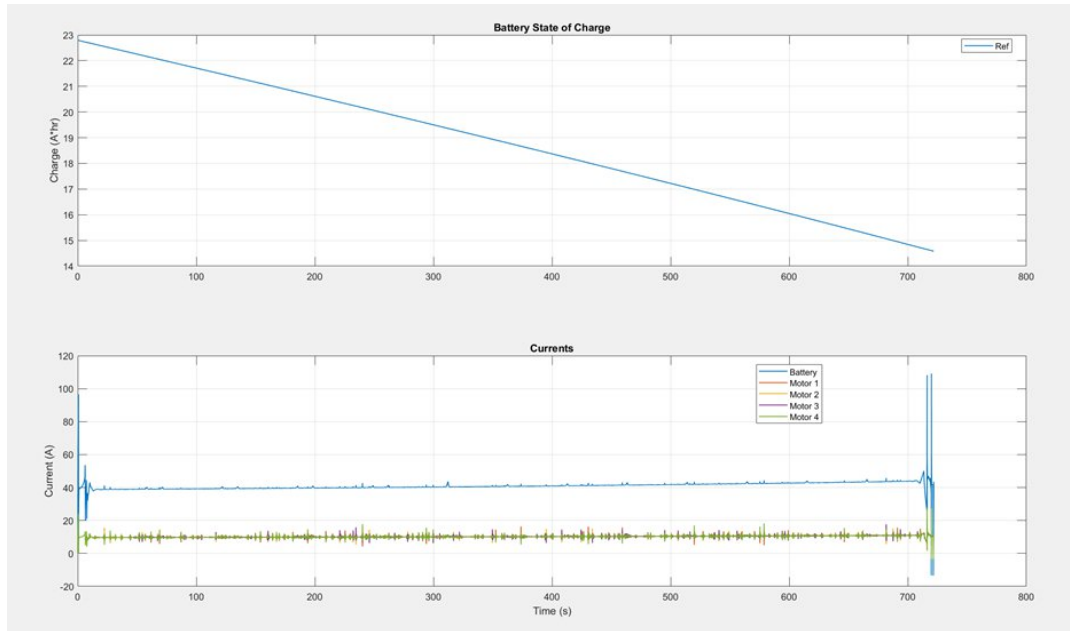


Figure 105: Battery Discharge and Motor Power Dynamics for Quadrilateral Multi-pass Zigzag

4.2.7.5.4. Staggered Continuous Zigzag

Waypoint Map:

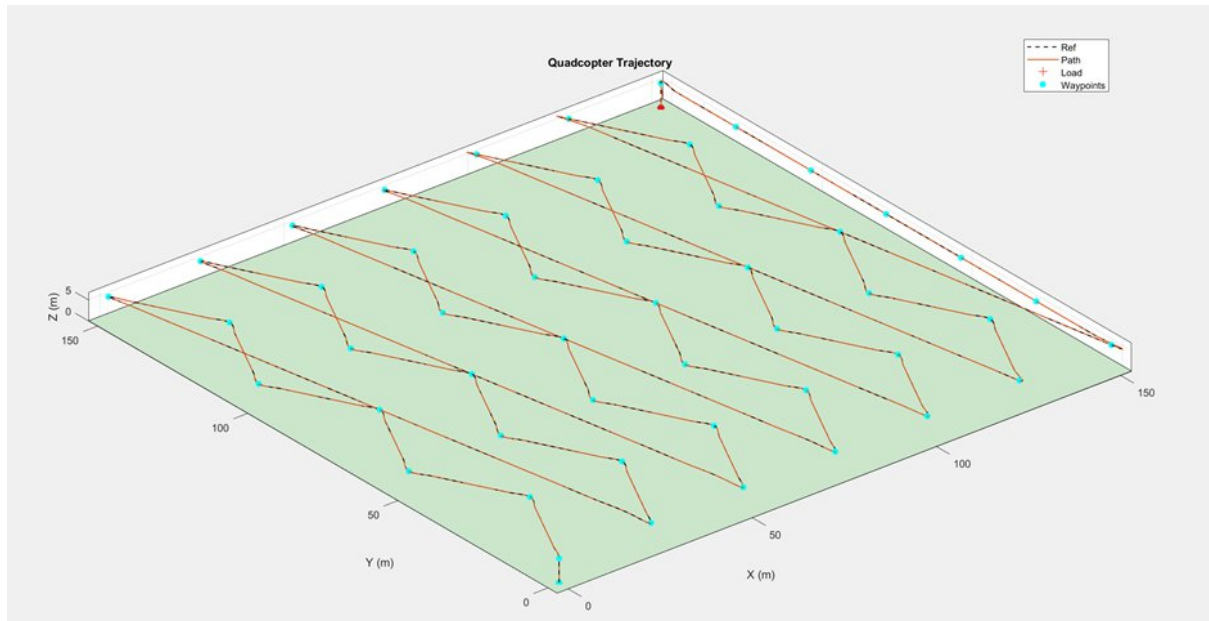


Figure 106: Quadcopter Trajectory for Staggered Continuous Zigzag

Spatial Position Sampling:

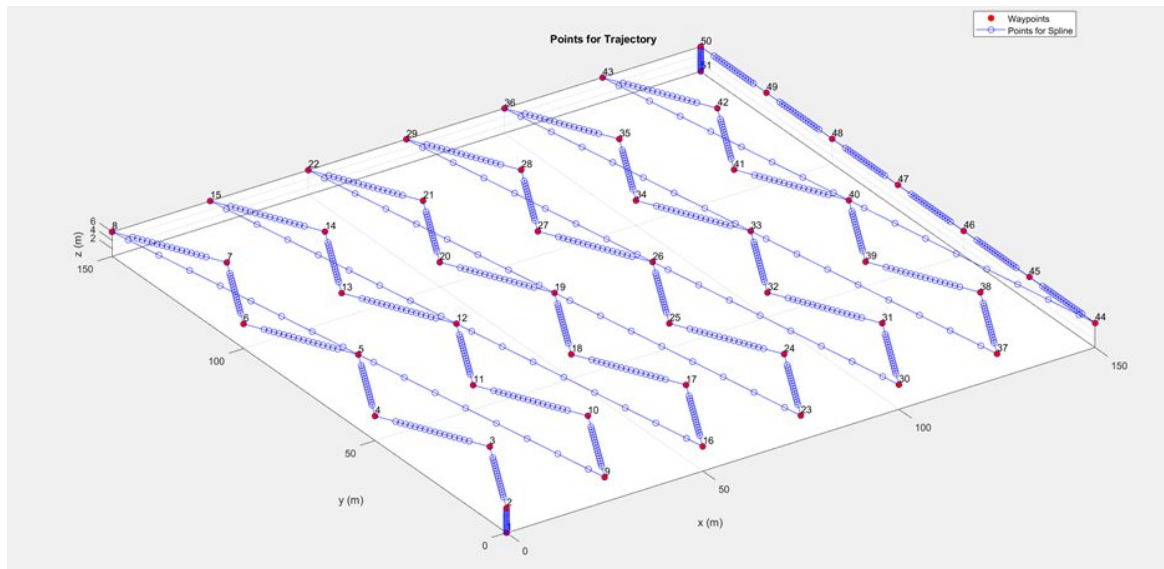


Figure 107: Trajectory Verification Points During Drone Flight for Staggered Continuous Zigzag

Spatial Error Correction:

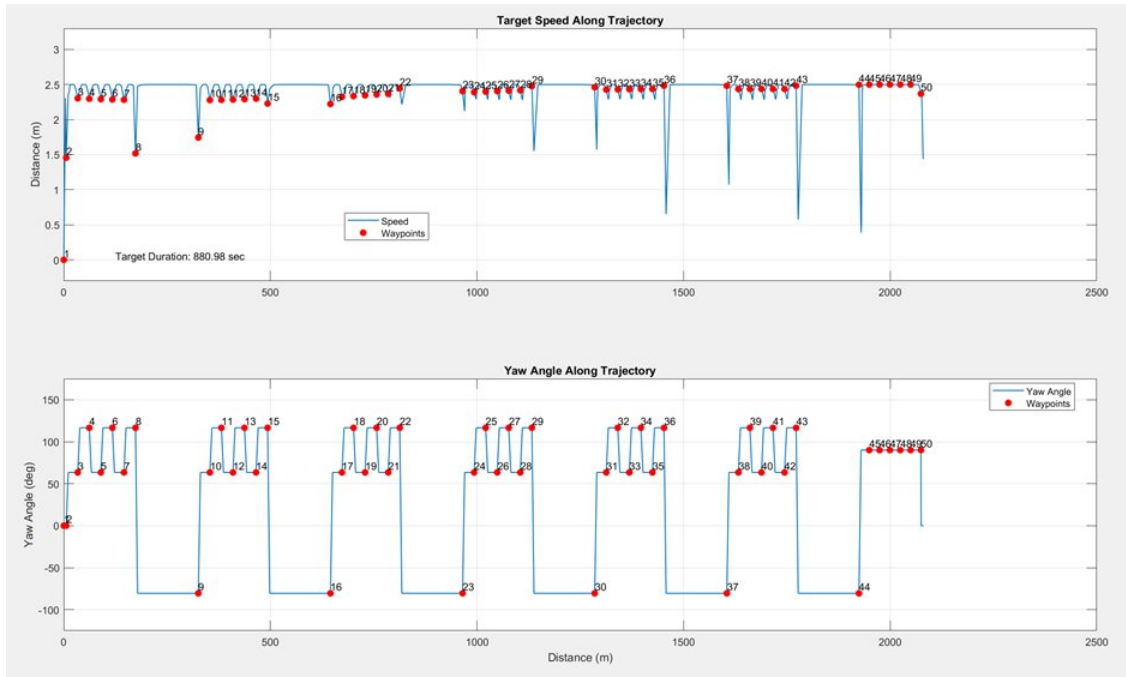


Figure 108: Trajectory Error Correction by Speed & Yaw Adjustment for Staggered Continuous Zigzag

Aerodynamic Profile Analysis:

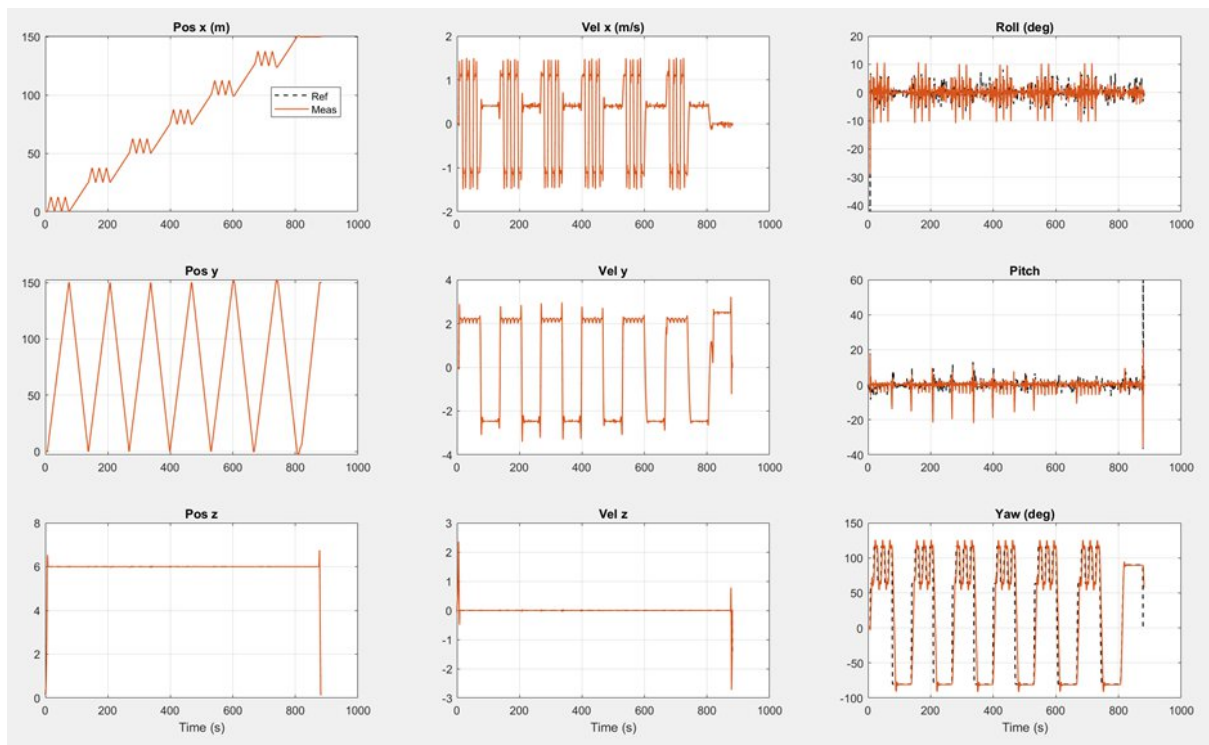


Figure 109: Aerodynamic Profile and Trajectory Tracking of the Drone for Staggered Continuous Zigzag

Battery Behavior Over Flight Duration:

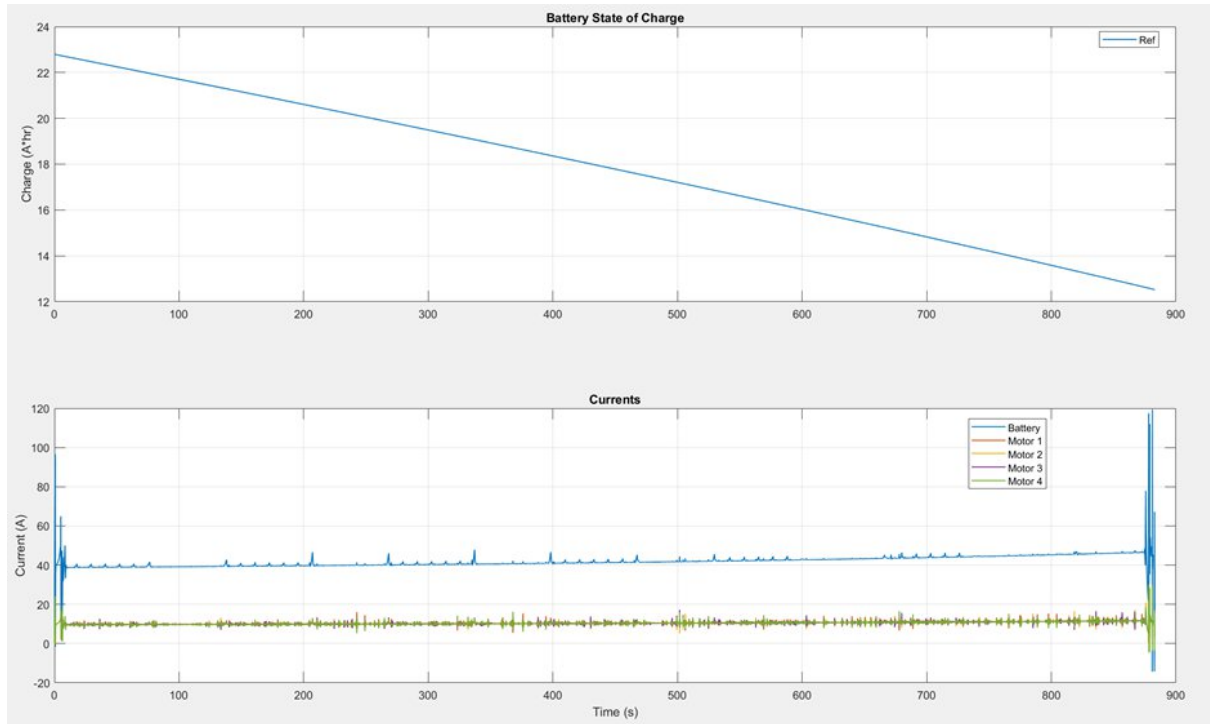


Figure 110: Battery Discharge and Motor Power Dynamics for Staggered Continuous Zigzag

4.2.7.5.5. Maximal Width Staggered Continuous Zigzag

Waypoint Map:

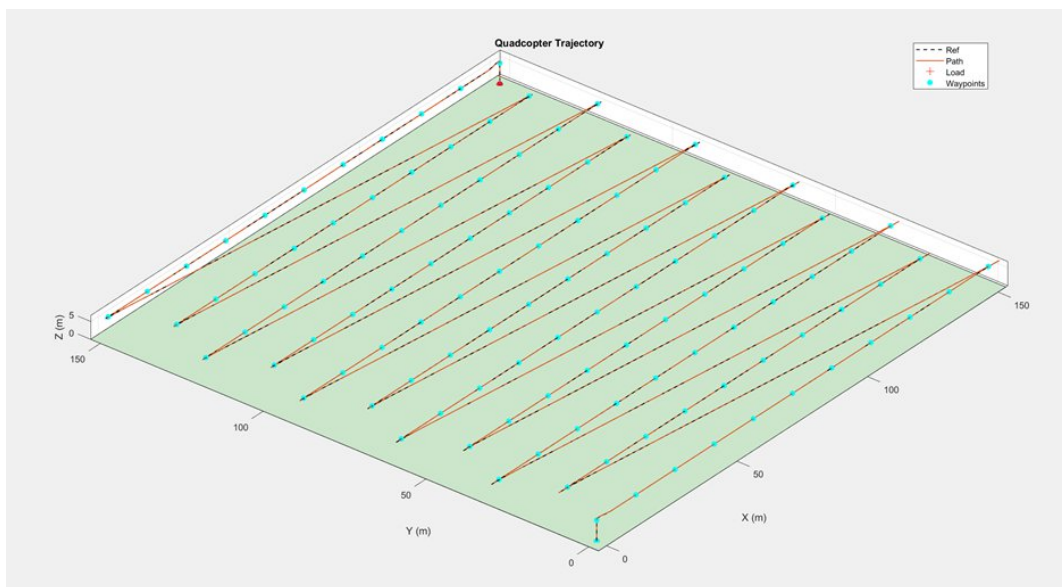


Figure 111: Quadcopter Trajectory for Maximal Width Staggered Continuous Zigzag

Spatial Position Sampling:

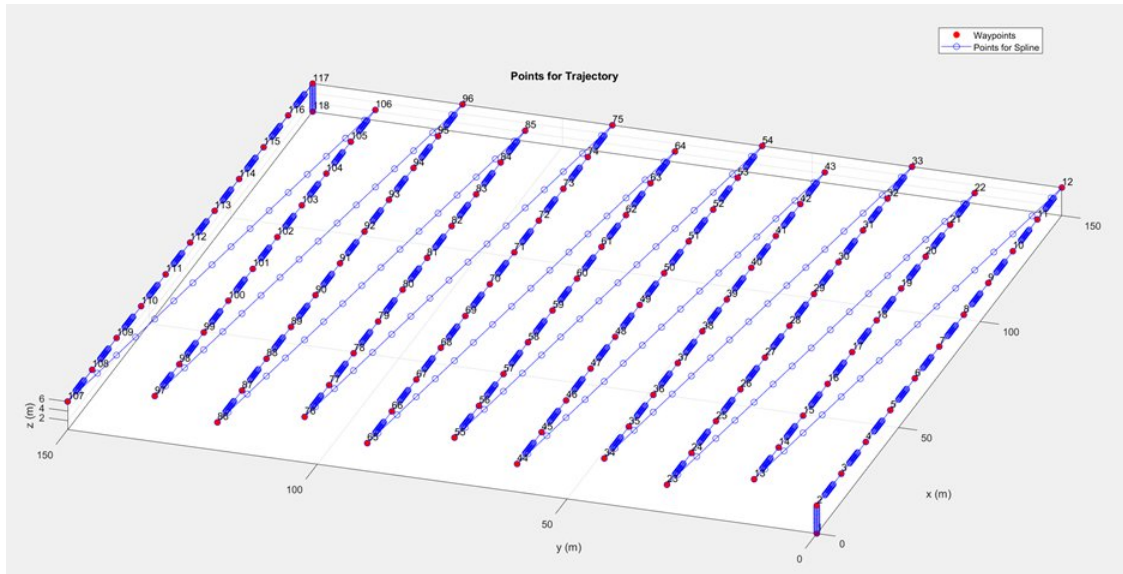


Figure 112: Trajectory Verification Points During Drone Flight for Maximal Width Staggered Continuous Zigzag

Spatial Error Correction:

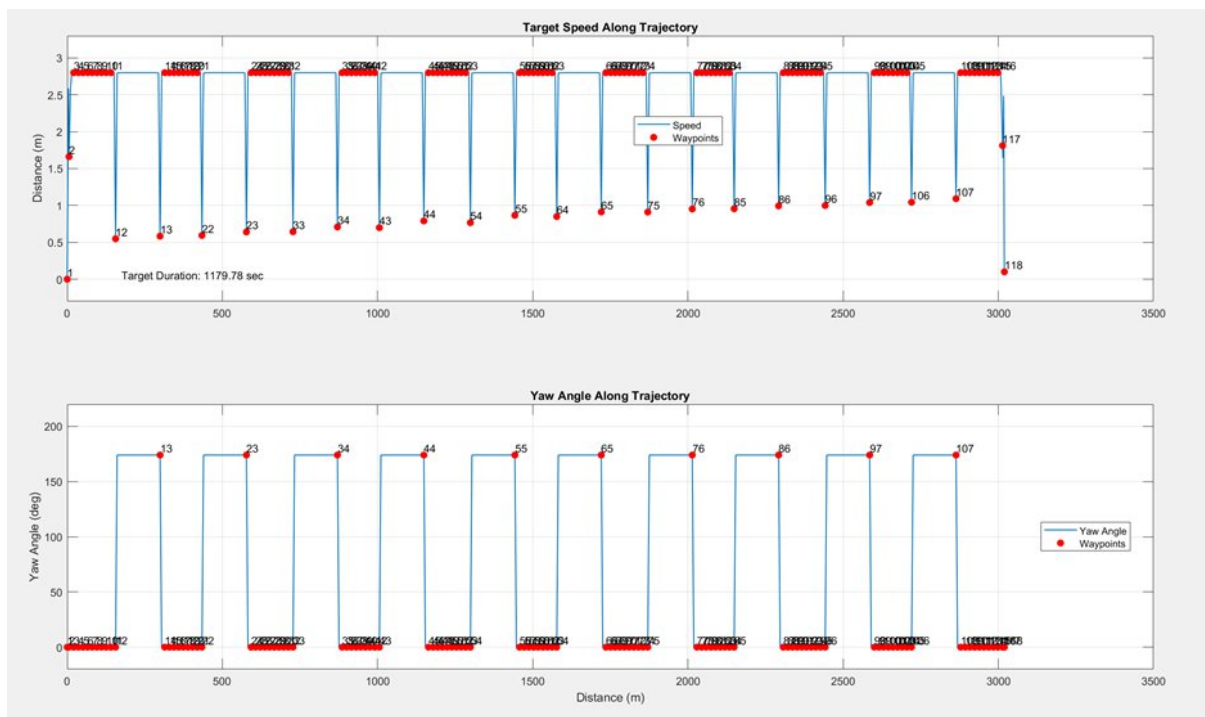


Figure 113: Trajectory Error Correction by Speed & Yaw Adjustment for Maximal Width Staggered Continuous Zigzag

Aerodynamic Profile Analysis:

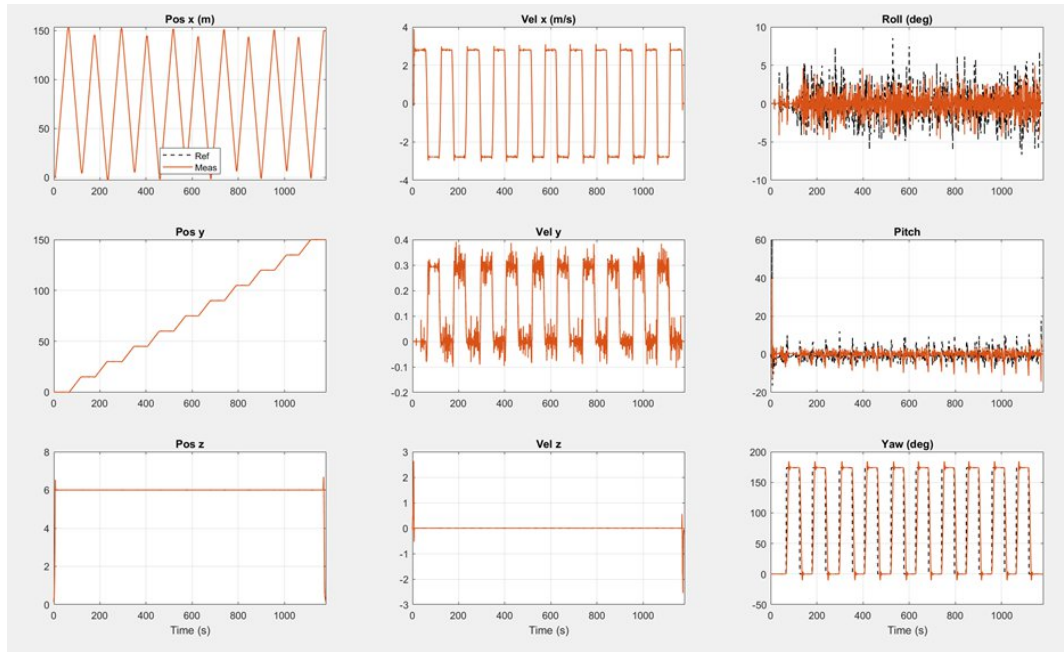


Figure 114: Aerodynamic Profile and Trajectory Tracking of the Drone for Maximal Width Staggered Continuous Zigzag

Battery Behavior Over Flight Duration:

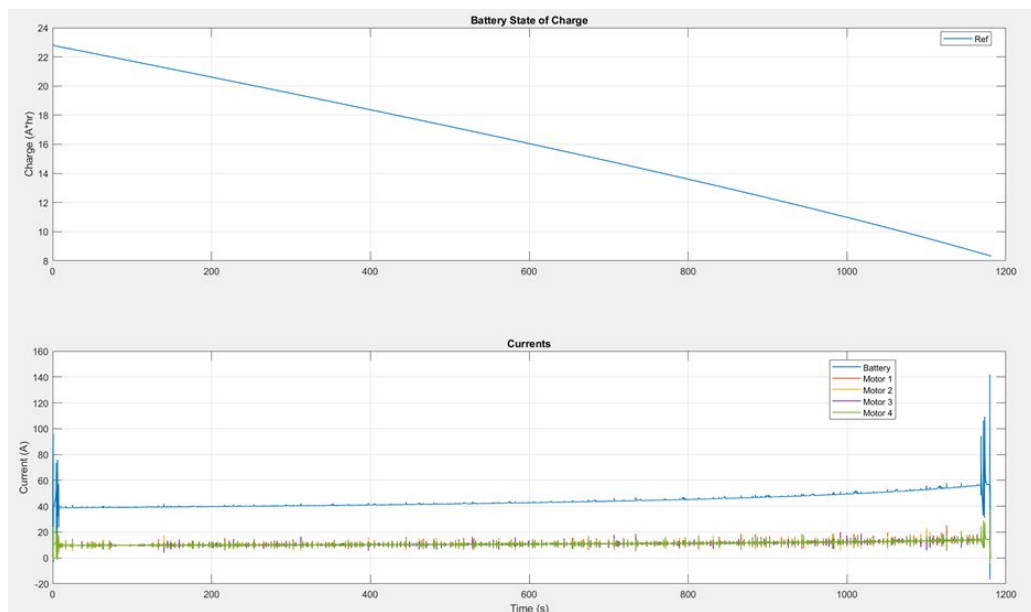


Figure 115: Battery Discharge and Motor Power Dynamics for Maximal Width Staggered Continuous Zigzag

4.2.7.5.6. Pseudo Penrose (Minimal Scattering)

Waypoint Map:

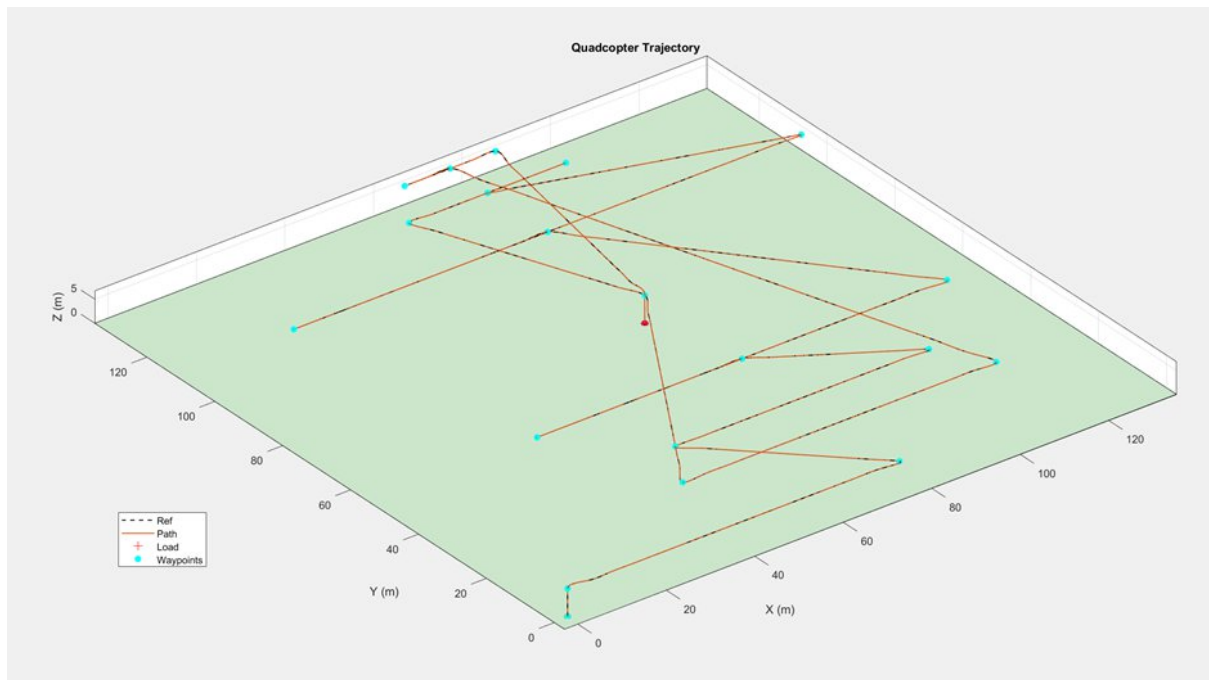


Figure 116: Quadcopter Trajectory for Pseudo Penrose Pattern with Minimal Scattering

Spatial Position Sampling:

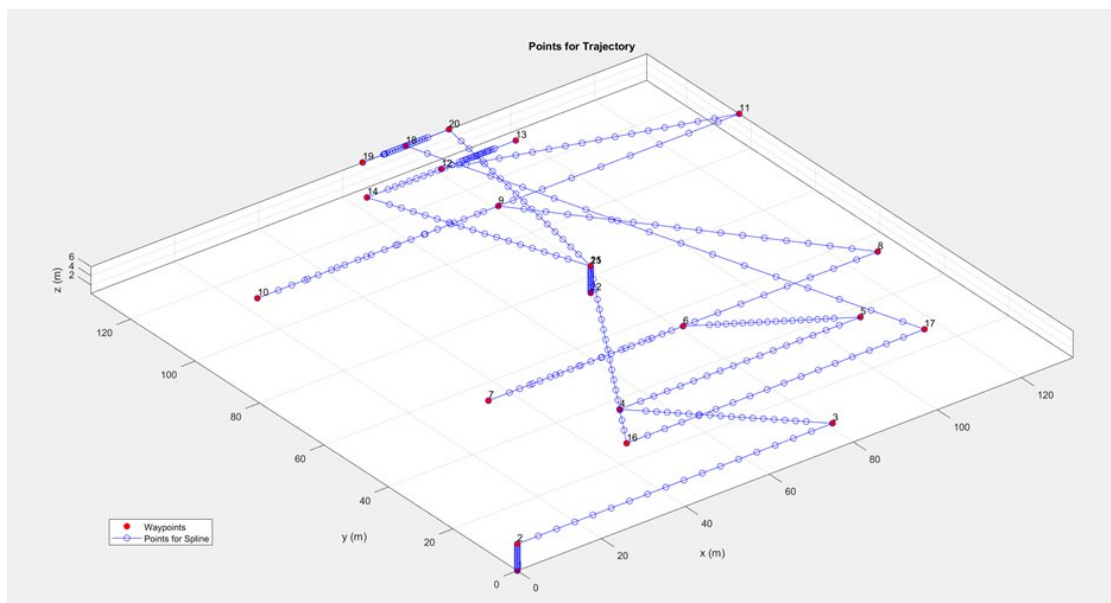


Figure 117: Trajectory Verification Points During Drone Flight for Pseudo Penrose Pattern with Minimal Scattering

Spatial Error Correction:

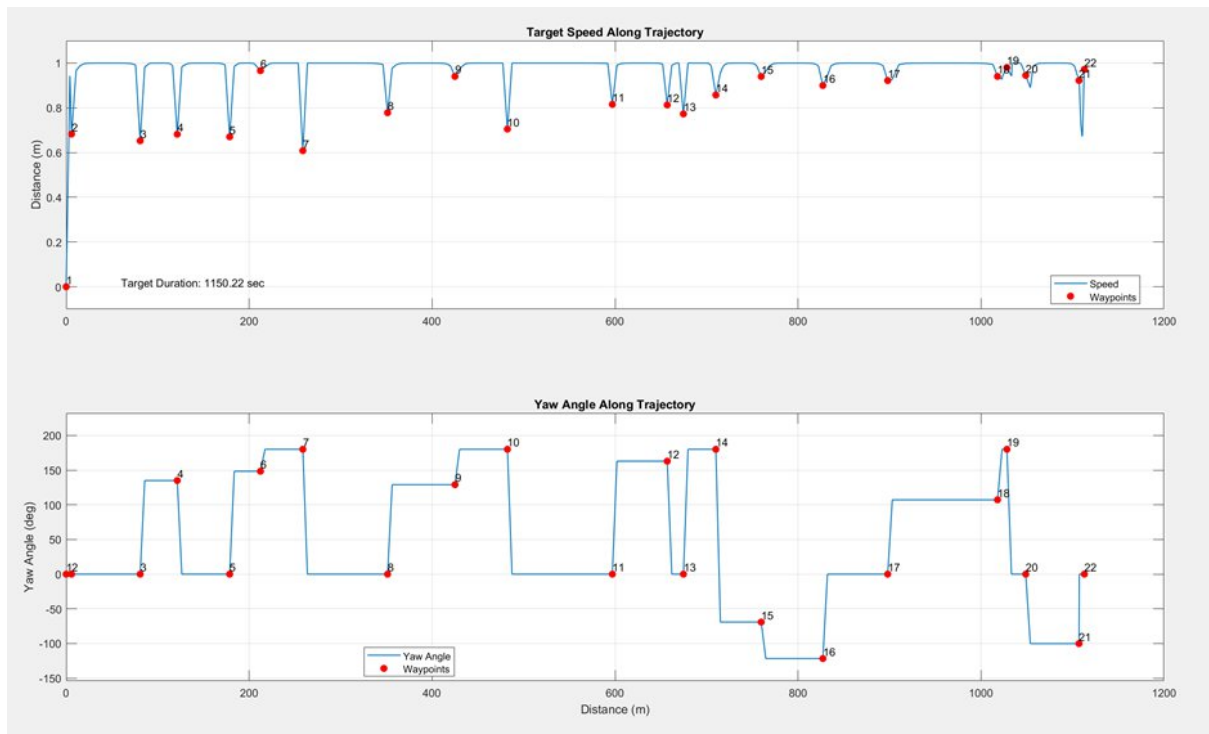


Figure 118: Trajectory Error Correction by Speed & Yaw Adjustment for Pseudo Penrose Pattern with Minimal Scattering

Aerodynamic Profile Analysis:

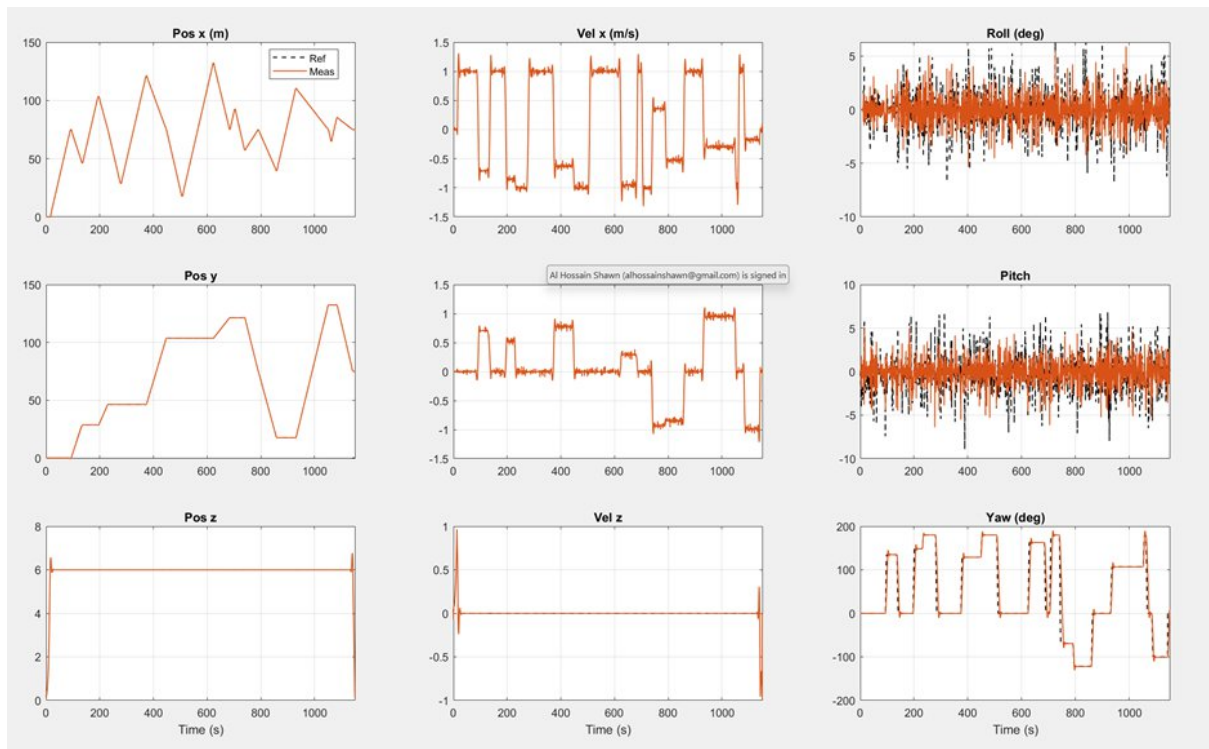


Figure 119: Aerodynamic Profile and Trajectory Tracking of the Drone for Pseudo Penrose Pattern with Minimal Scattering

Battery Behavior Over Flight Duration:

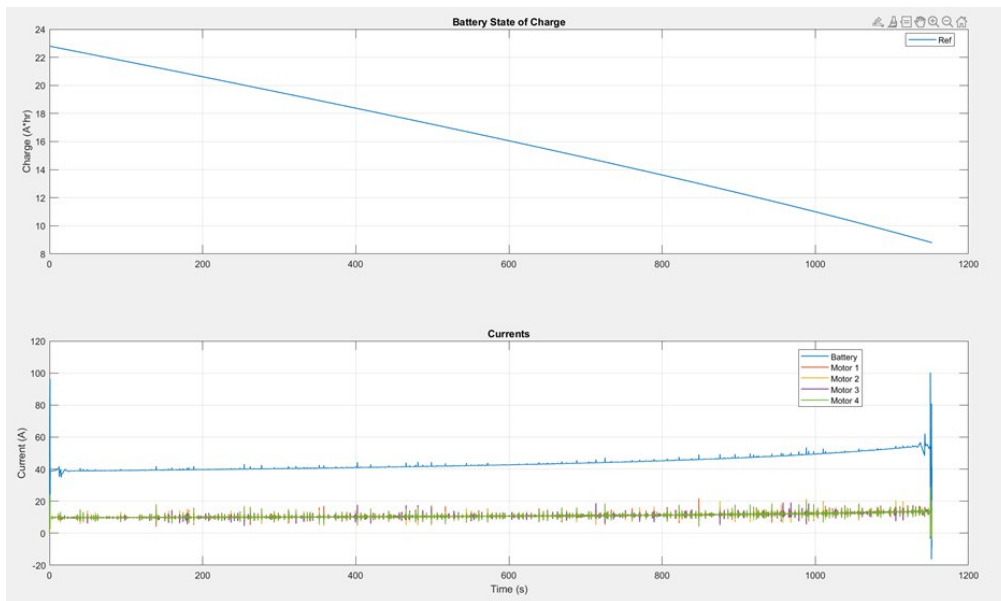


Figure 120: Battery Discharge and Motor Power Dynamics for Pseudo Penrose Pattern with Minimal Scattering

4.2.7.5.7. Pseudo Penrose (Moderate Scattering)

Waypoint Map:

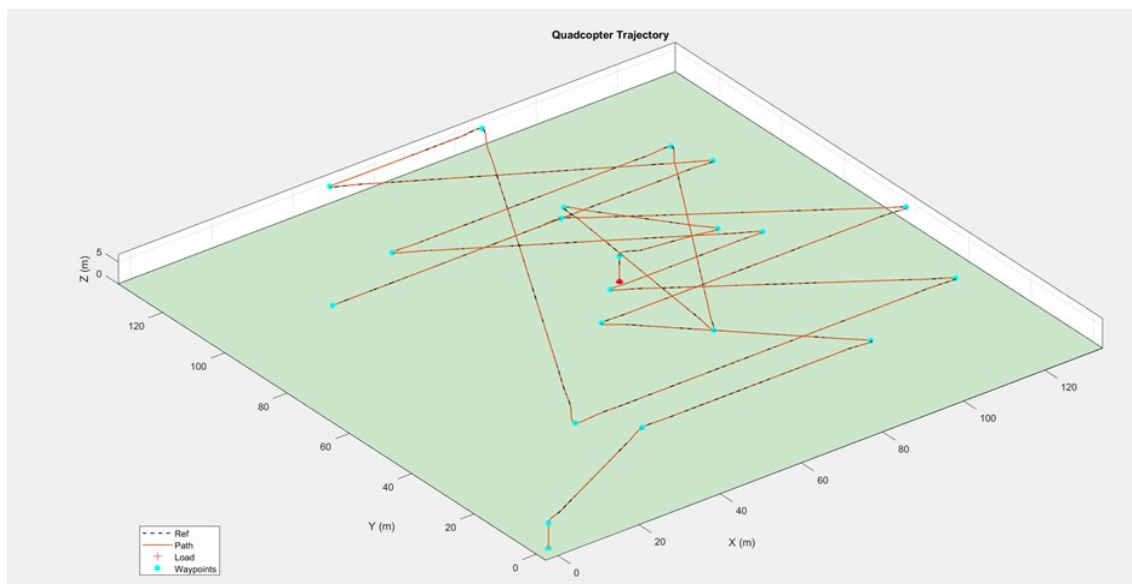


Figure 121: Quadcopter Trajectory for Pseudo Penrose Pattern with Moderate Scattering

Spatial Position Sampling:

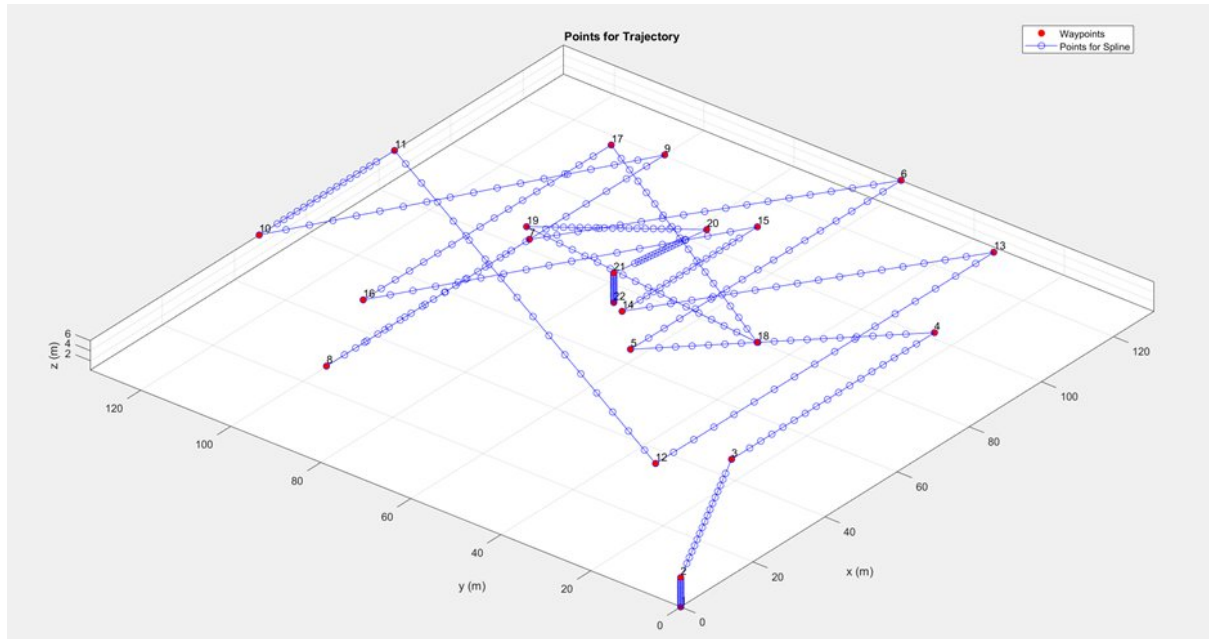


Figure 122: Trajectory Verification Points During Drone Flight for Pseudo Penrose Pattern with Moderate Scattering

Spatial Error Correction:

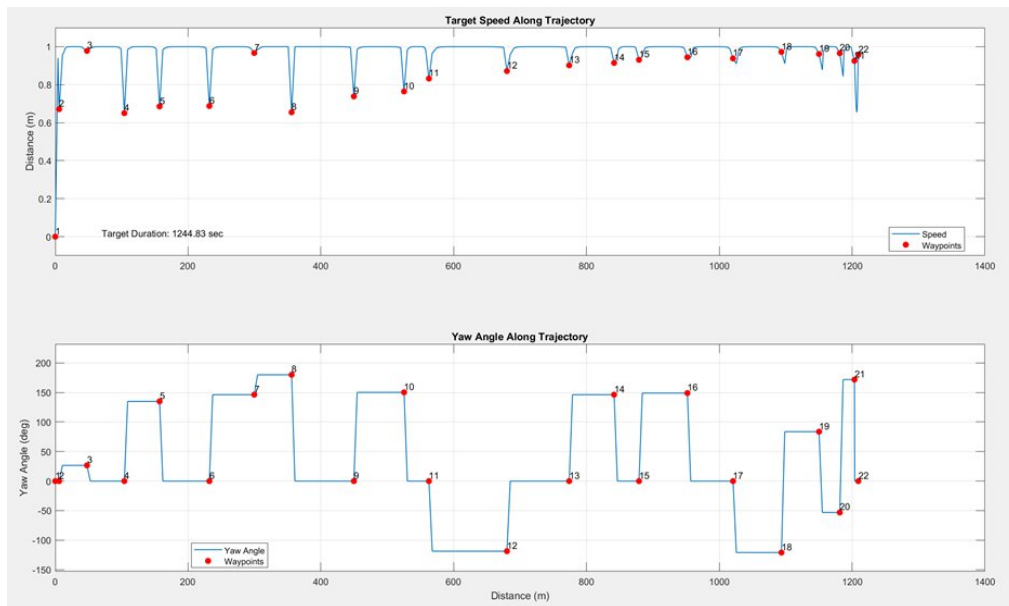


Figure 123: Trajectory Error Correction by Speed & Yaw Adjustment for Pseudo Penrose Pattern with Moderate Scattering

Aerodynamic Profile Analysis:

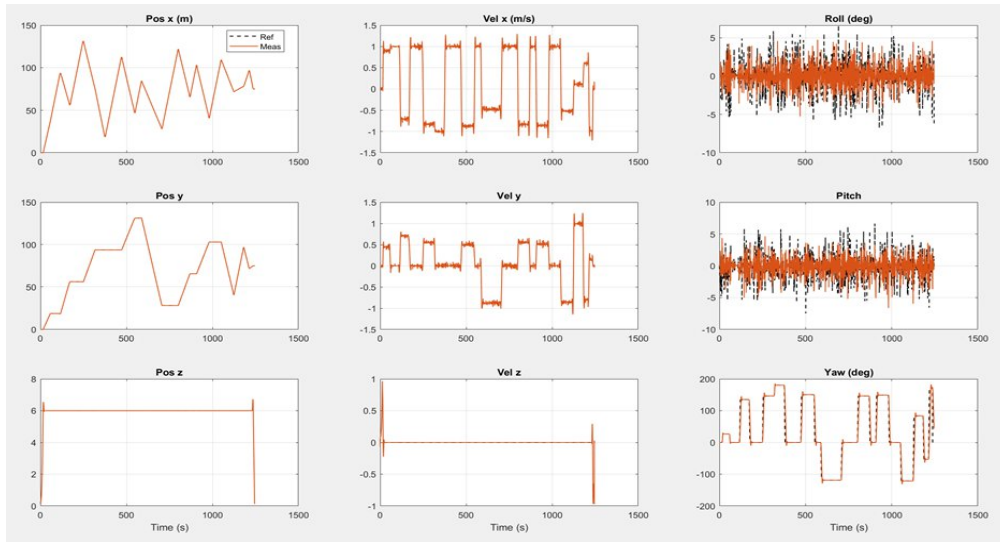


Figure 124: Aerodynamic Profile and Trajectory Tracking of the Drone for Pseudo Penrose Pattern with Moderate Scattering

Battery Behavior Over Flight Duration:

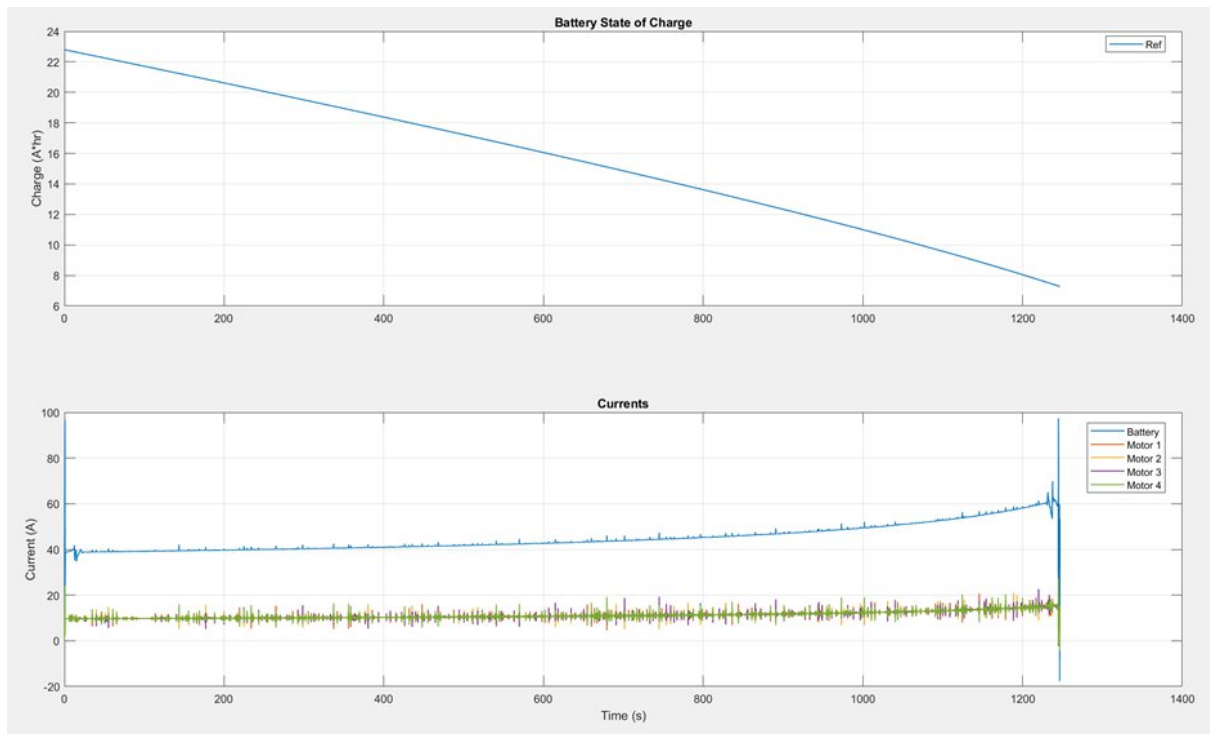


Figure 125: Battery Discharge and Motor Power Dynamics for Pseudo Penrose Pattern with Moderate Scattering

4.2.7.5.8. Weighted Area Pan

Waypoint Map:

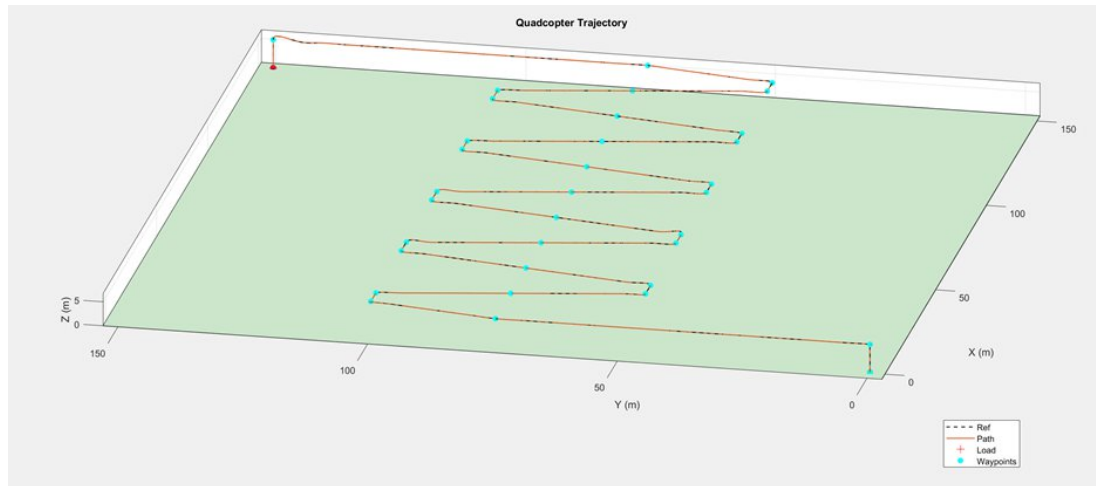


Figure 126: Quadcopter Trajectory for Weighted Area Pan

Spatial Position Sampling:

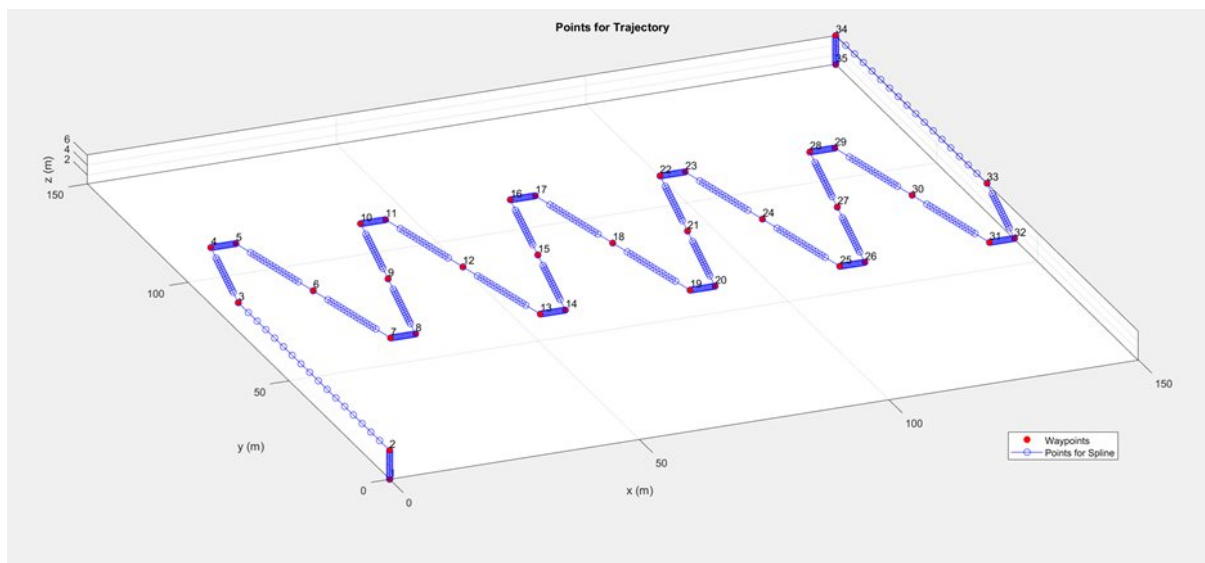


Figure 127: Trajectory Verification Points During Drone Flight for Weighted Area Pan

Spatial Error Correction:

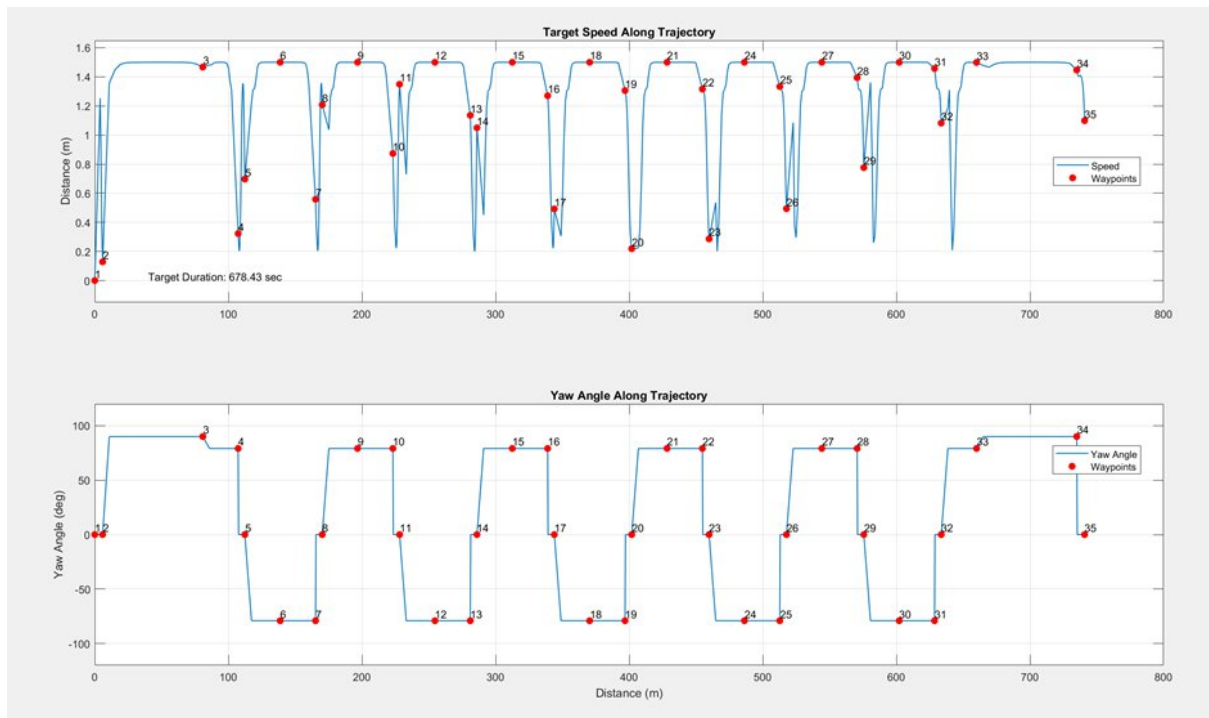


Figure 128: Trajectory Error Correction by Speed & Yaw Adjustment for Weighted Area Pan

Aerodynamic Profile Analysis:

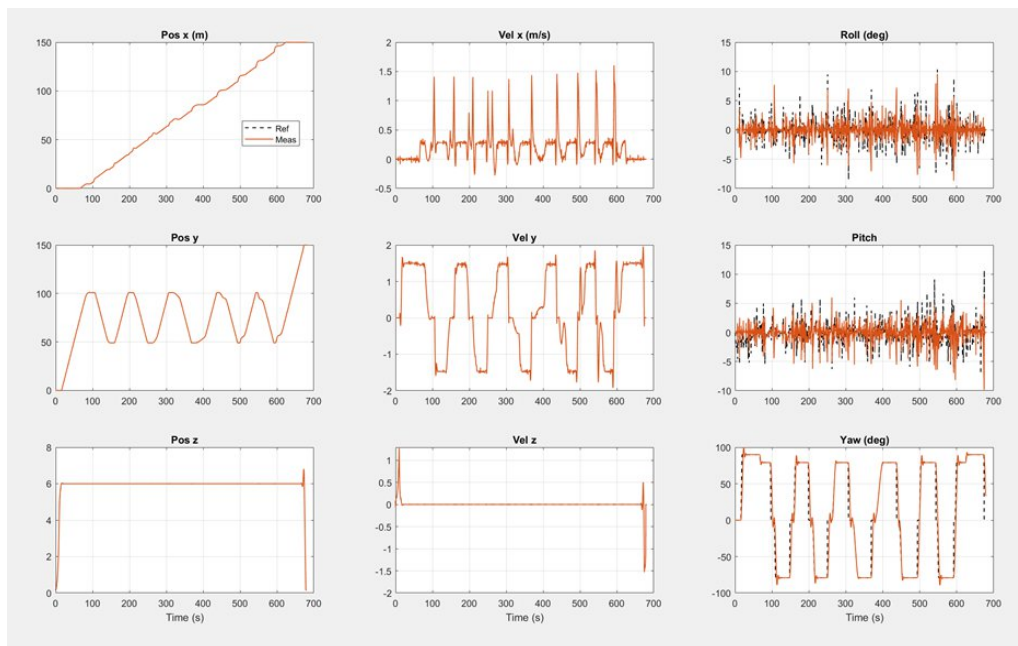


Figure 129: Aerodynamic Profile and Trajectory Tracking of the Drone for Weighted Area Pan

Battery Behavior Over Flight Duration:

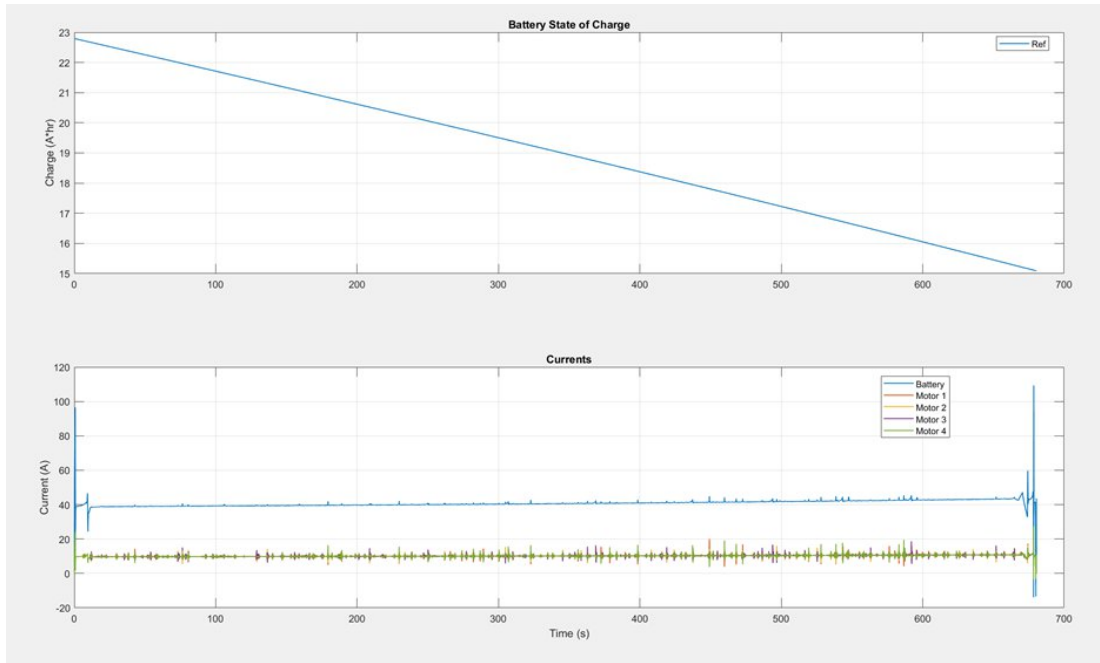


Figure 130: Battery Discharge and Motor Power Dynamics for Weighted Area Pan

4.2.7.5.9. Adaptive Weighted Area Pan

Waypoint Map:

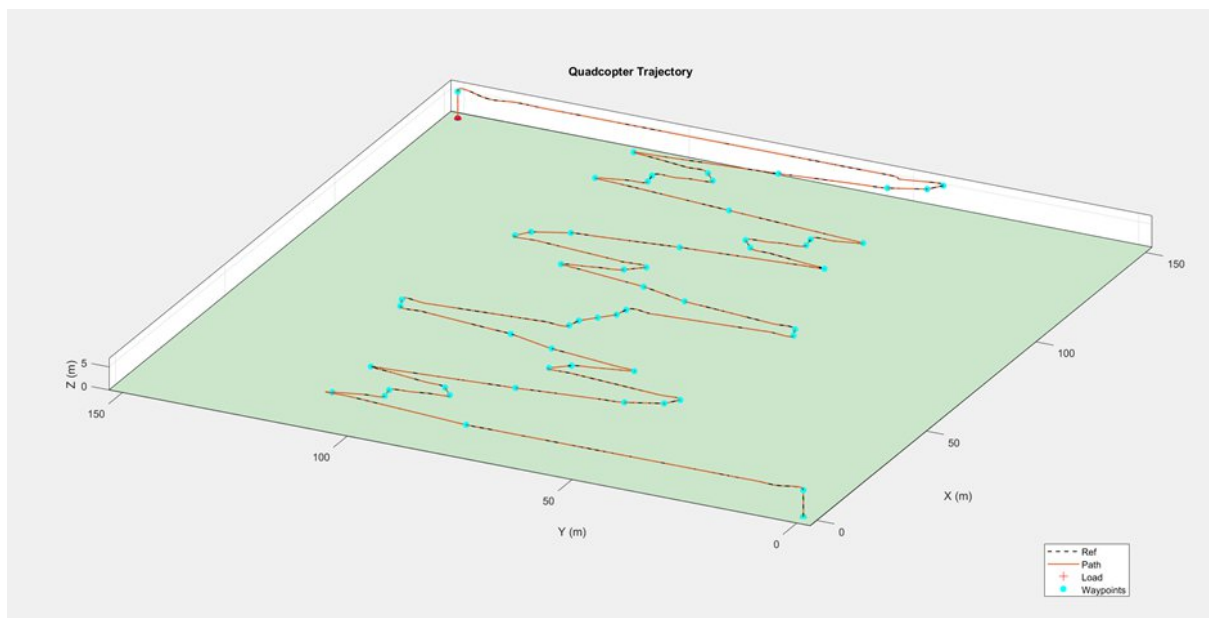


Figure 131: Quadcopter Trajectory for Adaptive Weighted Area Pan

Spatial Position Sampling:

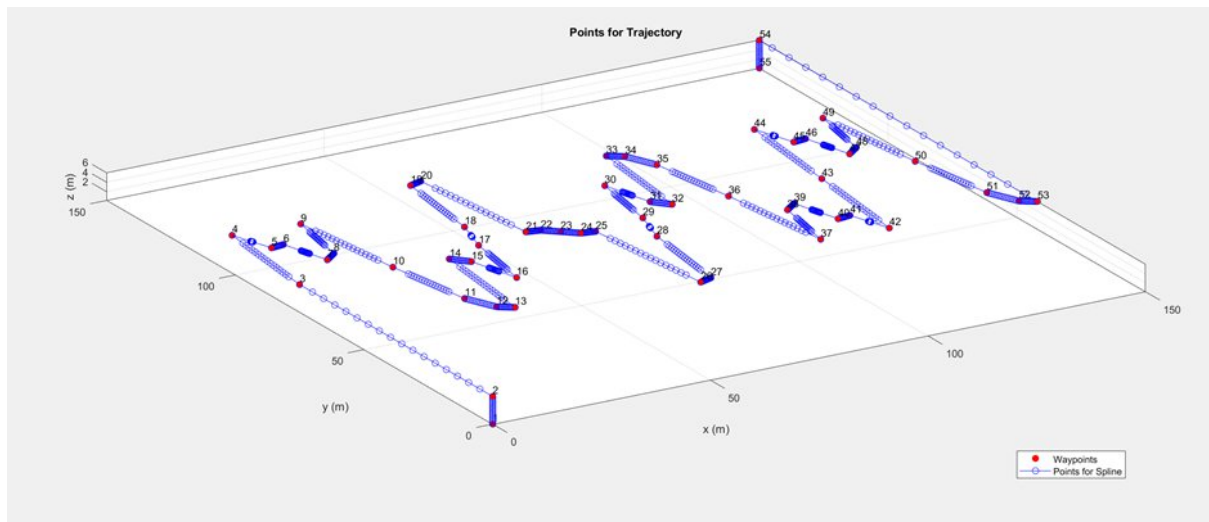


Figure 132: Trajectory Verification Points During Drone Flight for Adaptive Weighted Area Pan

Spatial Error Correction:

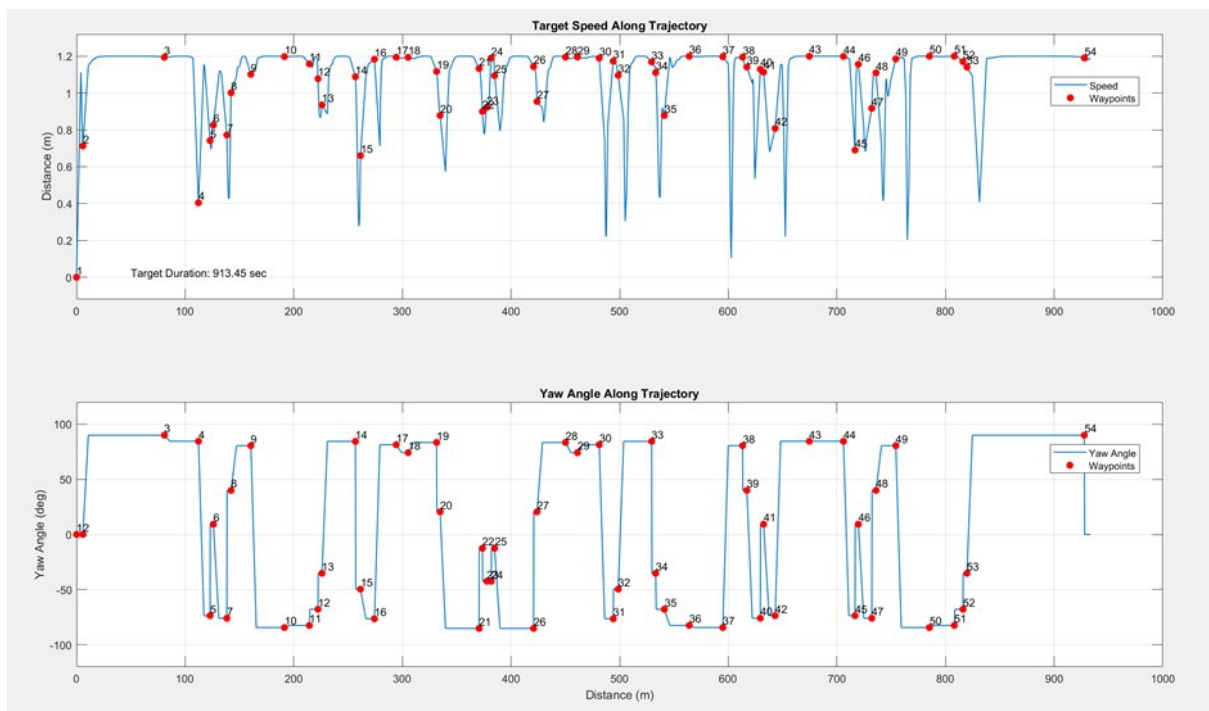


Figure 133: Trajectory Error Correction by Speed & Yaw Adjustment for Adaptive Weighted Area Pan

Aerodynamic Profile Analysis:

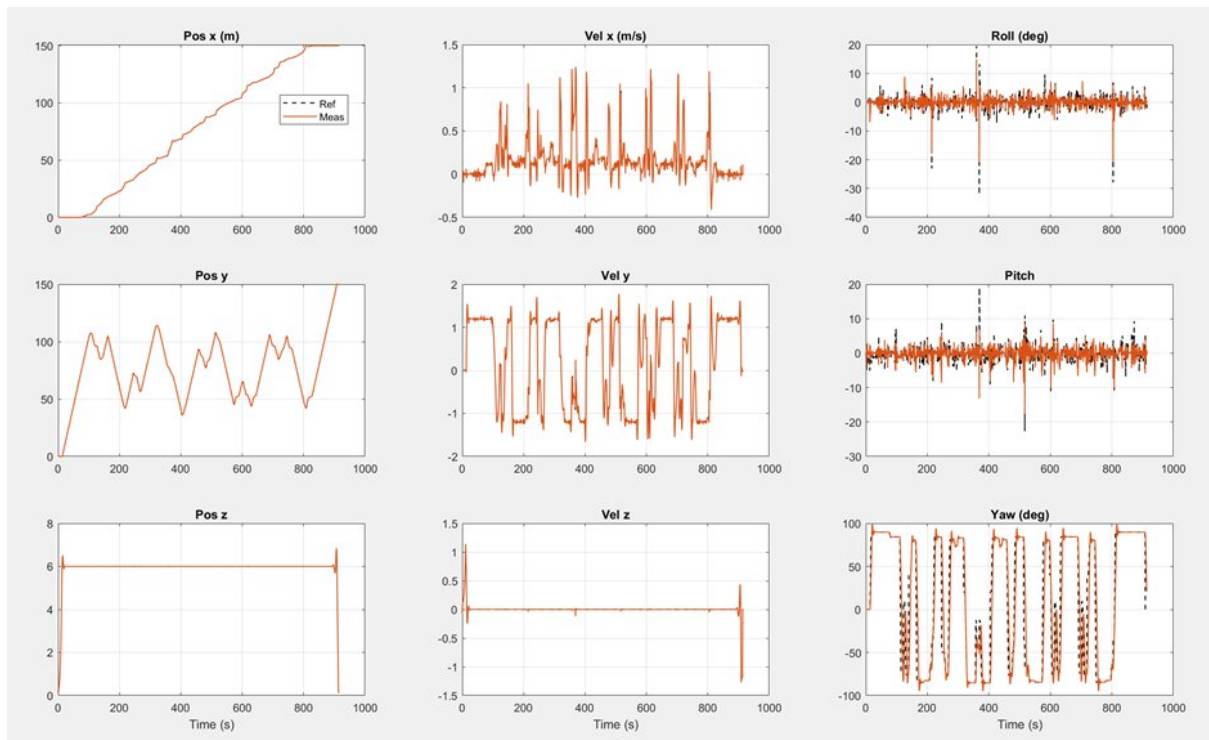


Figure 134: Aerodynamic Profile and Trajectory Tracking of the Drone for Adaptive Weighted Area Pan

Battery Behavior Over Flight Duration:

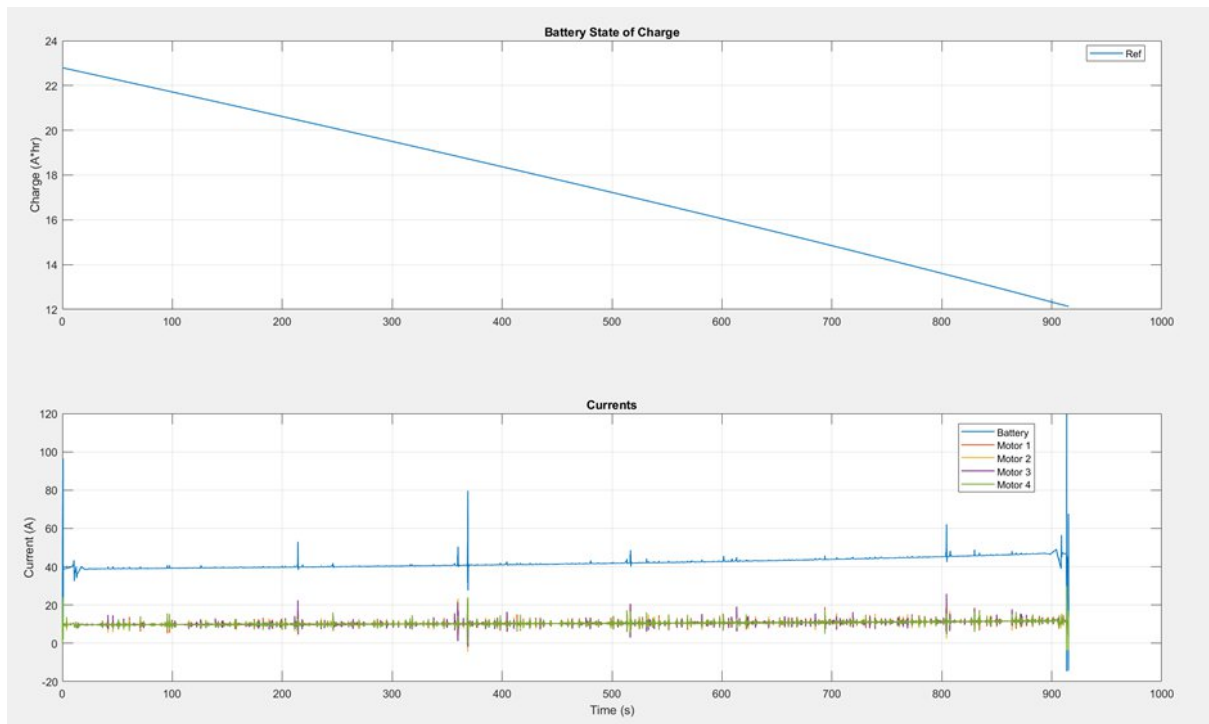


Figure 135: Battery Discharge and Motor Power Dynamics for Adaptive Adaptive Weighted Area Pan

4.2.7.5.10. Simple Topographic Path

Waypoint Map:

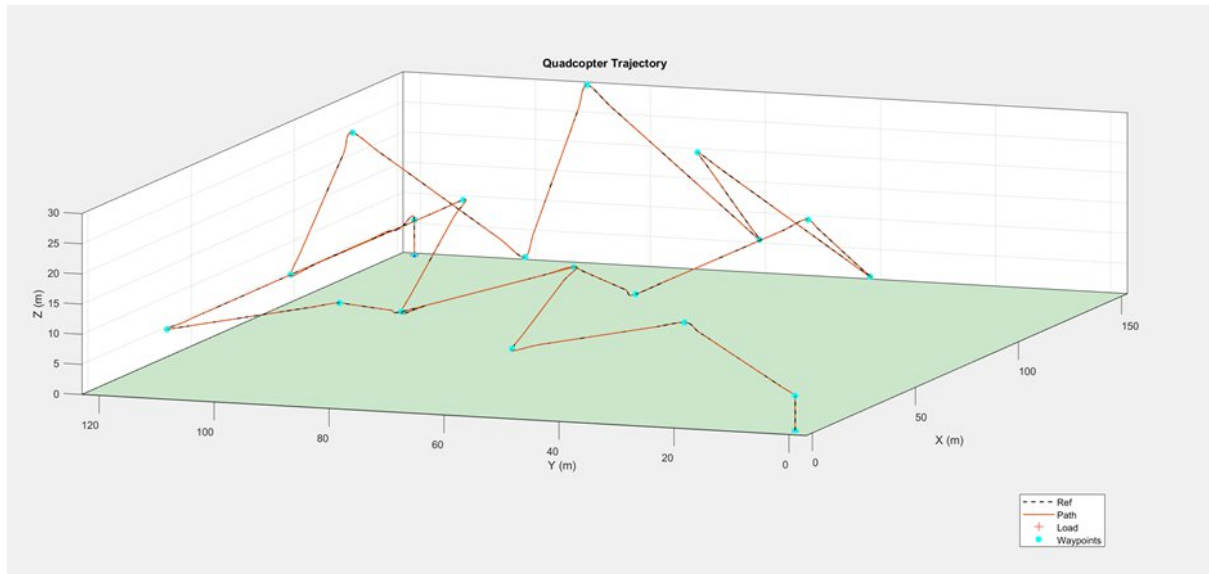


Figure 136: Quadcopter Trajectory for Simple Topographic Path

Spatial Position Sampling:

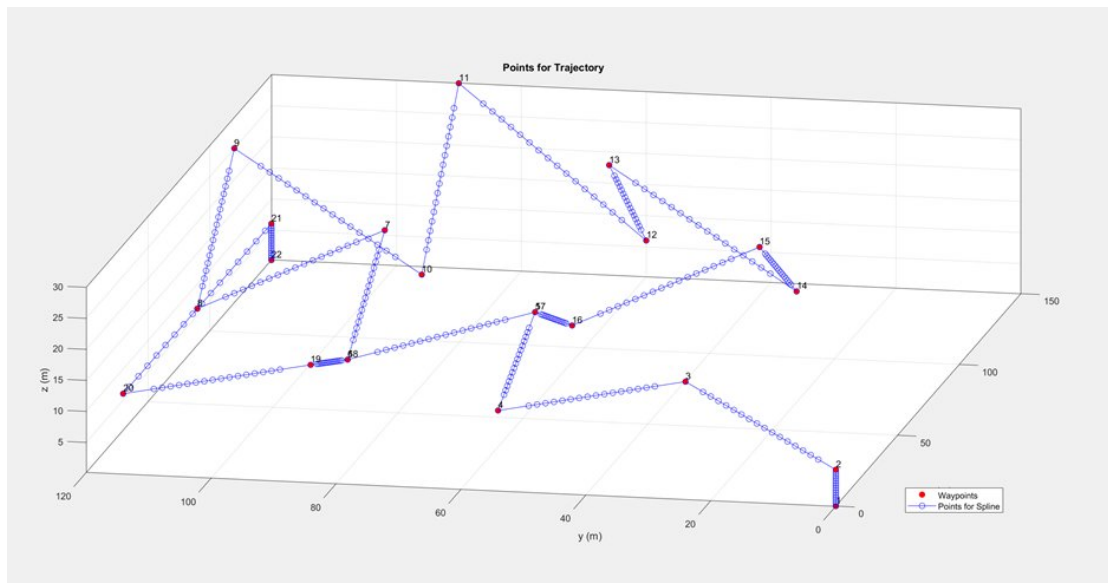


Figure 137: Trajectory Verification Points During Drone Flight for Simple Topographic Path

Spatial Error Correction:

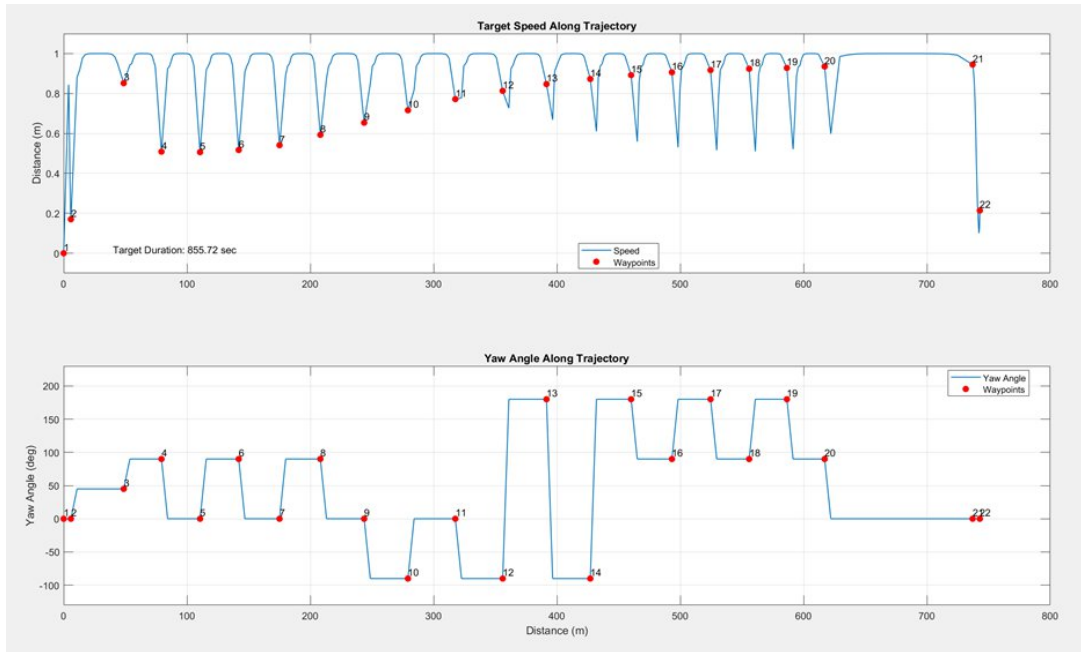


Figure 138: Trajectory Error Correction by Speed & Yaw Adjustment for Simple Topographic Path

Aerodynamic Profile Analysis:

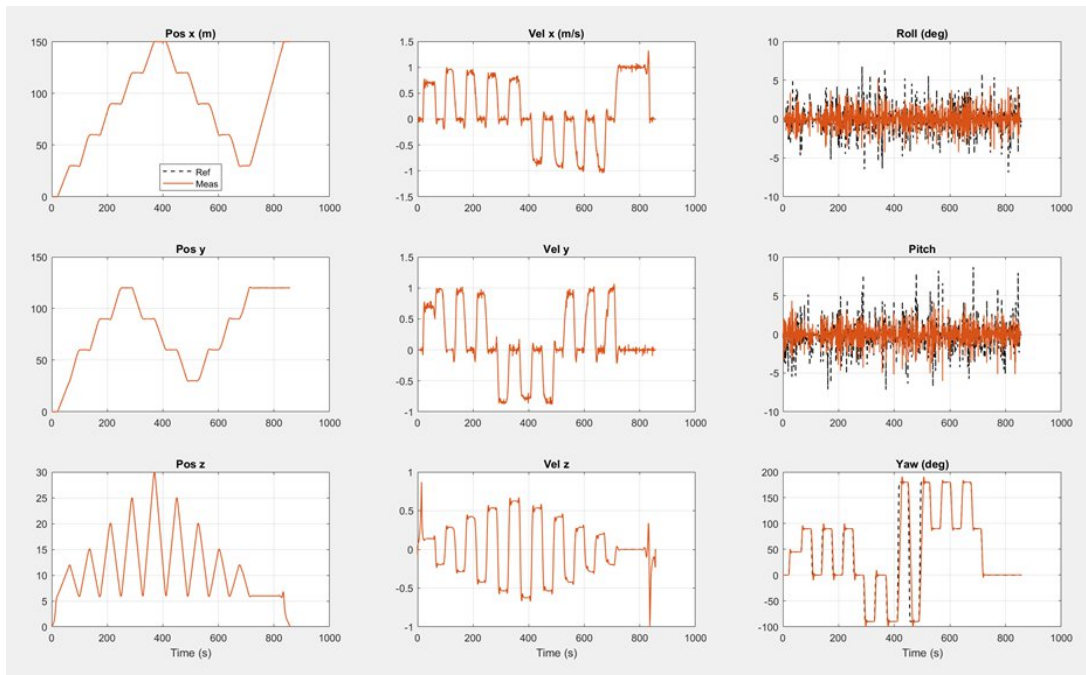


Figure 139: Aerodynamic Profile and Trajectory Tracking of the Drone for Simple Topographic Path

Battery Behavior Over Flight Duration:

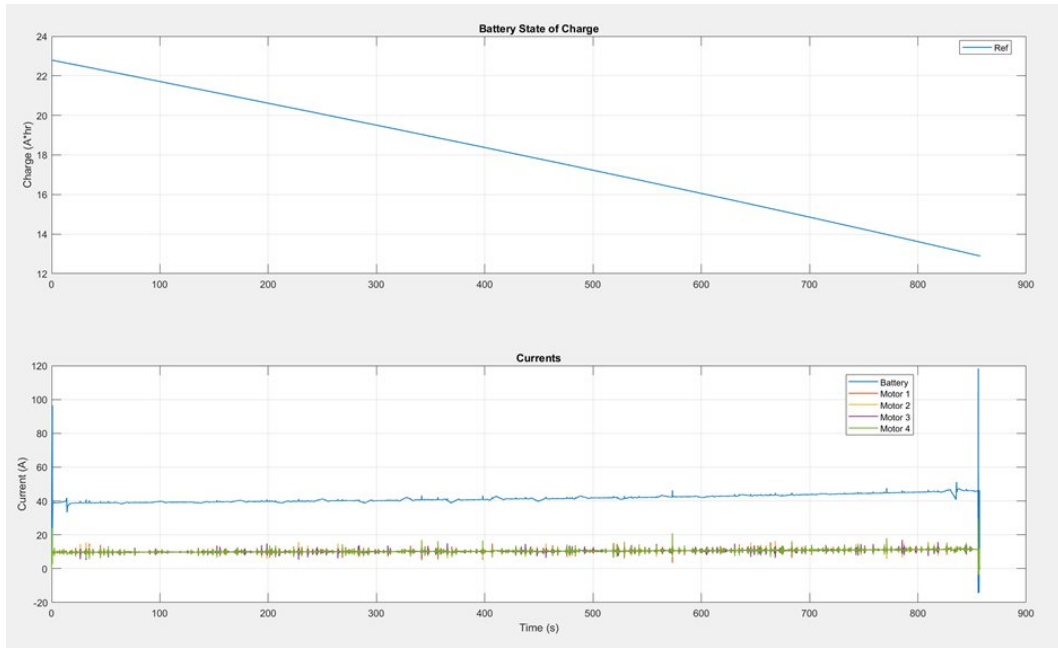


Figure 140: Battery Discharge and Motor Power Dynamics for Simple Topographic Path

4.2.7.5.11. Dynamic Topographic Path

Waypoint Map:

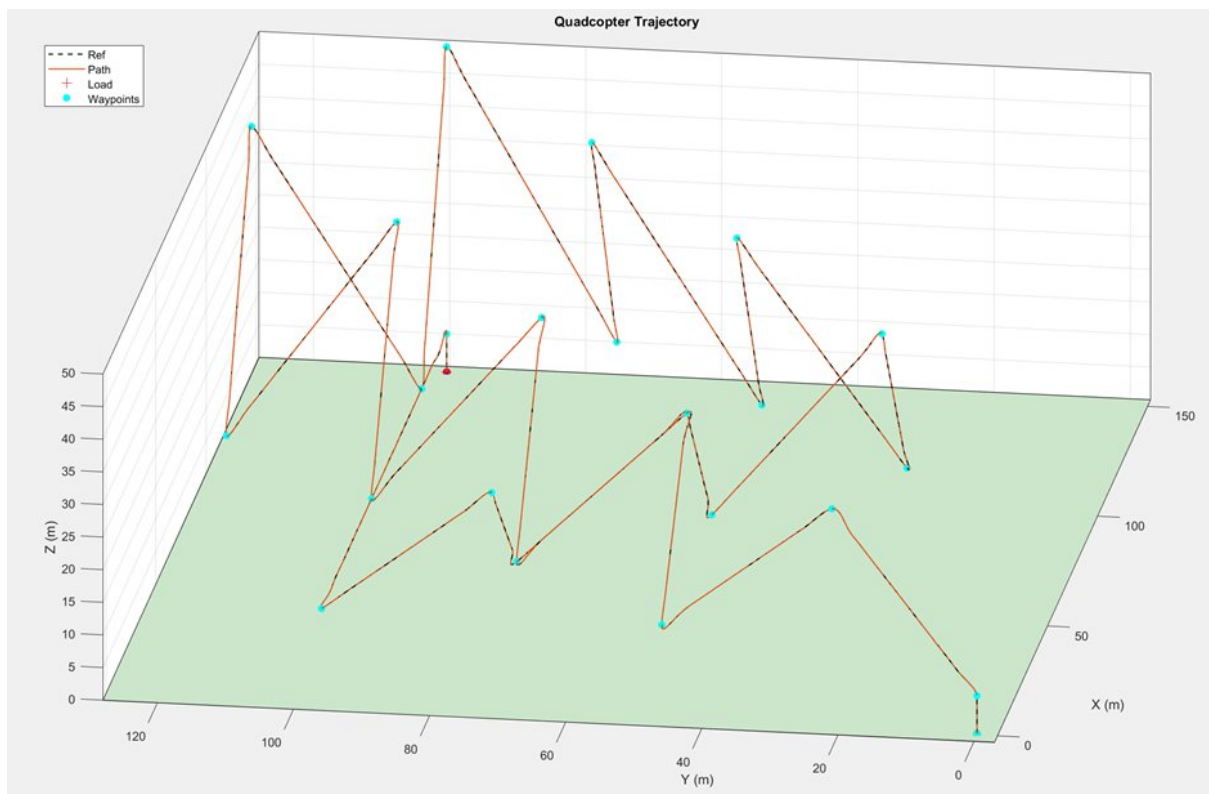


Figure 141: Quadcopter Trajectory for Dynamic Topographic Path

Spatial Position Sampling:

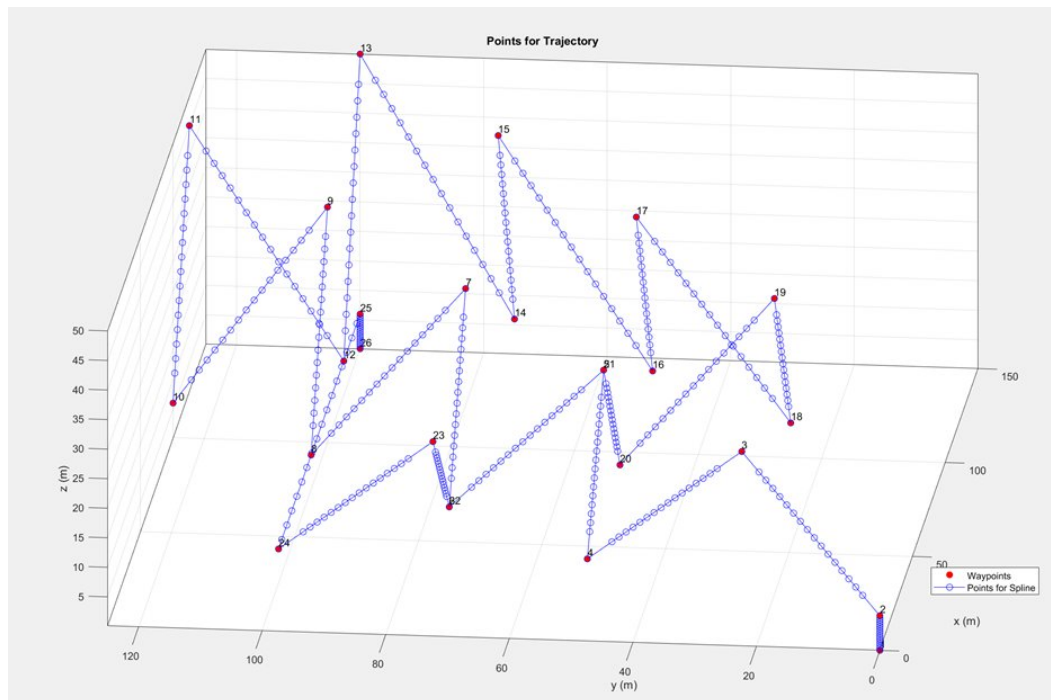


Figure 142: Trajectory Verification Points During Drone Flight for Dynamic Topographic Path

Spatial Error Correction:

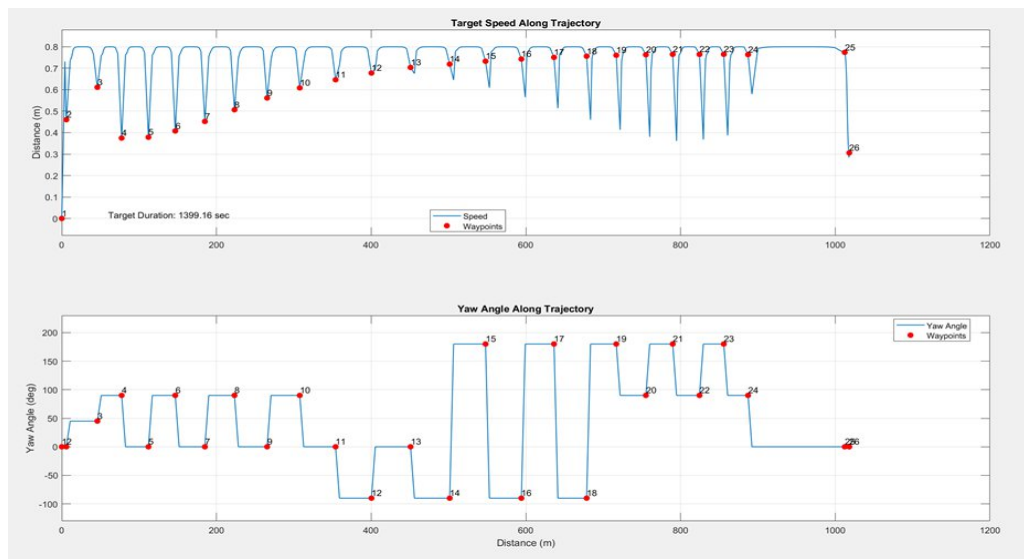


Figure 143: Trajectory Error Correction by Speed & Yaw Adjustment for Dynamic Topographic Path

Aerodynamic Profile Analysis:

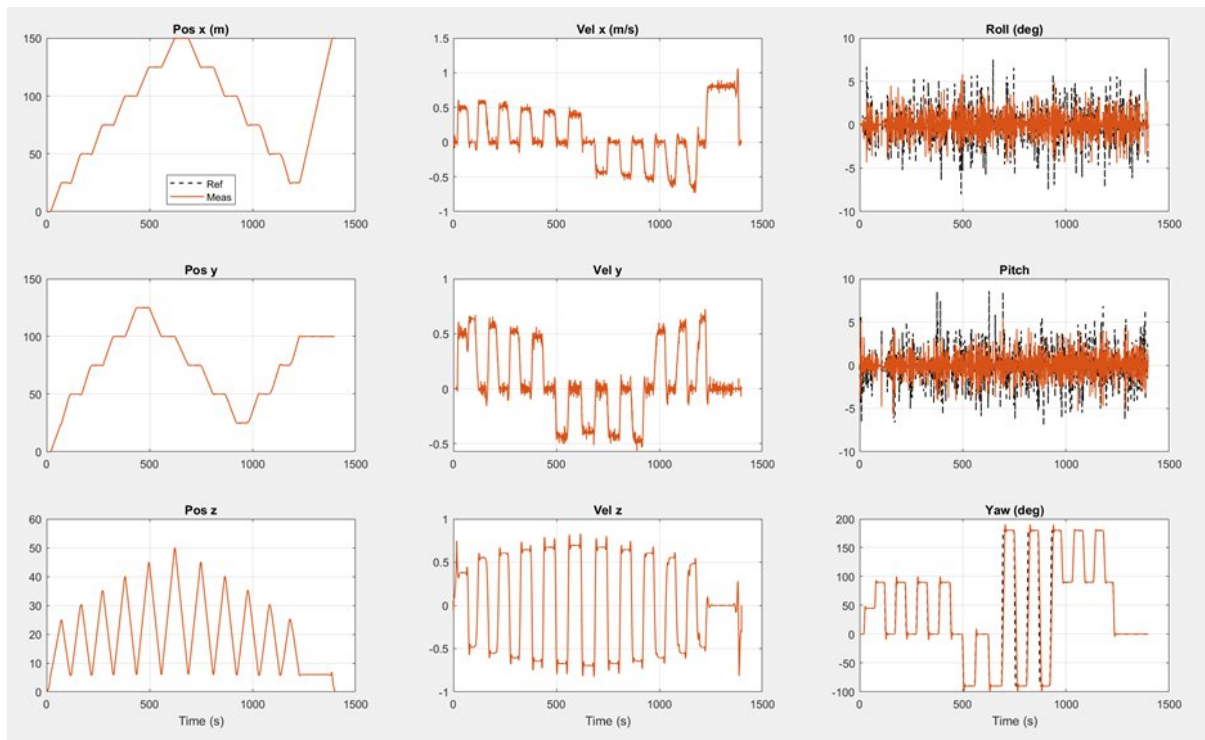


Figure 144: Aerodynamic Profile and Trajectory Tracking of the Drone for Dynamic Topographic Path

Battery Behavior Over Flight Duration:

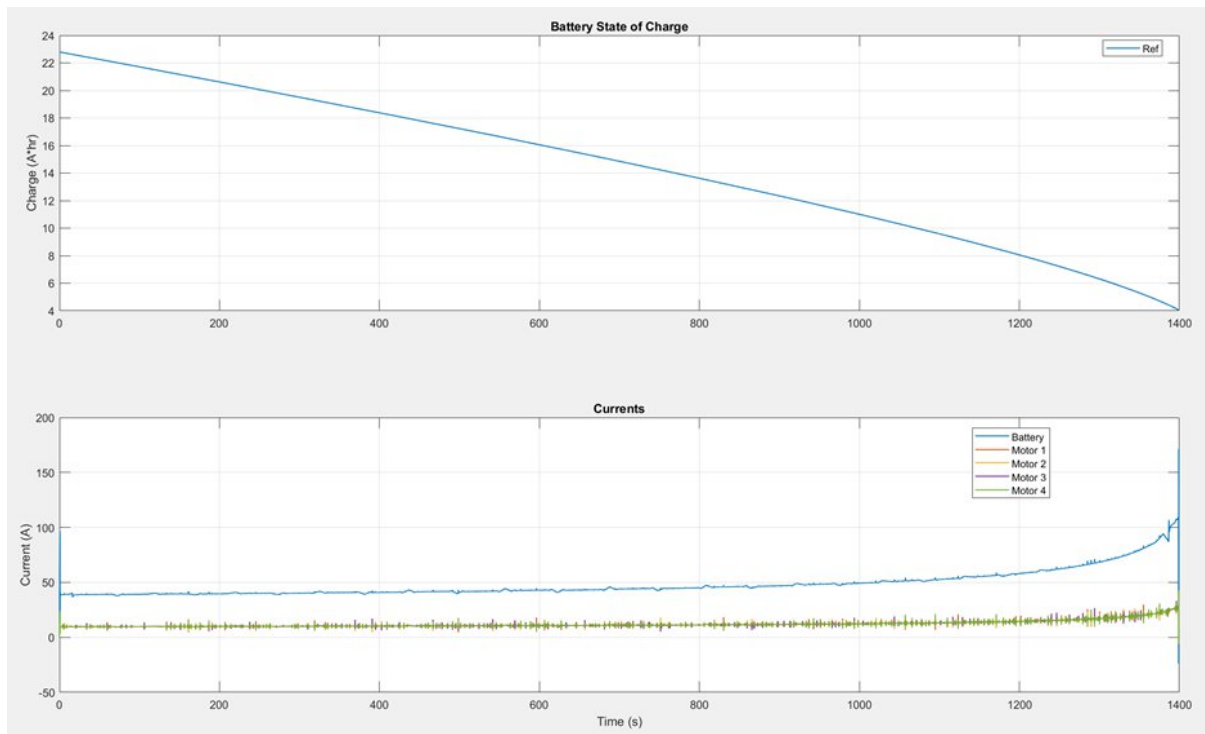


Figure 145: Battery Discharge and Motor Power Dynamics for Dynamic Topographic Path

4.3. Identify Optimal Design Approach

4.3.1. Computational Performance

Accuracy Curves of All Classification Models:

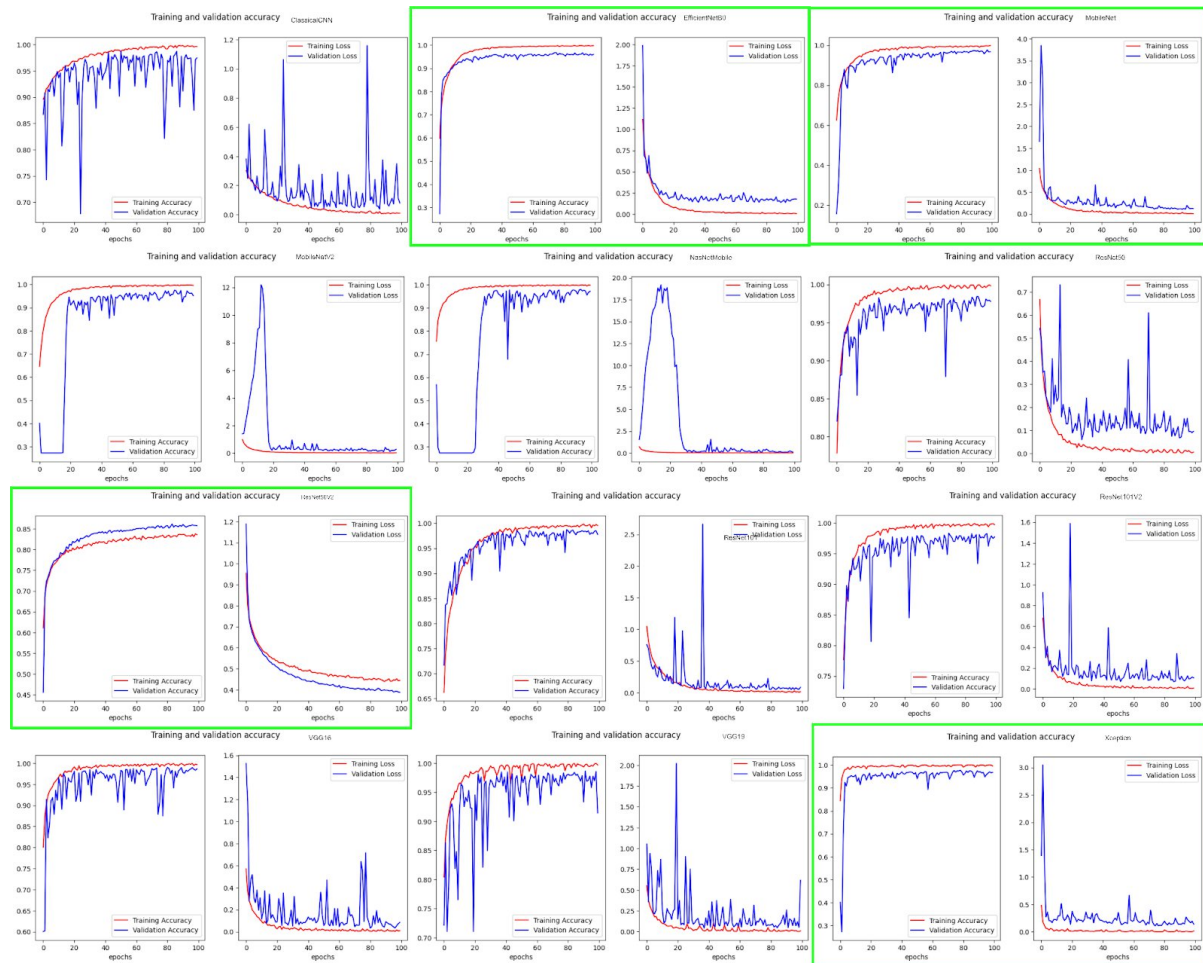


Figure 146: Training vs Validation Accuracy Curves for Top 4 Performing Models

From this model we have selected the best models, where we have considered which models are performing based on training and validation models. We selected four of the best models. Which are EfficientNetB0, MobileNet, ResNet50V2 and Xception.

Power Consumption Curves of All Models (x86_64):

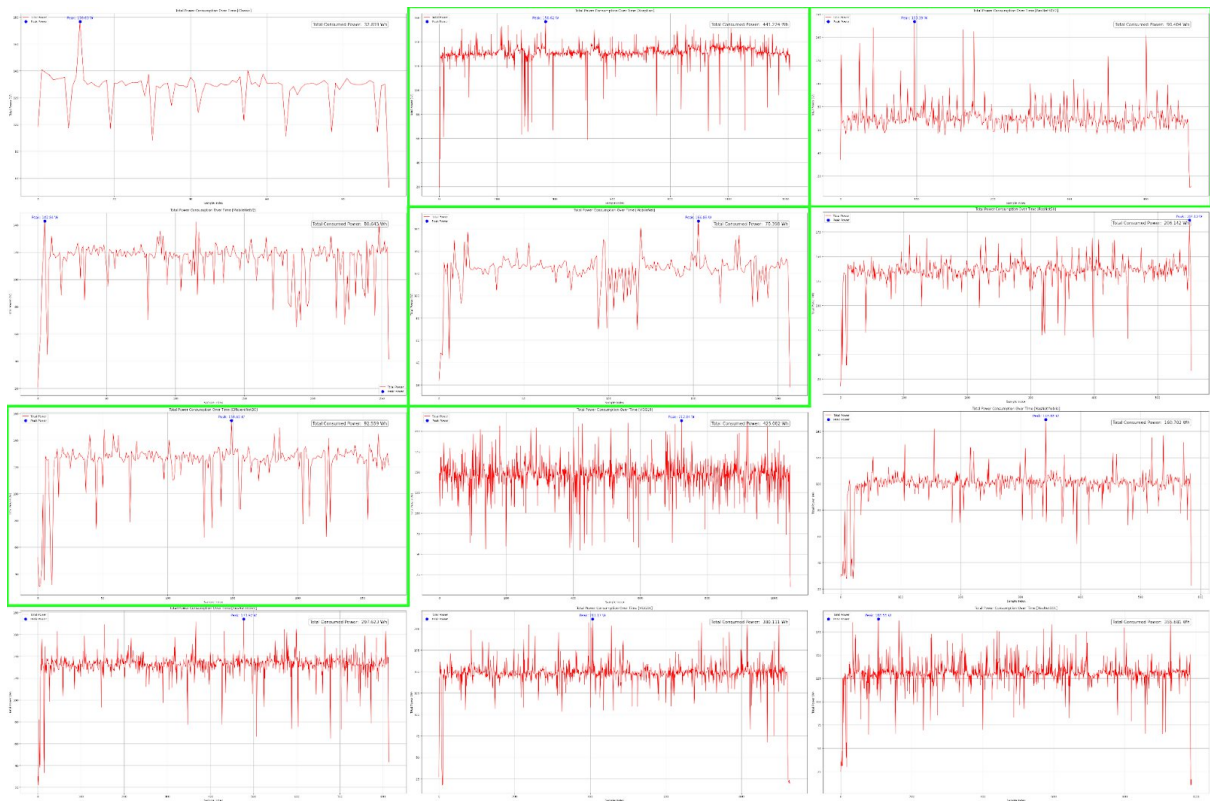


Figure 147: Power Draw Profiles for Top 4 Performing Models

From this model we have selected the best models, where we have considered which models are performing based on power consumption curves. We selected four of the best models. Which are EfficientNetB0, MobileNet, ResNet50V2, and Xception.

4.3.1.1. Time Required to Process 1k Frames

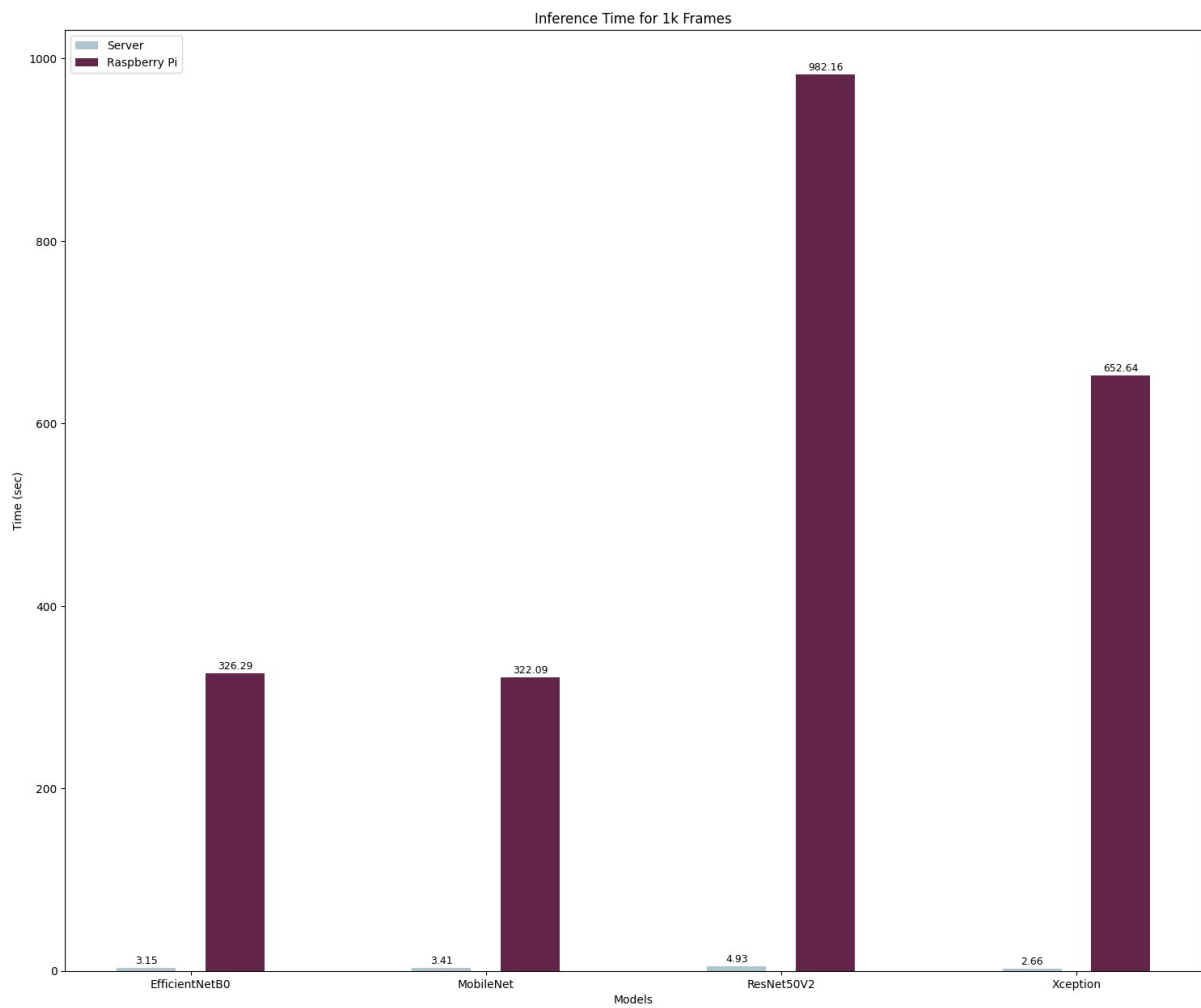


Figure 148: Time Deference to Process 1k Frames

The time required to process 1K frames of server and Raspberry Pi of the top four models is shown in this figure. EfficientNetB0 on the server took 3.15 seconds and raspberry pi showed 326.29 seconds. On the other hand, MobileNet on the server took 3.41 seconds and raspberry pi indicated 322.09 seconds. Moreover, ResNet50V2 on the server requires 4.93 seconds and raspberry pi shows 982.16 seconds. At last, Xception on the server required 2.66 seconds, again, raspberry pi showed 652.64 seconds. In this case MobileNet is doing better.

4.3.1.2. Time Required to Process One Frame

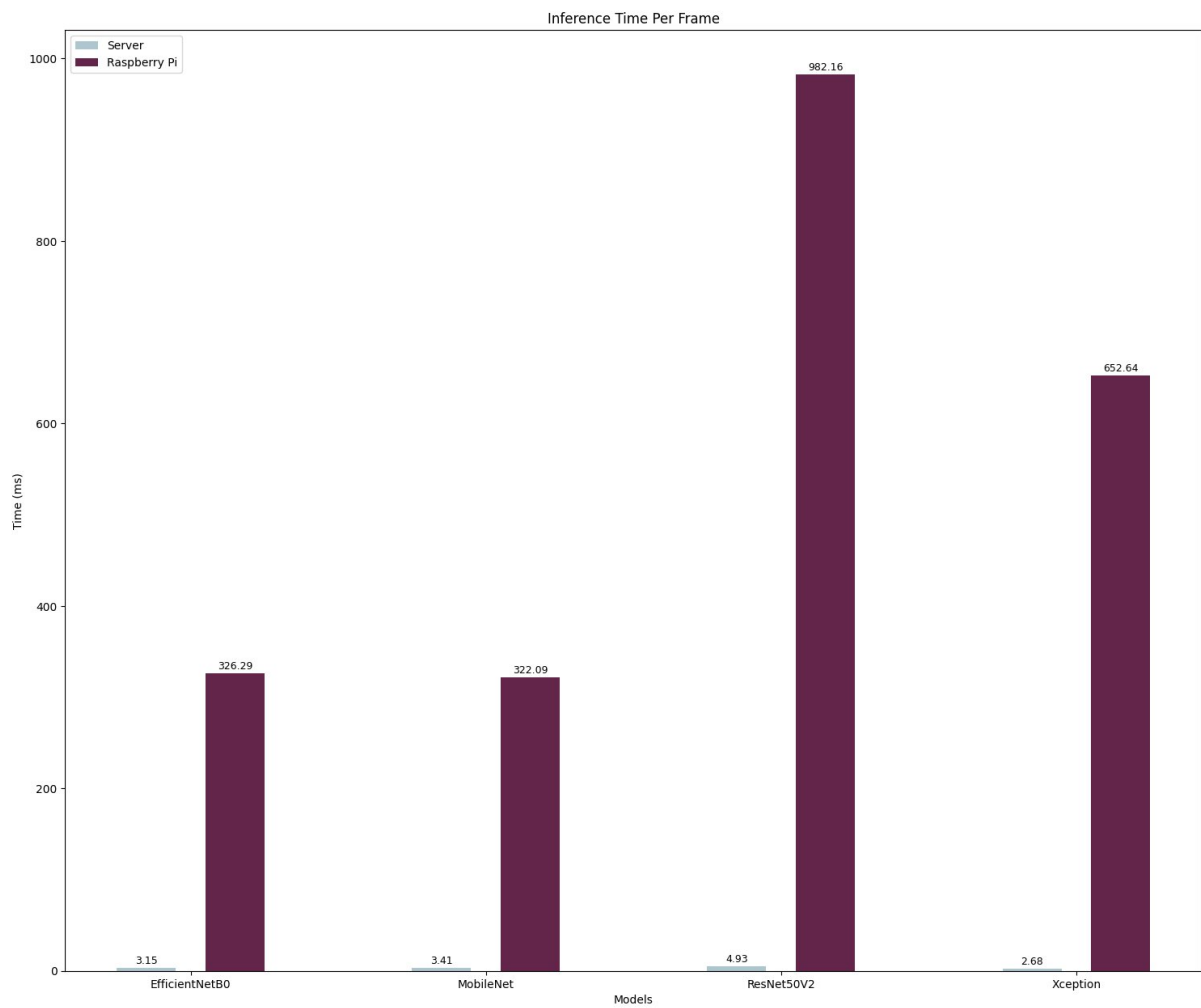


Figure 149: Time Deference to Process One Frame

The above bar chart shows the time taken for processing per frame. The values from this bar chart show that the server always takes less time than Raspberry Pi. The time required to process per frame for the top four models of both the server and Raspberry Pi are given in the figure. Inside the server EfficientNetB0 took 3.15 seconds, but our raspberry pi showed 326.29 seconds. On the other hand, on the server MobileNet took 3.41 seconds, while Raspberry Pi took 322.09 seconds. Moreover, ResNet50V2 on the server would have taken 4.93 seconds and the Raspberry Pi would have taken 982.16 seconds. At last, on the server Xception took 2.66 seconds, but the Raspberry Pi took 652.64 seconds. So in this case, MobileNet is doing better.

4.3.1.3. Inference Efficiency (Frames Per Watt)

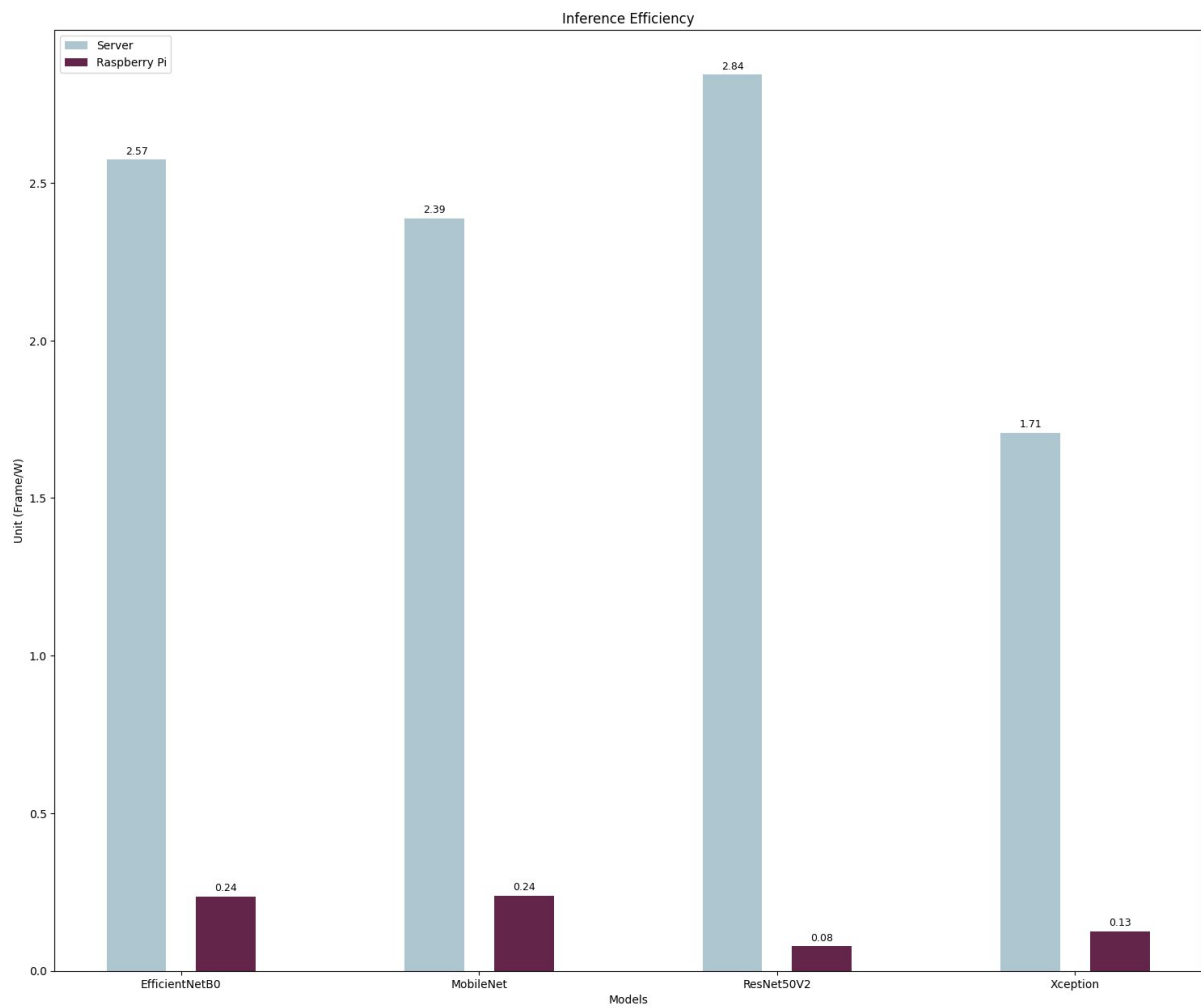


Figure 150: Inference Efficiency Comparison

Inference efficiency of the server and Raspberry Pi of the top four models are shown in this figure. EfficientNetB0 on the server took 2.57 frame/W and 0.24 frame/W is on the raspberry pi. On the other hand, MobileNet on the server was 2.39 and raspberry pi was 0.24 frames/W. Moreover, ResNet50V2 on the server used 2.84 frame/W comparing to the pi showing 0.08 frame/W. At last, Xception on the server is 1.71 frame/W, raspberry pi again displayed 0.13 frame/W. Here, we can say that MobileNet are doing better.

4.3.1.4. Inference Energy Consumption

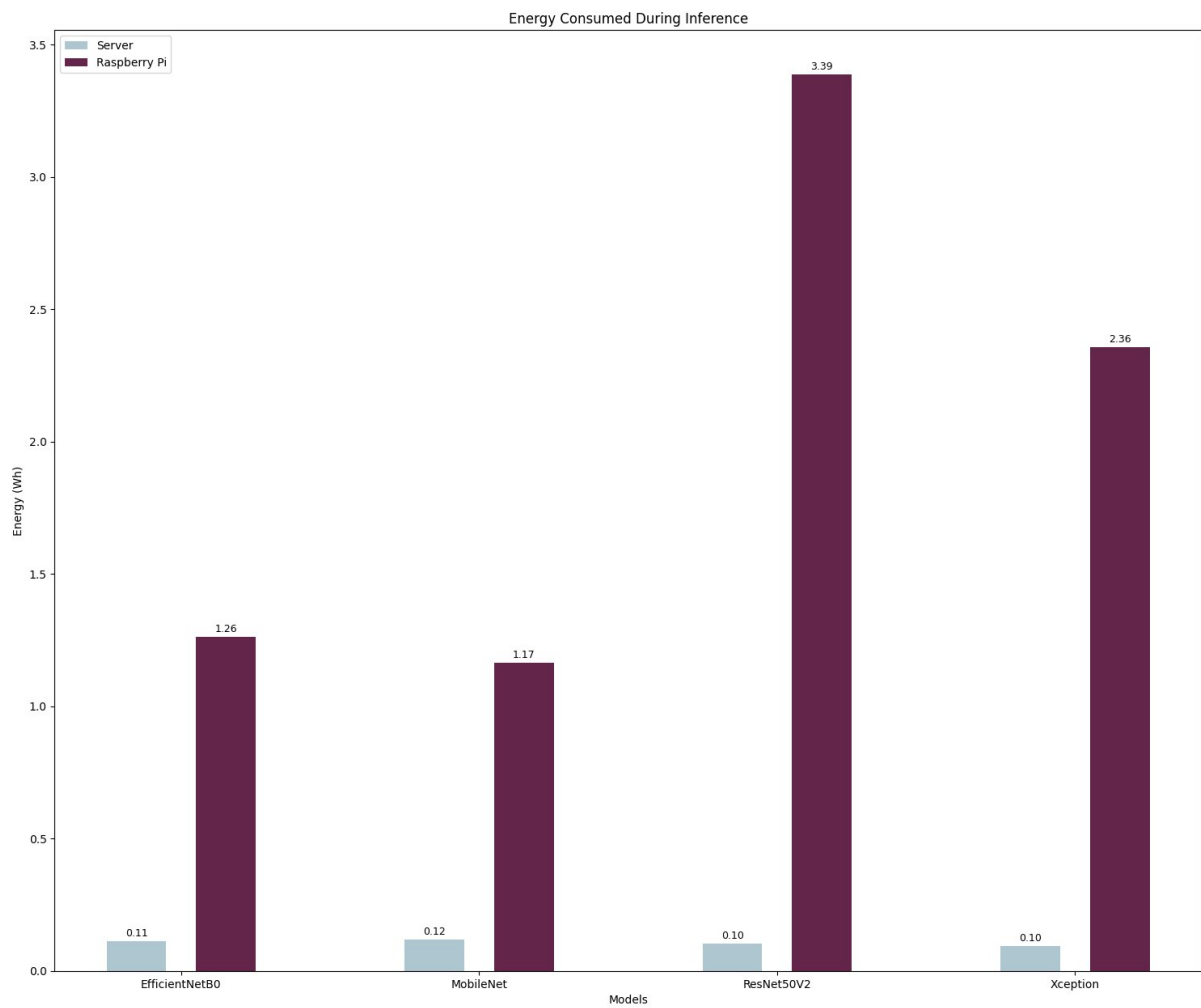


Figure 151: Inference Energy Consumption Comparision

Inference energy consumption of the server and Raspberry Pi of the top four models is shown in this figure. In EfficientNetB0 on the server, it consumed 0.11 Wh, and the Pi showed 1.26 Wh. On the other hand, MobileNet on the server required 0.12 Wh, the Pi consumed 1.17Wh. Moreover, ResNet50V2 on the server required 0.10 Wh, and Raspberry Pi showed 3.39Wh. At last, Xception on the server required 0.10 Wh, and Raspberry Pi showed 0.10. In this case, MobileNet is doing better.

4.3.1.5. Video Stream Handling Capability (Theoretical Maximum)

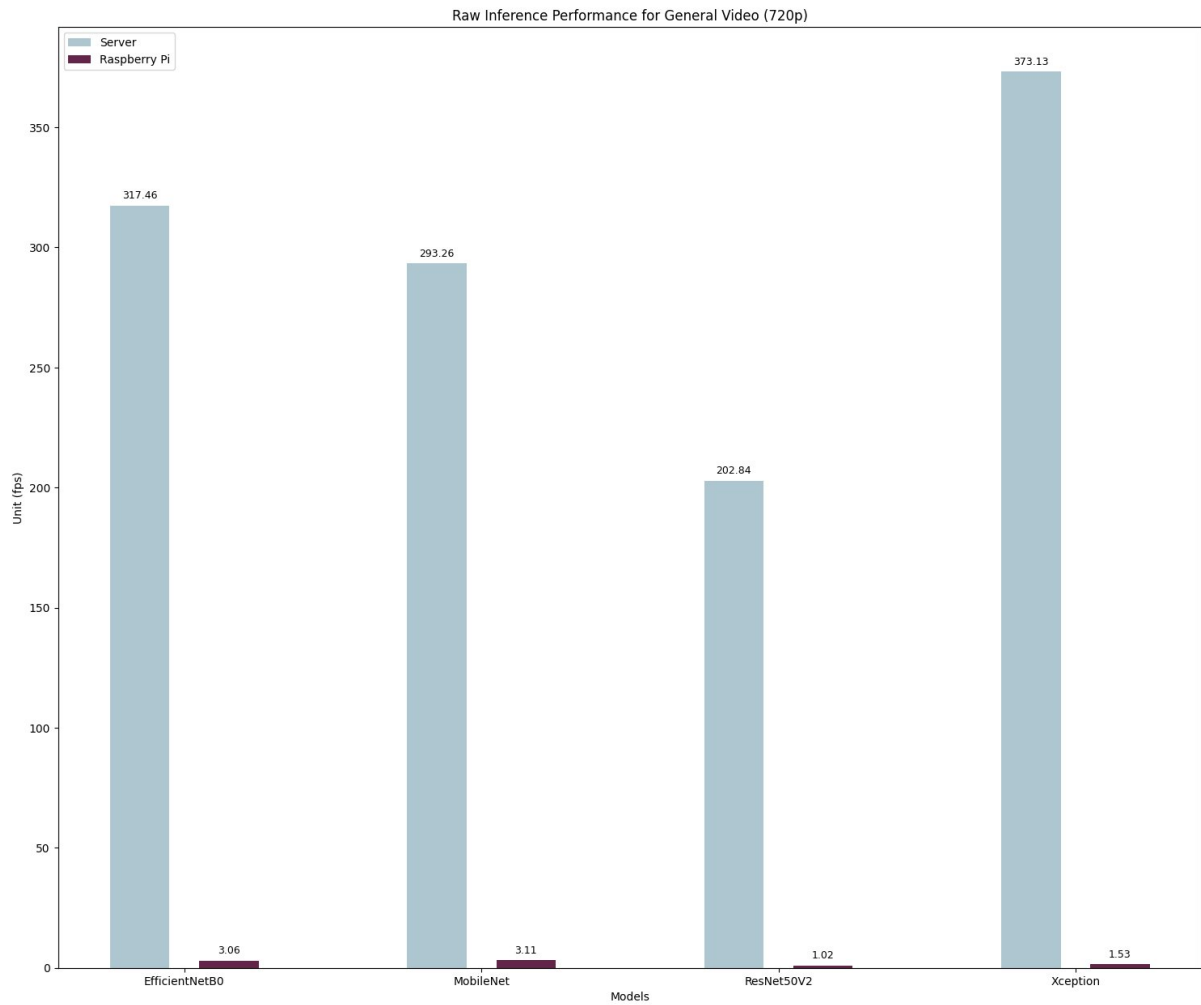


Figure 152: Comparision of Maximum Theoretical Framerate

Video Stream Handling The capability of the server and Raspberry Pi of the top four models are shown in this figure. In EfficientNetB0 on the server, it took 317.46 fps and it took on the Raspberry Pi 3.06 fps. On the other hand, MobileNet on server required 293.26 fps, Raspberry Pi showed 3.11 fps. Moreover, ResNet50V2 on the server took 202.84 fps, while on Raspberry Pi it resulted in 1.02 fps. At the end, Xception on the server was 373.13 fps, and Raspberry Pi was 1.53 fps. In this case, ResNet50V2 is doing better.

4.3.1.6. Peak Power Draw During Training

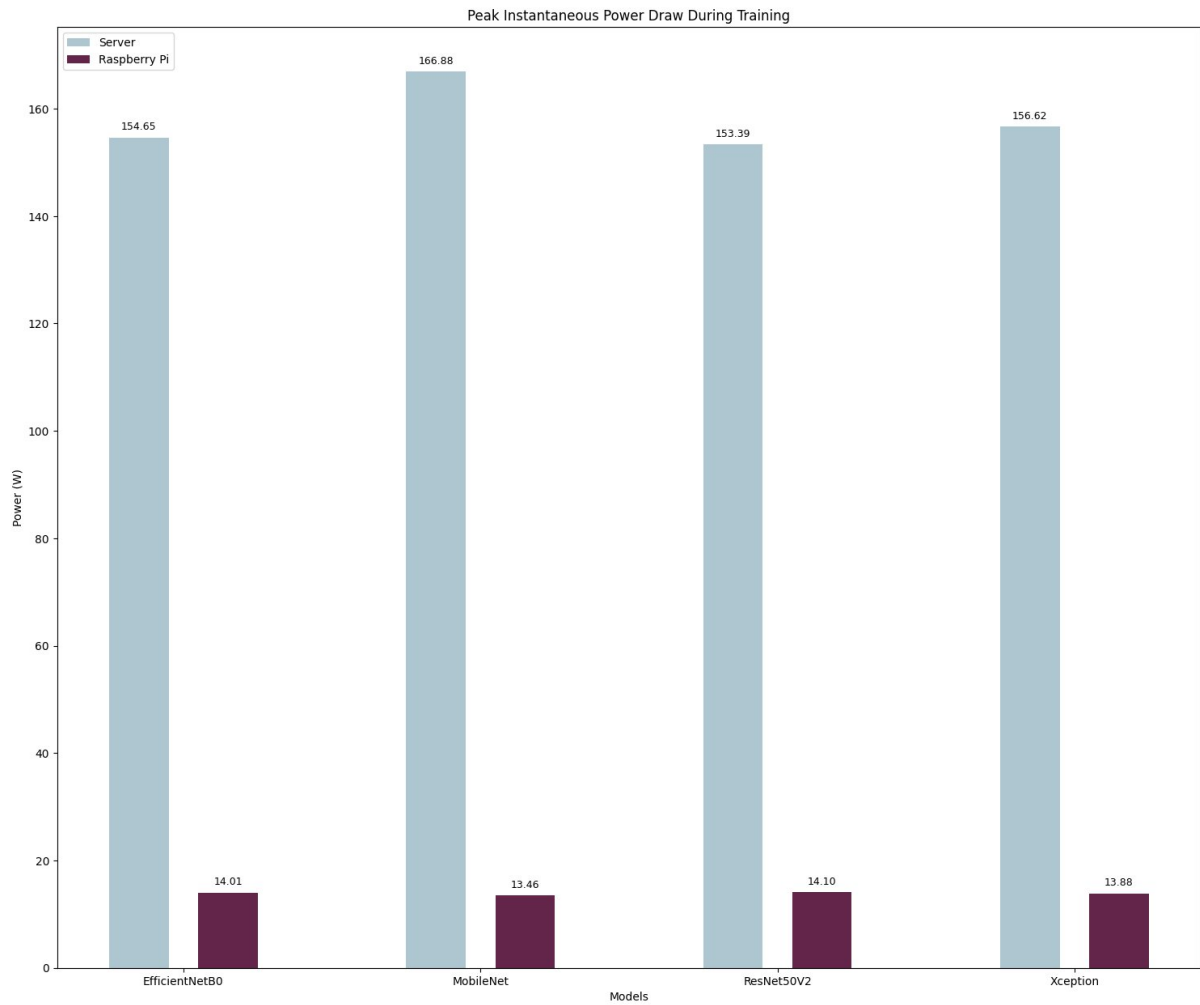


Figure 153: Peak Power Draw Comparison During Training

Peak powerdraw of the server and Raspberry Pi of the top four models is demonstrated in this figure. EfficientNetB0, has drawn 154.65 W, but the Raspberry Pi draws 14.01 W . On the other hand, MobileNet on the server required 166.88 W, Pi showed 13.46 W. Again, ResNet50V2 on the server required 153.39 W, again Raspberry Pi showed 14.10 W. At last, Xception on the server required 156.62 W, and Raspberry Pi shows 13.88 W. In this case, ResNet50V2 is doing better.

4.3.1.7. Average Power Draw During Training

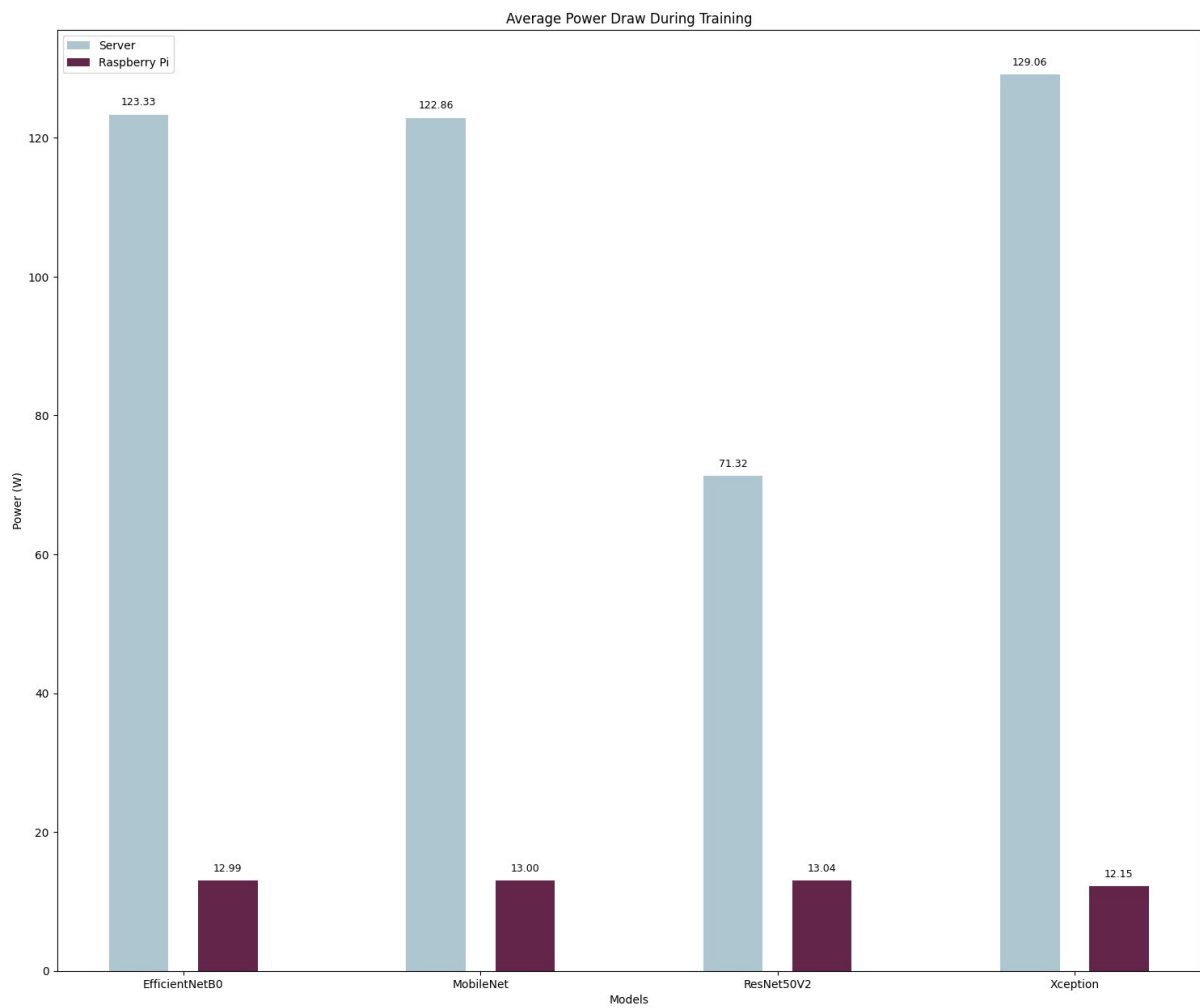


Figure 154 Average Power Draw Comparision During Training

Average Power during training of the server and Raspberry Pi of the top four models is shown in this figure. In EfficientNetB0 on the server, it required 123.33 W, whereas the Raspberry Pi has drawn 12.99 W. On the other hand, MobileNet on the server required 122.86W, but Raspberry Pi drew 13.00 W. While ResNet50V2 on the server required 71.32W, and Pi showed 13.04 W. At last, Xception on the server required 129.06W, again Raspberry Pi showed 12.15 W. In this case, ResNet50V2 is doing better.

4.3.1.8. Total Energy Consumed During Training

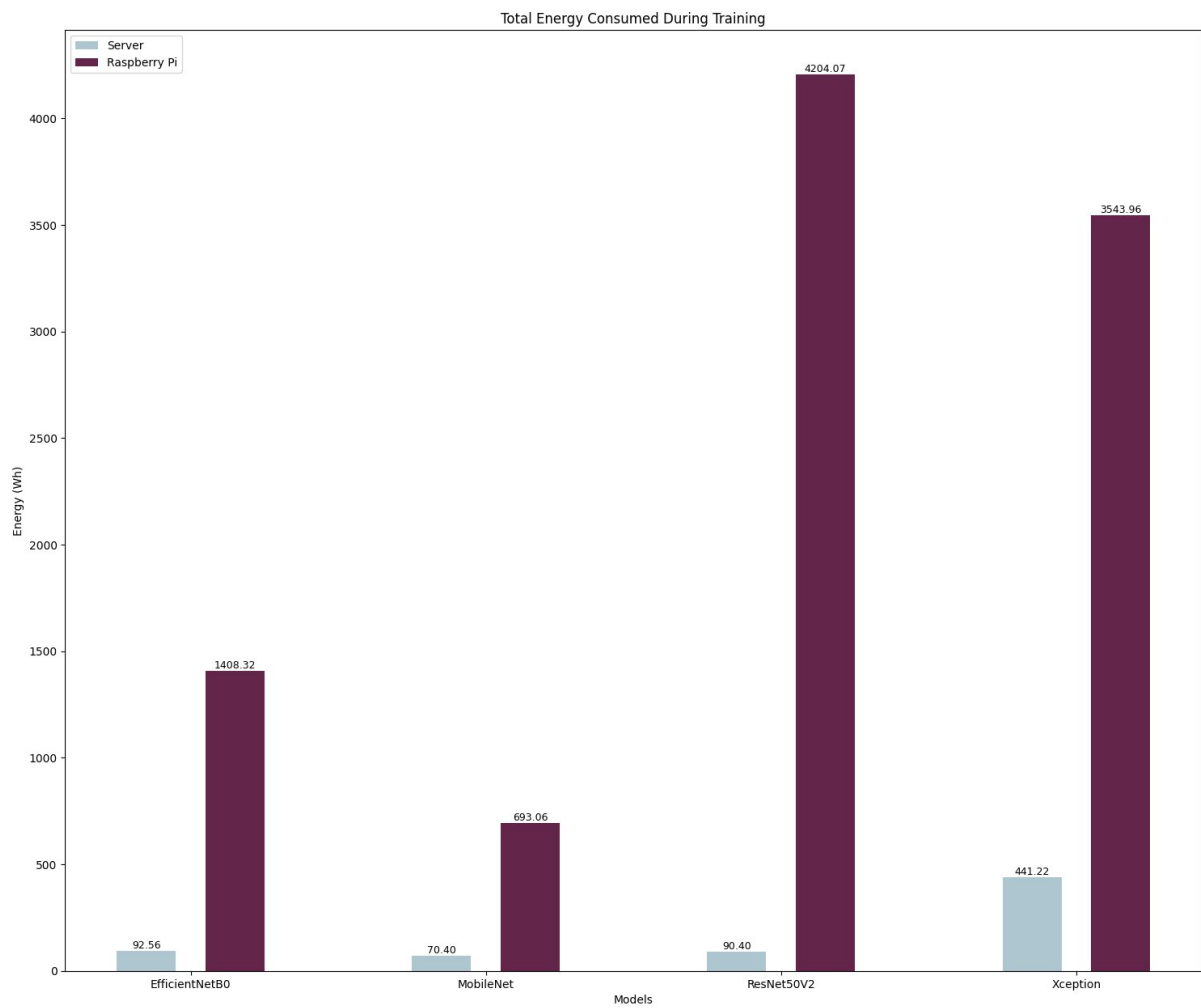


Figure 155: Comparison of Total Energy Consumption During Training

Total energy consumed training of server and Raspberry Pi of the top four models is shown in this figure. In EfficientNetB0 on the server, it required 92.56 Wh. On the other hand Raspberry Pi consumed 1,408.32 Wh. Again, MobileNet on the server required 70.40 Wh, Raspberry Pi showed 693.06Wh. ResNet50V2 on the server required 90.40 Wh, Pi showed 4,204.07 Wh. At last, Xception on the server required 441.22 Wh, and Pi showed 3,543.96 Wh. In this case, MobileNet is doing better.

4.3.1.9. Peak Power Draw During Inference

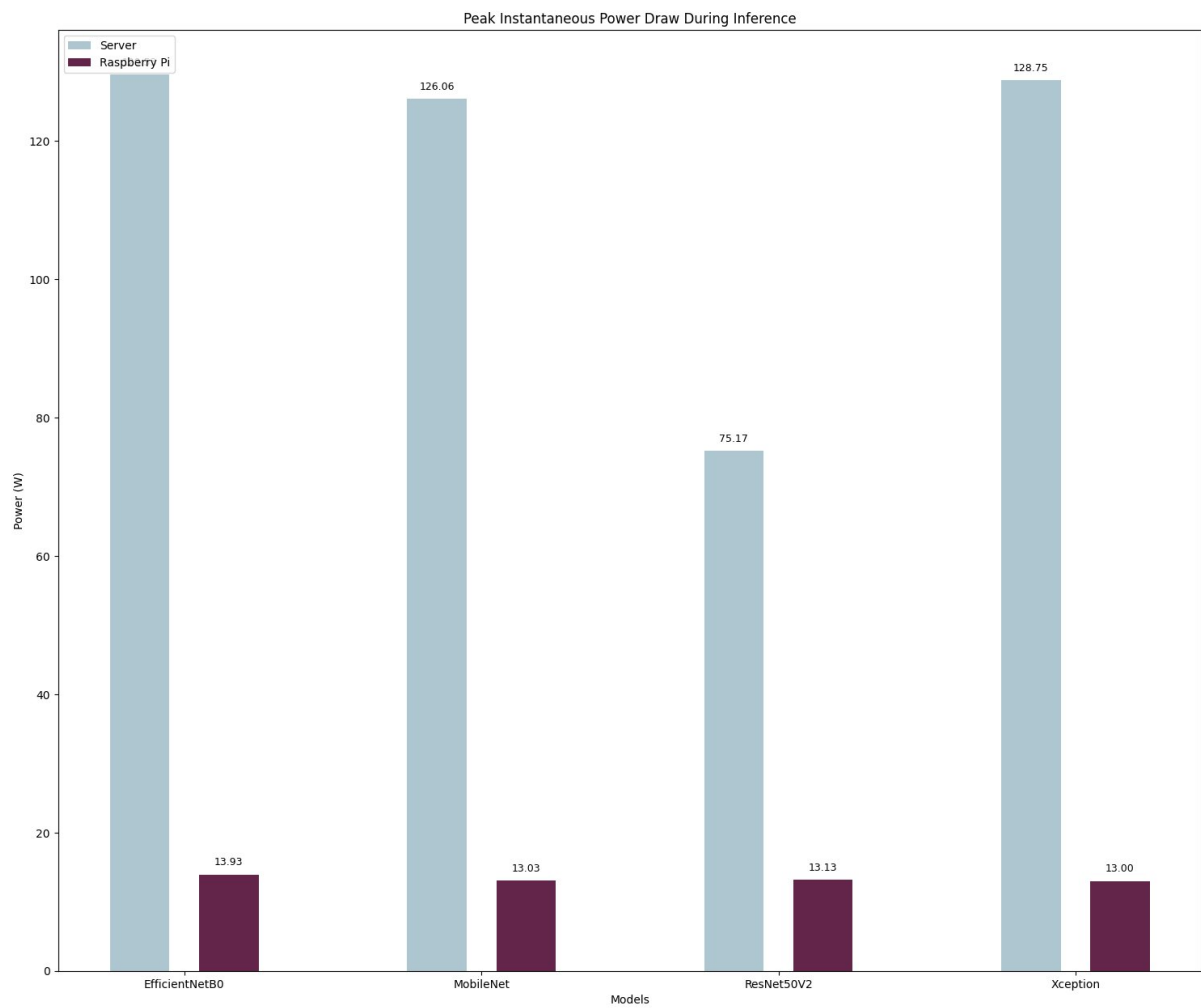


Figure 156: Peak Power Draw Comparison During Inference

Peak power draw for inference of server and Raspberry Pi of the top four models is shown in this figure. In EfficientNetB0 on the server, it required 126.96W, and the Raspberry Pi showed 13.93W. On the other hand, MobileNet on the server required 126.06W, but Raspberry Pi showed 13.03W. Moreover, ResNet50V2 on the server required 75.17W, again Raspberry Pi showed 13.15W. At last, Xception on the server required 128.75W, and Raspberry Pi showed 13.00W. In this case, ResNet50V2 is doing better.

4.3.1.10. Comparison of Power Curves During Training for Both Design Approaches

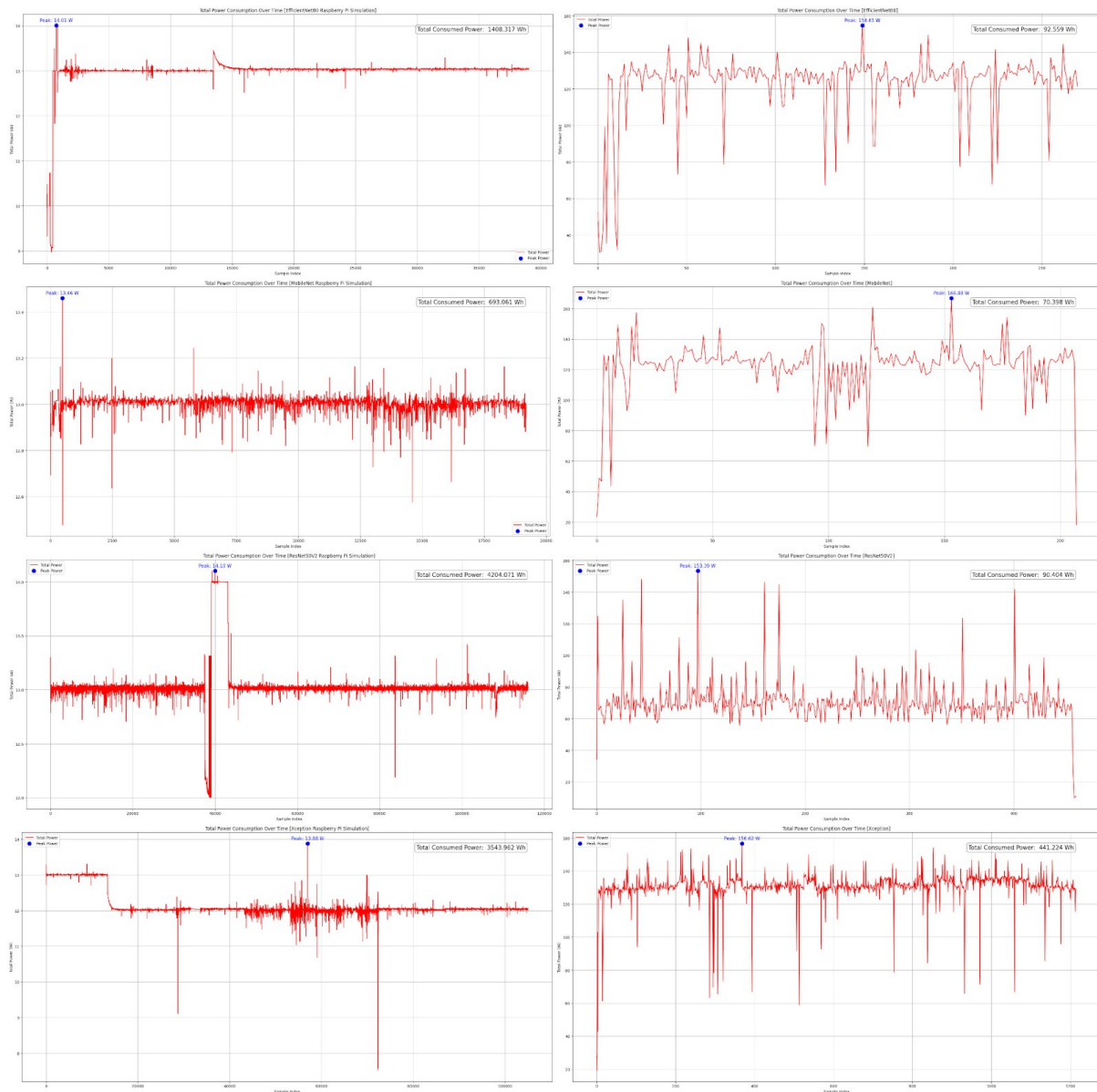


Figure 157: Power Draw Profile Comparison

In this figure is the power curve training of Pi vs. x86_64 of the top model in the server and Raspberry Pi system. In the case of EfficientNetB0 on Raspberry Pi, the peak is 14.01 W. On the other hand, the server is 154.65W. On the other hand, if we look at the total consumed power, it is 1408.317 Wh and 92.559Wh, respectively. In the case of MobileNet, the peak in Raspberry Pi and the server is 13.46W and 146.88W individually in case of power curve. If we look at the total consumed power, it is 693.061Wh and 70.398Wh. Then, looking at ResNet50V2 in the case of studying the Raspberry Pi and server model, the peak spike of the power curve is at 14.10 W and 153.39W. The total power consumption is 4,204.071Wh and 90.404 Wh individually. Looking at the last model called Xception the peak in Raspberry Pi and the server is 13.46 W and 146.88W each. Moreover, the total power consumption is 3543.962 Wh and 441.224 Wh separately.

4.3.1.11. Total Training Time

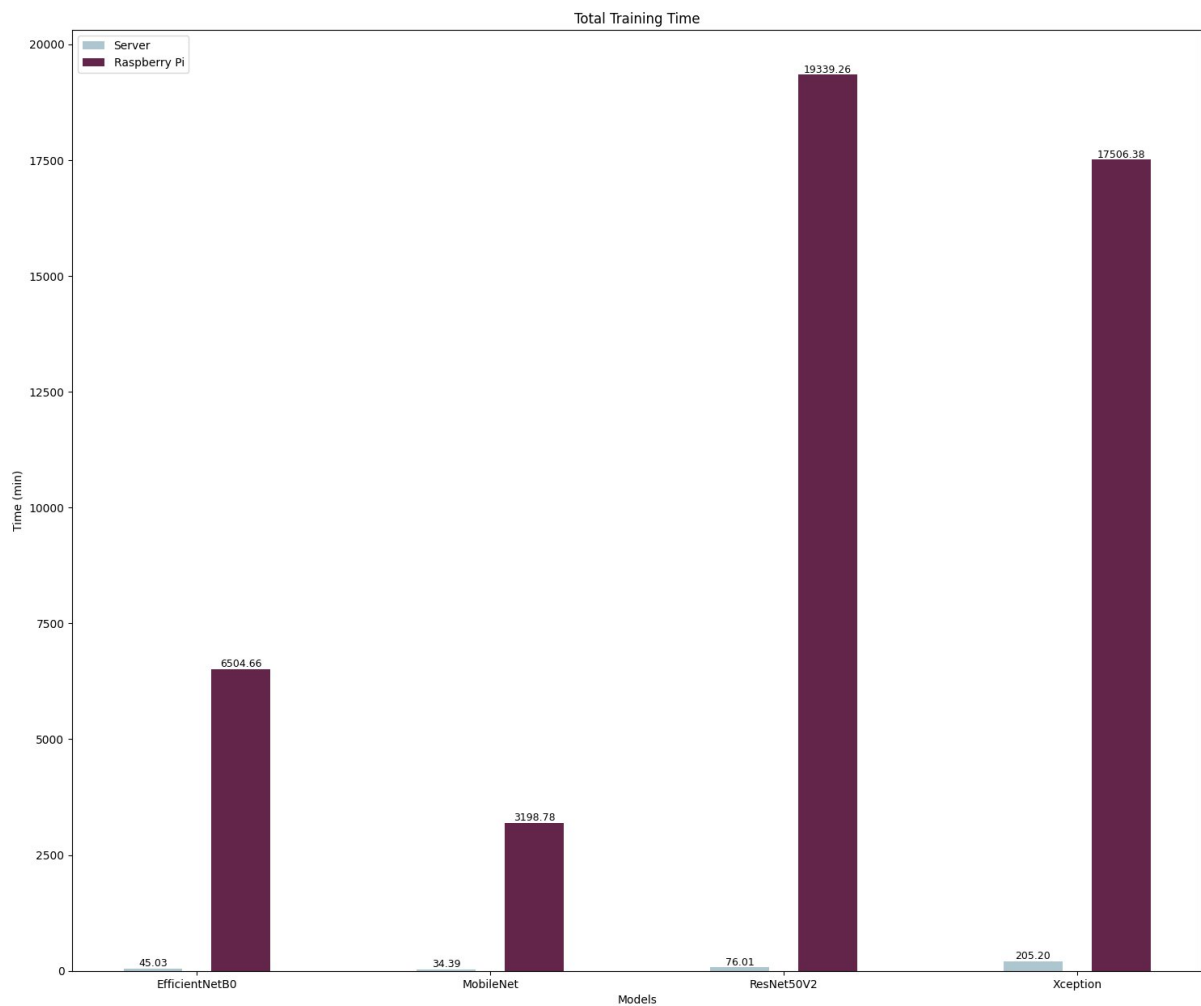


Figure 158: Total Training Time Comparison

Total training time of server and Raspberry Pi of the top four models is shown in this figure. In EfficientNetB0 on the server, it required 45.03 seconds, and the Raspberry Pi showed 6,504.66 seconds. On the other hand, MobileNet on the server required 34.39 seconds, and Raspberry Pi showed 3,198.78 seconds. Moreover, ResNet50V2 on the server required 76.01 seconds, and Raspberry Pi showed 19,339.26 seconds. At last, Xception on the server required 205.20 seconds, and Raspberry Pi showed 17,506.38 seconds. In this case, MobileNet is doing better.

4.4. Performance Evaluation of Developed Solution

In the following table a detailed comparison between both the solutions is conducted. Each criteria is given one point and a draw will net both designs one point. At this state from the 12 models that we have trained the top four have been chosen. Among the four top models we have averaged the score across all models to balance out them and to even out the comparison criteria. Every comparison criteria has a significance attribute (denoted by an asterisk). One asterisk means that lower is better and two means higher is preferred. Based on test results and

by following these ground rules we have compiled every key aspect into one easily readable table for direct one to one comparison. The aim is to aid in the finalization of the design approach that we will follow in the manufacturing phase of this project.

TABLE V Optimal Design Comparison Criteria for Choosing the Final Solution

Comparison Criteria	Design Approach 1	Design Approach 2
Training Time*	90.16 min	11,524.77 min
Training Energy Consumption*	173.65 Wh	2,462.35 Wh
Inference Energy Consumption*	0.11 Wh	2.05 Wh
Frame Rate Handling**	296.67 FPS	2.18 FPS
Budget*	BDT 48,500	BDT 46,650
Accuracy**	~85%	~85%
1k Frame Time*	3.54 sec	570.80 sec
Time per Frame*	3.544 ms	570.78 ms
Inference Efficiency**	2.38 frame/Watt	0.17 frame/Watt
Training Peak Power Draw*	157.89 W	13.86 W
Training Average Power Draw*	111.64 W	12.64
Inference Peak Power Draw*	115.00 W	13.27 W
Final Score	8/12	5/12

From the comprehensive testing regime and side by side comparison we can conclude that the first design approach is the optimal solution.

4.5. Conclusion

The project has been done successfully through the design and later simulation of the proposed system. Simulation helped us to anticipate performance of the system and ensure that it works before it is implemented physically. This procedure allowed discovering the possible disadvantages at the initial stage and adjusting the design to them. However, on the whole, the simulation stage proved sufficient to note that we have come up with the desired solution to our intended output.

Chapter 5: Completion of Final Design and Validation [CO8]

5.1. Introduction

The most important part after proving the projects viability is to finalize our selected design and give it a physical form. During this stage, a few alternative architectures were proposed and designed to attain the desirable results. Upon considering all the designs in terms of feasibility, performance and practicality we chose the one that was the most effective. After the selection, we were able to determine the materials, tools, and components needed to be constructed. After the sourcing was settled on, we were to construct the entire system and subsequently compare the actual physical outcome with the project plan.

5.2. Completion of Final Design

5.2.1. Software Stack

The full software stack involves a plethora of tools that are used to gather data and test the functionality of our devices. Almost all of the tools mentioned before were an active part of our completion phase excluding MATLAB.

5.2.1.1. High Level Layout

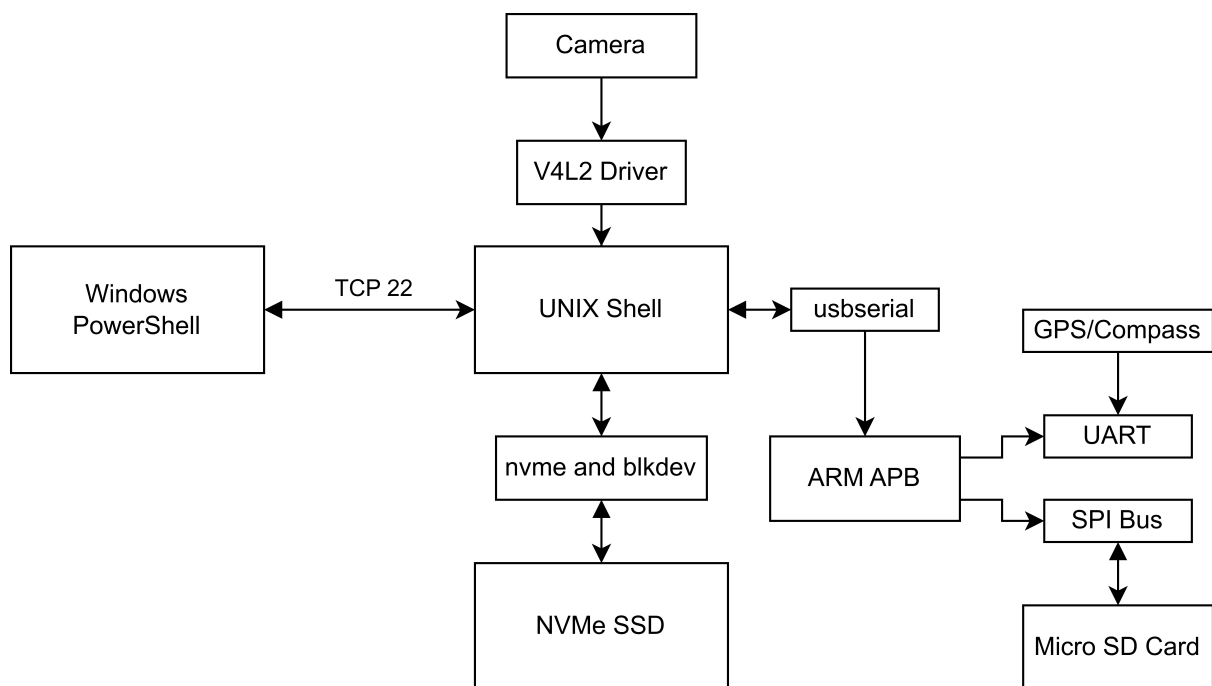


Figure 159: High Level System Software Overview

The Windows shell acts as a command and control server that sends commands via a SSH tunnel. This has been set up so that it does not require entering the passphrase every time. A public key cryptography system has been implemented to enable this feature. The command and control server communicates directly to the device via the UNIX shell. It then further commands to control recording, position logging, time drift correction as well as fine tuned control of the on board image processor.

5.2.1.2. Recording System

When the predefined path or manual mode is engaged and the operator wants to record video, the following command is sent via SSH.

```
pkill rpicas-vid; cd /home/comp_module/workspace/data_dump; nohup rpicas-vid  
--camera 1 --width 1000 --height 1000 --framerate 120 -t 0 --nopreview -o  
$timestamp.h264 > /dev/null 2>&1 &
```

This initiates a recording session with the proper name, H.264 (also known as AVC or MPEG-4 Part 10) is used in case of a catastrophic failure. If the drone can be salvaged and the data from the onboard NVMe SSD can be extracted then the footage will not be corrupted (given the SSD itself is not damaged and does not require advanced data recovery techniques). The file that is generated in this method is essentially images in serial unlike other encoding formats where no graceful end will cause a corrupted video (which can also be recovered but there is a greater chance of data loss in that case). Additionally this format allows the operator to push the sensor to ultra fast frame rates as the CPU is not overloaded by both recording and transcoding to another format (this is a native format of the Sony IMX series sensors used in the camera module).

Property	Value
Description	
Title	
Subtitle	
Rating	☆☆☆☆☆
Tags	
Comments	
Video	
Length	00:00:00
Frame width	352
Frame height	352
Data rate	11871kbps
Total bitrate	11871kbps
Frame rate	1000.00 frames/second
Audio	
Bit rate	
Channels	
Audio sample rate	
Media	
Contributing artists	

Figure 160: Max Raw Usable FPS

Windows often time fails to properly report the lengths of high frame rate H.264 video files which is the reason why it is saying zero video length (the system has no audio recording capability thus there is no audio track).

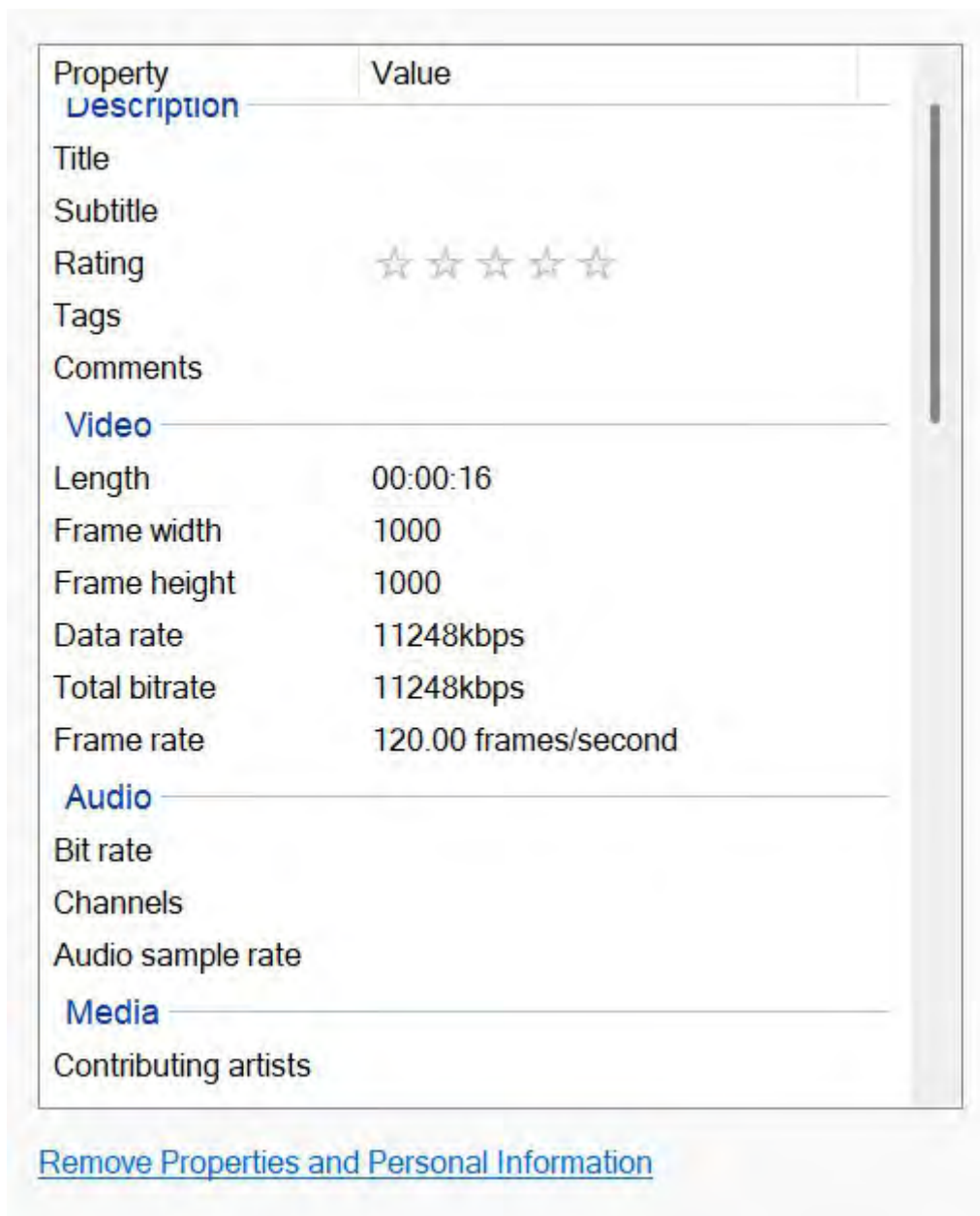


Figure 161: Actual Useful FPS

Where as after conversion to a known format MP4 AVC1 (as shown below), note this is accelerated with an Nvidia GPU to speed up the encoding process

```
ffprobe -hide_banner OUTPUT.mp4
Input #0, mov,mp4,m4a,3gp,3g2,mj2, from 'OUTPUT.mp4':
  Metadata:
    major_brand      : isom
    minor_version    : 512
    compatible_brands: isomiso2avc1mp41
    encoder          : Lavf62.6.103
  Duration: 00:00:16.19, start: 0.000000, bitrate: 11120 kb/s
  Stream #0:0[0x1](und): Video: h264 (Main) (avc1 / 0x31637661), yuv420p(tv, bt709, progressive), 1000x1000 [SAR 1:1 DAR 1:1], 11115 kb/s, 120 fps, 120 tbr, 15360 tbn (default)
```

```

Metadata:
  handler_name      : VideoHandler
  encoder           : Lavc62.21.100 h264_nvenc

```

5.2.1.3. Time Drift Mitigation System

The on board imaging computer has no system clock battery. Thus if the device is power cycled it loses time. In order to sync with the network time (for precise positioning and feature extraction) the following process takes place.

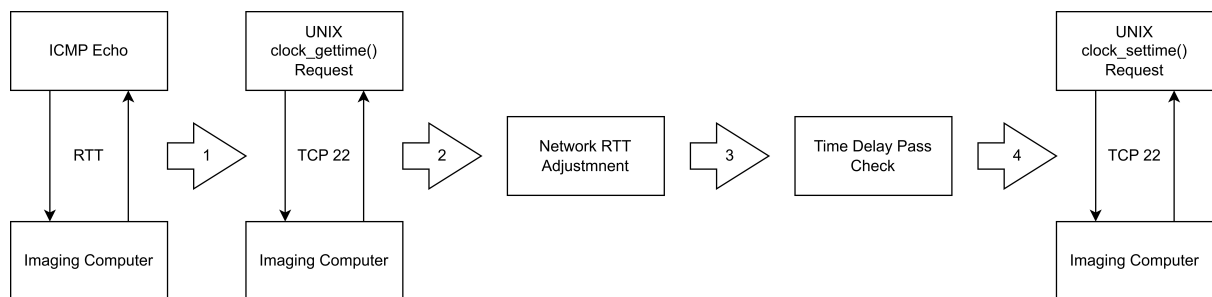


Figure 162: Time Sync Architecture

The command and control server first sends a ICMP echo request and measures network RTT. After that a UNIX time request is sent and the response is recorded. When adjusted with the RTT if the dilation is greater than 1ms then it sets the current network time (adjusted for network RTT, thus the time is in the future, current time plus the RTT time so that when the radio wave reaches the imaging computer it will be way more accurate).

5.2.1.4. Transfer Pipeline

All recorded data (from the cameras and the flight data recorder) are transferred over the secure copy protocol or SCP. Once all data is copied the files the following command is sent over SSH (TCP 22) to over write the memory cells so that no data cannot be recovered.

```

ssh user_mod@[REDACTED] 'cd /home/comp_module/workspace/data_dump && shred -n 2 -z -u *'

```

This first over writes the memory cells with random noise in two passes and finally writes zeroes over them. Followed by erasure of all residual files within that dump directory. There is a counter argument that can be made here that it would drastically reduce the life span of the NAND flash memory used as the scratch and boot disk. The write endurance is nowhere near enterprise solid state storage but it is still a cost that we are willing to make in order to ensure security of our data in case the device needed to be deployed to collect data of restricted locations.

5.2.1.5. Location Data Recorder

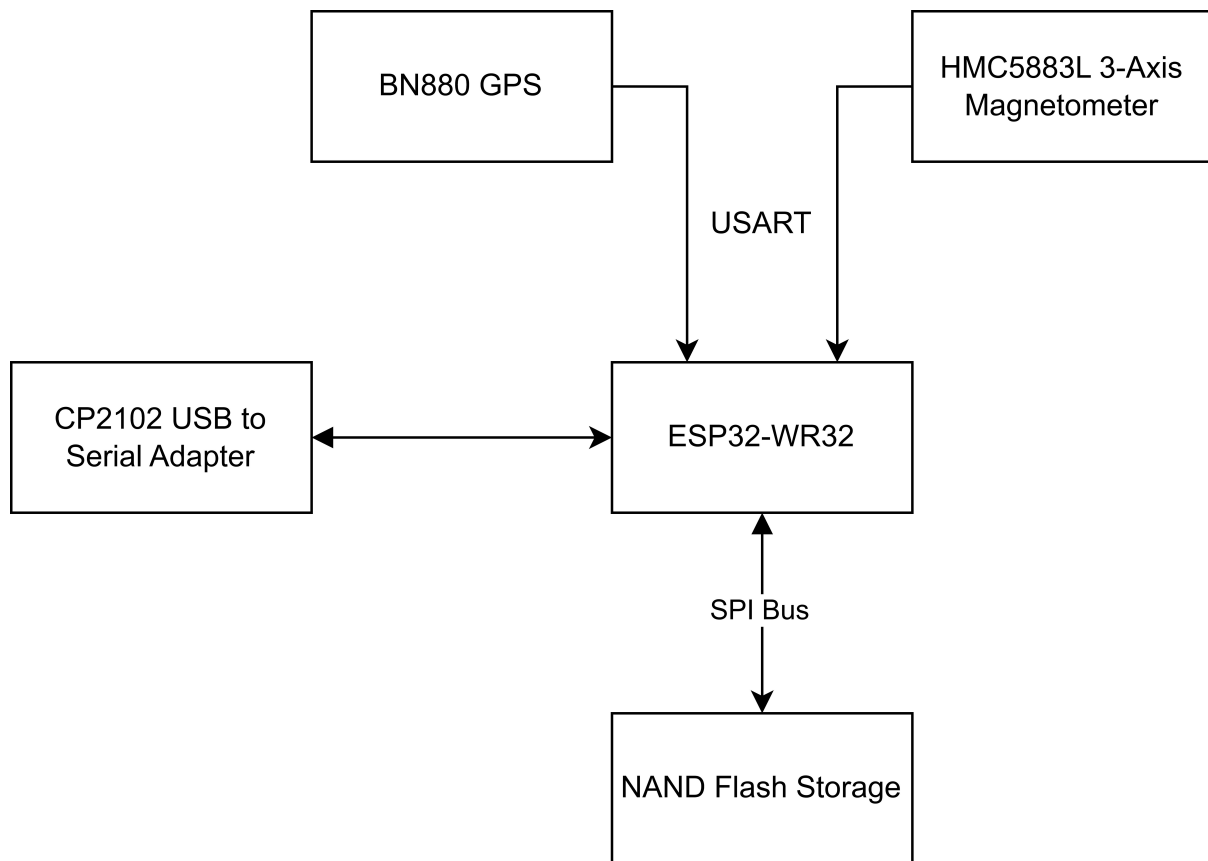


Figure 163: Location Data Recording System Overview

The command and control host sends instructions over USART at 115.2kbps to initiate GPS and compass tracking. All the generate files are date and time stamped so it is easier to process once imported into the processing suite. The capture data is stored into a flash storage device (formatted to FAT32 in order to meet the requirements of the SD bus and associated software libraries).

5.2.1.6. Distance Measurement

There is an ability to measure distance of the landing surface or to perform specific and precise targeting for either training and landing tasks. This is done primarily via OpenCV and the rplicam application stack.

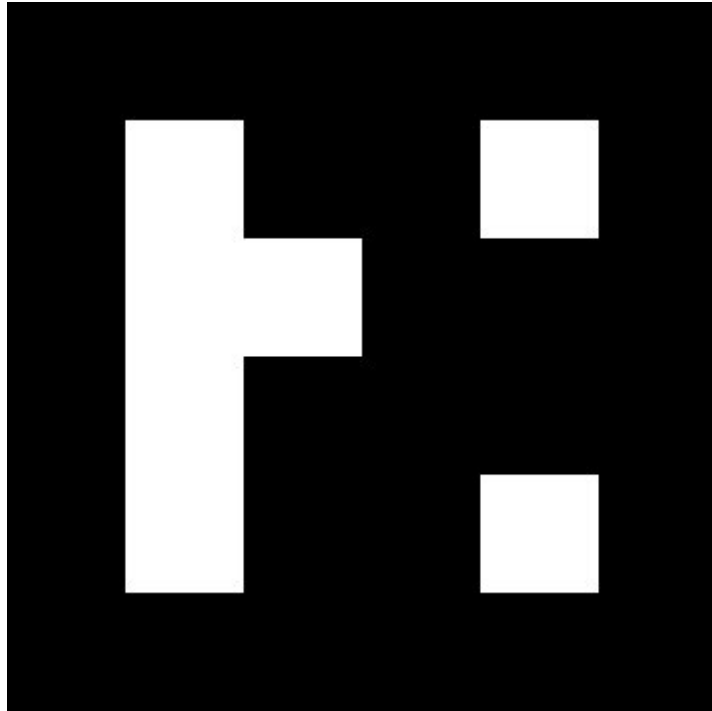


Figure 164: 4x4 ArUco Marker

This is an image of the markers that has been used to make the precise optical measurement possible. ArUco (Augmented Reality University of Córdoba) is the name of the standard and has been chosen instead of object detection as it is baked into OpenCV. Thus there is significantly less processing overhead that could increase latency and reduce responsiveness to operator commands.

5.2.1.7. Integrated Tool Chain

The following tool was made using the open source Windows Forms (WinForms) .NET (dotNET) framework provided by Microsoft. This enabled us to avoid tedious custom UI design using C++ libraries (SDL) or QT.

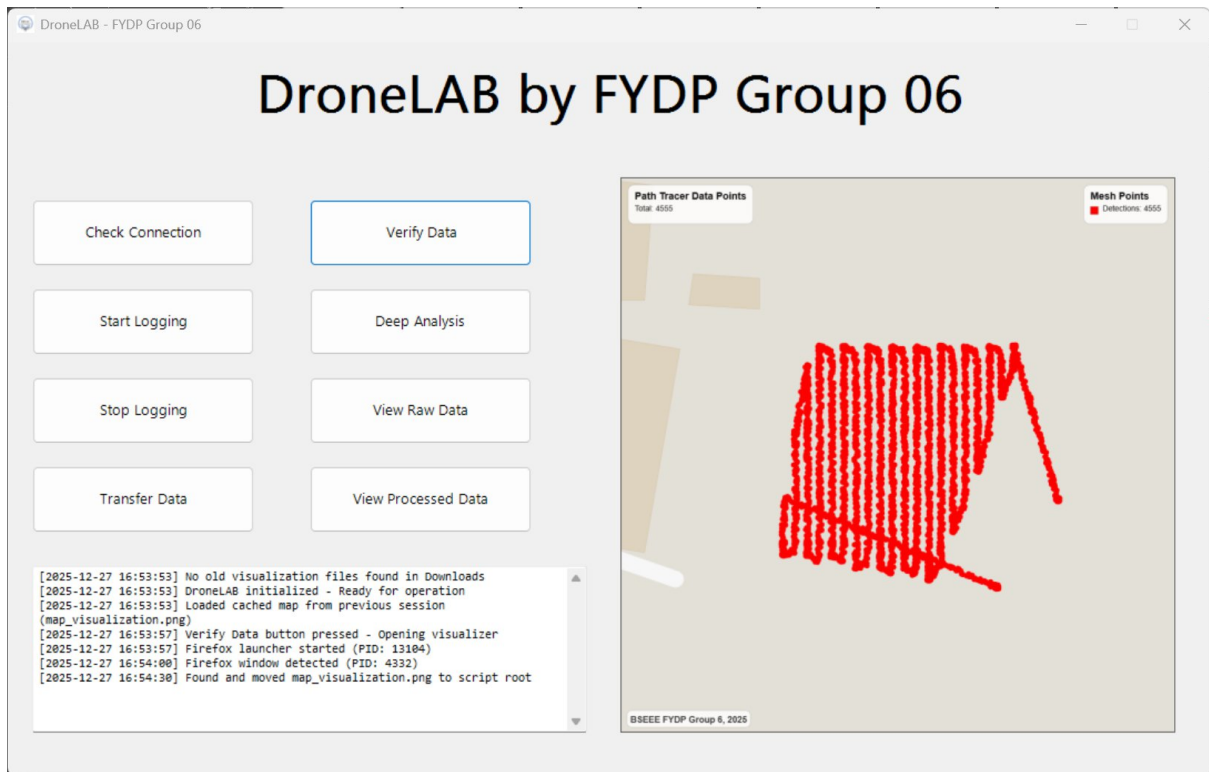


Figure 165: Full Version of DroneLAB

This is the fully operational mode of the DroneLAB suite. This integrates all the functionality of our custom MapDVT (Data Verification Tool), MapLAB and the detection utilities in one place.

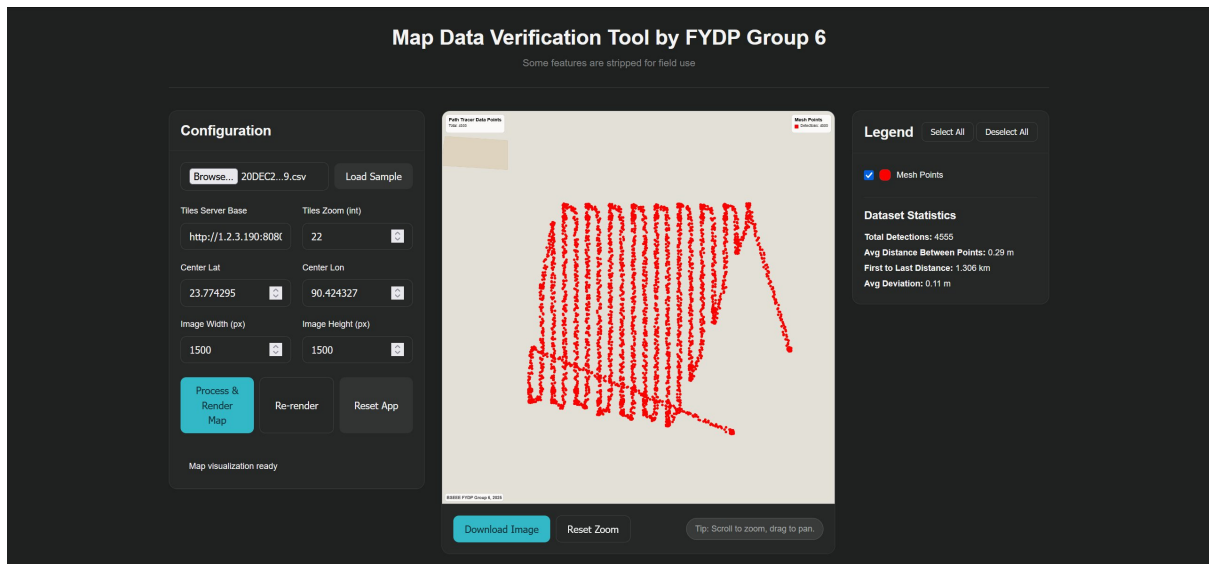
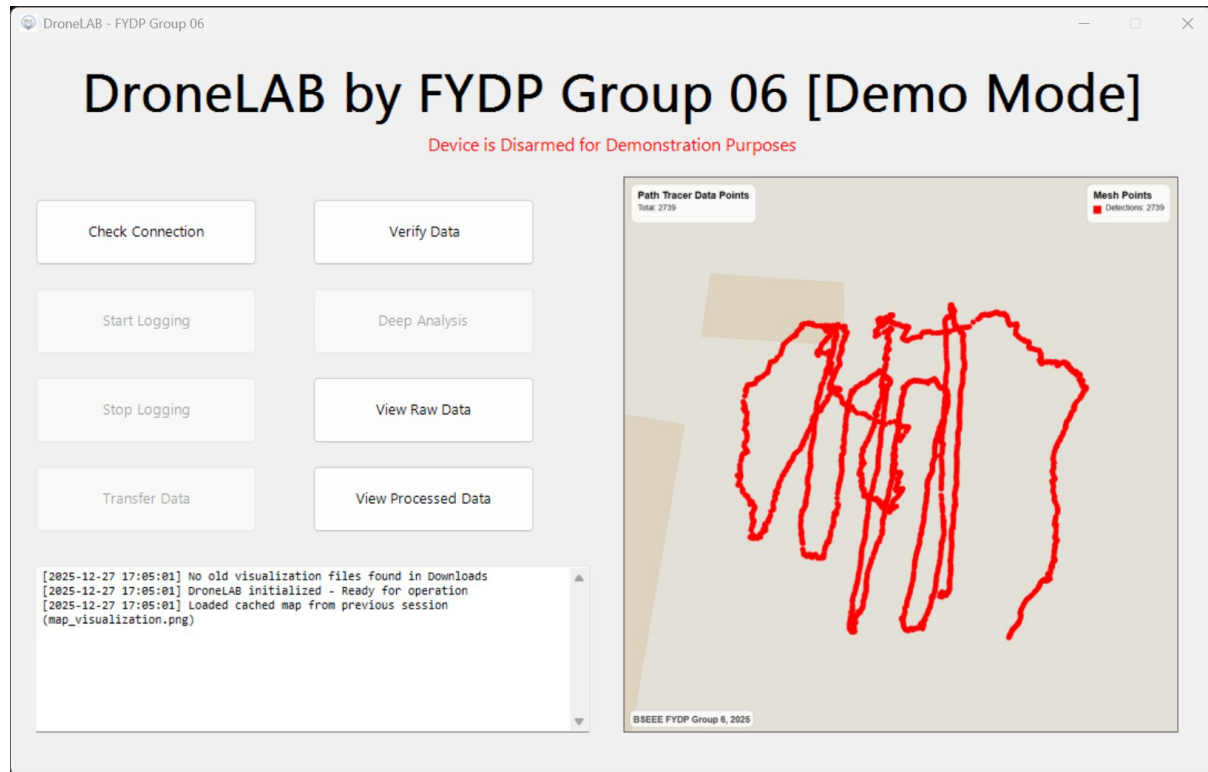


Figure 166: Modified Version of MapDVT

The MapDVT tool has been modified to show that the recorded data has been accurately collected and sent back to the command and control server. Some features have been removed

from the tool that was originally developed during the simulation phase of the project. As we have determined that it takes quite some time in order to process the data even with hardware acceleration. Thus the MapDVT tool has been adapted to work with raw CSV data.



This is the demo mode for the same utility that we implemented for safely displaying the device without taking the propellers off. It is an additional measure to ensure the safety where an unplanned power on event could put people in danger.

5.2.2. Hardware Components

5.2.2.1. Component Level Specification

TABLE VI Components

Component	Model	Specification
Imaging/Compute Platform	Raspberry Pi 4 Model B	SOC: Broadcom BCM2711 Instruction Set: ARMv8-A Instructions: 64-bit Architecture: ARM Cortex-A72 Core Count: 4 Clock Speed: 1.5GHz (Nominal) to 1.8GHz (C0 Stepping) TDP: 7.5W Process Node: 28nm CMOS L1 Cache: 32kB DAT + 48kB INS L2 Cache: 1MB shared SDRAM: 4G LPDDR4-3200 iGPU: VideoCore IV iGPU Clock: 500MHz Rendering API: OpenGL ES 3.1, Vulkan 1.0 Storage: Expandable (SD) Power Draw: 15W @ 5V

		WLAN Chip: Cypress CYW43455 WLAN Features: IEEE 802.11ac Dual Band (2.4/5G)
sRGB Camera	RPI-CAM-V2	Sensor: Sony IMX219PQ Type: Back Illuminated (APS) On Chip A/D: Sony Exmor R™ Resolution: 8MP Shutter: Electronic (Rolling) Color Filter: Bayer Primary CFM Frame Rate: 30Fps (Max. All Pixel Scan Mode), 60Fps (FHD V-Crop), 180Fps (HD 2x2 Analog Binning) Pixel Rate: 280MP/s Data Rate: 755Mbps (4 Lane), 912Mbps (2 Lane) Sensor Size: 4.60mm Diagonal (Type 1/4.0) Pixel Count: 8.28M (Total), 8.17M (Effective), 8.08M (Active) Unit Cell Size: 1.12µm (Square) Substrate Material: Silicon Power Rails: 1.2V, 1.8V and 2.8V
Sub-Routine Handler	ESP32	MCU: ESP32 Architecture: Xtensa LX6 32-bit (Dual-Core) Program Memory: 4MB Flash (typical, external) RAM: 520kB SRAM EEPROM: Emulated (in Flash) Pin Count: 48 QFN Operating Voltage: 2.2V~3.6V ADC Channels: 18 ADC Resolution: 12-bit Comm. Protocols: I²C, SPI, UART, PWM, I2S Special Features: Wi-Fi (802.11 b/g/n), Bluetooth (Classic + BLE)
GPS Tracking Module	BN-880	Chipset: M8030-KT Compass: HMC5883L Memory: 4M Flash Channels: 72 Formats: GPS, GLONASS, Galileo, BeiDou, QZSS and SBAS
Drone Frame	LJI X500	Diagonal Wheelbase: 500mm Dimensions: 355×355×220mm Weight: 453g Build Material: Full Carbon Fiber Type: 3K Fiber Motor Support: 2212, 2213, 2216
Motor	TMotor AS2820 KV880 BLDC Motor for Fixed Wings	Brand: T-motor Series: AS2820 kV Rating: 880kV Configuration: 12N14P Rated Voltage: 3-4S Idle Current: 1.98A @10V Peak Current: 45A, 180s Max Power: 626W, 180s Motor Dimension: Ø36.2×62.2mm Weight: 136g
ESC	40A 2-4S Brushless Motor Electronic Speed Controller	Continuous Current: 40A Burst Current: 55A, >10s BEC output: 5V@3A Voltage Input: 2-4S (LiPo) Net Weight: 30g Size: 68×25×8mm
Propeller	1255 Carbon Fiber	Length: 12"

	Propeller	Pitch: 5.5" Weight: 25gm Mounting Hole: Ø4mm Aux Mount Distance: 12mm (from center)
Flight Controller	Pixhawk 2.4.8	Integration: PX4FMU + PX4IO Platform: Pixhawk, 32-bit chip Processor: STM32F427 Cortex-M4 RAM: 256kB, 168MHz Backup Coprocessor: 32-bit STM32F103 Gyroscope: L3GD20 3-axis digital 16-bit Primary IMU: LSM303D 3-axis 14-bit Aux IMU: MPU6000 6-axis Barometer: MS5611 UART Ports: 5 (1 HV, 2 RTS/CTS) CAN Ports: 2 Radio Compatibility: Spektrum DSM/DSM2/DSM-X/SBUS Input Interfaces: PPM/RSSI/I2C/SPI ADC Input Voltages: 3.3V and 6.6V USB Interface: External Micro USB
Battery	LiPo	Capacity: 5400mAh Voltage: 11.1V 3S Max. Discharge Rate: 60C Max. Charging Rate: 5C Main Plug: XT60 Weight: 315gm Size: 140×42×28mm Product Certification: CE, FCC, ROHS, ISO:9001, SGS System
Power Distribution Board	5 in 1 PDB+Shock Absorber+Power Module+5&12V BEC for APM/Pixhawk	Special feature: Multi-Copter battery connection ESC/Motor support: Up to 16 Maximum Input Current: 100A Weight: 6.7gm Size: 55×55mm

5.2.2.2. Flight Subsystem

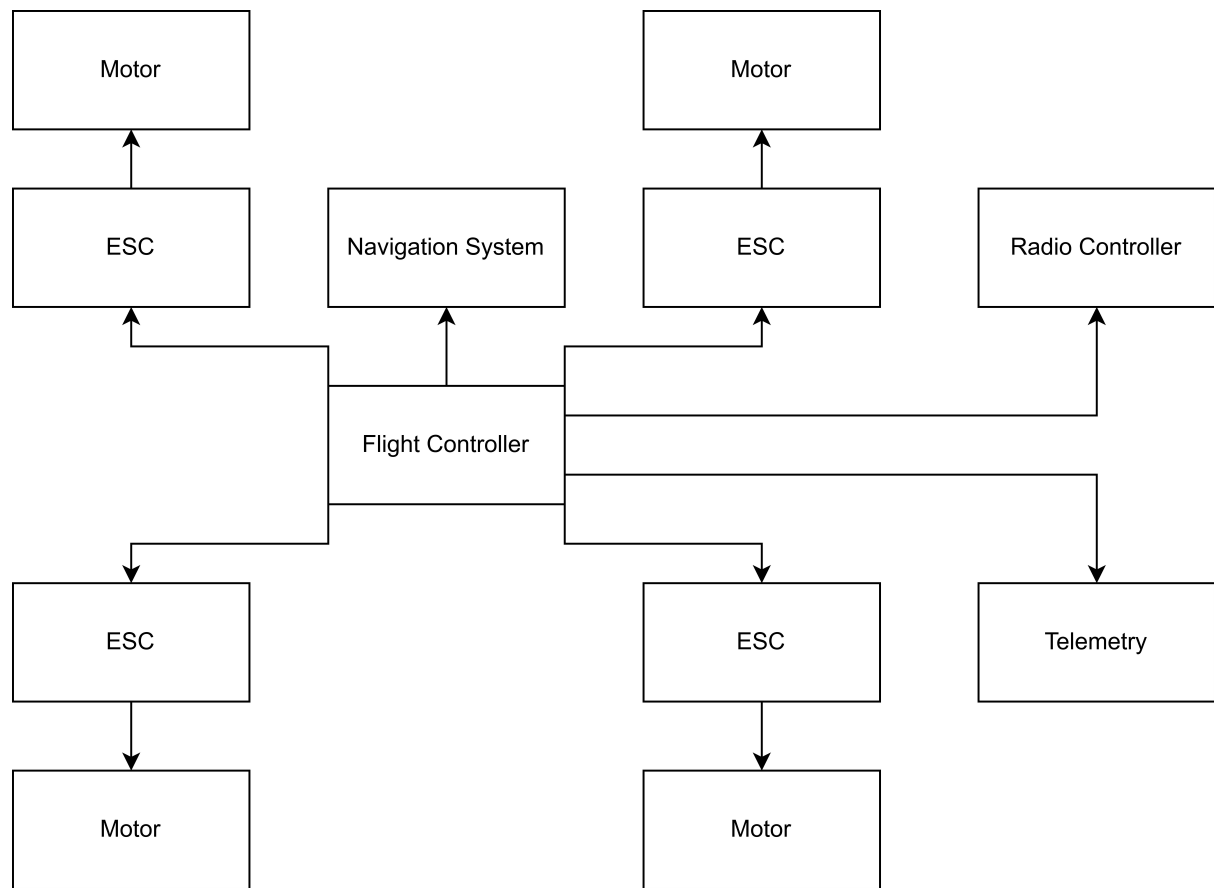


Figure 167: High Level Overview of Flight System

The heart of our device is the flight controller as this is the autonomous part of our device. We actively decided to keep this part modular to prevent complication as modular design was also a part of our design process. The flight controller takes care of balancing and maintaining altitude, speed and position in 3D space. The backup radio is apart from the telemetry system. This allows the operator to take control over the drone at any moment. The telemetry sensor operates at ISM 433MHz and the manual override radio system works at ISM 2.4GHz. Both boasting a range of at least 3.5 kilometers.

5.2.2.3. Imaging Subsystem

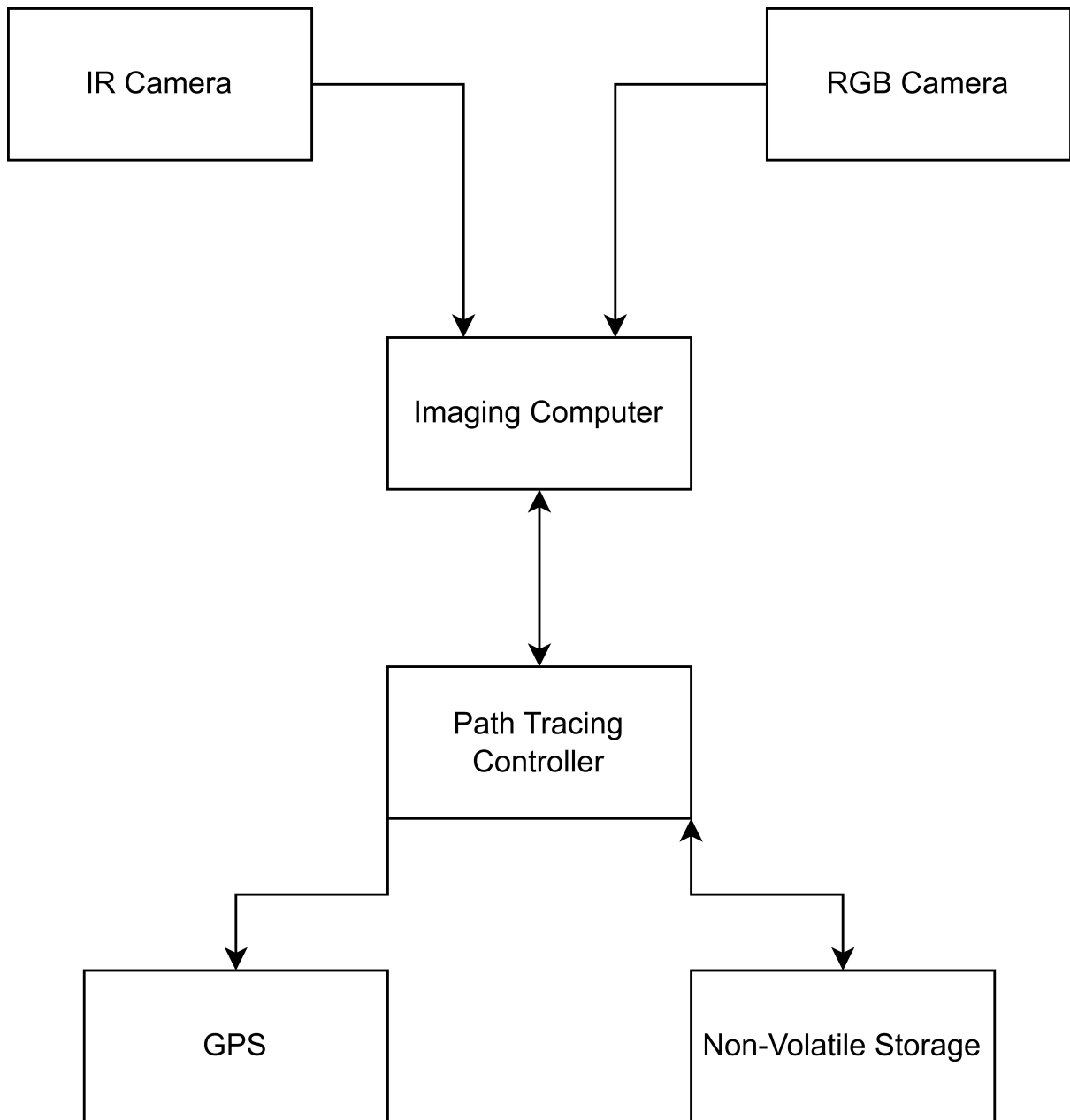


Figure 168: High Level Overview of Imaging Sub-System

The imaging computer uses two MIPI CSI-2 (Mobile Industry Processor Interface Camera Serial Interface Version 2) for each channel. It records straight to H.264 which is stored on the local PCIe Gen 3 NAND flash solid state drive. Once transferred the files are wiped from the imaging computer



Figure 169: Camera Modules

These are the cameras used for imaging as they are currently the best camera available withing a reasonable budget that meets all of our needs. Both of them feature a 12 megapixel Sony IMX708 sensor (the black NoIR variant has its IR cut filter removed at the time of manufacturing).

5.2.2.4. Power Delivery Subsystem

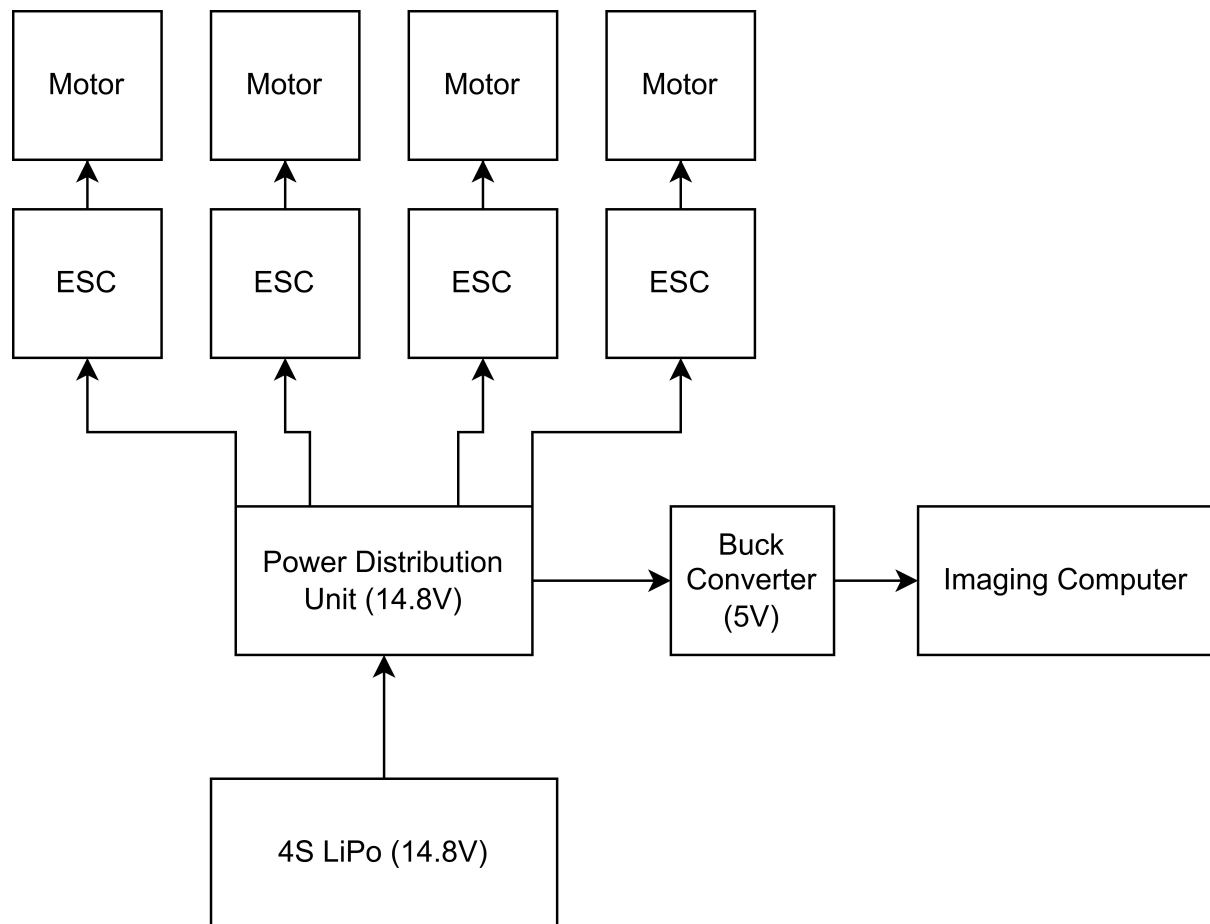


Figure 170: High Level Overview of Power Delivery System

The power delivery system has been quite simple as there is one voltage converter module to power the imaging computer. Inside the Linux Kernel we had to change a flag in order to make the imaging computer not throttle the SoC. The imaging computer expects a special handshake that the Renesas DA9091 PMIC (Power Management IC) processes and the OS then does not limit the power of the SoC. Once that flag has been set it then draws as much power it needs to give the SoC the 15 watts of power that it was designed to operate and reach the normal boost and base clock.

5.2.2.5. Communication

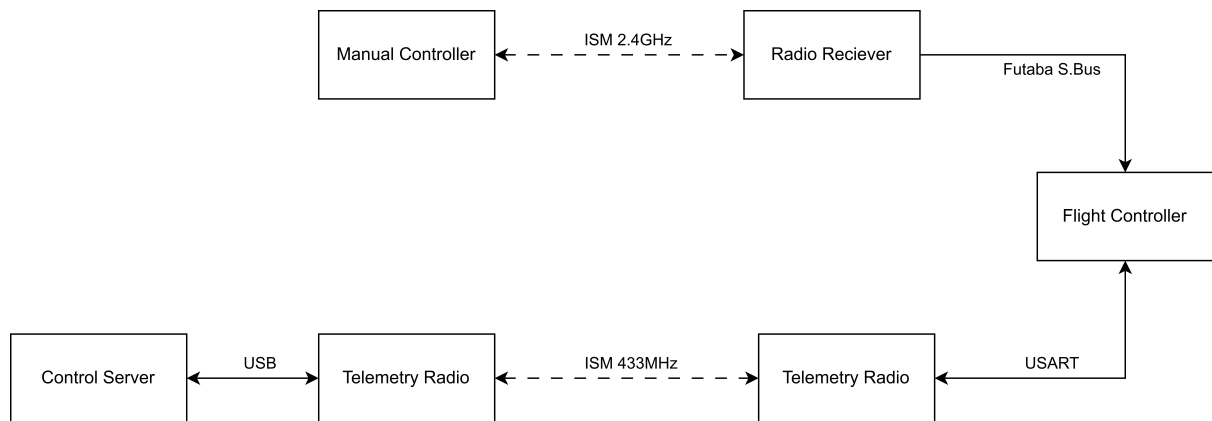


Figure 171: Communication Architecture

As mentioned before the command and control server talks to the radio over serial (via the USB Port). Then it gets transmitted in the ISM 433MHz band (this communication is bidirectional). Meanwhile the manual radio is connected in the ISM 2.44GHz band although it uses proprietary inverted USART logic designed by Futaba. This is called S.Bus.



Figure 172: Telemetry Radio

When both of the radio devices establish communication it displays a blinking led (which can be visible through the translucent shell of the radio itself). This signifies that a heartbeat package is established.

5.2.2.6. Flight Data Recorder

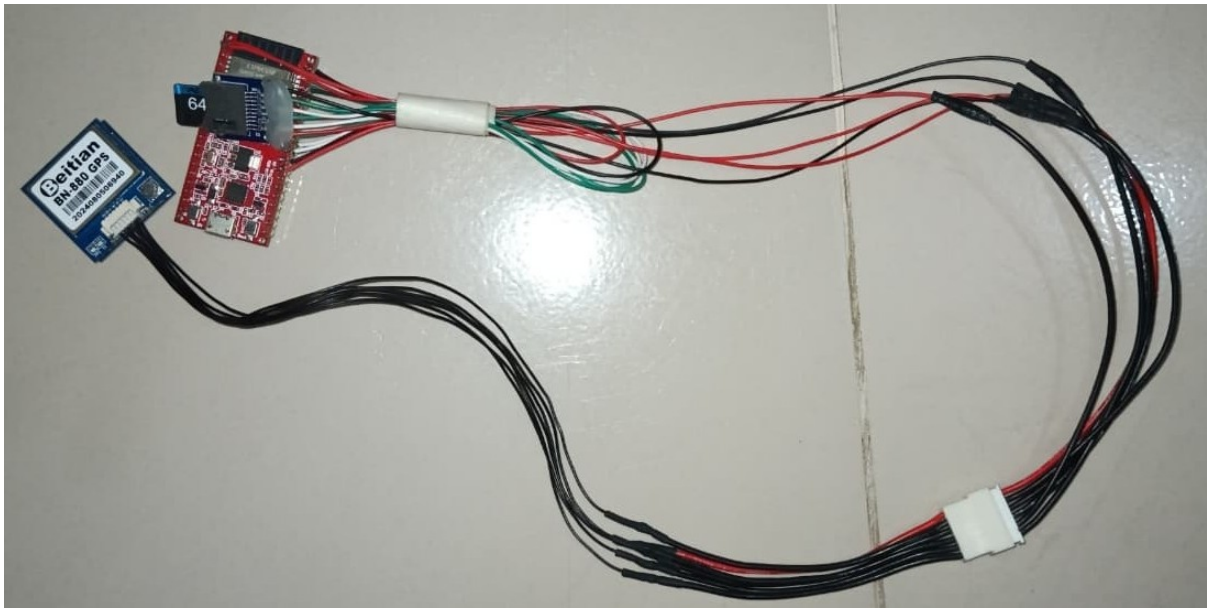


Figure 173: Flight Data Recorder

A generic ESP32 WROOM32 module that uses a Silicon Labs CP2102 USB to serial chip to communicate with the imaging computer. The backup USART is used to communicate with a BN880 GPS module (which has a Honeywell HMC5883L triple axis I²C compass on board). All of the real time location data and magnetic heading with a timestamp. This data is stored on a micro SD card that communicates over the SPI bus.

5.3. Evaluate the Solution to Meed Desired Need

5.3.1. Optical Correction

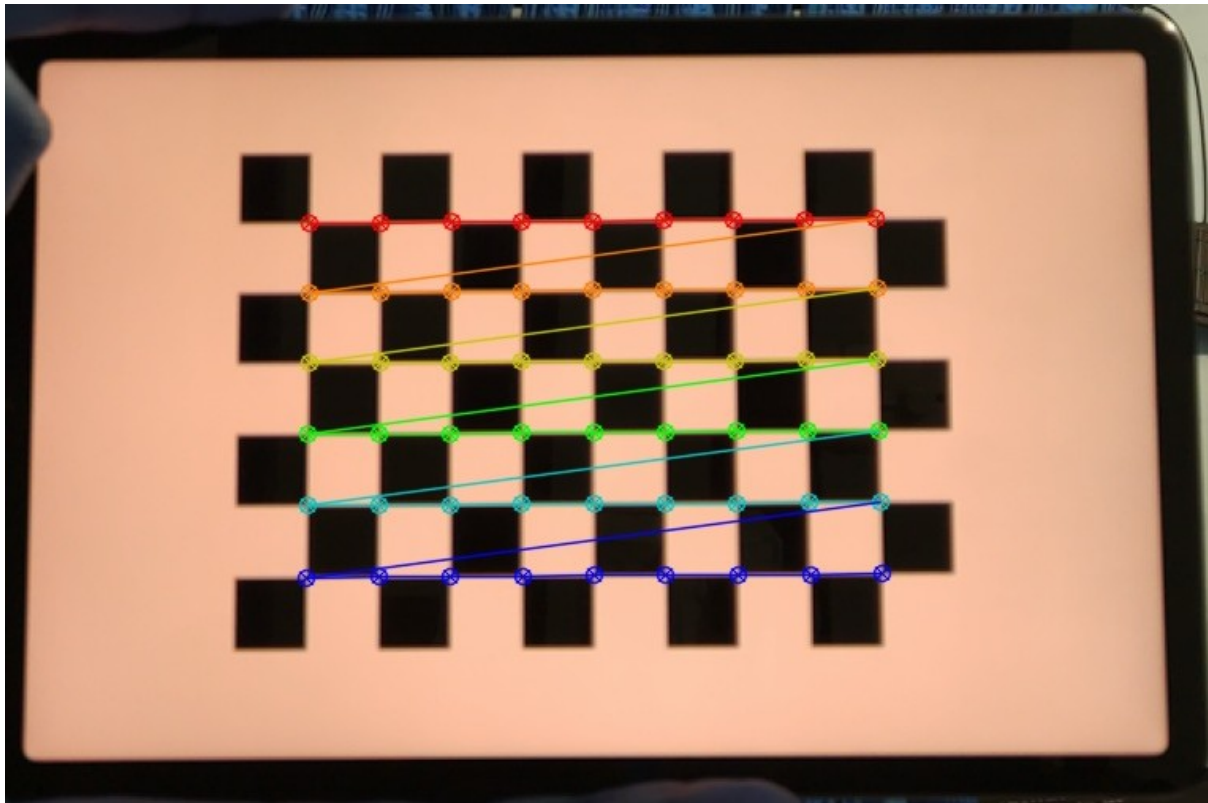


Figure 174: Optical Image Correction Using Checker Board Pattern

All lenses have some form of aberration (some more than others). It can either be chromatic or spherical (both related to the unique property of optical lenses, one is caused by the change of material density which results in the convergence of shorter wavelengths before the focusing plane thus a fringe of color is noticed on the edges of captured images and the other one is the distortion caused by the geometry). The correction is applied for barrel distortion. As the color shift and other factors are not quite as important as others but if the image is distorted on the edge of the sensor it will seriously impact the measurement accuracy. A checker board pattern of known dimensions is used to create a correction profile (a sort of pincushion distortion) to negate the effect barrel distortion. This is stored in a pickle format which is used by OpenCV

5.3.2. Prototype Flight Test



Figure 175: Initial Flight Test

This is the first prototype flight test used to check the stability of the whole platform. It was very stable but there was significant drift in the reported GPS position of both the systems. Aside from this all telemetry and onboard data was being received by the command and control server seamlessly.

5.3.3. Manual Flight



Figure 176: Manual Flight and Telemetry Test

The manual flight data was also being displayed on the command and control server without any issue.

5.3.4. Autonomous Flight



Figure 177: Autonomous Flight Test

Although there was significant drift on the recorded location tracking subsystem there was negligible error in the system. This implied the secondary GPS calibration was invalid and it needed calibration and a correction profile applied on the MCU so that it can null the error before writing to the scratch disk (the flash storage used for location tracking).

5.3.5. Flight Data Preview



Figure 178: GPS Drift

Our suspicions were confirmed when we saw that there was some serious drift on the system. As the image above shows how it has interpreted a simple lawnmower pattern into a garbled mess.

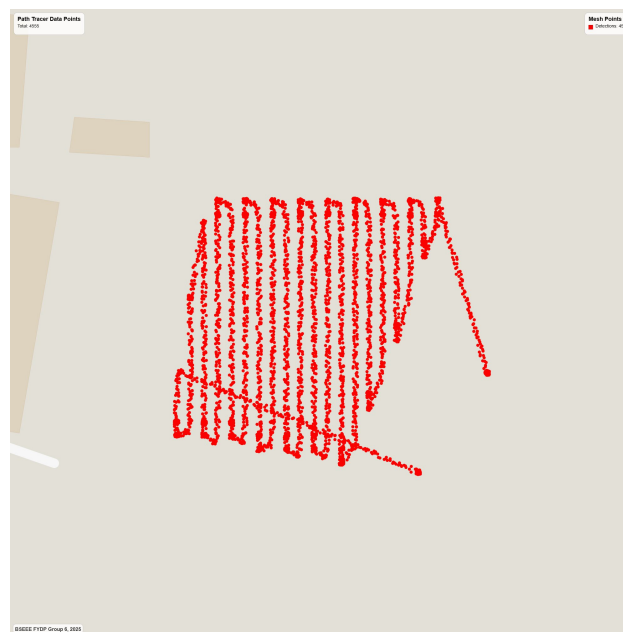


Figure 179: Post Calibration GPS Performance

After applying a calibration according to the GPS manufacturer instruction the error was reduced to an absolute error of 25 centimeters. Which was an acceptable amount of drift. A

few tests after power cycling and leaving it powered off for a few days did not affect the calibration profile. So, we can conclude that the calibration process was successful.

5.3.6. Detection Capability

5.3.6.1. Street Deflection

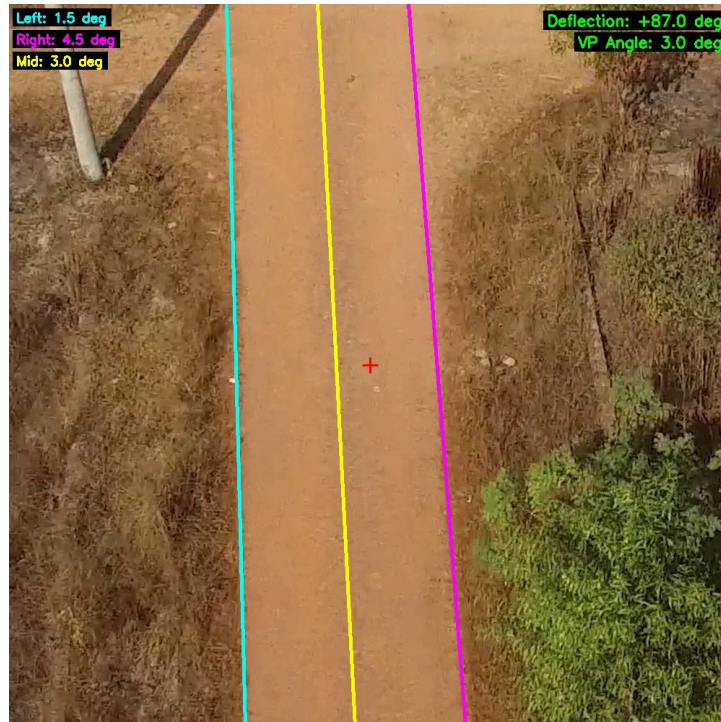


Figure 180: Road Tracking RGB

In order to check the deflection and edges of the street first the image was reduced to a 1-bit monochrome image. Which helped the CV model to identify the street easier.



Figure 181: Road Tracking 1 Bit

This 1-bit image was required as this was an unpaved road. Thus prebuilt OpenCV lane detection algorithms were not applicable for our test purposes. Due to current issues we were not able to access the required infrastructure to show the full detection capability of the device as we did not get permission to fly a prototype device in populated or up and running installation to perform a test on our full fledged model.

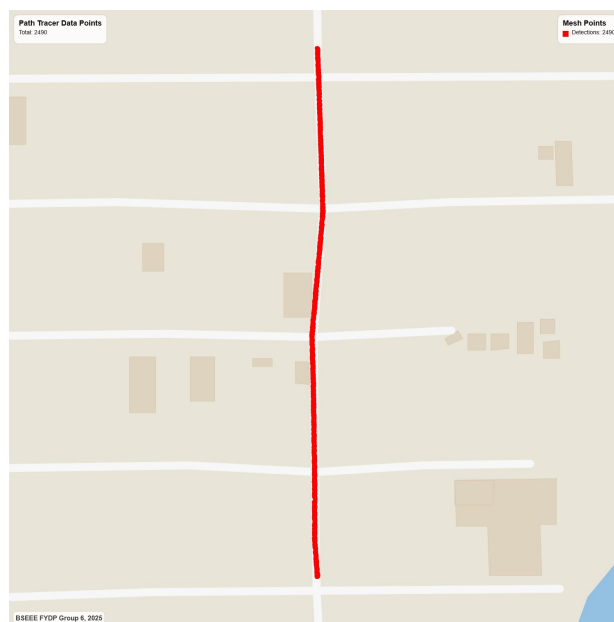


Figure 182: Path Tracing

5.3.6.2. Shallow Angle Scanning



Figure 183: Shallow Angle Detection

On the shallow angle sweep we can detect debris left on the road that could cause accidents if left as is.



Figure 184: Detection Performance

The frame rate and confidence scores were satisfactory. As it was more than the 30fps requirement that we have set as a baseline (as smooth motion is perceived the higher the frame rate is but there are diminishing returns at higher rates).

5.3.6.3. Manual Control Interrupt



Figure 185: Manual Override

In this frame we can see that there are people and vehicles entering the road. At that time the operator tool control away from the automatic routine which was a key safety feature demonstration as we mentioned that at any point during the automatic flight the drone operator can take over at a moments notice.

5.3.6.4. Precise Manual Control



Figure 186: Real Time Measurement Assist

ArUco marker based precise measurement with the live feed this way operators can monitor, track or precisely land on a surface. The end user is free to use this fiducial marker tracking capability in flexible and creative ways to further extend the usefulness of this device.

5.4. Conclusion

Once we completed the project construction, we realized that the finished construction was more or less similar to our design. There were limited cases of inconsistencies, which were predominantly caused by the constraints of reality (material availability, budgetary constraints, etc.). We also had a situation that was being affected by external factors such as the current socio political condition of the country. Due to that, the drone tests were limited to some areas, thus we could not perform a complete practical test.

Chapter 6: Impact Analysis and Project Sustainability [CO3, CO4]

6.1. Introduction

A unique aspect of both of our solutions has created an unprecedented precondition for impact assessment. As none of the designs drastically perform the physical characteristics of the end product. Thus there is no measurable change that can be linked with the chosen solution. Since intangible factors such as performance (which is the main differentiating factor for the design variants of this project) do not influence the visible outcome of any key metric.

6.2. Assess the Impact of Solution [CO3]

Deploying an Autonomous Aerial Survey Platform (AASP) will have great impacts across technical, environmental, economic, and social dimensions. This platform will change the way data is collected and equipment is watched. Additionally, it will help bring about a systemic change in how resource-limited areas, like Bangladesh, do surveys in the field.

- 1. Technological:** Field conditions data collecting, processing, and interpretation are fundamentally changed by the AASP. High-resolution spatial data collecting that surpasses conventional surveying techniques in accuracy, density, and temporal efficiency is made possible by UAVs fitted with sophisticated sensors, GPS tracking systems, and onboard data processors. For example, drone-based SfM (Structure from Motion) methods have demonstrated to provide up to nine times the point cloud density when compared to complete station surveys while greatly lowering operational time [30]. These developments in technology not only enhance survey quality but also increase the capacity to regions once unreachable or too hazardous for manual surveyors. Moreover, the AASP may independently identify, classify, and react to features in real time by combining artificial intelligence-based image recognition with computer vision (CV), therefore enabling applications such as real-time security surveillance, critical infrastructure inspection, and wildlife monitoring. Real-time data collecting combined with post-processing capabilities also implies that actionable insights may be produced fast and effectively, hence improving decision-making processes in many different industries.
- 2. Economic:** Economically speaking, the AASP can significantly lower the running expenses of survey projects. All of which mean more project expenses, traditional techniques sometimes call for big teams, significant field equipment, and time-consuming data processing. By comparison, UAV-based systems need little staff, are faster to install, and produce data that is usually ready for analysis almost after collection. FAO claims that more precise irrigation and pest control techniques in drone-based precision agriculture have caused a 20 to 25% rise in agricultural production [33]. The system's price and efficiency also appeal to public sector entities and small- and medium-scale businesses (SMEs). Its multi-role features maximize return on investment and reduce the barrier for entry into data-driven decision-making by allowing a single platform to carry out activities in agricultural, urban planning,

infrastructure monitoring, and disaster response.

3. **Environmental:** The traditional methods of surveying frequently involve doing many vehicle-based or piloted aerial surveys, which contribute to the production of large levels of carbon emissions. These are replaced with lightweight electric unmanned aerial vehicles (UAVs) by AASP, which results in a significant reduction in the carbon footprint. Solar-powered drones, when combined with energy-efficient electronics, have the potential to reduce operational carbon emissions by as much as forty percent, according to studies [35]. Measuring biodiversity using drones is also beneficial to conservation efforts since it allows for the detection of wildlife populations from a safe distance, hence reducing the amount of ecological disturbance [36].
4. **Societal:** Using AASP has far-reaching social consequences. UAVs can close these gaps in a nation like Bangladesh, where topographical obstacles and infrastructure shortcomings impede service delivery in distant areas. Drones, for instance, can bring digital learning tools or educational resources to flood-affected or remote places, hence guaranteeing educational continuity during natural catastrophes [37]. Drones can be utilized in healthcare for the fast delivery of life-saving supplies like vaccines, blood, and Automated External Defibrillators (AEDs), particularly in roadless or emergency situations [38]. This not only increases survival rates but also enhances emergency response systems. The AASP also offers hope for skills development and employment generation. Opportunities for vocational and technical training will also rise as the need for maintenance technicians, data analysts, and drone pilots develops. The correct policy incentives can help to lower young unemployment and support a technically knowledgeable workforce.

6.3. Evaluate the Sustainability [CO4]

6.3.1. Environmental Factors

1. **Carbon Footprint:** The total net carbon emission factor is 530-570t CO₂/GWh in recent years which is a significant challenge for Bangladesh, which is much higher than developed countries [39]. The carbon footprint in autonomous drones is considerable in the context. The reduction in carbon footprint can be reduced significantly by the usage of autonomous drones. By the usage of geoscience investigation and by usage of aerial surveying the development is approx 60% better in data of productivity and efficiency. Which helps us to reduce the carbon footprint for conventional operation [40]. This effort can be used in agriculture operations and in disaster response. On the other hand, manufacturing and operating autonomous drones will still create carbon emissions for usage of a fossil fuel dependent electricity grid. From 2014-2015 the total amount of greenhouse gas was approx 23 millions tons for the production of 43 TWh electricity. For this reason the charging station for autonomous drones will produce high carbon intensity. A better solution is solar-powered autonomous drones. These can operate with less carbon emission compared with conventional battery-powered

sources [41]. In Bangladesh there is a geographical advantage where the environment gets a good amount of sunlight in much of the year.

2. **Energy Efficiency:** There is a critical condition in energy efficiency in Bangladesh. Which makes it a big challenge to face power generation. These drones have a good configuration, having advanced energy management systems. Having good research on solar power drones can extend flight time with good efficiency [41]. Having minimal energy input, these drones can be operated with agricultural monitoring, border surveillance, and disaster response. Theoretically, multirotor robots are more power-consuming than bioinspired revolving winged drones, which use less power, and they are more advanced [42]. These drones are more sustainable considering the energy resources in Bangladesh. It is less dependent on current energy resources. The combination of energy-efficient drones and renewable energy is very much crucial for designing. Bangladesh's diverse geographical conditions can be helped by the development of energy consumption minimization algorithms, especially the UAV assist network which has been discussed in recent years [43].
3. **Material Sourcing:** One of the biggest challenges for Bangladesh is the sustainability of the materials used in autonomous drone manufacturing. Most of the equipment for example composites, lightweight metal, electric metals, and specialized electronic components, needs to be imported. An extensive analysis of materials used in drone construction is divided based on the three different applications, such as frames, propellers, wings and structural support [44]. For Bangladesh the dependency on import would increase the increment of carbon and would impact the environment. One of the sustainable alternatives is bio based composites and recycled carbon fiber [44] which is more environmentally friendly but the local production is much more limited in the country. The shipbreaking industries in Chittagong district can provide a certain amount of drone components, but quality control and material selection challenges should be addressed properly [45]. Bangladesh can make policies that can be used to encourage the use of local material, which would establish a sustainable supply chain. Similarly, we have seen this in other technological sectors.
4. **E-Waste Management:** One of the biggest challenges of deploying autonomous drones in Bangladesh is E-waste management because it creates significant environmental challenges. The country has improper municipal solid waste management, which creates environmental pollution [46]. The drones carry advanced electronic components, for example, circuit boards, sensors, and communication systems, which would affect waste streams. Current waste management in Bangladesh is not well structured to take advantage of electronic components. Although Dhaka has door-to-door community-based waste collection, half of the wastage is not collected [46]. The responsibility of waste management operators, generators, and transporters has a lack of technical guidance [45]. Special care should be taken to manage autonomous drone waste to avoid toxic substances from entering society. Household segregation would be needed to handle drone components. They would face a

significant challenge in places like Khulna [47]. Policy gaps, limited budgets, proper facilities, and social constraints create a difficulty in implementing proper waste management [47]. Implementing an ERP system can create an extended problem-solving system; having end-of-life solar photovoltaic panels [48] can create a drone-related e-waste management system.

5. **Toxic Substance Use (RoHS):** These drones create toxic substances that damage the environment. It is more dangerous for Bangladesh, which has limited regulation. Batteries use cobalt, lithium, and heavy metals, which damage the health of the environment. The circuit board contains mercury and cadmium, and also the frame has hazardous elements in it. Due to fewer rules and regulations and a judicial system, the concern about health always remains questionable [49]. The local regulatory system should be making it strong and effectively monitoring and controlling the imports and should be looking after the e-waste management.
6. **Renewable Energy Use:** There is a good opportunity for Bangladesh to integrate renewable energy and autonomous drones. A solar-powered drone can create long-flight drones, supplying power to the system and not depending on the power grid system [41]. Bangladesh has already used sunlight as solar power which helps to light up houses in rural areas. These solar-powered drones would help to fulfill the sustainability goals of Bangladesh. Fuel-based propulsion has been used in recent times, which has opened the gates of renewable energy that reduce the impact on the environment [50]. That is the best option for Bangladesh to achieve her goals. Deployment of renewable energy-powered autonomous drones can help Bangladesh to achieve its goals. This adaptation would help Bangladesh to save 13.55 GWh/day which would save 0.6 million dollars/day. It will help to fulfill the sustainable development plans.
7. **Biodiversity:** In Bangladesh the impact of autonomous drones on biodiversity is very complex. The usage of drones can be used for environmental monitoring. But the presence can disrupt the natural habitat and create stress in wildlife, which can create behavioral changes that can create a bad impact in reproduction and survival rates [53]. Biodiversity loss is done by human activities [53]. The usage of drones for biodiversity impacts is limited for Bangladesh. More research needs to be done. Protective measures should be taken so that it doesn't create any bad impact.

6.3.2. Economic Factors

1. **Initial Capital Investment:** The initial cost creates a big challenge for the adoption of autonomous drones. As Bangladesh is a developing country with limited technology infrastructure, the initial investment creates a barrier. A low-cost drone is a better option for Bangladesh's economic conditions [54]. Affordable drones are a good choice for this country. Agriculture is one of the economic pillars of the economy of Bangladesh. Affordable drones can create a good impact for small farmers [55]. Government

subsidies are necessary to fulfill this dream. Educational institutions need to give necessary training to run these drones. Moreover, reducing import dependence is a must thing to do.

2. **Domestic Manufacturing Potential:** A growing manufacturing sector creates opportunities for local production of drones which will reduce potential costs. The growing expertise in these fields will gradually decrease the cost of production [57].
3. **Operational Cost:** Technical expertise, availability of energy infrastructure, and the regulatory environment will cut the production of drones. Low-cost labor creates good economic conditions for the production of drones more easily than developed nations [58]. Bangladesh's frequent load shedding can create a big problem of charging the batteries of the drones. The supply of electricity should be maintained for continuous flight operation [59]. The usage of solar energy should be encouraged in autonomous drones.
4. **Data Management:** Another operation expense is data cost. Bangladesh Data management is developing but limited, and we need more access to data for drone [55]. Edge computing technology is needed for reducing operational costs.
5. **Maintenance Cost:** The country's climate, availability of spare parts and technical expertise will create unique challenges for Bangladesh in order to maintain the autonomous drones. High flood situations and dusty environments in dry seasons will create damage to drone components. Bangladesh has to modernize her supply chain management system for drone components and increase investment in inventory management to extend downtimes for repairs. These developments will target modular drone design that allows for easier components and filling up the demand by using sources produced locally [61].
6. **Domestic Technical Capability:** For sustainable implementation, developing local technical capacity for drones is very essential. In engineering institutions, Bangladesh can implement drone technology and later on can create a working environment for further development, repair, and maintenance [62][58]. These will change cost centers into potential sources of technical jobs and expertise.
7. **Lifespan:** The lifespan of drones in Bangladesh will be less because of the environment in Bangladesh. Moreover, the rainy season in Bangladesh will stay for most of the year. On the other hand, equipment would be damaged by moisture in nature [55]. Adding high temperature and humidity levels could accelerate electronic equipment degradation. Research shows that operating drones in Bangladesh is challenging [63][54].
8. **Adaptation Strategies:** To maximize the lifespan of drones, the drones should incorporate enhanced drone protection and corrosion resistance with climate control-

sensitive electronics. Forestry applications of drones should be maintained routinely because of frequent climate change in temperature [63].

- 9. Productivity:** Autonomous drones have significant potential in multiple sectors in Bangladesh. In agriculture, 40% of employment is done. Drones can be used to revolutionize crop monitoring, precision agriculture, and pest management [55]. COVID-19 disruption in agriculture has shown the necessity of drone technology in this sector [55]. The growing industrial sector in Bangladesh can see a completely different scenario by using aerial drone technologies. The challenging environment, such as numerous rivers and flooded areas where traditional transportation doesn't work, makes monitoring difficult, which can be solved with aerial drones [62][57].
- 10. Disaster Management Application:** Bangladesh is a disaster prone country (not to a severe degree like other regions but this is being exacerbated by climate change), which makes using drones more useful to operate in such situations. Moreover, autonomous drones can be used in such a situation where local supplies cannot be transported [54]. This technology will significantly enhance the saving of other lives.
- 11. Market Potential:** Bangladesh has the potential to build autonomous drone technology with private and public partnerships. With a population of 180 million and rapidly growing technological adaptation, it is a promising emerging market for drone services [56]. There are 87,000 square kilometers of agricultural land that could benefit from drone-based monitoring systems and management [55]. Stakeholders in Bangladesh should ensure the safety of the people and give funding to innovation, particularly in the defense sector and our face management infrastructure.
- 12. Regulatory Development:** The regulatory body of Bangladesh needs to have a proper framework in order to develop the drone market. The regulatory body can be similar to the European EASA framework, which has a clear indication of drone-related stuff and accelerates the market growth [64].
- 13. Job Creation:** There is a complex effect on employment in Bangladesh for autonomous drone technology. While having many displacements in some sectors, the overall impact could be positive with proper policies and training programs. New jobs will emerge in drone operation maintenance data analysis and software development Bangladesh has a huge young Tech survey Workers' home can be well positioned for these roles with target training programs [58][59]. The country will frankenize its IT sector for the growing software development and drone-related issues.
- 14. Skill Requirements:** For Bangladesh to maximize Job creation of potential autonomous drones requires a significant amount of investment in technical education. Universities and technical colleges need to cooperate with government and non-government institutions to develop drones [62]. Education investment essentially

ensures job creation and creates a proper workforce.

15. Domestic Economy: For Bangladesh to maximize Job creation of potential autonomous drones requires a significant amount of investment in technical education. Universities and technical colleges need to cooperate with government and non-government institutions to develop drones [62]. Education investment essentially ensures job creation and creates a proper workforce.

16. Technological Transfer: The impact of autonomous drones adaptation in Bangladesh would affect positively in different sectors' supply chain operations and potentially create business opportunities for jobs and specialized services [57][61]. Agricultural economics would benefit significantly, especially with improved agriculture and better market connections. Financial research, industry profitability and GDP growth of Bangladesh will increase. Similar technological growth will increase the overall economy [56].

6.3.3. Societal Factors

- 1. Health and Safety:** Have a significant potential to change the health care delivery system and emergency response in the country. The supplies can be delivered using deployed drones in South addresses where medical facilities can be taken easily. especially in flood-prone areas. In the COVID-19 pandemic, logistic systems have been seen to fail due to the limitation of manpower, and for this reason, medical supplies were delivered late [65][66]. In Bangladesh, national disasters, including floods and cyclones, and autonomous drones can play a significant role, for example, supplying medical kits that will eventually save lives.
- 2. Community Engagement:** The introduction of autonomous drones could change the system, and robust community engagement could ensure maximum benefit. Community involvement by implementing drones could gather actual success, particularly in rural areas where technology might face disturbance. Experts worldwide agree that drone operations, to remain socially acceptable, must classify crucial criteria, including safety, security, environmental sustainability, and legal requirements [67]. In Bangladesh it is necessary to increase engagement with the community to address privacy concerns, noise issues, and cultural sensitivities related to drone operation. To build community acceptance, demonstration of projects in public showcasing for successful drone applications can play a vital role. Showing drones can deliver medical supplies to remote healthcare facilities, which can be implemented in developing nations, could do tangible benefits [68]. In the Bangladeshi context, it is necessary to increase educational programs and awareness campaigns to familiarize the community with drones and to overcome the potential resistance and foster a sense of ownership over this technological innovation. Social equity challenges in Bangladesh for autonomous drones have a big potential, which should be improved by taking essential

services and delivering them to the unserved population. In Bangladesh, numerous rivers and remote areas create disturbance in service availability. Medical drones can significantly improve the Healthcare system by ensuring the essential medical supplies reach art areas.

3. **Education:** Unique opportunities can be created using autonomous drones for educational purposes in Bangladesh. During emergencies, drones can play a vital role, especially in remote areas. Disaster affected schools face a big problem in disturbed education during natural disasters, as shown in documented studies [71]. The normal approach focuses on identifying alternative learning places. Autonomous drones can play a vital role by delivering educational material to isolated communities during emergencies; ensuring a proper educational environment for the isolated communities will make a difference. Drones with imaging capabilities could support educational research and fieldwork in various disciplines. For example, archeology to environmental science. Moreover, Malaysia is using photogrammetry mapping for archeological research [72] which can help and benefit educational institutions in Bangladesh by providing access to study remote sites and enhancing research capabilities. Bangladeshi students can emerge as a factor of pride in the technological field by the integration of drone technology in the educational field. By learning drone operations, maintenance and application increases the opportunity to develop valuable stem skills that can help to achieve the country's digital transformation goals [69]. By focusing on drone based educational systems will help to create a new workforce and create opportunity. If we need to integrate drone technology into education we need to build new infrastructures, proper teacher training and focus on students so that all can be benefited by it.
4. **Cultural Effects:** The autonomous drones in Bangladesh will create a multidirectional impact on the cultural landscape in Bangladesh. Which will eventually affect the traditional practices to society norms and societal interactions. In Bangladesh traditional lifestyle is still prioritized. Sudden drone appliance could create cultural disturbance. The people in the community could create suspicion and potentially create resistance. Research on drone operation will create social accessibility in acceptability which will be difficult to be successful [67]. Privacy concerns are very important from the perspective of society. Especially in the case of women, the camera in drones will create resistance. Clear guidelines need to be created in order to solve this problem. These drones will also help to preserve and protect this country's cultural heritage and these drones can be used for archaeological mapping for other countries [72]. The potential for drones to support cultural events. It can be supported by applying aerial photography support.
5. **Displacement:** These drones raise important considerations regarding potential displacement effects which can be both economically and physically. Economic displacement, mainly job displacement, will be a significant concern. Fear of losing a job will be a big obstacle for executing the 4th industrial revolution [69]. Traditional

jobs will vanish as the drones take over those places. In Bangladesh, where unemployment is a big factor, it's getting challenging. Drone technology would create new employment opportunities for drone manufacturing, maintenance, operation, and support service. The main challenge is to prepare the Bangladeshi workforce and prepare for a new role that the whole world is going to face in the future, giving them proper education and focusing on technological skills [69]. Establishing this drone-related infrastructure will eventually create jobs for humans. This process needs a lot of land, but eventually, it will create the opening door of opportunities for this community. We have to be careful while planning to minimize the displacement impact. The noise in the drone is one of the main factors for sensory displacement, which can disturb the traditional soundscape.

6.4. Conclusion

As engineers, it is our duty to know the impacts of our work concerning the society, the environment and the economy. These impacts were analyzed during this project to make sure that our solution is beneficial and causes minimal risks. We have considered things like privacy issues, the noise of the drones affecting the environment, and the cost-effectiveness of aerial surveying. According to our judgment, we feel our platform is responsible in its engineering behavior and has possible advantages to the modern day surveys.

Chapter 7: Engineering Project Management [CO11, CO14]

7.1. Introduction

The success of engineering projects rely strongly on project management. It allows for effective coordination between members working towards the desired outcome, done with adherence to schedules. Project management and logistics ensure that together the project is executed in accordance to the required results.

7.2. Define, Plan and Manage Engineering Project

Our project took place over the course of three consecutive semesters. The following are the key periods that our project took place over.

7.2.1. EEE499P The Planning Phase

TABLE VII Project Plan EEE499P

ID	Task Title	Start Date	End Date	Duration
1	Topic Selection & Research	16/02/25	18/02/25	3 days
2	Find Complex Engineering Problem	20/02/25	20/02/25	1 day
3	Related Papers and Journal	20/02/25	27/02/25	8 days
4	Selected Final Topic	28/02/25	28/02/25	1 day
5	Project concept note	01/03/25	03/03/25	4 days
6	Tentative Problem Statement and Objective	06/03/25	7/3/25	2 days
7	Multiple Design Approach	08/03/25	10/03/25	3 days
8	Specification Requirement and Constraint	11/03/25	16/03/25	7 days
9	Applicable Standards and Codes	20/03/25	27/03/25	8 days
10	Project Proposal Note	29/03/25	05/04/25	8 days
11	Background Research and Project Plan	06/04/25	12/04/25	7 days
12	Methodology, Budget, Impact, Expected Outcome	13/04/25	22/04/25	10 days
13	Sustainability, Ethical Consideration	24/04/25	27/04/25	4 days
14	Risk Factor Management, Safety Consideration	28/04/25	03/05/25	6 days
15	Final Project Proposal Note	04/05/25	05/05/25	2 days
16	Preparation of slide for Final Presentation	05/05/25	05/05/25	1 days
17	Mock Presentation	06/05/25	06/05/25	1 days
18	Final Presentation	08/05/25	08/05/25	1 days
Total				77 days

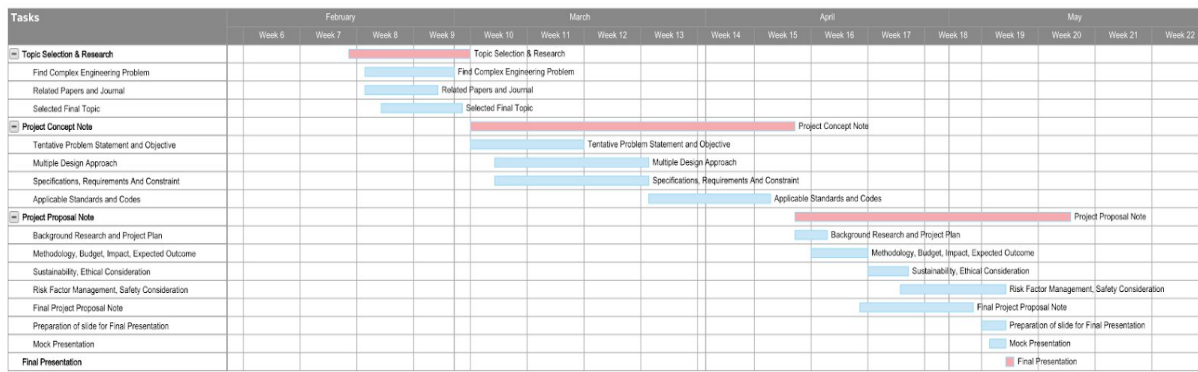


Figure 187: EEE499P Gantt Chart

7.2.2. EEE499D The Design Phase

TABLE VIII Project Plan EEE499D

ID	Task Title	Start Date	End Date	Duration
1	Data Set Selection	28/06/25	30/06/25	3 days
2	Data Set Annotation	30/06/25	12/07/25	13 days
3	Task Distribution	07/07/25	07/07/25	1 day
4	Drone Simulation	08/07/25	13/07/25	6 days
5	Flight Pattern Analysis	13/07/25	21/07/25	9 days
6	Software Simulation	22/07/25	05/08/25	15 days
7	Machine Learning Simulation	3/8/25	27/8/25	25 days
8	Prepare Progress Presentation Slide	23/07/25	23/07/25	1 day
9	Progress Presentation	24/07/25	24/07/25	1 day
10	Project Report	20/08/25	11/09/25	23 days
11	Final Presentation Slides Preperation	25/8/25	25/8/26	1 day
12	Final Presentation	28/8/25	28/8/26	1 day
13	Draft Report Submission	10/9/25	10/9/25	1 day
14	Project Report Submission	11/9/25	11/9/25	1 day
Total				95 days

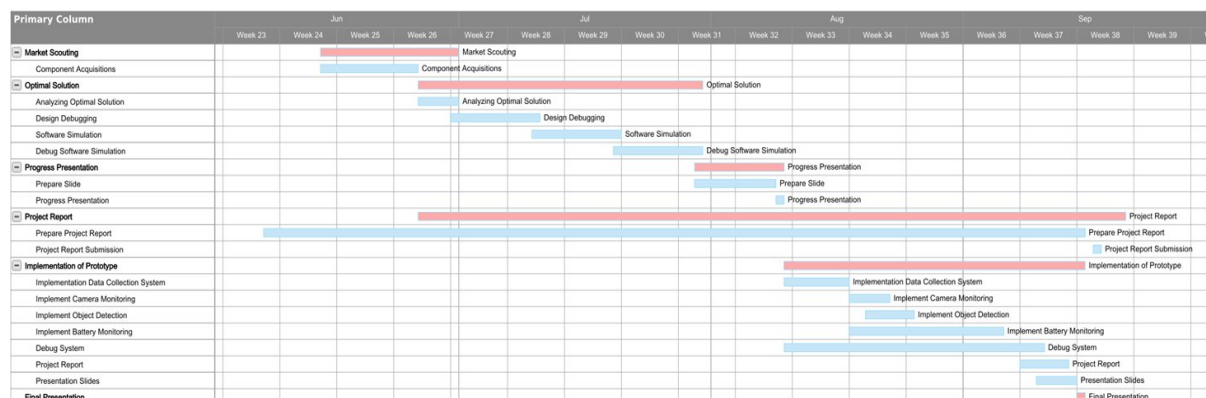


Figure 188: EEE499D Gantt Chart

7.2.3. EEE499C The Completion Phase

TABLE IX Project Plan EEE499C

ID	Task Title	Start Date	End Date	Duration
1	Market Scouting	04/10/25	06/10/25	3 days
2	Component Collection	7/10/25	19/10/25	13 days
3	Building Airframe	20/10/25	23/10/25	4 day
4	Manual Flight	1/11/25	1/11/25	1 day
5	Automatic Flight	6/12/25	6/12/25	1 day
6	Progress Presentation Slide	8/12/25	9/12/25	2 days
7	Progress Presentation	10/12/25	10/12/25	1 day
8	Sensor Denoising, Camera Calibration and Documentation	11/12/25	30/12/25	20 days
9	Project Debugging	31/12/25	3/1/26	4 days
10	Poster Creation	2/1/26	2/1/26	1 day
11	Video Creation	2/1/26	3/1/26	2 days
12	Project Showcase	5/1/26	5/1/26	1 day
13	Draft Project Report Submission	9/1/26	9/1/26	1 day
Total				54 Days

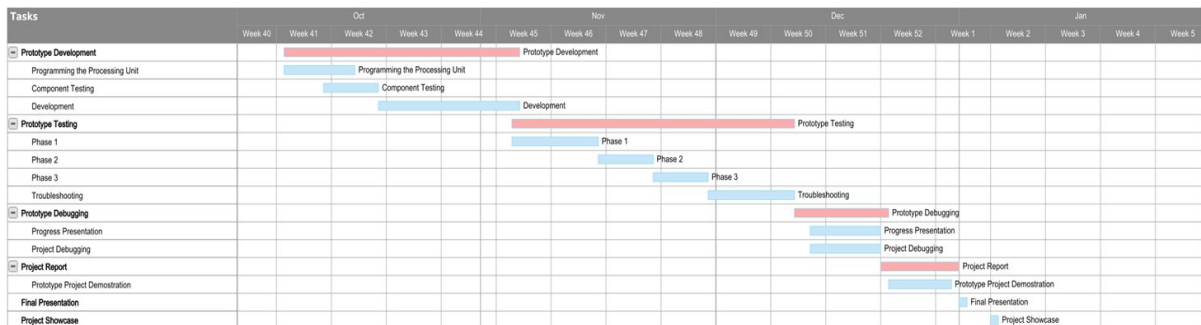


Figure 189: EEE499C Gantt Chart

7.2.4. Risk Management Plan

TABLE X Risk Response Matrix

Event	Likelihood	Impact	Risk Level	Response Strategy
Hardware Failure (Drone crash, sensor malfunction)	Medium	High	High	- Implementation of redundancy (backup sensors, dual GPS) - Regular pre-flight checks and calibration - Maintain spare parts and repair tools on-site
Data Loss or Corruption	Medium	High	High	- Use real-time edge backups + cloud sync - Adding error-checking features (checksums, redundancy in packets) - Test storage modules under field conditions
Inaccurate Fault Detection (False positives/negatives by AI model)	High	Medium	High	- Improve dataset diversity and size - Use confidence scoring in detection - Incorporation of a human validation process for critical areas

Regulatory or Airspace Restrictions	Low	Medium	Medium	▪ Obtain necessary UAV flight permits early ▪ Pre-define geo-fenced zones ▪ Use simulation or smaller drones where restricted
Power Supply/Charging Issues at Remote Stations	Low	High	High	▪ Use hybrid power options (solar + battery + grid fallback) ▪ Monitor battery health with IoT telemetry ▪ Include manual override and on-site fallback charging

7.2.5. Additional Fault Mitigation Scheme:

1. Mission Failure: Prevention of electrical hazards and mitigation of damage if the drone is unable to carry out its mission as planned.

Battery Disconnection and Isolation: The battery has to be automatically disconnected, thus isolating the power supply, in case the mission is aborted due to system errors, mechanical faults, or external interference, for risks of short circuits might develop, while overheating and even fires are other potential hazards.

Implementation: A fail-safe circuit may be installed to detect when mission-failure conditions take place so that a relay or an electronic switch will cut off battery power in a safe manner. The system shall also log the failure for subsequent after-mission analysis.

2. Complete Power Loss: Alerting persons in the vicinity and prematurely eradicating or minimizing the dangers of serious injury or damage to properties in any unforeseen loss of power.

Audible Warning Signal: During the complete loss of power, the drone should raise a loud and distinct audible alarm (for instance, buzzer or siren) prior to and during its descent/crash. Such a warning will enable bystanders to acknowledge the emergency and get away from the scene.

Implementation: A specialized low-power backup circuit or a capacitor-driven energy reserve shall provide power for the audible alarm, albeit for a limited time, during sudden failure of the main battery.

3. Navigation Failure: To ensure that when the drone faces navigation errors, it at least can handle these errors safely and accountability.

Safe Parking Procedure: Upon navigation system failure (e.g., loss of GPS, sensor malfunction), the drone either attempts to land or return to base (if possible).

7.3. Evaluate Project Progress

The progress and overall condition of the project was closely monitored and regularly evaluated. There have periodically been minor setbacks primarily due to the short schedule at hand. Such conflicts and issues were dealt with using extreme caution and awareness. The task at hand has always been prioritised appropriately and worked on in a unified methodical approach. Despite these issues, significant strides have been made towards reaching the desired outcome allowing the project to reach a positive condition where it can be presented for use or further research.

In the first phase we spent a great deal of time planning and researching for all viable modes of operation that would minimize our time required in the design phase. We were made aware from senior faculty members that we should properly plan and utilize this time as the following semesters will have a shorter time limit to work around.

During the FYDP phase D we encountered some technical issues but they were quickly resolved as we have spent a lot of time in the first semester planning for it. At that time one of our kind teammates pointed out that he had read about this certain issue being very common. So, out of sheer curiosity we decided to do some research and test out the software beforehand. This was our saving grace. Aside from this small hurdle this phase went without any major events.

As we progressed to the final phase of our project and neared completion we had to wait for quite some time for the parts to arrive to Bangladesh. Although the most expensive component i.e the imaging computer and the whole flight system (Flight Controller, ESC, telemetry and radio subsystem) were procured before as one of our group mates went to China on a business trip. This drastically reduced the parts cost. Once we had brought everything on our hands we encountered a problem that took us the longest time to solve and thus was single handedly the most difficult issue to solve. The camera had some serious geometric issues which needed correction (which is quite common for small form factor cameras to have as their lenses are very small thus have high levels of distortion near the edges of the frames) and most importantly we had GPS issues. Our GPS module had some serious drift issues. First we assumed it was a faulty module. Only after spending an embarrassing amount of time we realized that it was a calibration fault. Once that was resolved it was very straight forward to move on with our project and take it to completion.

7.4. Conclusion

The emphasis and importance of timely progression cannot be stated enough. This is the most crucial part of any projects success. Due to our highly skilled teams efforts and proper measures we managed to finish the project in record time. A tight feedback loop and generous assistance from our ATC we were able to resolve challenges as they arose.

Chapter 8: Economical Analysis [CO12]

8.1. Introduction

Economic analysis is an important part of project evaluation that examines the financial feasibility, sustainability, and long-term variability of a proposed system or product. It shows a structured assessment of cost, revenue, benefit and risk associated with the project over its operational life plan. It is used in government and corporate decision making and economic analysis. Which shows that resources are relocated efficiently and that investments deliver measurable results. The analysis considers key financial indicators such as total production cost, revenue generation, and return on investment. Moreover, cost benefit ratio and net present value (NPV) and break even point. The impact of inflation and discount rate is also shown to reflect real world economic conditions and the time value of money. This shows collective support informed decision making, project acceptance pricing strategy and long term planning. During the research process of any project it is necessary to identify demand and possibilities of realistic implementation. It is imperative that the financial aspect of feasibility is identified as soon as possible. The backing for such projects may not always be present but the possible demand for such technology in the future must be considered as well.

8.2. Economic analysis

Budget breakdown for Design Approach 1

TABLE XI Design Approach 1 Budget

Component Name	Quantity	Unit price	Price (BDT)
Raspberry Pi 3B 4GB	1	12,500	12,500
Raspberry Pi Camera V2	2	4,600	9,200
120GB USB SSD	1	2,350	2,350
64GB Micro SD Card	1	500	500
2200kV BLDC Motor	4	1,150	4,600
30A ESC	4	650	2,600
Quad Copter Frame	1	4,500	4,500
Betaflight F405 v3	1	3,800	3,800
4S LiPo Battery 3500mAh	1	3,500	3,500
Drone Remote	1	4,200	4,200
ESP32	1	750	750
Total			48,500

Budget breakdown for Design Approach 2

TABLE XII Design Approach 2 Budget

Component Name	Quantity	Unit price	Price (BDT)
Raspberry Pi 3B 4GB	1	12,500	12,500
Raspberry Pi Camera V2	2	4,600	9,200
128GB Micro SD Card	1	1,000	1,000
2200kV BLDC Motor	4	1,150	4,600
30A ESC	4	650	2,600

Quad Copter Frame	1	4,500	4,500
Betaflight F405 v3	1	3,800	3,800
4S LiPo Battery 3500mAh	1	3,500	3,500
Drone Remote	1	4,200	4,200
ESP32	1	750	750
Total			46,650

8.3. Cost Benefit Analysis

When regarding our project's development the cost benefit analysis is crucial for various reasons. This evaluation considers financial and other outcomes regarding the deployment of this device to various reasons.

When comparing existing solutions in the market we have found the closest solution to be a proprietary product from DJI (a technology company based in Nanshan, Shenzhen China). The operating mode and features are almost one to one, which costs BDT 73,800 (from local vendors on existing older stock). However newer stock with updated firmware can go upwards to BDT 100,000.

TABLE XIII Cost of Existing Market Solution

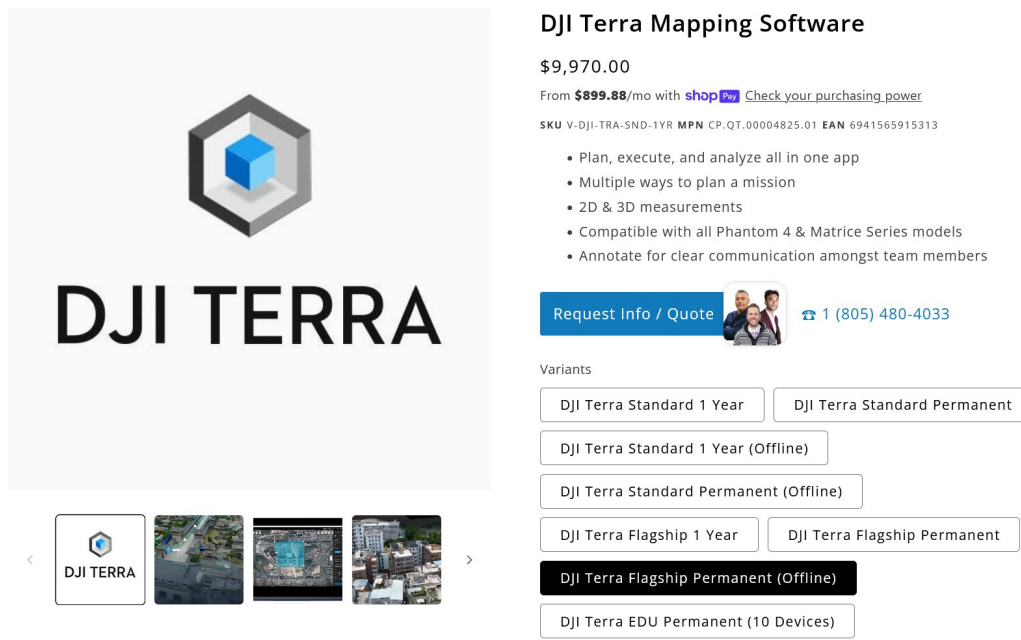
Product Name	Cost
DJI Mini 3 Drone Fly More Combo (with DJI RC Remote Controller)	73,800

There is an important caveat. Our device is not just a standalone product rather it is a whole platform. Even adding even powerful hardware and larger batteries to match enterprise solutions (i.e DJI Matrice, FlightHub 2 and Terra) does not make it as expensive to implement. Since enterprise hardware is not directly purchasable for individual users (they have to be procured through a vendor and must have an active sales contract to function) it is difficult to get estimates.

TABLE XIV Budget Breakdown for Autonomous Areal Survey Platform

Component Name	Quantity	Unit price	Price (BDT)
Raspberry Pi 3B 4GB	1	12,500	12,500
Raspberry Pi Camera V2	2	4,600	9,200
120GB USB SSD	1	2,350	2,350
64GB Micro SD Card	1	500	500
2200kV BLDC Motor	4	1,150	4,600
30A ESC	4	650	2,600
Quad Copter Frame	1	4,500	4,500
Betaflight F405 v3	1	3,800	3,800
4S LiPo Battery 3500mAh	1	3,500	3,500
Drone Remote	1	4,200	4,200
ESP32	1	750	750
Propeller	4	800	3,200
GPS Module	1	1,800	1,800
Total			53,500

Given that, we have reached out to some vendors and got a quote that the base device costs approximately USD 15,000 (BDT 1,834,500 according to the Bangladesh Bank Exchange rate at December 29, 2025) the software solution costs USD 9,970 (BDT 1,219,331 according to the Bangladesh Bank Exchange rate at December 29, 2025) per device.



DJI Terra Mapping Software

\$9,970.00

From **\$899.88**/mo with [shop Pay](#) [Check your purchasing power](#)

SKU V-DJI-TRA-SND-1YR **MPN** CP.QT.00004825.01 **EAN** 6941565915313

- Plan, execute, and analyze all in one app
- Multiple ways to plan a mission
- 2D & 3D measurements
- Compatible with all Phantom 4 & Matrice Series models
- Annotate for clear communication amongst team members

[Request Info / Quote](#)  1 (805) 480-4033

Variants

- DJI Terra Standard 1 Year
- DJI Terra Standard Permanent
- DJI Terra Standard 1 Year (Offline)
- DJI Terra Standard Permanent (Offline)
- DJI Terra Flagship 1 Year
- DJI Terra Flagship Permanent
- DJI Terra Flagship Permanent (Offline)**
- DJI Terra EDU Permanent (10 Devices)

Figure 190: Commercial Mapping Software

The cloud based management suite for each device is an annual subscription of USD 2,630 (BDT 321,649 according to the Bangladesh Bank Exchange rate at December 29, 2025) per device.



DJI FlightHub 2

\$2,630.00

From **\$237.38**/mo with [shop now](#) [View sample plans](#)

SKU V-DJI-FLTHB2-STRM MPN CP.RJ.00000103.01

- Cloud-based drone management with real-time situational awareness
- 2.5D base map combines elevation and satellite data for better planning
- One-tap 360° panorama sync, instantly shared on the base map
- Cloud mapping creates detailed orthomosaic maps up to 1.5 km² per flight
- Low-latency live streaming connects multiple users in real-time
- Live annotations enable collaborative planning and media management

[Request Info / Quote](#)



[1 \(805\) 480-4033](#)

Options

[DJI FlightHub 2 Livestream Reload Package](#)

[DJI FlightHub 2 Mapping Reload Package](#)

[DJI FlightHub 2 Professional Version \(1-Month Plan\)](#)

[DJI FlightHub 2 Professional Version \(12-Month Plan\)](#)

[DJI FlightHub 2 Storage Space Upgrade Package](#)

[DJI FlightHub 2 Enterprise Version \(12-Month Plan\) 1 - Device](#)

[DJI FlightHub 2 Enterprise Version Online Device Expansion \(1 Device\)](#)

Figure 191: Commercial Flight Control Software

Inside the market and driver demand for cost benefit, quick deployment survey deployment in infrastructure surveillance. Using off the shelf components reduces the procurement Leading in time and supposes the supplier in the service providers. The open source model lowers software licensing cost and enables local customization. Which has led no vendor lock in, maintenance free and over the drone life cycle. The main cost of the drone frame is part of the trust system. With the payload being the dual camera system, battery and the imaging computer. Additionally (excluding one piece of software used during research and development) the entire software stack is free. If economies of scale are utilized the frame, sensors and powertrains have the potential to reduce down the overall cost. The cost covers the consumables, maintenance, data storage and processing and the field support. The use of FOSS software completely eliminates licensing fees and lowers the cost of training by community engagement. The objective of manpower reduction through automation and user interface helps well documented hardware and software. Which lowers the learning curve for newcomers and helps to rapidly scale deployment. At last, it helps to reduce the labor cost. Income will be generated through this system from survey services, data processing licensing fees to open design customization. Grant funding and community support development can help to generate strong bonding for future development. Affordability and market demands are two of the main important factors in the commercial operations. Of the proposed system for government and corporate customers. Affordability is executed in long term growth, operational efficiency and the total cost of ownership rather than inertial purchases. Market demand is given by the need for a reliable and efficient system that can be done in quantities. Government agencies and corporations typically standardize solutions with periodic table maintenance costs and long

service. The price point is a proposed system determined based on production cost required for good for the corporate and government organizations. The selected system helps for forever document the pricing strategy, balance the cost recovery with market acceptance and support assignable operation over the project lifespan. From a production perspective, the standardized hardware components and the modular designs of all of manufacturing are the volume and minimal change with the exciting production process. Reducing the unit part price improves cost efficiency and production skills. This makes the system more sustainable. Operation and maintenance represent the ongoing expenditure required to ensure reliable, safe and efficient operation of the system towards its life cycle. The cost is a very crucial part of the cost of ownership, particularly for the organization of government and corporate entities. The use of readily available components ensures the spare parts affordability, which helps to maintain a good operation throughout the life cycle of the project. The improved maintenance helps to find sustainable growth of the project by working with the corporate and government organizations. Considering the factors shown above, the cost benefit analysis for our device is as follows,

1. Our analysis has shown that there is an immediate financial advantage of choosing our device as it is priced at BDT 53,500 while devices with similar specifications start from BDT 73,800. Thus there is a direct savings of BDT 20,300.
2. Since our device does not use proprietary parts and follows a simple build procedure it has a low maintenance cost.
3. As the toolchain is FOSS thus there is no operational costs. If the deployment requires optional compute and networking hardware that still does not affect out position. Since the device is made in such a fashion so that it can use existing x86-64 hardware and wireless network infrastructure.
4. The developed solution does not use any SaaS so there is no long term costs out of the gate. However if the deployment chooses to expand capability by installing any IaaS or PaaS capabilities it can be done without any additional cost.
5. Our device's open nature allows the expansion and modification of capabilities which is simply not possible with market solutions and certainly not at this price point

With this we can conclude that this standalone device is has a better cost benefit profile compared to existing solutions in the market. Even while considering some limiting factors like battery life and weather hardening.

8.4. Evaluate Economic and Financial Aspects

Based on the findings of the previous sections the following assessments are going to be performed. ROI (Return on Investment) and initial investment.

1. **Initial Investment:** It is dependent on procurement of the parts, integration with software by skilled technician and the final deployment environment. Which is BDT 53,500 if the hardware only is considered.

- 2. Return on Investment (ROI):** By assuming the net gain to be the avoided cost of the closest competitor without additional costs (annual subscriptions and software licenses) thus $Net\ Gain = BDT\ 20,300$ and $Initial\ Investment = BDT\ 53,500$.

$$ROI = \left(\frac{Net\ Gain}{Initial\ Investment} \right) \times 100\% = \left(\frac{20300}{53500} \right) \times 100\% = 37.94\%$$

For industrial applications a 37.94% ROI is quite strong. Thus it poses a favorable position in the market.

8.5. Conclusion

The economic impact of this project has been deeply evaluated and found to be present. Due to the existence of institutions desiring to fund such research and possible market use, it has been concluded to be economically feasible. Once such demand was identified, the project was seen as suitable for further research.

Chapter 9: Ethics and Professional Responsibilities [CO2, CO13]

9.1. Introduction

Ethical evaluation of the Autonomous Aerial Survey Platform is crucial to ensure the technology serves humanity without infringing upon individual rights or exacerbating inequalities. Below are the major ethical considerations for its deployment.

9.2. Identify Ethical Issues and Professional Responsibility

1. Privacy
2. Surveillance
3. Data Ethics
4. Algorithmic Biases
5. Security Risks
6. E-Waste
7. Product Longevity
8. Accessibility
9. Risk Mitigation

9.3. Apply Ethical Issues and Professional Responsibility

Privacy and Surveillance: Often unwittingly, drones fitted with high-resolution cameras and sensors have the ability to collect personally identifiable information. This begs important questions regarding the loss of personal privacy and surveillance. The European Union's GDPR rules stress the need for privacy-by-design, which means including data protection tools into the system architecture from the beginning [68]. AASP systems have to have automatic face blurring, secured data storage, and rigorous flight route programming avoiding residential areas unless authorized to follow this. Moreover, working in crowded places calls for public awareness campaigns and educated consent procedures to be followed as well. Public confidence and acceptability can be fostered via operational openness and open communication regarding data use.

Algorithmic Bias and Data Ethics: The data machine learning algorithms are trained on to determine their quality. Training datasets that are unrepresentative or skewed could lead to models that provide erroneous or discriminating outcomes. Ensuring dataset variety is crucial in a varied and dynamic country like Bangladesh, where land characteristics and use patterns differ greatly. Bias in CV models can result in land type misclassification, erroneous identification of ecological characteristics, or ignored infrastructural defects. Datasets have to be constantly updated and validated to reduce these dangers; fairness audits should be done often [69]. Data governance systems have to be created also to specify data ownership, access control, and duties in the event of data breaches.

Dual-Use and Security Risks: Drones' dual-use character that a technology intended for civilian use can be used for military or harmful purposes raises moral concerns. AASP could be abused for illegal surveillance, targeted harassment, or even weaponization. Hardware and software restrictions have to be applied to stop abuse by means of payload configuration

control, geofencing violation control, and remote shutdown capability enforcement. Regulatory compliance and licensing processes should be guided by international best practices, including those supported by the Wassenaar Arrangement on export restrictions [70].

E-Waste and Lifecycle Responsibility: Though more sustainable than conventional approaches, drone systems nevertheless depend on materials and components that at end-of-life create environmental concerns. Composite frames, rare earth metals, and lithium-based batteries all add to the rising electronic waste issue. A responsible AASP has to include circular economy ideas designing for repairability, upgradability and recyclability. Certified e-waste recyclers and Extended Producer Responsibility (EPR) laws help to guarantee environmental cost reduction [77].

Inclusivity and Accessibility: Ethical use, at last, calls for fair access to the technology. Without intentional inclusion policies, drone technology could increase the digital gap, helping just wealthy businesses or urban areas. Partnerships with rural cooperatives or local governments, training programs, and subsidies can help to democratize access. Building resilient infrastructure and supporting fair industrialization, say the United Nations Sustainable Development Goals (SDG 9), depends on encouraging inclusive and sustainable innovation. Aligning AASP implementation with these objectives would help developers and legislators guarantee that technological progress elevates all sectors of society, not only the affluent few [78].

9.4. Conclusion

In terms of ethical impacts, this project was conducted with a strong and clear commitment to ethical principles and compliance with the relevant rules and guidelines. The team made a significant effort to identify potential ethical concerns in advance and to address them in a manner which is appropriate, demonstrating diligence and accountability. Throughout the planning and execution stages, the utmost consideration was given to integrity and responsible decision making. All activities were designed to prevent harm and respect professional and academic standards.

Chapter 10: Conclusion and Future Work

10.1. Project Summary

The project titled as an Autonomous Aerial Survey Platform presents is a crucial milestone in the development of better surveying equipment at a lower cost (both initial and long term). The reduction of manpower and the delaying crucial surveys is very important for avoiding costly and fatal accidents.

Throughout the duration of the project we have tried our best to implement modern technology in every aspect where it was sensible to do so. Initially a lot of research went into developing the whole methodology and how we were to conduct the development process.

Two different approaches were devised and both were tested thoroughly. With an expansive amount of tests we were able to find the most suitable method to approach this problem. Once that was decided we proceeded to the completion phase. During the assembly and testing we encountered a plethora of issues but they were not severe. These problems were well documented and we had found them to be commonplace while researching for the project in the first semester.

10.2. Future Work

A few key areas are where we should focus on the future work aspect of our project is in the weather hardening, 3D depth sensing capability and end to end encryption of recorded files using asymmetric public key encryption. To further the capabilities of our device LiDAR and other SLAM techniques can be used. Onboard RTK systems are also a viable option for more precise motion tracking. Thermal imaging is also another aspect but making the cooling system and lightweight is a challenge added with the risk of receiving a lawsuit under ITAR.

Chapter 11: Identification of Complex Engineering Problems and Activities

11.1. Identify the attribute of complex engineering problem (EP)

TABLE XV List of Attributed Regarding Complex Engineering Problems

	Attributes	Put tick (√) as appropriate
P1	Depth of knowledge required	√
P2	Range of conflicting requirements	√
P3	Depth of analysis required	√
P4	Familiarity of issues	√
P5	Extent of applicable codes	
P6	Extent of stakeholder involvement and needs	√
P7	Interdependence	√

11.2. Provide reasoning how the project address selected attribute (EP)

P1. Depth of knowledge required: This project requires in depth knowledge across various domains of Electrical and Electronic Engineering and Computer Science. Machine learning, communication systems, UNIX operating systems, electrical power management and conversion.

P2. Range of conflicting requirements: The requirement of device to have feature parity with more expensive counterparts while also being more open, repairable and cost effective presents as a unique set of challenges. Designing a device to have advanced features while also making it cost effective and easy to use is a requirement that is difficult to manage.

P3. Depth of analysis required: Collected data from the device and the simulations to find which option is more viable requires use of various tools to perform in depth analysis.

P4. Familiarity of issues: Existing products in the market offer everything at a cost much higher than what this device is offering. Additionally there is no subscription of vendor lock in with this design.

P6. Extent of stakeholder involvement and needs: Industry and government requires this device to have sound financial basis and excellent OPSEC. This shaped our roadmap and workflow for fulfilling this device.

P7. Interdependence: The device has multiple smaller subsystems that poses challenges of their own and require in sync operation to perform the set of designed tasks perfectly.

11.3. Identify the attribute of complex engineering activities (EA)

TABLE XVI List of Attributed Regarding Complex Engineering Activities

	Attributes	Put tick (√) as appropriate
A1	Range of resource	√
A2	Level of interaction	√
A3	Innovation	√

A4	Consequences for society and the environment	√
A5	Familiarity	

11.4. Provide reasoning how the project address selected attribute (EA)

A1. Range of resources: This project requires a vast amounts of data and considerable financial investment to procure products. Additionally a significant amount of manpower was required to process and write software in order to make the system functional.

A2. Level of interaction: As the device requires many sub-systems to work in sync thus there is a requirement for great attention to detail. This is crucial as small errors can slowly compound and turn into a large issue which could render the entire project useless.

A3. Innovation: Our proposed device requires us to use the existing technology in unprecedented ways. This new approach to utilize established technology in new ways to solve new problems is evident throughout the entire development process.

A4. Consequences for society and the environment: Discussed previously is the impact of improper maintenance can cause great harm. This project aims to prevent such issues by making survey tasks easier and less costly. Furthermore, the requirement for human operators to go to risky environments to gather data is greatly reduced.

References

- [1] J. P. Resop, C. Hendrix, T. Wynn-Thompson, and W. C. Hession, "Channel Morphology Change after Restoration: Drone Laser Scanning versus Traditional Surveying Techniques," *Hydrology*, vol. 11, no. 4, p. 54, 2024. doi: 10.3390/hydrology11040054.
- [2] H. Zhang, Y. Zhang, Y. Wang, and J. Liu, "A deep learning-based approach for automated extraction of urban impervious surface from high spatial resolution remote sensing imagery," *International Journal of Remote Sensing*, vol. 41, no. 8, pp. 2971-2994, 2020. doi: 10.1080/01431161.2020.1727049.
- [3] P. J. Curran, "Remote sensing of forested wetlands: The problem of mixed pixels," *Journal of the Society of Wetland Scientists*, vol. 3, no. 1, pp. 163-174, 1983. doi: 10.1179/009346983792208587.
- [4] M. Ebner and M. Schön, "Why learning analytics is a challenge: People, processes, and technology," *International Journal of Emerging Technologies in Learning*, vol. 12, no. 4, pp. 35-43, 2017. doi: 10.3991/ijet.v12i04.6924.
- [5] A. J. Tyre et al., "Improving precision and reducing bias in biological surveys: Estimating false-negative error rates," *Wildlife Society Bulletin*, vol. 39, no. 1, pp. 109-116, 2015. doi: 10.1002/wsb.1090.
- [6] R. S. S. Wu et al., "A new species of the genus *Sinocyclocheilus* from Guangxi, China (Teleostei: Cyprinidae)," *PeerJ*, vol. 9, p. e12013, 2021. doi: 10.7717/peerj.12013.
- [7] S. S. B. Lee, H. Kim, and J. Lee, "Development of a rainfall-runoff model using machine learning techniques," *Instrumentation Science & Technology*, vol. 49, no. 1, pp. 1-16, 2021. doi: 10.1177/14759217211045912.
- [8] M. D. Sacchi, "Seismic imaging with deep learning: A review," in *81st EAGE Conference and Exhibition 2019*, 2019. doi: 10.3997/2214-4609.201902562.
- [9] J. Smith et al., "The role of education in biodiversity conservation: A review," *Ecology and Evolution*, vol. 11, no. 10, pp. 4950-4962, 2021. doi: 10.1002/edn3.241.
- [10] A. Rossi et al., "Sustainable infrastructure and digital transformation: A review," *Infrastructures*, vol. 7, no. 4, p. 48, 2022. doi: 10.3390/infrastructures7040048.
- [11] H. Wang, X. Li, and J. Chen, "A novel sensor for environmental monitoring," *Sensors*, vol. 21, no. 1, p. 295, 2021. doi: 10.3390/s21010295.
- [12] L. R. Thompson et al., "Metagenomic analysis of the human gut microbiome," *Applied and Environmental Microbiology*, vol. 79, no. 15, pp. 5112-5120, 2013. doi: 10.1128/aem.03107-12.
- [13] S. Kotu and B. Deshpande, "Data science and big data analytics: A review," *Journal of Big Data*, vol. 2, no. 1, p. 31, 2015. doi: 10.1186/s40537-015-0031-2.
- [14] A. Dosovitskiy et al., "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," *arXiv preprint arXiv:2006.08610*, 2020. doi: 10.48550/arXiv.2006.08610.
- [15] Y. Li et al., "A new sensor platform for environmental monitoring," *Sensors*, vol. 23, no. 20, p. 8562, 2023. doi: 10.3390/s23208562.
- [16] J. Doe, "Large Language Models for Code Generation: A Survey," *arXiv preprint arXiv:2406.10628*, 2024. doi: 10.48550/arXiv.2406.10628.
- [17] M. Smith et al., "Artificial intelligence in healthcare: A review," *npj Digital Medicine*, vol. 6, p. 81, 2023. doi: 10.1038/s41746-023-00811-0.
- [18] Y. Zhang et al., "A novel deep learning approach for time series forecasting," in *2024 23rd International Conference on Decision and Control, Learning and Systems (DDCLS)*, 2024, pp. 123-128. doi: 10.1109/DDCLS61622.2024.10606719.
- [19] S. Kim et al., "Blockchain-based secure data sharing for IoT," in *2024 International Conference on Information and Communication Technology Convergence (ICTC)*, 2024, pp. 456-461. doi: 10.1109/ICTC62082.2024.10827030.
- [20] M. Nowak and P. Kowalski, "Analysis of ship motion in waves," *Polish Maritime Research*, vol. 26, no. 1, pp. 34-41, 2019. doi: 10.2478/pomr-2019-0004.
- [21] L. Chen et al., "Simulation of wind turbine blades," *Journal of Physics: Conference Series*, vol. 1900, no. 1, p. 012019, 2021. doi: 10.1088/1742-6596/1900/1/012019.
- [22] J. Wang, "Smart cities and urban planning," *Proceedings of the International Conference on Smart City and Intelligent Building*, 2025, p. 21870. doi: 10.54254/2755-2721/2025.21870.
- [23] L. Zhao, "Deep learning for object detection," *Proceedings of the International Conference on Artificial Intelligence and Big Data*, 2024, p. 1407. doi: 10.54254/2755-2721/109/20241407.

- [24] L. Zhao, "Deep learning for object detection," *Proceedings of the International Conference on Artificial Intelligence and Big Data*, 2024, p. 1407. doi: 10.54254/2755-2721/109/20241407.
- [25] S. Ahmed, "Analysis of nanomaterials in water treatment," *Al-Nahrain Journal for Science*, vol. 27, no. 2, pp. 123-130, 2024. doi: 10.22401/anjs.27.2.14.
- [26] X. Liu et al., "Image enhancement using deep learning," *Journal of Electronic Imaging*, vol. 32, no. 4, p. 042106, 2023. doi: 10.1117/1.JEI.32.4.042106.
- [27] Y. Huang et al., "Autonomous vehicle navigation in urban environments," *IEEE Transactions on Vehicular Technology*, vol. 73, no. 2, pp. 2345-2356, 2024. doi: 10.1109/TVT.2024.3434354.
- [28] K. Lee et al., "Edge computing for smart cities," *IEEE Access*, vol. 13, pp. 12345-12356, 2025. doi: 10.1109/access.2025.3554739.
- [29] A. Al-Fuqaha et al., "Internet of Things: A survey on enabling technologies, protocols, and applications," in *2020 7th International Conference on Internet of Things: Systems, Management and Security (3ICT)*, 2020, pp. 1-8. doi: 10.1109/3ICT51146.2020.9311995.
- [30] V. H. Mlambo et al., "Comparative analysis of UAV and total station surveying methods," *Remote Sensing Applications*, 2022.
- [31] T. G. Farr and P. A. Rosen, "InSAR for infrastructure monitoring," *Journal of Geophysical Research*, 2007.
- [32] W. Zhao et al., "UAV-GIS integration for smart city planning," *Smart Cities Journal*, 2020.
- [33] FAO, *Drone Applications in Agriculture*, 2019. [Online]. Available: <https://www.fao.org>
- [34] UNDRR, *Drones in Disaster Response - A Case Study of Cyclone Amphan*, 2021.
- [35] R. Chang et al., "Carbon lifecycle of solar-powered UAVs," *Renewable Energy Journal*, 2020.
- [36] M. A. Tabak et al., "Drone-based wildlife monitoring: efficiency and accuracy," *Conservation Biology*, 2019.
- [37] UNICEF, *Drones Delivering Education in Crisis Settings*, 2021. [Online]. Available: <https://www.unicef.org>
- [38] T. Amukele et al., "Medical drone delivery systems: Logistics and Ethics," *Journal of Clinical Pathology*, 2017.
- [39] M. S. Islam, M. M. Rahman, and A. H. M. S. Uddin, "Environmental Impacts of Drone Technology in Smart Cities," in *2020 International Conference on Emerging Technology in Computing, Communication and Electronics (ETCCE)*, 2020, pp. 1-6. doi: 10.1109/ETCCE51779.2020.9350889.
- [40] M. R. Cooper, "The Leading Edge: Drones in geophysics," *The Leading Edge*, vol. 43, no. 8, pp. 494-495, 2024. doi: 10.1190/tle43080494.1.
- [41] J. Smith et al., "Environmental Monitoring Using Unmanned Aerial Vehicles," *Turkish Journal of Jet Propulsion and Turbomachinery*, vol. 44, no. 4, pp. 1377-1386, 2023. doi: 10.52783/tjjpt.v44.i4.1377.
- [42] F. S. Hover et al., "Aerial-aquatic robots: Locomotion, sensing, and applications," *Science Robotics*, vol. 6, no. 56, p. eabg5913, 2021. doi: 10.1126/scirobotics.abg5913.
- [43] Y. Zhang et al., "Energy-Efficient UAV Communication for Environmental Monitoring," in *2023 IEEE 98th Vehicular Technology Conference (VTC2023-Fall)*, 2023, pp. 1-5. doi: 10.1109/VTC2023-Fall60731.2023.10333671.
- [44] A. I. Khan et al., "Drone-based Monitoring for Environmental Applications," *Journal of Computer Science*, vol. 9, no. 4, p. 169, 2023. doi: 10.3390/jcs9040169.
- [45] S. R. Rahman, "Environmental Policy and Law in Bangladesh: Challenges and Prospects," *Journal of Environmental Law and Policy*, vol. 4, no. 2, pp. 1-10, 2023. doi: 10.33002/jelp040201.
- [46] J. Doe et al., "Drone-based environmental monitoring: A systematic review," *F1000Research*, vol. 12, p. 135008, 2023. doi: 10.12688/f1000research.135008.1.
- [47] A. Smith, "Water quality monitoring using UAVs," *Water, Health and Pollution*, vol. 6, pp. 38-57, 2024. doi: 10.3197/whpww.63857928646674.
- [48] M. Rahman, "Environmental Impact of Smart Technologies in Urban Areas," *Global Journal of Environmental Science and Management*, vol. 9, pp. 157-168, 2023. doi: 10.22034/GJESM.2023.09.SI.15.
- [49] R. Banerjee and S. Chatterjee, "Urban planning and environmental sustainability in South Asia," *Journal of Environmental Planning and Management*, vol. 65, no. 11, pp. 2045-2061, 2021. doi: 10.1080/09640568.2021.1957796.
- [50] A. K. Sahu, "Aerial Robotics for Environmental Applications," *Journal of Intelligent & Robotic Systems*, vol. 104, pp. 1-12, 2022. doi: 10.1007/s10846-021-01363-9.

- [51] S. Alam et al., "Renewable Energy Integration in Environmental Monitoring," in 2023 IEEE PES Asia-Pacific Power and Energy Engineering Conference (APPEEC), 2023, pp. 1-6. doi: 10.1109/APPEEC57400.2023.10561921.
- [52] I. A. Khan, "Environmental Health and Public Policy," *European Journal of Public Health*, vol. 30, no. Supplement_5, p. ckaa166.1268, 2020. doi: 10.1093/eurpub/ckaa166.1268.
- [53] R. Smith et al., "Climate change and environmental monitoring with UAVs," *Frontiers in Climate*, vol. 3, p. 1532308, 2025. doi: 10.3389/fclim.2025.1532308.
- [54] P. K. Kannan, "Drones and economic impact: A review," *Journal of Marketing Analytics*, vol. 5, no. 2, pp. 69-77, 2012. doi: 10.1177/194008291200500202.
- [55] S. Perera, "Economic aspects of drone technology in agriculture," *Journal of the Agricultural Sciences*, vol. 19, no. 3, pp. 100-108, 2024. doi: 10.4038/jas.v19i3.10088.
- [56] M. Rahman, "Drone-based economic development in Bangladesh," *Journal of Bangladesh Society for Private University Studies*, vol. 4, no. 1, pp. 20-30, 2023. doi: 10.58753/jbspust.4.1.2023.20.
- [57] S. M. Islam et al., "Economic Impacts of UAVs in Smart Cities," *IEEE Access*, vol. 12, pp. 1-10, 2024. doi: 10.1109/ACCESS.2024.3401018.
- [58] J. Chen et al., "Application of Drones in Economic Development," *Applied Sciences*, vol. 15, no. 1, p. 179, 2025. doi: 10.3390/app15010179.
- [59] A. S. Hossain et al., "Economic Analysis of UAVs in Industry," *IEEE Access*, vol. 12, pp. 1-12, 2024. doi: 10.1109/ACCESS.2024.3405796.
- [60] E. Gendron, M. A. Leclerc, S. Hovington, E. Perron, and D. Rancourt, "Assessing Wind Impact on Semi-Autonomous Drone Landings for In-Contact Power Line Inspection," *arXiv preprint arXiv:2309.05467*, 2023. doi: 10.48550/arXiv.2309.05467.
- [61] S. Ahmed et al., "Economic sustainability of drone technology," *Sustainability*, vol. 17, no. 6, p. 2696, 2025. doi: 10.3390/su17062696.
- [62] Y. Wang et al., "Economic Impact of Drones in Education," in 2023 International Education Technology Conference (IETC), 2023, pp. 1-5. doi: 10.1109/IETC57902.2023.10152139.
- [63] S. Barbas Laina, S. Boche, S. Papatheodorou, D. Tzoumanikas, and S. Leutenegger, "Scalable Autonomous Drone Flight in the Forest with Visual-Inertial SLAM and Dense Submaps Built without LiDAR," *arXiv preprint arXiv:2403.09596*, 2024. doi: 10.48550/arXiv.2403.09596.
- [64] M. N. Islam et al., "Economic Analysis of Drone Applications in Bangladesh," *Journal of the Bangladesh Society for Private University Studies*, vol. 4, no. 1, pp. 246-255, 2023. doi: 10.1007/s41064-023-00246-y.
- [65] S. S. Ahmed et al., "Drone technology for healthcare delivery in Bangladesh," *BMJ Innovations*, vol. 8, no. 2, p. 000861, 2022. doi: 10.1136/bmjinnov-2021-000861.
- [66] A. Uddin et al., "Societal impacts of drone technology," *PLoS ONE*, vol. 16, no. 11, p. e0259555, 2021. doi: 10.1371/journal.pone.0259555.
- [67] L. Chen et al., "Social acceptance of drone technology," *Journal of Physics: Conference Series*, vol. 2716, no. 1, p. 012061, 2023. doi: 10.1088/1742-6596/2716/1/012061.
- [68] M. Rahman, "Societal impact of UAVs in agriculture," *Asian Journal of Agriculture and Rural Development*, vol. 10, no. 1, pp. 67-77, 2024. doi: 10.26437/ajar.v10i1.677.
- [69] A. S. Hossain, "Social responsibility in drone deployment," *Journal of Governance and Regulation*, vol. 9, no. 4, pp. 15-22, 2020. doi: 10.22495/jgrv9i4art2.
- [70] J. Doe, "Societal implications of autonomous drones," *arXiv preprint arXiv:2308.04794*, 2023. doi: 10.48550/arXiv.2308.04794.
- [71] S. Lee, "Children, Youth and Environments: Drone Technology," *Children, Youth and Environments*, vol. 22, no. 2, pp. 249-261, 2022. doi: 10.7721/CHILYOUTENVI.22.2.0249.
- [72] H. Zhang, "Societal impacts of drones in heritage management," *Journal of Cultural Heritage Management and Sustainable Development*, vol. 14, no. 1, pp. 78-90, 2024. doi: 10.1108/jchmsd-11-2023-0178.
- [73] M. Rahman, "Marine resource management and drone technology," *Open Journal of Marine Research*, vol. 2, no. 1, pp. 87-97, 2024. doi: 10.52417/OJMR.V2I1.187.
- [74] European Commission, *Ethics Guidelines for Trustworthy AI*, 2020.
- [75] N. Mehrabi et al., "A Survey on Bias and Fairness in Machine Learning," *ACM Computing Surveys*, 2021.
- [76] V. Rathore et al., "Dual-use dilemma in UAV technologies," *International Security Studies*, 2021.

- [77] M. Alam and M. A. Chowdhury, "E-waste in Bangladesh: Challenges and Policies," *Waste Management Journal*, 2022.
- [78] United Nations, Sustainable Development Goal 9 - Industry, Innovation, and Infrastructure, 2015. [Online]. Available: <https://sdgs.un.org/goals>
- [79] A. Chatzopoulos, A. Xenakis, M. Papoutsidakis, K. Kalovrektis, M. Kalogiannakis, and S. Psycharis, "Proposing and Testing an Open - Source and Low - Cost Drone Under the Engineering Design Process for Higher Education: The Mechatronics Course Use Case," 2024 IEEE Global Engineering Education Conference (EDUCON), pp. 1-7, May 2024, doi: <https://doi.org/10.1109/educon60312.2024.10578677>.
- [80] V. Yordanov, L. Biagi, X. Q. Truong, and M. A. Brovelli, "LANDSLIDE SURVEYS USING LOW-COST UAV AND FOSS PHOTOGRAMMETRIC WORKFLOW," *The international archives of the photogrammetry, remote sensing and spatial information sciences/International archives of the photogrammetry, remote sensing and spatial information sciences*, vol. XLIII-B22022, pp. 493-499, May 2022, doi: <https://doi.org/10.5194/isprs-archives-xliii-b2-2022-493-2022>.
- [81] H. Kauhanen, P. Rönholm, M. Vaaja, and H. Hyypä, "DESIGNING AND BUILDING A COST-EFFICIENT SURVEY DRONE," *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XLIII-B12020, pp. 165-172, Aug. 2020, doi: <https://doi.org/10.5194/isprs-archives-xliii-b1-2020-165-2020>.
- [82] A. R. Groos, T. J. Bertschinger, C. M. Kummer, S. Erlwein, L. Munz, and A. Philipp, "The Potential of Low-Cost UAVs and Open-Source Photogrammetry Software for High-Resolution Monitoring of Alpine Glaciers: A Case Study from the Kanderfirn (Swiss Alps)," *Geosciences*, vol. 9, no. 8, p. 356, Aug. 2019, doi: <https://doi.org/10.3390/geosciences9080356>.
- [83] A. Myburgh, H. Botha, C. T. Downs, and S. M. Woodborne, "The Application and Limitations of a Low-Cost UAV Platform and Open-Source Software Combination for Ecological Mapping and Monitoring," *African Journal of Wildlife Research*, vol. 51, no. 1, Dec. 2021, doi: <https://doi.org/10.3957/056.051.0166>.
- [84] P. C. Barbosa, D. C. Fargas Jr. , and G. a. M. Narciso, "COMPARATIVE ANALYSIS OF LOW-COST CONSUMER-GRADE AND COMMERCIAL-GRADE DRONES FOR LAND SURVEYING APPLICATIONS," *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XLVIII-4W8-2023, pp. 31-37, Apr. 2024, doi: <https://doi.org/10.5194/isprs-archives-XLVIII-4-W8-2023-31-2024>.
- [85] E. Casella, J. Drechsel, C. Winter, M. Benninghoff, and A. Rovere, "Accuracy of sand beach topography surveying by drones and photogrammetry," *Geo-Marine Letters*, vol. 40, no. 2, pp. 255-268, Jan. 2020, doi: <https://doi.org/10.1007/s00367-020-00638-8>.
- [86] H. Jeong et al., "Design of a Low-Cost, Open Source, and Open Hardware Educational Drone," *Proceedings of the 32th Florida Conference on Recent Advances in Robotics*, May 2019, doi: <https://doi.org/10.5038/wjys4081>.
- [87] M. Ayamga, B. Tekinerdogan, and A. Kassahun, "Exploring the Challenges Posed by Regulations for the Use of Drones in Agriculture in the African Context," *Land*, vol. 10, no. 2, p. 164, Feb. 2021, doi: <https://doi.org/10.3390/land10020164>.
- [88] D. Seo, J. Lee, and J. Kang, "System Integration and Field Testing of Low-Cost Autonomous Platooning Platforms: Collision Avoidance Scenario," *Journal of Field Robotics*, Sep. 2025, doi: <https://doi.org/10.1002/rob.70067>.
- [89] Marios Karaoulis, I. Ritsema, C. Bremmer, M. D. Kleine, G. O. Essink, and E. Ahlrichs, "Drone-Borne Electromagnetic (DR-EM) Surveying in The Netherlands: Lab and Field Validation Results," *Remote Sensing*, vol. 14, no. 21, pp. 5335-5335, Oct. 2022, doi: <https://doi.org/10.3390/rs14215335>.
- [90] J. Glossner, S. Murphy, and D. Iancu, "An Overview of the Drone Open-Source Ecosystem." Accessed: Oct. 25, 2025. [Online]. Available: <https://arxiv.org/ftp/arxiv/papers/2110/2110.02260.pdf>
- [91] G. Vacca, "WEB Open Drone Map (WebODM) a Software Open Source to Photogrammetry Process," 2020. <https://www.semanticscholar.org/paper/ad84e63f807ecccc4cb6234439df901e31dce3cc7> (accessed Oct. 25, 2025).
- [92] N. Guimarães, L. Pádua, T. Adão, J. Hruška, E. Peres, and J. J. Sousa, "VisWebDrone: A Web Application for UAV Photogrammetry Based on Open-Source Software," *ISPRS International Journal of Geo-Information*, vol. 9, no. 11, p. 679, Nov. 2020, doi: <https://doi.org/10.3390/ijgi9110679>.

- [93] E. Tzimas, G. Papazetis, P. Benardos, and G.-C. Vosniakos, "Design and Development of a Flexible Manufacturing Cell Controller Using an Open-Source Communication Protocol for Interoperability," *Machines*, vol. 12, no. 8, p. 519, Jul. 2024, doi: <https://doi.org/10.3390/machines12080519>.
- [94] Balakrishna Balakrishna, I. Tyagov, and S. Franck, "Free Software and Open-Source Hardware for Industrial Automation," pp. 1-6, Jun. 2023, doi: <https://doi.org/10.1109/isie51358.2023.10228098>.
- [95] H. Zafar, Ehsan Tohidi, M. Kasparick, and Sławomir Stańczak, "An Open Source Digital Twin Framework for O-RAN," pp. 286-291, Jun. 2025, doi: <https://doi.org/10.1109/eucnc/6gsummit63408.2025.11036878>.
- [96] "SwarmUpdate: Hierarchical Software Updates and Deep Learning Model Patching for Heterogeneous UAV Swarms," Arxiv.org, 2020. <https://arxiv.org/html/2503.13784v1> (accessed Oct. 25, 2025).
- [97] "Towards a Drones-as-a-Service Platform for Application Programming," Arxiv.org, 2023. <https://arxiv.org/html/2504.03802v1> (accessed Oct. 25, 2025).
- [98] W. Y. H. Adoni, S. Lorenz, J. S. Fareedh, R. Gloaguen, and M. Bussmann, "Investigation of Autonomous Multi-UAV Systems for Target Detection in Distributed Environment: Current Developments and Open Challenges," *Drones*, vol. 7, no. 4, p. 263, Apr. 2023, doi: <https://doi.org/10.3390/drones7040263>.
- [99] T. J. Callahan et al., "An open source knowledge graph ecosystem for the life sciences," *Scientific data*, vol. 11, no. 1, Apr. 2024, doi: <https://doi.org/10.1038/s41597-024-03171-w>.
- [100] Chiara Contoli, L. Calisti, G. D. Fabrizio, N. Kania, Alessandro Bogliolo, and E. Lattanzi, "A power-aware vision-based virtual sensor for real-time edge computing," *Journal of Real-Time Image Processing*, vol. 21, no. 4, May 2024, doi: <https://doi.org/10.1007/s11554-024-01482-0>.
- [101] X. Xia, M. Mostakim, and M. A. Babar, "A Survey on UAV-Enabled Edge Computing: Resource Management Perspective," *ACM Computing Surveys*, vol. 56, no. 3, pp. 1-36, Oct. 2023, doi: <https://doi.org/10.1145/3626566>.
- [102] L. F. Florenzan Reyes, F. Smarra, A. D'Innocenzo, and M. Levorato, "CADET: Control-Aware Dynamic Edge Computing for Real-Time Target Tracking in UAV Systems," *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1-5, Jun. 2023, doi: <https://doi.org/10.1109/icassp49357.2023.10095902>.
- [103] M. Janßen, T. Pfandzelter, M. Wang, and D. Bermbach, "Supporting UAVs with Edge Computing: A Review of Opportunities and Challenges," 2023. Accessed: Oct. 25, 2025. [Online]. Available: <https://arxiv.org/pdf/2310.11957.pdf>
- [104] W. Wu, F. Zhou, B. Wang, Q. Wu, C. Dong, and R. Q. Hu, "Unmanned Aerial Vehicle Swarm-Enabled Edge Computing: Potentials, Promising Technologies, and Challenges," *IEEE Wireless Communications*, pp. 1-17, 2022, doi: <https://doi.org/10.1109/mwc.103.2100286>.
- [105] V. Karjalainen, T. Hakala, A. George, N. Koivumäki, J. Suomalainen, and E. Honkavaara, "A DRONE SYSTEM FOR AUTONOMOUS MAPPING FLIGHTS INSIDE A FOREST - A FEASIBILITY STUDY AND FIRST RESULTS," *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XLVIII-1/W22023, pp. 597-603, Dec. 2023, doi: <https://doi.org/10.5194/isprs-archives-xlvi-1-w2-2023-597-2023>.
- [106] L. Watkins, D. Hamilton, K. Kornegay, and A. Rubin, "Triaging Autonomous Drone Faults By Simultaneously Assuring Autonomy and Security," pp. 1-6, Mar. 2021, doi: <https://doi.org/10.1109/ciss50987.2021.9400286>.
- [107] H. Xiao, P. Krisshnakumar, Pothuri, Jagadeswara P K V, P. Soni, E. Butcher, and S. Chowdhury, "An Open-source Hardware/Software Architecture and Supporting Simulation Environment to Perform Human FPV Flight Demonstrations for Unmanned Aerial Vehicle Autonomy," arXiv.org, 2024. <https://arxiv.org/abs/2407.06320v1> (accessed Oct. 25, 2025).
- [108] S. Mocanu, E. Slusanschi, and Marius Leordeanu, "Self-Supervised Learning to Fly using Efficient Semantic Segmentation and Metric Depth Estimation for Low-Cost Autonomous UAVs," 2025. <https://www.semanticscholar.org/paper/e629a5105e0ab2a54b861b7fc9500b93d1663a7c> (accessed Oct. 25, 2025).
- [109] T. Rohe, B. Böhm, M. Kölle, J. Stein, R. Müller, and C. Linnhoff-Popien, "Coconut Palm Tree Counting on Drone Images with Deep Object Detection and Synthetic Training Data," arXiv.org, 2024. <https://arxiv.org/abs/2412.11949> (accessed Oct. 25, 2025).

- [110] K. Doherty et al., "Leafy Spurge Dataset: Real-world Weed Classification Within Aerial Drone Imagery," arXiv.org, 2024. <https://arxiv.org/abs/2405.03702> (accessed Oct. 25, 2025).
- [111] G. Rizzoli, F. Barbato, M. Caligiuri, and P. Zanuttigh, "SynDrone -Multi-modal UAV Dataset for Urban Scenarios." Available: <https://arxiv.org/pdf/2308.10491.pdf>
- [112] W. Xu et al., "UAV-VisLoc: A Large-scale Dataset for UAV Visual Localization," 2024. Accessed: Oct. 25, 2025. [Online]. Available: <http://arxiv.org/pdf/2405.11936.pdf>
- [113] R. Samanta, B. Saha, and S. K. Ghosh, "TinyML-On-The-Fly: Real-Time Low-Power and Low-Cost MCU-Embedded On-Device Computer Vision for Aerial Image Classification," 2024 IEEE Space, Aerospace and Defence Conference (SPACE), pp. 194-198, Jul. 2024, doi: <https://doi.org/10.1109/space63117.2024.10667906>.
- [114] A. Setyanto et al., "Knowledge Distillation in Object Detection for Resource-Constrained Edge Computing," IEEE Access, pp. 1-1, 2025, doi: <https://doi.org/10.1109/access.2025.3534020>.
- [115] K. Zheng and X. Yang, "Design and Implementation of Lightweight Vision Transformer for Low-Power Edge Devices," pp. 1-4, Feb. 2025, doi: <https://doi.org/10.1109/iccece65250.2025.10985633>.
- [116] P. Jeevan and A. Sethi, "Which Backbone to Use: A Resource-efficient Domain Specific Comparison for Computer Vision," arXiv.org, 2024. <https://arxiv.org/abs/2406.05612>
- [117] C. Li, S. C. Sun, Z. Wei, Antonios Tsourdos, and W. Guo, "Scarce data driven deep learning of drones via generalized data distribution space," Neural computing & applications, vol. 35, no. 20, pp. 15095-15108, Apr. 2023, doi: <https://doi.org/10.1007/s00521-023-08522-z>.
- [118] Anis Koubâa, A. Ammar, M. Abdelkader, Yasser Alhabashi, and Lahouari Ghouti, "AERO: AI-Enabled Remote Sensing Observation with Onboard Edge Computing in UAVs," Remote Sensing, vol. 15, no. 7, pp. 1873-1873, Mar. 2023, doi: <https://doi.org/10.3390/rs15071873>.
- [119] S. Bertin, B. Levy, T. Gee, and P. Delmas, "Geomorphic Change Detection Using Cost-Effective Structure-from-Motion Photogrammetry: Evaluation of Direct Georeferencing from Consumer-Grade UAS at Orewa Beach (New Zealand)," Photogrammetric Engineering & Remote Sensing, vol. 86, no. 5, pp. 289-298, May 2020, doi: <https://doi.org/10.14358/pers.86.5.289>.
- [120] S. I. Jiménez-Jiménez, W. Ojeda-Bustamante, M. Marcial-Pablo, and J. Enciso, "Digital Terrain Models Generated with Low-Cost UAV Photogrammetry: Methodology and Accuracy," ISPRS International Journal of Geo-Information, vol. 10, no. 5, p. 285, Apr. 2021, doi: <https://doi.org/10.3390/ijgi10050285>.
- [121] J. Shan, Z. Li, D. Lercel, K. Tissue, J. Hupy, and J. Carpenter, "Democratizing photogrammetry: an accuracy perspective," Geo-spatial Information Science, vol. 26, no. 2, pp. 175-188, Apr. 2023, doi: <https://doi.org/10.1080/10095020.2023.2178336>.
- [122] A. Gressin, J. Vallet, and M. Bron, "ABOUT PHOTOGRAMMETRIC UAV-MAPPING: WHICH ACCURACY FOR WHICH APPLICATION?," The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. XLIII-B22020, pp. 1081-1089, Aug. 2020, doi: <https://doi.org/10.5194/isprs-archives-xliii-b2-2020-1081-2020>.
- [123] V. G. S, Pathipati Srihari, U. Kumar, Ashwin Nandagiri, Bethi Pardhasaradhi, and Linga Reddy Cenkeramaddi, "Real-Time UAV Altitude Estimation and Data Transmission via mmWave Radar and Edge Computing," pp. 1-6, Dec. 2024, doi: <https://doi.org/10.1109/indicon63790.2024.10958360>.
- [124] H. H. Jeong, J. W. Park, J. S. Kim, and C. U. Choi, "Assessing the Accuracy of Ortho-image using Photogrammetric Unmanned Aerial System," ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. XLI-B1, pp. 867-872, Jun. 2016, doi: <https://doi.org/10.5194/isprsarchives-xli-b1-867-2016>.
- [125] G. S. N. Perera and H. A. Nalani, "UAVS FOR A COMPLETE TOPOGRAPHIC SURVEY," The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. XLIII-B22022, pp. 441-447, May 2022, doi: <https://doi.org/10.5194/isprs-archives-XLIII-B2-2022-441-2022>.
- [126] S. J. Carter, N. C. Tsagkopoulos, G. Clawson, and C. Fox, "OpenScout: Open Source Hardware Mobile Robot," Journal of Open Hardware, vol. 7, no. 1, Dec. 2023, doi: <https://doi.org/10.5334/joh.54>.
- [127] M. Perez-Jimenez, P. Ramon-Soria, B. Arrue, and A. Ollero, "Hecatonquiros: Open-source hardware for aerial manipulation applications," International Journal of Advanced Robotic Systems, vol. 17, no. 2, p. 172988142092162, Mar. 2020, doi: <https://doi.org/10.1177/1729881420921622>.

- [128] R. Awate, A. Pardeshi, B. Singh, P. Malpani, and Dr. P. Khadkikar, "Comprehensive Survey of Autonomous Driving Systems and Advanced Technologies: Literature Review, Gap Analysis and Future Directions," *International Journal for Research in Applied Science and Engineering Technology*, vol. 12, no. 5, pp. 229-238, May 2024, doi: <https://doi.org/10.22214/ijraset.2024.61360>.
- [129] R. H. Jacobsen et al., "Design of an Autonomous Cooperative Drone Swarm for Inspections of Safety Critical Infrastructure," *Applied Sciences*, vol. 13, no. 3, p. 1256, Jan. 2023, doi: <https://doi.org/10.3390/app13031256>.
- [130] Y. Hong, S. Jung, S. Kim, and J. Cha, "Autonomous Mission of Multi-UAV for Optimal Area Coverage," *Sensors*, vol. 21, no. 7, p. 2482, Apr. 2021, doi: <https://doi.org/10.3390/s21072482>.
- [131] R. Alkadi, N. Alnuaimi, C. Y. Yeun, and A. Shoufan, "Blockchain Interoperability in Unmanned Aerial Vehicles Networks: State-of-the-Art and Open Issues," *IEEE Access*, vol. 10, pp. 14463-14479, 2022, doi: <https://doi.org/10.1109/ACCESS.2022.3145199>.
- [132] T. Lagkas, V. Argyriou, S. Bibi, and P. Sarigiannidis, "UAV IoT Framework Views and Challenges: Towards Protecting Drones as 'Things,'" *Sensors*, vol. 18, no. 11, p. 4015, Nov. 2018, doi: <https://doi.org/10.3390/s18114015>.
- [133] E. A. C. Moreno, J. F. U. Cruz, M. B. R. Esmeral, J. E. O. Rodriguez, M. A. R. Cubillos, and C. D. A. Rincón, "Methodology for the Implementation of an Intelligent Drone: From Hardware to Technical Deployment," *Revista de Gestão Social e Ambiental*, vol. 19, no. 4, p. e012047, Apr. 2025, doi: <https://doi.org/10.24857/rgsa.v19n4-129>.
- [134] E. A. C. Moreno, J. F. U. Cruz, M. B. R. Esmeral, J. E. O. Rodriguez, M. A. R. Cubillos, and C. D. A. Rincón, "Methodology for the Implementation of an Intelligent Drone: From Hardware to Technical Deployment," *Revista de Gestão Social e Ambiental*, vol. 19, no. 4, p. e012047, Apr. 2025, doi: <https://doi.org/10.24857/rgsa.v19n4-129>.
- [135] A. Brighente, M. Conti, M. Schotsman, and S. Sciancalepore, "Obfuscated Location Disclosure for Remote ID Enabled Drones." Accessed: Oct. 25, 2025. [Online]. Available: <http://arxiv.org/pdf/2407.14256.pdf>
- [136] "Scenario-Based Field Testing of Drone Missions," Arxiv.org, 2015. <https://arxiv.org/html/2407.08359v1> (accessed Oct. 25, 2025).
- [137] "DroneWiS: Automated Simulation Testing of small Unmanned Aerial Systems in Realistic Windy Conditions," Arxiv.org, 2024. <https://arxiv.org/html/2408.16559> (accessed Oct. 25, 2025).
- [138] S. Sarkar, S. Shafaei, T. S. Jones, and M. W. Totaro, "Secure Communication in Drone Networks: A Comprehensive Survey of Lightweight Encryption and Key Management Techniques," *Drones*, vol. 9, no. 8, p. 583, Aug. 2025, doi: <https://doi.org/10.3390/drones9080583>.
- [139] C. Zhang and A. M. Maga, "An Open-Source Photogrammetry Workflow for Reconstructing 3D Models," *Integrative organismal biology*, vol. 5, no. 1, Jan. 2023, doi: <https://doi.org/10.1093/iob/obad024>.
- [140] G. Muradás Odriozola, K. Pauly, S. Oswald, and D. Raymaekers, "Automating Ground Control Point Detection in Drone Imagery: From Computer Vision to Deep Learning," *Remote Sensing*, vol. 16, no. 5, p. 794, Jan. 2024, doi: <https://doi.org/10.3390/rs16050794>.
- [141] B. Yang et al., "Developing an Introductory UAV/Drone Mapping Training Program for Seagrass Monitoring and Research," *Drones*, vol. 4, no. 4, p. 70, Nov. 2020, doi: <https://doi.org/10.3390/drones4040070>.
- [142] csbeuii, "DEVELOPMENT AND UTILIZATION OF AERIAL PHOTOGRAMMETRY SURVEY USING NONMETRIC CAMERA DRONE WITH OPEN-SOURCE SOFTWARE: CASE STUDY ON TUGU DAM PROJECT - 8th ICSBE 2025," Uii.ac.id, Dec. 13, 2023. <https://icsbe.uui.ac.id/2023/12/13/development-and-utilization-of-aerial-photogrammetry-survey-using-nonmetric-camera-drone-with-open-source-software-case-study-on-tugu-dam-project/> (accessed Oct. 25, 2025).
- [143] M. Lehoczy and Z. Abdurakhmonov, "Present Software of photogrammetric processing of digital images," *E3S Web of Conferences*, vol. 227, p. 04001, 2021, doi: <https://doi.org/10.1051/e3sconf/202122704001>.
- [144] N. Wang, H. Yang, and C. D. Wang, "FinGPT: Instruction Tuning Benchmark for Open-Source Large Language Models in Financial Datasets," *arXiv.org*, Oct. 07, 2023. <https://arxiv.org/abs/2310.04793>
- [145] S. Glüge, M. Nyfeler, A. Aghaebrahimian, N. Ramagnano, and C. Schüpbach, "Robust Low-Cost Drone Detection and Classification Using Convolutional Neural Networks in Low SNR Environments," *IEEE*

- Journal of Radio Frequency Identification, vol. 8, pp. 821-830, 2024, doi: <https://doi.org/10.1109/jrfid.2024.3487303>.
- [146] J. P. Duffy et al., "Location, location, location: considerations when using lightweight drones in challenging environments," *Remote Sensing in Ecology and Conservation*, vol. 4, no. 1, pp. 7-19, Aug. 2017, doi: <https://doi.org/10.1002/rse2.58>.
- [147] C. H. Singh, V. Mishra, and K. Jain, "HIGH-RESOLUTION MAPPING OF FORESTED HILLS USING REAL-TIME UAV TERRAIN FOLLOWING," *ISPRS annals of the photogrammetry, remote sensing and spatial information sciences*, vol. X-1/W12023, pp. 665-671, Dec. 2023, doi: <https://doi.org/10.5194/isprs-annals-x-1-w1-2023-665-2023>.
- [148] M-Mahdi Naddaf-Sh, A. Lee, K. Yen, E. Amini, and I. Soltani, "Low-Cost Infrared Vision Systems for Improved Safety of Emergency Vehicle Operations Under Low-Visibility Conditions," *IEEE Access*, pp. 1-1, Jan. 2025, doi: <https://doi.org/10.1109/access.2025.3615117>.
- [149] K. Mason and D. Kelly, "Lo-MARVE: A Low Cost Autonomous Underwater Vehicle for Marine Exploration," pp. 421-426, Nov. 2024, doi: <https://doi.org/10.1109/iccma63715.2024.10843932>.
- [150] A. M. Knoblauch et al., "Bi-directional drones to strengthen healthcare provision: experiences and lessons from Madagascar, Malawi and Senegal," *BMJ Global Health*, vol. 4, no. 4, p. e001541, Jul. 2019, doi: <https://doi.org/10.1136/bmjgh-2019-001541>.
- [151] R. Reyes-Ramirez et al., "A Decentralized, Low-Cost Indoor Autonomous Aerial Swarm With Open-Source Intelligence," pp. 1-6, Apr. 2025, doi: <https://doi.org/10.1109/dcas65331.2025.11045462>.
- [152] R. Shrestha, J. Zevenbergen, U. S. Panday, B. Awasthi, and S. Karki, "REVISITING THE CURRENT UAV REGULATIONS IN NEPAL: A STEP TOWARDS LEGAL DIMENSION FOR UAVS EFFICIENT APPLICATION," *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XLII-5/W3, pp. 107-114, Dec. 2019, doi: <https://doi.org/10.5194/isprs-archives-xlii-5-w3-107-2019>.
- [153] K. Harding et al., "A mobile health model supporting Ethiopia's eHealth strategy," *Digital Medicine*, vol. 4, no. 2, p. 54, 2018, doi: https://doi.org/10.4103/digm.digm_10_18.
- [154] E. Öz, E. Heikkilä, and R. Tiusanen, "Development of an Organisational Certification Process for Specific Category Drone Operations," *Drones*, vol. 6, no. 10, p. 278, Sep. 2022, doi: <https://doi.org/10.3390/drones6100278>.
- [155] E. Öz, E. Heikkilä, and R. Tiusanen, "Development of an Organisational Certification Process for Specific Category Drone Operations," *Drones*, vol. 6, no. 10, p. 278, Sep. 2022, doi: <https://doi.org/10.3390/drones6100278>.
- [156] M. R. Bennion, R. Spencer, R. K. Moore, and R. Kenyon, "Digital capability, open-source use, and interoperability standards within the NHS in England: A survey of healthcare trusts (Preprint)," *JMIR Human Factors*, Sep. 2024, doi: <https://doi.org/10.2196/66398>.
- [157] O. A. Erin and O. A. Bamigboye, "Evaluation and Analysis of SDG Reporting: Evidence from Africa," *Journal of Accounting & Organizational Change*, vol. 18, no. 3, pp. 369-396, Jul. 2021, doi: <https://doi.org/10.1108/jaoc-02-2020-0025>.
- [158] A. Lockhart, A. While, S. Marvin, M. Kovacic, N. Odendaal, and C. Alexander, "Making space for drones: The contested reregulation of airspace in Tanzania and Rwanda," *Transactions of the Institute of British Geographers*, vol. 46, no. 4, pp. 850-865, May 2021, doi: <https://doi.org/10.1111/tran.12448>.
- [159] Q. Mokoena, I. Daniyan, K. Mpofu, and O. Abisuga, "TOWARDS STRATEGIC MANAGEMENT OF DRONE APPLICATION PROCESS AND REGULATION IN SOUTH AFRICA," *South African Journal of Industrial Engineering*, vol. 33, no. 4, 2022, doi: <https://doi.org/10.7166/33-4-2691>.
- [160] M. D. Watson and F. Gonzalez, "Design and Implementation of a Low-Cost Folding-Wing UAV for Model Rocket Deployment," *QUT ePrints (Queensland University of Technology)*, Mar. 2023, doi: <https://doi.org/10.1109/aero55745.2023.10115901>.
- [161] R. Cannizzaro, R. Howard, P. Lewinska, and L. Kunze, "Towards Probabilistic Causal Discovery, Inference & Explanations for Autonomous Drones in Mine Surveying Tasks," *arXiv.org*, 2023. <https://arxiv.org/abs/2308.10047> (accessed Dec. 26, 2025).

Appendix

Onboard Location Tracker Code

```
#include <SPI.h>
#include <SD.h>
#include <time.h>
#include <WiFi.h>
#include <WebServer.h>
#include <TinyGPS++.h>
#include <Wire.h>
#include <Adafruit_HMC5883_U.h>
#define VSPI_MISO 19
#define VSPI_MOSI 23
#define VSPI_SCLK 18
#define VSPI_CS 5
#define BUFFER_SIZE 10

#define RXD2 16
#define TXD2 17
#define GPS_BAUD 115200
#define I2C_SDA 21
#define I2C_SCL 22

const char* ssid = "██████";
const char* password = "██████";

File dataFile;
String currentFilename = "";
uint32_t currentID = 0;
bool sdInitialized = false;
bool loggingActive = false;
String ramBuffer[BUFFER_SIZE];
uint8_t bufferPos = 0;

unsigned long loggingStartTime = 0;
uint32_t logsWritten = 0;

WebServer server(80);
bool wifiConnected = false;
bool serverActive = false;

const float MAG_X_OFFSET = -50.41;
const float MAG_Y_OFFSET = -39.64;
const float MAG_X_SCALE = 1.0000;
const float MAG_Y_SCALE = 0.7627;
float declinationAngle = -1.4024;

TinyGPSPlus gps;
HardwareSerial gpsSerial(2);
Adafruit_HMC5883_Unified mag = Adafruit_HMC5883_Unified(12345);
bool gpsLocked = false;
bool realTimeMode = false;

float getCompassHeading() {
    sensors_event_t event;
    mag.getEvent(&event);
    float xCorrected = event.magnetic.x - MAG_X_OFFSET;
```

```

float yCorrected = event.magnetic.y - MAG_Y_OFFSET;
xCorrected *= MAG_X_SCALE;
yCorrected *= MAG_Y_SCALE;
float heading = atan2(-yCorrected, xCorrected);
heading += declinationAngle;

if (heading < 0) {
    heading += 2 * PI;
}

if (heading > 2 * PI) {
    heading -= 2 * PI;
}

return heading * 180 / PI;
}

String getCardinalDirection(float heading) {
    const char* directions[] = {"N", "NNE", "NE", "ENE", "E", "ESE", "SE",
    "SSE", "S", "SSW", "SW", "WSW", "W", "WNW", "NW", "NNW"};
    int index = int((heading + 11.25) / 22.5) % 16;
    return String(directions[index]);
}

String twoDigits(uint32_t v) {
    if (v < 10) return "0" + String(v);
    return String(v);
}

String gpsUtcIso8601() {
    if (gps.date.isValid() && gps.time.isValid()) {
        uint16_t yr = gps.date.year();
        uint8_t mo = gps.date.month();
        uint8_t dy = gps.date.day();
        uint8_t hr = gps.time.hour();
        uint8_t mi = gps.time.minute();
        uint8_t se = gps.time.second();
        return String(yr) + "-" + twoDigits(mo) + "-" + twoDigits(dy) + "T" +
        twoDigits(hr) + ":" + twoDigits(mi) + ":" + twoDigits(se) + "Z";
    }
    return "No UTC";
}

String getFormattedTime() {
    if (gps.date.isValid() && gps.time.isValid()) {
        uint16_t yr = gps.date.year();
        uint8_t mo = gps.date.month();
        uint8_t dy = gps.date.day();
        uint8_t hr = gps.time.hour();
        uint8_t mi = gps.time.minute();
        uint8_t se = gps.time.second();

        String result = twoDigits(mo) + twoDigits(dy) + twoDigits(hr) +
        twoDigits(mi) + String(yr) + "." + twoDigits(se);
    }
}

```

```

    return result;
}
return "No GPS Time";
}

void printHelp() {
    Serial.println("\n=== Available Commands ===");
    Serial.println("help          - Display this help message");
    Serial.println("start        - Start GPS data logging to SD card");
    Serial.println("stop         - Stop GPS data logging");
    Serial.println("stat         - Display SD card statistics");
    Serial.println("list         - List all CSV files on SD card");
    Serial.println("get_time     - Get current GPS time (MMDDhhmmYYYY.ss
format)");
    Serial.println("real_time_start - Start real-time GPS data display");
    Serial.println("real_time_stop - Stop real-time GPS data display");
    Serial.println("sys          - Show GPS system status and satellite
info");
    Serial.println("connect      - Connect to WiFi and start web
server");
    Serial.println("disconnect   - Disconnect from WiFi");
    Serial.println("=====\n");
}

void printSystemStatus() {
    Serial.println("\n=== GPS System Status ===");
    Serial.print("GPS Lock: ");
    Serial.println(gpsLocked ? "YES" : "NO");

    Serial.print("Satellites: ");
    if (gps.satellites.isValid()) {
        Serial.println(gps.satellites.value());
    } else {
        Serial.println("N/A");
    }

    Serial.print("HDOP: ");
    if (gps.hdop.isValid()) {
        Serial.println(gps.hdop.hdop(), 2);
    } else {
        Serial.println("N/A");
    }

    Serial.print("Update Rate: ");
    Serial.println("Using saved flash config");

    Serial.print("Baud Rate: ");
    Serial.println(GPS_BAUD);

    Serial.print("Location Valid: ");
    Serial.println(gps.location.isValid() ? "YES" : "NO");

    Serial.print("Date/Time Valid: ");
    Serial.println((gps.date.isValid() && gps.time.isValid()) ? "YES" :
"NO");
}

```

```

Serial.print("Age of Fix: ");
if (gps.location.isValid()) {
    Serial.print(gps.location.age());
    Serial.println(" ms");
} else {
    Serial.println("N/A");
}

Serial.print("Characters Processed: ");
Serial.println(gps.charsProcessed());

Serial.print("Sentences with Fix: ");
Serial.println(gps.sentencesWithFix());

Serial.print("Failed Checksums: ");
Serial.println(gps.failedChecksum());

Serial.println("=====\n");
}

void displayRealTimeData() {
    if (!gps.location.isValid()) {
        Serial.println("Waiting for valid location data...");
        return;
    }

    float magneticHeading = getCompassHeading();
    String cardinalDir = getCardinalDirection(magneticHeading);

    Serial.print("MAG: ");
    Serial.print(magneticHeading, 2);
    Serial.println(" DEG");

    Serial.print("LAT: ");
    Serial.println(gps.location.lat(), 8);

    Serial.print("LON: ");
    Serial.println(gps.location.lng(), 8);

    Serial.print("DIR: ");
    Serial.println(cardinalDir);

    Serial.print("VEL: ");
    if (gps.speed.isValid()) {
        Serial.print(gps.speed.kmph(), 2);
        Serial.println(" km/h");
    } else {
        Serial.println("N/A");
    }

    Serial.print("ALT: ");
    if (gps.altitude.isValid()) {
        Serial.print(gps.altitude.meters(), 2);
        Serial.println(" m");
    } else {
        Serial.println("N/A");
    }
}

```



```

    Serial.println("---");
}

void setup() {
    Serial.begin(230400);
    delay(2000);
    Serial.println("ESP32 SD Card Data Logger with GPS");

    Wire.begin(I2C_SDA, I2C_SCL);

    if (!mag.begin()) {
        Serial.println("ERROR: HMC5883L not detected!");
    }

    gpsSerial.begin(GPS_BAUD, SERIAL_8N1, RXD2, TXD2);
    delay(500);
    Serial.println("GPS initialized with flash-saved configuration");

    SPI.begin(VSPI_SCLK, VSPI_MISO, VSPI_MOSI, VSPI_CS);
    delay(100);

    Serial.println("Attempting SD card initialization...");
    if (!SD.begin(VSPI_CS)) {
        Serial.println("SD CARD INITIALIZATION FAILED!");
        while(1) { delay(1000); }
    }
    Serial.println("PASS SD card initialized successfully!");
    sdInitialized = true;

    uint8_t cardType = SD.cardType();
    if(cardType == CARD_NONE){
        Serial.println("No SD card detected!");
        while(1) { delay(1000); }
    }
    Serial.print("SD Card Type: ");
    if(cardType == CARD_MMC) Serial.println("MMC");
    else if(cardType == CARD_SD) Serial.println("SDSC");
    else if(cardType == CARD_SDHC) Serial.println("SDHC");
    else Serial.println("UNKNOWN");
    Serial.printf("SD Card Size: %lluMB\n", SD.cardSize() / (1024 * 1024));
    Serial.printf("Total space: %lluMB\n", SD.totalBytes() / (1024 * 1024));
    Serial.printf("Used space: %lluMB\n", SD.usedBytes() / (1024 * 1024));
    Serial.println("\nWaiting for GPS lock...");
}

void loop() {
    while (gpsSerial.available() > 0) {
        gps.encode(gpsSerial.read());
    }
    if (!gpsLocked) {
        if (gps.location.isValid() && gps.date.isValid() && gps.time.isValid()
        && gps.satellites.value() >= 4) {
            gpsLocked = true;

```

```

    Serial.println("\nGPS LOCK SUCCESS");
    Serial.print("SAT COUNT: ");
    Serial.println(gps.satellites.value());
    Serial.println("Type 'help' to see available commands");
} else {
    static unsigned long lastPrint = 0;
    if (millis() - lastPrint > 2000) {
        Serial.println("NO GPS LOCK");
        lastPrint = millis();
    }
    delay(100);
    return;
}

}

if (realTimeMode && gps.location.isUpdated()) {
    displayRealTimeData();
}

serialCommandListener();
if (serverActive) server.handleClient();
if (loggingActive && sdInitialized) {
    if (!dataFile) {
        dataFile = SD.open(currentFilename, FILE_APPEND);
        if (!dataFile) {
            Serial.println("FAIL ERROR: Failed to open file for logging!");
            loggingActive = false;
            return;
        }
    }
    if (gps.location.isUpdated()) {
        double lat = gps.location.lat();
        double lon = gps.location.lng();
        float magneticHeading = getCompassHeading();
        String utcTime = gpsUtcIso8601();
        ramBuffer[bufferPos] = csv_gen_line(currentID, utcTime, lat, lon,
magneticHeading);
        bufferPos++;
        currentID++;
        logsWritten++;
        if (bufferPos >= BUFFER_SIZE) {
            for (uint8_t i=0; i<BUFFER_SIZE; ++i)
dataFile.println(ramBuffer[i]);
            dataFile.flush();
            bufferPos = 0;
        }
    }
    delay(100);
} else {
    if (dataFile) {
        for (uint8_t i=0; i<bufferPos; ++i) dataFile.println(ramBuffer[i]);
        dataFile.flush();
        dataFile.close();
        dataFile = File();
        bufferPos = 0;
    }
    delay(20);
}
}

```

```
}
```

```
void serialCommandListener() {
    if (Serial.available()) {
        String cmd = Serial.readStringUntil('\n');
        cmd.trim();
        if (cmd.equalsIgnoreCase("help")) {
            printHelp();
        } else if (cmd.equalsIgnoreCase("sys")) {
            printSystemStatus();
        } else if (cmd.equalsIgnoreCase("real_time_start")) {
            if (!gpsLocked) {
                Serial.println("ERROR: GPS not locked! Cannot start real-time
display.");
                return;
            }
            realTimeMode = true;
            Serial.println("Real-time display started. Type 'real_time_stop' to
stop.");
        } else if (cmd.equalsIgnoreCase("real_time_stop")) {
            realTimeMode = false;
            Serial.println("Real-time display stopped.");
        } else if (cmd.equalsIgnoreCase("start")) {
            if (!gpsLocked) {
                Serial.println("ERROR: GPS not locked! Cannot start logging.");
                return;
            }
        }
        if (!loggingActive) {
            currentFilename = getNextFilename();
            if (SD.exists(currentFilename)) SD.remove(currentFilename);
            File file = SD.open(currentFilename, FILE_WRITE);
            if (file) {
                file.println("ID,UTC_TIME,LAT,LON,MAG_HEAD");
                file.close();
                delay(50);
                String displayName = currentFilename;
                if (displayName.startsWith("/")) displayName =
displayName.substring(1);
                Serial.println(displayName);
            } else {
                Serial.println("FAIL Error creating file!");
                return;
            }
            currentID = 0;
            logsWritten = 0;
            loggingStartTime = millis();
            loggingActive = true;
        }
        } else if (cmd.equalsIgnoreCase("get_time")) {
            Serial.println(getFormattedTime());
        } else if (cmd.equalsIgnoreCase("stop")) {
            if (loggingActive) {
                loggingActive = false;
                unsigned long duration = (millis() - loggingStartTime) / 1000;
                Serial.println("Logging stopped.");
                Serial.print("Duration: ");
                Serial.print(duration);
            }
        }
    }
}
```

```

        Serial.println(" seconds");
        Serial.print("Logs written: ");
        Serial.println(logsWritten);
    }
} else if (cmd.equalsIgnoreCase("stat")) {
    printCardStats();
} else if (cmd.equalsIgnoreCase("list")) {
    listCSVFiles();
} else if (cmd.equalsIgnoreCase("connect")) {
    connectWiFi();
} else if (cmd.equalsIgnoreCase("disconnect")) {
    disconnectWiFi();
}
}
}

void connectWiFi() {
    if (wifiConnected) {
        Serial.println("Already connected to WiFi!");
        Serial.print("IP Address: ");
        Serial.println(WiFi.localIP());
        return;
    }

    Serial.println("Connecting to WiFi");
    Serial.print("SSID: ");
    Serial.println(ssid);
    WiFi.mode(WIFI_STA);
    WiFi.begin(ssid, password);
    int attempts = 0;
    while (WiFi.status() != WL_CONNECTED && attempts < 20) {
        delay(500);
        Serial.print(".");
        attempts++;
    }

    if (WiFi.status() == WL_CONNECTED) {
        wifiConnected = true;
        Serial.println("WiFi connected successfully!");
        Serial.print("IP Address: ");
        Serial.println(WiFi.localIP());
        setupWebServer();
        server.begin();
        serverActive = true;
        Serial.println("Web server started!");
        Serial.println("Access the file browser at: http://" +
            WiFi.localIP().toString());
    } else {
        Serial.println("Failed to connect to WiFi!");
    }
}

void disconnectWiFi() {
    if (!wifiConnected) {
        Serial.println("Not connected to WiFi!");
        return;
    }
}

```

```

Serial.println("Disconnecting WiFi");
if (serverActive) {
    server.stop();
    serverActive = false;
    Serial.println("Web server stopped.");
}
WiFi.disconnect(true);
WiFi.mode(WIFI_OFF);
wifiConnected = false;
Serial.println("WiFi disconnected.");
}

void setupWebServer() {
    server.on("/", HTTP_GET, []() {
        String html = generateFileListHTML();
        server.send(200, "text/html", html);
    });
    server.on("/download", HTTP_GET, []() {
        if (server.hasArg("file")) {
            String filename = server.arg("file");
            if (filename.indexOf("..") != -1 || !filename.endsWith(".csv")) {
                server.send(403, "text/plain", "Access denied");
                return;
            }
            if (!filename.startsWith("/")) {
                filename = "/" + filename;
            }
            if (!SD.exists(filename)) {
                server.send(404, "text/plain", "File not found");
                return;
            }
            File file = SD.open(filename, FILE_READ);
            if (!file) {
                server.send(500, "text/plain", "Failed to open file");
                return;
            }
            String header = "attachment; filename=\"" + filename.substring(1) +
"\n";
            server.sendHeader("Content-Disposition", header);
            server.streamFile(file, "text/csv");
            file.close();
        } else {
            server.send(400, "text/plain", "Missing file parameter");
        }
    });
    server.onNotFound([]() {
        server.send(404, "text/plain", "404: Not Found");
    });
}

String generateFileListHTML() {
    String html = "<!DOCTYPE html><html><head>";
    html += "<meta charset='UTF-8'>";
    html += "<meta name='viewport' content='width=device-width, initial-
scale=1.0'>";
    html += "<title>ESP32 SD Card File Browser</title>";
    html += "<style>";

```

```

html += "body { font-family: Arial, sans-serif; margin: 20px; background-
color: #f0f0f0; }";
html += "h1 { color: #333; }";
html += ".container { background-color: white; padding: 20px; border-
radius: 8px; box-shadow: 0 2px 4px rgba(0,0,0,0.1); }";
html += "table { width: 100%; border-collapse: collapse; margin-top:
20px; }";
html += "th, td { padding: 12px; text-align: left; border-bottom: 1px
solid #ddd; }";
html += "th { background-color: #4CAF50; color: white; }";
html += "tr:hover { background-color: #f5f5f5; }";
html += ".download-btn { background-color: #008CBA; color: white;
padding: 8px 16px; text-decoration: none; border-radius: 4px; display:
inline-block; }";
html += ".download-btn:hover { background-color: #006B8F; }";
html += ".info { background-color: #e7f3fe; padding: 10px; border-left:
4px solid #2196F3; margin-bottom: 20px; }";
html += ".refresh-btn { background-color: #4CAF50; color: white; padding:
10px 20px; text-decoration: none; border-radius: 4px; display: inline-
block; margin-top: 10px; }";
html += ".refresh-btn:hover { background-color: #45a049; }";
html += "</style></head><body>";
html += "<div class='container'>";
html += "<h1>ESP32 SD Card File Browser</h1>";
html += "<div class='info'>";
html += "<strong>SD Card Info:</strong><br>";
html += "Total Space: " + String(SD.totalBytes() / (1024 * 1024)) + "
MB<br>";
html += "Used Space: " + String(SD.usedBytes() / (1024 * 1024)) + "
MB<br>";
html += "Free Space: " + String((SD.totalBytes() - SD.usedBytes()) /
(1024 * 1024)) + " MB";
html += "</div>";
html += "<table>";
html +=
"<tr><th>Filename</th><th>Size</th><th>Lines</th><th>Action</th></tr>";
File root = SD.open("/");
if (root) {
  File file = root.openNextFile();
  int fileCount = 0;
  while (file) {
    String filename = String(file.name());
    if (filename.endsWith(".csv") && isValidLogFilename(filename)) {
      size_t fileSize = file.size();
      uint32_t csvLength = countCSVLines(filename);
      String displayName = filename;
      if (displayName.startsWith("/")) displayName =
displayName.substring(1);
      html += "<tr>";
      html += "<td>" + displayName + "</td>";
      html += "<td>" + formatFileSize(fileSize) + "</td>";
      html += "<td>" + String(csvLength) + "</td>";
      html += "<td><a class='download-btn' href='/download?file=" +
displayName + "'>Download</a></td>";
      html += "</tr>";
      fileCount++;
    }
    file = root.openNextFile();
  }
}

```

```

    }
    root.close();
    if (fileCount == 0) html += "<tr><td colspan='4' style='text-align: center; color: #999;'>No CSV files found</td></tr>";
}
html += "</table>";
html += "<a class='refresh-btn' href='/'>Refresh</a>";
html += "</div></body></html>";
return html;
}

void printCardStats() {
    Serial.println("SD Card Statistics");
    uint8_t cardType = SD.cardType();
    Serial.print("Card Type: ");
    if(cardType == CARD_MMC) Serial.println("MMC");
    else if(cardType == CARD_SD) Serial.println("SDSC");
    else if(cardType == CARD_SDHC) Serial.println("SDHC");
    else Serial.println("UNKNOWN");
    uint64_t totalBytes = SD.totalBytes() / (1024 * 1024);
    uint64_t usedBytes = SD.usedBytes() / (1024 * 1024);
    uint64_t freeBytes = totalBytes - usedBytes;
    Serial.printf("Total Space: %lluMB\n", totalBytes);
    Serial.printf("Used Space: %lluMB\n", usedBytes);
    Serial.printf("Free Space: %lluMB\n", freeBytes);
}

void listCSVFiles() {
    Serial.println("CSV File List");
    Serial.println("ID\tName\t\t\t\t\tSize\tLength");
    Serial.println("-----");
    File root = SD.open("/");
    if (!root) {
        Serial.println("ERROR: Failed to open root directory");
        return;
    }
    int fileCount = 0;
    File file = root.openNextFile();
    while (file) {
        String filename = String(file.name());
        if (filename.endsWith(".csv") && isValidLogFilename(filename)) {
            size_t fileSize = file.size();
            uint32_t csvLength = countCSVLines(filename);
            Serial.print(fileCount);
            Serial.print("\t");
            Serial.print(filename);
            Serial.print("\t");
            if (filename.length() < 24) Serial.print("\t");
            Serial.print(formatFileSize(fileSize));
            Serial.print("\t");
            Serial.println(csvLength);
            fileCount++;
        }
        file = root.openNextFile();
    }
    root.close();
}

```

```

    Serial.println("-----
-");
    Serial.print("Total files: ");
    Serial.println(fileCount);
}

bool isValidLogFilename(String filename) {
    if (filename.startsWith("/")) filename = filename.substring(1);
    if (!filename.endsWith(".csv")) return false;
    int firstUnderscore = filename.indexOf('_');
    int secondUnderscore = filename.indexOf('_', firstUnderscore + 1);
    if (firstUnderscore == -1 || secondUnderscore == -1) return false;
    String afterSecond = filename.substring(secondUnderscore + 1);
    afterSecond.replace(".csv", "");
    if (afterSecond.length() != 3) return false;
    for (unsigned int i = 0; i < afterSecond.length(); i++) {
        if (!isDigit(afterSecond.charAt(i))) return false;
    }
    return true;
}

uint32_t countCSVLines(const String& pathIn) {
    String path = pathIn;
    if (!path.startsWith("/")) path = "/" + path;
    File f = SD.open(path, FILE_READ);
    if (!f) return 0;
    uint32_t lines = 0;
    bool headerSkipped = false;
    bool inContent = false;
    while (f.available()) {
        char c = f.read();
        if (c == '\r') continue;
        else if (c == '\n') {
            if (!headerSkipped) headerSkipped = true;
            else if (inContent) lines++;
            inContent = false;
        } else inContent = true;
    }
    if (inContent && headerSkipped) lines++;
    f.close();
    return lines;
}

String formatFileSize(size_t bytes) {
    if (bytes < 1024) return String(bytes) + "B";
    else if (bytes < 1024 * 1024) return String(bytes / 1024.0, 1) + "kB";
    else return String(bytes / (1024.0 * 1024.0), 1) + "MB";
}

String csv_gen_line(uint32_t id, String utcTime, double lat, double lon,
double mag) {
    return String(id) + "," + utcTime + "," + String(lat, 16) + "," +
String(lon, 16) + "," + String(mag, 8);
}

String getNextFilename() {
    int highestSerial = 0;
    File root = SD.open("/");

```



```

if (!root) {
    Serial.println("Failed to open root directory");
    return "/31OCT2025_000000AM_001.csv";
}
File file = root.openNextFile();
while (file) {
    String filename = String(file.name());
    if (filename.endsWith(".csv")) {
        int lastUnderscore = filename.lastIndexOf('_');
        if (lastUnderscore != -1) {
            String serialPart = filename.substring(lastUnderscore + 1);
            serialPart.replace(".csv", "");
            int serial = serialPart.toInt();
            if (serial > highestSerial) highestSerial = serial;
        }
    }
    file = root.openNextFile();
}
root.close();
int nextSerial = highestSerial + 1;
String datePrefix = getCurrentDatePrefix();
char serialStr[4];
sprintf(serialStr, "%03d", nextSerial);
String filename = "/" + datePrefix + "_" + String(serialStr) + ".csv";
return filename;
}

String getCurrentDatePrefix() {
    if (gps.date.isValid() && gps.time.isValid()) {
        const char* months[] = {"JAN", "FEB", "MAR", "APR", "MAY", "JUN",
"JUL", "AUG", "SEP", "OCT", "NOV", "DEC"};
        uint16_t yr = gps.date.year();
        uint8_t mo = gps.date.month();
        uint8_t dy = gps.date.day();
        uint8_t hr = gps.time.hour();
        uint8_t mi = gps.time.minute();
        uint8_t se = gps.time.second();
        String dateStr = String(dy) + months[mo - 1] + String(yr);
        char timeStr[10];
        sprintf(timeStr, "%02d%02d%02ds", hr, mi, se, "UTC");
        return dateStr + "_" + String(timeStr);
    }
    String dateStr = "31OCT2025";
    time_t now = millis() / 1000;
    unsigned long seconds = now % 86400;
    int hours = (seconds / 3600) % 12;
    if (hours == 0) hours = 12;
    int minutes = (seconds % 3600) / 60;
    int secs = seconds % 60;
    String ampm = ((seconds / 3600) < 12) ? "AM" : "PM";
    char timeStr[10];
    sprintf(timeStr, "%02d%02d%02ds", hours, minutes, secs, ampm.c_str());
    return dateStr + "_" + String(timeStr);
}

```

Drone UI Code

```
Add-Type -Name Window -Namespace Console -MemberDefinition '  
[DllImport("Kernel32.dll")]  
public static extern IntPtr GetConsoleWindow();  
  
[DllImport("user32.dll")]  
public static extern bool ShowWindow(IntPtr hWnd, Int32 nCmdShow);  
'  
  
$consolePtr = [Console.Window]::GetConsoleWindow()  
[void][Console.Window]::ShowWindow($consolePtr, 0)  
Add-Type -AssemblyName System.Windows.Forms  
Add-Type -AssemblyName System.Drawing  
[System.Windows.Forms.Application]::EnableVisualStyles()  
Add-Type -TypeDefinition @"  
using System;  
using System.Runtime.InteropServices;  
  
public class AppID {  
    [DllImport("shell32.dll", SetLastError = true)]  
    public static extern int  
SetCurrentProcessExplicitAppUserModelID([MarshalAs(UnmanagedType.LPWStr)]  
string AppID);  
}  
"@  
[AppID]::SetCurrentProcessExplicitAppUserModelID("BRACU.FYDP.G6.2025")  
$scriptPath = Split-Path -Parent $MyInvocation.MyCommand.Definition  
$form = New-Object System.Windows.Forms.Form  
$form.Text = 'DroneLAB - FYDP Group 06'  
$form.Size = New-Object System.Drawing.Size(1100, 700)  
$form.StartPosition = 'CenterScreen'  
$form.FormBorderStyle = 'FixedSingle'  
$form.MaximizeBox = $false  
$iconPath = Join-Path $scriptPath 'logo.ico'  
if (Test-Path $iconPath) {  
    $form.Icon = New-Object System.Drawing.Icon($iconPath)  
}  
$titleLabel = New-Object System.Windows.Forms.Label  
$titleLabel.Text = 'DroneLAB by FYDP Group 06'  
$titleLabel.Font = New-Object System.Drawing.Font("Segoe UI", 36,  
[System.Drawing.FontStyle]::Bold)  
$titleLabel.AutoSize = $true  
$titleLabel.Location = New-Object System.Drawing.Point(0, 10)  
$titleLabel.TextAlign = 'MiddleCenter'  
$titleLabel.Left = [int](($form.ClientSize.Width -  
$titleLabel.PreferredWidth) / 2)  
$form.Controls.Add($titleLabel)  
$warningLabel = New-Object System.Windows.Forms.Label  
$warningLabel.Text = 'Device is Disarmed for Demonstration Purposes'  
$warningLabel.Font = New-Object System.Drawing.Font("Segoe UI", 12,  
[System.Drawing.FontStyle]::Regular)  
$warningLabel.ForeColor = [System.Drawing.Color]::Red  
$warningLabel.AutoSize = $true  
$warningLabel.Location = New-Object System.Drawing.Point(0, 80)
```

```

$warningLabel.TextAlign = 'MiddleCenter'
$warningLabel.Left = [int](($form.ClientSize.Width -
$warningLabel.PreferredWidth) / 2)
$form.Controls.Add($warningLabel)
$imageBox = New-Object System.Windows.Forms.PictureBox
$imageBox.Size = New-Object System.Drawing.Size(500, 500)
$imageBox.Location = New-Object System.Drawing.Point(552, 122)
$imageBox.BorderStyle = 'FixedSingle'
$imageBox.SizeMode = 'Zoom'
$logBox = New-Object System.Windows.Forms.TextBox
$logBox.Multiline = $true
$logBox.ScrollBars = 'Vertical'
$logBox.Size = New-Object System.Drawing.Size(500, 150)
$logBox.Location = New-Object System.Drawing.Point(22, 472)
$logBox.ReadOnly = $true
$logBox.Font = New-Object System.Drawing.Font("Consolas", 8)
$logBox.BackColor = [System.Drawing.Color]::White
$form.Controls.Add($logBox)

$UserName = '██████'
$ServerAddress = '██████'
$ThresholdSeconds = 1

function Get-TimeDriftMessage {
    param (
        [double]$Seconds
    )
    $status = if ($Seconds -ge 0) { "ahead of" } else { "behind" }
    $totalSeconds = [math]::Abs($Seconds)
    $secondsInMinute = 60
    $secondsInHour = 60 * $secondsInMinute
    $secondsInDay = 24 * $secondsInHour
    $secondsInMonth = 30 * $secondsInDay
    $secondsInYear = 365 * $secondsInDay
    $years = [math]::Floor($totalSeconds / $secondsInYear)
    $totalSeconds -= $years * $secondsInYear

    $months = [math]::Floor($totalSeconds / $secondsInMonth)
    $totalSeconds -= $months * $secondsInMonth

    $days = [math]::Floor($totalSeconds / $secondsInDay)
    $totalSeconds -= $days * $secondsInDay

    $hours = [math]::Floor($totalSeconds / $secondsInHour)
    $totalSeconds -= $hours * $secondsInHour

    $minutes = [math]::Floor($totalSeconds / $secondsInMinute)
    $secondsRemaining = [math]::Round($totalSeconds - ($minutes *
$secondsInMinute), 2)
    $parts = @()
    $parts = @()

```

```

        if ($years -gt 0) {
            $parts += "$years year" + $(if ($years -gt 1) { "s" } else { "" })
        }
        if ($months -gt 0) {
            $parts += "$months month" + $(if ($months -gt 1) { "s" } else
{ "" })
        }
        if ($days -gt 0) {
            $parts += "$days day" + $(if ($days -gt 1) { "s" } else { "" })
        }
        if ($hours -gt 0) {
            $parts += "$hours hour" + $(if ($hours -gt 1) { "s" } else { "" })
        }
        if ($minutes -gt 0) {
            $parts += "$minutes minute" + $(if ($minutes -gt 1) { "s" } else
{ "" })
        }
        if ($secondsRemaining -gt 0 -or $parts.Count -eq 0) {
            $parts += "$secondsRemaining second" + $(if ($secondsRemaining -ne
1) { "s" } else { "" })
        }
        if ($parts.Count -gt 1) {
            $timeString = ($parts[0..($parts.Count - 2)] -join ", ") + " and "
+ $parts[-1]
        }
        else {
            $timeString = $parts[0]
        }

        return "The onboard flight computer is $status $timeString from local
time."
    }
}

```

```

function Get-RemoteTimeDifference {
    param (
        [Parameter(Mandatory = $true)]
        [string]$UserName,

        [Parameter(Mandatory = $true)]
        [string]$ServerAddress
    )
    $remoteTimeStr = ssh "$UserName@$ServerAddress" "date
'+%Y-%m-%d %I:%M:%S %p'"

    if (-not $remoteTimeStr) {
        Write-Error "Failed to retrieve remote time."
        return $null
    }

    try {
        $remoteTime = [datetime]::ParseExact($remoteTimeStr.Trim(), "yyyy-
MM-dd hh:mm:ss tt", $null)
    }
    catch {
        Write-Error "Failed to parse remote time: $_"
    }
}

```

```

        return $null
    }
    $localTime = (Get-Date).ToUniversalTime()
    $diffSeconds = ($remoteTime.ToUniversalTime() -
$localTime).TotalSeconds
    return $diffSeconds
}

function Update-RemoteHostTime {
    param (
        [Parameter(Mandatory = $true)]
        [string]$UserName,

        [Parameter(Mandatory = $true)]
        [string]$ServerAddress,

        [int]$OffsetSeconds = 6,
        [string]$RemoteWorkspace = "/home/USER_PROFILE/workspace",
        [string]$VenvPath = ".venv/bin/activate"
    )

    $future = (Get-Date).AddSeconds($OffsetSeconds)
    $tmp_tim = $future.ToString("dd MMM yyyy_h_m_s_tt")
    $formatted_time = "set_time=$tmp_tim"

    $cmd_set_time = "cd $RemoteWorkspace; source $VenvPath; python3 -m
flying_machine $formatted_time; deactivate;"
    $cmd_update_time = "cd $RemoteWorkspace; source $VenvPath; python3 -m
flying_machine update_time; deactivate;"

    try {
        ssh "$UserName@$ServerAddress" "$cmd_set_time"
        ssh "$UserName@$ServerAddress" "$cmd_update_time"
        Write-Output "Remote host time updated successfully."
    }
    catch {
        Write-Error "Failed to update remote host time: $_"
    }
}

function TimeSync {
    $diffSeconds = Get-RemoteTimeDifference -UserName $UserName -
ServerAddress $ServerAddress

    if ($null -ne $diffSeconds) {
        $driftMessage = Get-TimeDriftMessage -Seconds $diffSeconds
        Write-Output $driftMessage

        $absDiffSeconds = [math]::Abs($diffSeconds)
        Write-Output "Time difference is $absDiffSeconds seconds."

        if ($absDiffSeconds -gt $ThresholdSeconds) {
            Write-Output "Difference exceeds $ThresholdSeconds second(s).
Updating remote host time..."

```

```

        Update-RemoteHostTime -UserName $UserName -ServerAddress
$ServerAddress
    }
    else {
        Write-Output "Remote host time is in sync. No update needed."
    }
}
else {
    Write-Error "Could not retrieve time difference."
}
}

function Process-LatestCsv {
    param(
        [Parameter(Mandatory = $true)]
        [int]$LogID
    )
    $latestCsv = Get-ChildItem -Filter "*.csv" | Sort-Object LastWriteTime
-Descending | Select-Object -First 1

    if (-not $latestCsv) {
        Write-Error "No CSV files found in current directory"
        return
    }

    $csvFileName = $latestCsv.Name
    Write-Host "Processing: $csvFileName" -ForegroundColor Green
    $desktopPath = "$env:USERPROFILE\Desktop"
    Get-ChildItem -Path $desktopPath -Filter "*.csv" | Remove-Item -Force
    $baseName = [System.IO.Path]::GetFileNameWithoutExtension($csvFileName)
    $simplifiedFileName = "${baseName}_SIMPLIFIED.csv"
    $outputPath = Join-Path $desktopPath $simplifiedFileName
    $sourceData = Import-Csv $csvFileName
    $csvLines = [System.Collections.Generic.List[string]]::new()
    $csvLines.Add("class,lat,lon")

    foreach ($row in $sourceData) {
        $csvLines.Add("0,$($row.LAT),$($row.LON)")
    }

    [System.IO.File]::WriteAllLines($outputPath, $csvLines)
    Write-Host "Created: $outputPath ($($sourceData.Count) rows)" -
ForegroundColor Green
    $scriptRoot = $PSScriptRoot
    $rawDumpPath = Join-Path $scriptRoot "raw_dump"
    $logDirPath = Join-Path $rawDumpPath $LogID
    if (Test-Path $logDirPath) {
        Remove-Item -Path $logDirPath -Recurse -Force
        Write-Host "Removed existing directory: $logDirPath" -
ForegroundColor Yellow
    }

    New-Item -Path $logDirPath -ItemType Directory -Force | Out-Null
    Write-Host "Created directory: $logDirPath" -ForegroundColor Green
    $filesToMove = Get-ChildItem -Path $scriptRoot -File | Where-Object
{ $_.Extension -in '.h264', '.csv' }

```

```

foreach ($file in $filesToMove) {
    Move-Item -Path $file.FullName -Destination $logDirPath -Force
    Write-Host "Moved: $($file.Name)" -ForegroundColor Cyan
}

Write-Host "All files moved to: $logDirPath" -ForegroundColor Green
}
function Add-Log {
    param([string]$message)
    $timestamp = Get-Date -Format "yyyy-MM-dd HH:mm:ss"
    $logBox.AppendText("[ $timestamp] $message`r`n")
}
function Remove-OldVisualizationFiles {
    $downloadsPath = Join-Path $env:USERPROFILE "Downloads"
    $mapVizPattern = "map_visualization*.png"
    $rawMapPattern = "raw_map*.png"

    $filesDeleted = 0
    $mapVizFiles = Get-ChildItem -Path $downloadsPath -Filter
$mapVizPattern -ErrorAction SilentlyContinue
    foreach ($file in $mapVizFiles) {
        try {
            Remove-Item -Path $file.FullName -Force
            Add-Log "Deleted old file from Downloads: $($file.Name)"
            $filesDeleted++
        }
        catch {
            Add-Log "Failed to delete $($file.Name):
$($_.Exception.Message)"
        }
    }
    $rawMapFiles = Get-ChildItem -Path $downloadsPath -Filter
$rawMapPattern -ErrorAction SilentlyContinue
    foreach ($file in $rawMapFiles) {
        try {
            Remove-Item -Path $file.FullName -Force
            Add-Log "Deleted old file from Downloads: $($file.Name)"
            $filesDeleted++
        }
        catch {
            Add-Log "Failed to delete $($file.Name):
$($_.Exception.Message)"
        }
    }

    if ($filesDeleted -eq 0) {
        Add-Log "No old visualization files found in Downloads"
    }
    else {
        Add-Log "Cleaned up $filesDeleted old visualization file(s) from
Downloads"
    }
}
Remove-OldVisualizationFiles
$cachedMapPath = Join-Path $scriptPath 'map_visualization.png'

```

```

$defaultImagePath = Join-Path $scriptPath 'noVUE.jpg'

if (Test-Path $cachedMapPath) {
    $imageBox.Image = [System.Drawing.Image]::FromFile($cachedMapPath)
    $script:usingCachedMap = $true
}
elseif (Test-Path $defaultImagePath) {
    $imageBox.Image = [System.Drawing.Image]::FromFile($defaultImagePath)
    $script:usingCachedMap = $false
}
$form.Controls.Add($imageBox)
$buttonWidth = 200
$buttonHeight = 60
$buttonLeft = 22
$buttonTopRow = 142
$buttonSpacing = 50
$verticalSpacing = 80
$script:monitorJob = $null
$script:monitorTimer = $null
$button1 = New-Object System.Windows.Forms.Button
$button1.Location = New-Object System.Drawing.Point($buttonLeft,
$buttonTopRow)
$button1.Size = New-Object System.Drawing.Size($buttonWidth, $buttonHeight)
$button1.Text = 'Check Connection'
$button1.Font = New-Object System.Drawing.Font("Segoe UI", 10)
$button1.Add_Click({
    Add-Log "Check Connection button pressed"
    $pingResult = Test-Connection -ComputerName $ServerAddress -Count 1
-Quiet

    if (-not $pingResult) {
        [System.Windows.Forms.MessageBox]::Show(
            "Failed to reach the drone at $ServerAddress`n`nPlease
check if the drone is connected to the network.",
            "Connection Error",
            [System.Windows.Forms.MessageBoxButtons]::OK,
            [System.Windows.Forms.MessageBoxIcon]::Error
        )
        Add-Log "Connection check failed - Unable to ping
$ServerAddress"
        return
    }

    Add-Log "Connection successful - Checking time synchronization"
    try {
        $time_diff = Get-RemoteTimeDifference -UserName $UserName -
ServerAddress $ServerAddress

        if ($null -eq $time_diff) {
            [System.Windows.Forms.MessageBox]::Show(
                "Failed to retrieve time information from the drone.",
                "Time Check Error",
                [System.Windows.Forms.MessageBoxButtons]::OK,
                [System.Windows.Forms.MessageBoxIcon]::Error
            )
            Add-Log "Failed to retrieve remote time difference"
            return
        }
    }
}

```



```

    }

    $absDiff = [math]::Abs($time_diff)

    if ($absDiff -le $ThresholdSeconds) {
        $status = if ($time_diff -ge 0) { "ahead of" } else
{ "behind" }
        $message = "Onboard Flight Computer is $status local time
by $([math]::Round($absDiff, 2)) seconds.`n`nThis is acceptable."

        [System.Windows.Forms.MessageBox]::Show(
            $message,
            "Time Sync Status",
            [System.Windows.Forms.MessageBoxButtons]::OK,
            [System.Windows.Forms.MessageBoxIcon]::Information
        )
        Add-Log "Time drift acceptable: $([math]::Round($absDiff,
2)) seconds"
    }
    else {
        $driftMessage = Get-TimeDriftMessage -Seconds $time_diff
        $message = "$driftMessage`n`nTime synchronization is
required."

        $result = [System.Windows.Forms.MessageBox]::Show(
            $message,
            "Sync Time Now",
            [System.Windows.Forms.MessageBoxButtons]::OK,
            [System.Windows.Forms.MessageBoxIcon]::Warning
        )

        Add-Log "Time drift exceeds threshold:
$([math]::Round($absDiff, 2)) seconds"
        Add-Log "Starting time synchronization..."
        TimeSync
        Start-Sleep -Seconds 2
        $new_time_diff = Get-RemoteTimeDifference -UserName
$UserName -ServerAddress $ServerAddress

        if ($null -ne $new_time_diff) {
            $newAbsDiff = [math]::Abs($new_time_diff)
            $newStatus = if ($new_time_diff -ge 0) { "ahead of" }
else { "behind" }
            Add-Log "Time sync completed - New drift:
$([math]::Round($newAbsDiff, 2)) seconds ($newStatus local time)"
        }
        else {
            Add-Log "Time sync completed but could not verify new
time difference"
        }
    }
}
catch {
    [System.Windows.Forms.MessageBox]::Show(
        "An error occurred while checking time
synchronization:`n`n$($_.Exception.Message)",
        "Error",
        [System.Windows.Forms.MessageBoxButtons]::OK,

```

```

        [System.Windows.Forms.MessageBoxIcon]::Error
    )
    Add-Log "Error during time check: $($_.Exception.Message)"
}
})

$form.Controls.Add($button1)

$button5 = New-Object System.Windows.Forms.Button
$button5.Location = New-Object System.Drawing.Point(($buttonLeft +
$buttonWidth + $buttonSpacing), $buttonTopRow)
$button5.Size = New-Object System.Drawing.Size($buttonWidth, $buttonHeight)
$button5.Text = 'Verify Data'
$button5.Font = New-Object System.Drawing.Font("Segoe UI", 10)
$button5.Add_Click({
    $mapPath = Join-Path $scriptPath 'map_visualization.png'
    if ((Test-Path $mapPath) -and $imageBox.Image) {
        $oldImage = $imageBox.Image
        $imageBox.Image = $null
        $oldImage.Dispose()
        Start-Sleep -Milliseconds 100
    }
    Remove-Item $mapPath -ErrorAction SilentlyContinue
    $noVuePath = Join-Path $scriptPath 'noVUE.jpg'
    if (Test-Path $noVuePath) {
        $imageBox.Image = [System.Drawing.Image]::FromFile($noVuePath)
    }
    if ($script:monitorTimer) {
        $script:monitorTimer.Stop()
        $script:monitorTimer.Dispose()
        $script:monitorTimer = $null
    }

    if ($script:monitorJob) {
        Stop-Job -Job $script:monitorJob -ErrorAction SilentlyContinue
        Remove-Job -Job $script:monitorJob -Force -ErrorAction
        SilentlyContinue
        $script:monitorJob = $null
    }

    Add-Log "Verify Data button pressed - Opening visualizer"
    $firefoxPath = "C:\Program Files\Firefox Nightly\firefox.exe"
    $filePath = Join-Path $env:USERPROFILE
    "Documents\Uni\FYDP\2_EEE499C\C_things\Shell_Util\index.html"
    $firefoxArgs = "-private-window", $filePath
    $process = Start-Process -FilePath $firefoxPath -ArgumentList
    $firefoxArgs -PassThru

    Add-Log "Firefox launcher started (PID: $($process.Id))"
    $script:monitorJob = Start-Job -ScriptBlock {
        param($downloadsPath, $scriptRoot, $launcherPid)
        Start-Sleep -Seconds 2
        $firefoxWindowProc = $null
    }

```

```

        while (-not $firefoxWindowProc) {
            $firefoxWindowProc = Get-Process firefox -ErrorAction
SilentlyContinue |
            Where-Object { $_.MainWindowHandle -ne 0 }
            Start-Sleep -Milliseconds 500
        }

        $realPid = $firefoxWindowProc.Id
        Write-Output "FIREFOX_PID:$realPid"

        $targetFile = "map_visualization.png"
        $sourceFile = Join-Path $downloadsPath $targetFile
        $destFile = Join-Path $scriptRoot $targetFile
        while ($true) {
            try {
                $p = Get-Process -Id $realPid -ErrorAction Stop

                if ($p.MainWindowHandle -eq 0) {
                    Write-Output "LOG:Firefox window closed - Stopping
monitor"
                    break
                }
            }
            catch {
                Write-Output "LOG:Firefox process terminated (PID
$realPid) - Stopping monitor"
                break
            }
            if (Test-Path $sourceFile) {
                try {
                    Start-Sleep -Milliseconds 200
                    Move-Item -Path $sourceFile -Destination $destFile
-Force

                    Write-Output "LOG:Found and moved $targetFile to
script root"

                    Write-Output "IMAGE_UPDATE"
                }
                catch {
                    Write-Output "LOG:Error moving file:
$(($_.Exception.Message))"
                }
            }

            Start-Sleep -Seconds 1
        }
        $rawMapPattern = "raw_map*.png"
        $rawMapFiles = Get-ChildItem -Path $downloadsPath -Filter
$rawMapPattern -ErrorAction SilentlyContinue

        if ($rawMapFiles.Count -gt 0) {
            foreach ($file in $rawMapFiles) {
                try {
                    Remove-Item -Path $file.FullName -Force
                    Write-Output "RAW_MAP_DELETED:$($file.Name)"
                }
                catch {
                    Write-Output
"RAW_MAP_DELETE_ERROR:$($file.Name):$(($_.Exception.Message))"

```

```

        }
    }
    Write-Output
"RAW_MAP_CLEANUP_COMPLETE:$($rawMapFiles.Count)"
    }
    else {
        Write-Output "RAW_MAP_CLEANUP_COMPLETE:0"
    }

    Write-Output "MONITOR_STOPPED"
} -ArgumentList "$env:USERPROFILE\Downloads", $scriptPath,
$process.Id
$script:monitorTimer = New-Object System.Windows.Forms.Timer
$script:monitorTimer.Interval = 500
$script:monitorTimer.Add_Tick({
    if ($script:monitorJob -and $script:monitorJob.State -eq
'Running') {
        $jobOutput = Receive-Job -Job $script:monitorJob -
ErrorAction SilentlyContinue

        foreach ($line in $jobOutput) {
            if ($line -match "^FIREFOX_PID:(\d+)$") {
                $realPid = $matches[1]
                Add-Log "Firefox window detected (PID:
$realPid)"
            }
            elseif ($line -match "^LOG:(.+)$") {
                Add-Log $matches[1]
            }
            elseif ($line -eq "IMAGE_UPDATE") {
                $mapPath = Join-Path $scriptPath
'visualization.png'

                if (Test-Path $mapPath) {
                    if ($imageBox.Image) {
                        $oldImage = $imageBox.Image
                        $imageBox.Image = $null
                        $oldImage.Dispose()
                    }
                    $imageBox.Image =
[System.Drawing.Image]::FromFile($mapPath)
                }
            }
            elseif ($line -match "^RAW_MAP_DELETED:(.+)$") {
                Add-Log "Deleted from Downloads:
$(($matches[1]))"
            }
            elseif ($line -match
"^RAW_MAP_DELETE_ERROR:([:^:]+):(.+)$") {
                Add-Log "Failed to delete $(($matches[1])):
$(($matches[2]))"
            }
            elseif ($line -match
"^RAW_MAP_CLEANUP_COMPLETE:(\d+)$") {
                $count = [int]$matches[1]
                if ($count -gt 0) {
                    Add-Log "Cleaned up $count raw_map file(s)
from Downloads"
                }
            }
        }
    }
}

```

```

        else {
            Add-Log "No raw_map files found in
Downloads"
        }
    }
    elseif ($line -eq "MONITOR_STOPPED") {
        $script:monitorTimer.Stop()
        Remove-Job -Job $script:monitorJob -Force -
ErrorAction SilentlyContinue
        $script:monitorJob = $null
        $noVuePath = Join-Path $scriptPath 'noVUE.jpg'
        if (Test-Path $noVuePath) {
            if ($imageBox.Image) {
                $oldImage = $imageBox.Image
                $imageBox.Image = $null
                $oldImage.Dispose()
            }
            $imageBox.Image =
[System.Drawing.Image]::FromFile($noVuePath)
        }

        Add-Log "Image monitoring stopped"
    }
}
elseif ($script:monitorJob -and $script:monitorJob.State -
ne 'Running') {
    $script:monitorTimer.Stop()
    Add-Log "Monitor job stopped unexpectedly"
    Remove-Job -Job $script:monitorJob -Force -ErrorAction
SilentlyContinue
    $script:monitorJob = $null
}
})
$script:monitorTimer.Start()
})
$form.Controls.Add($button5)
$button2 = New-Object System.Windows.Forms.Button
$button2.Location = New-Object System.Drawing.Point($buttonLeft,
($buttonTopRow + $verticalSpacing))
$button2.Size = New-Object System.Drawing.Size($buttonWidth, $buttonHeight)
$button2.Text = 'Start Logging'
$button2.Font = New-Object System.Drawing.Font("Segoe UI", 10)
$button2.ForeColor = [System.Drawing.Color]::Red
$button2.Enabled = $false
$button2.Add_Click({
    Add-Log "Start Logging button pressed"
    Add-Log "Testing connection to $ServerAddress..."
    $pingResult = Test-Connection -ComputerName $ServerAddress -Count 1 -
Quiet

    if (-not $pingResult) {
        [System.Windows.Forms.MessageBox]::Show(
            "Failed to reach the drone at $ServerAddress`n`nPlease check if
the drone is connected to the network.",
            "Connection Error",
            [System.Windows.Forms.MessageBoxButtons]::OK,
            [System.Windows.Forms.MessageBoxIcon]::Error
    )
    }
}
)

```

```

    )
    Add-Log "Ping failed - Cannot reach $ServerAddress"
    return
}

Add-Log "Ping successful"
Add-Log "Running link test..."
try {
    $linkTestOutput = ssh USER_PROFILE@$ServerAddress "cd
/home/USER_PROFILE/workspace; source .venv/bin/activate; python3 -m
flying_machine link_test; deactivate" 2>&1
    Add-Log "Link test output: $linkTestOutput"
    $mavlinkFound = $false
    $monitorFound = $false
    $mavlinkCorrect = $false
    $monitorCorrect = $false

    foreach ($line in $linkTestOutput -split "`n") {
        $line = $line.Trim()
        if ($line -match "^MAVLINK=(USB\d+),\s*(\d+)") {
            $mavlinkFound = $true
            $mavlinkBaud = $matches[2]
            if ($mavlinkBaud -eq "57600") {
                $mavlinkCorrect = $true
                Add-Log "MAVLINK found: $($matches[1]), $mavlinkBaud
(OK)"
            } else {
                Add-Log "MAVLINK found but baud rate incorrect:
$mavlinkBaud"
            }
        }
        if ($line -match "^MONITOR=(USB\d+),\s*(\d+)") {
            $monitorFound = $true
            $monitorBaud = $matches[2]
            if ($monitorBaud -eq "230400") {
                $monitorCorrect = $true
                Add-Log "MONITOR found: $($matches[1]), $monitorBaud
(OK)"
            } else {
                Add-Log "MONITOR found but baud rate incorrect:
$monitorBaud"
            }
        }
    }
}
if (-not $monitorCorrect) {
    [System.Windows.Forms.MessageBox]::Show(
        "Monitoring GPS Subsystem General Failure",
        "Subsystem Error",
        [System.Windows.Forms.MessageBoxButtons]::OK,
        [System.Windows.Forms.MessageBoxIcon]::Error
    )
    Add-Log "Link test failed - MONITOR subsystem error"
    return
}

if (-not $mavlinkCorrect) {
    [System.Windows.Forms.MessageBox]::Show(
        "AutoLanding Subsystem General Failure",

```

```

        "Subsystem Error",
        [System.Windows.Forms.MessageBoxButtons]::OK,
        [System.Windows.Forms.MessageBoxIcon]::Error
    )
    Add-Log "Link test failed - MAVLINK subsystem error"
    return
}

Add-Log "Link test passed - All subsystems OK"

} catch {
    [System.Windows.Forms.MessageBox]::Show(
        "Error running link test: $($_.Exception.Message)",
        "Error",
        [System.Windows.Forms.MessageBoxButtons]::OK,
        [System.Windows.Forms.MessageBoxIcon]::Error
    )
    Add-Log "Link test error: $($_.Exception.Message)"
    return
}
Add-Log "Starting log..."
try {
    $logOutput = ssh USER_PROFILE@$ServerAddress "cd
/home/USER_PROFILE/workspace; source .venv/bin/activate; python3 -m
flying_machine start_log; deactivate" 2>&1
    Add-Log "Log start output: $logOutput"
    $logNumber = ""
    foreach ($line in $logOutput -split "`n") {
        if ($line -match "Log number:\s*(\d+)") {
            $logNumber = $matches[1]
            Add-Log "Extracted log number: $logNumber"
            break
        }
    }

    if ($logNumber -eq "") {
        [System.Windows.Forms.MessageBox]::Show(
            "Failed to extract log number from start_log output",
            "Error",
            [System.Windows.Forms.MessageBoxButtons]::OK,
            [System.Windows.Forms.MessageBoxIcon]::Error
        )
        Add-Log "Failed to extract log number"
        return
    }
    Add-Log "Updating sys_stat.json with log number: $logNumber"
    $jsonPath = Join-Path $scriptPath 'sys_stat.json'

    try {
        $jsonContent = Get-Content -Path $jsonPath -Raw | ConvertFrom-
Json
        $jsonContent.log_number = $logNumber
        $jsonContent | ConvertTo-Json | Set-Content -Path $jsonPath
        Add-Log "sys_stat.json updated successfully"
    } catch {
        Add-Log "Warning: Failed to update sys_stat.json:
$($_.Exception.Message)"
    }
}

```

```

        Add-Log "Starting video recording..."
        $timestamp = (Get-
Date).ToString("ddMMMyyyy_HH:mm:ss'UTC'_$logNumber").ToUpper()
        Add-Log "Video filename: $timestamp.h264"

        $videoCmd = "pkill rpicas-vid; cd
/home/USER_PROFILE/workspace/data_dump; nohup rpicas-vid --camera 1 --width
1000 --height 1000 --framerate 120 -t 0 --nopreview -o $timestamp.h264 >
/dev/null 2>&1 &"
        ssh USER_PROFILE@$ServerAddress $videoCmd

        Add-Log "Video recording started"
        [System.Windows.Forms.MessageBox]::Show(
            "Logging started successfully. You may start the flight.",
            "Success",
            [System.Windows.Forms.MessageBoxButtons]::OK,
            [System.Windows.Forms.MessageBoxIcon]::Information
        )
        Add-Log "All systems ready - Flight can begin"

    } catch {
        [System.Windows.Forms.MessageBox]::Show(
            "Error during logging setup: $($_.Exception.Message)",
            "Error",
            [System.Windows.Forms.MessageBoxButtons]::OK,
            [System.Windows.Forms.MessageBoxIcon]::Error
        )
        Add-Log "Logging setup error: $($_.Exception.Message)"
        return
    }
})
$form.Controls.Add($button2)

$button6 = New-Object System.Windows.Forms.Button
$button6.Location = New-Object System.Drawing.Point(($buttonLeft +
$buttonWidth + $buttonSpacing), ($buttonTopRow + $verticalSpacing))
$button6.Size = New-Object System.Drawing.Size($buttonWidth, $buttonHeight)
$button6.Text = 'Launch Data Explorer'
$button6.Font = New-Object System.Drawing.Font("Segoe UI", 10)
$button6.ForeColor = [System.Drawing.Color]::Black
$button6.Enabled = $true
$button6.Add_Click({
    Add-Log "Launch Data Explorer - Opening tool"
    $firefoxPath = "C:\Program Files\Firefox Nightly\firefox.exe"
    $url = "http://[REDACTED]:8501/"
    $firefoxArgs = "-private-window", $url
    $process = Start-Process -FilePath $firefoxPath -ArgumentList
$firefoxArgs -PassThru
    Start-Sleep -Milliseconds 500
    $wshell = New-Object -ComObject wscript.shell
    $wshell.AppActivate($process.Id)

    Add-Log "Firefox launcher started (PID: $($process.Id))"
    $script:explorerMonitorJob = Start-Job -ScriptBlock {
        param($launcherPid)
        Start-Sleep -Seconds 2
    }
})

```



```

$firefoxWindowProc = $null

while (-not $firefoxWindowProc) {
    $firefoxWindowProc = Get-Process firefox -ErrorAction
SilentlyContinue |
    Where-Object { $_.MainWindowHandle -ne 0 }
    Start-Sleep -Milliseconds 500
}

$realPid = $firefoxWindowProc.Id
Write-Output "FIREFOX_PID:$realPid"
$firefoxWindowProc.WaitForExit()
Write-Output "FIREFOX_CLOSED:$realPid"

} -ArgumentList $process.Id
$script:explorerMonitorTimer = New-Object System.Windows.Forms.Timer
$script:explorerMonitorTimer.Interval = 500
$script:explorerMonitorTimer.Add_Tick({
    if ($script:explorerMonitorJob -and
$script:explorerMonitorJob.State -eq 'Running') {
        $jobOutput = Receive-Job -Job $script:explorerMonitorJob -
ErrorAction SilentlyContinue

        foreach ($line in $jobOutput) {
            if ($line -match "^FIREFOX_PID:(\d+)$") {
                $realPid = $matches[1]
                Add-Log "Map Lab Deep Visualizer started (PID:
$realPid)"
            }
            elseif ($line -match "^FIREFOX_CLOSED:(\d+)$") {
                $closedPid = $matches[1]
                Add-Log "Map Lab Deep Visualizer closed gracefully
(PID: $closedPid)"
            }
        }
    }
    elseif ($script:explorerMonitorJob -and
$script:explorerMonitorJob.State -ne 'Running') {
        $jobOutput = Receive-Job -Job $script:explorerMonitorJob -
ErrorAction SilentlyContinue
        foreach ($line in $jobOutput) {
            if ($line -match "^FIREFOX_CLOSED:(\d+)$") {
                $closedPid = $matches[1]
                Add-Log "Map Lab Deep Visualizer closed gracefully
(PID: $closedPid)"
            }
        }
    }

    $script:explorerMonitorTimer.Stop()
    $script:explorerMonitorTimer.Dispose()
    $script:explorerMonitorTimer = $null

```

```

        Remove-Job -Job $script:explorerMonitorJob -Force -ErrorAction
SilentlyContinue
        $script:explorerMonitorJob = $null
    }
    })
    $script:explorerMonitorTimer.Start()
})
$form.Controls.Add($button6)
$button3 = New-Object System.Windows.Forms.Button
$button3.Location = New-Object System.Drawing.Point($buttonLeft,
($buttonTopRow + (2 * $verticalSpacing)))
$button3.Size = New-Object System.Drawing.Size($buttonWidth, $buttonHeight)
$button3.Text = 'Stop Logging'
$button3.Font = New-Object System.Drawing.Font("Segoe UI", 10)
$button3.ForeColor = [System.Drawing.Color]::Red
$button3.Enabled = $false
$button3.Add_Click({
    Add-Log "Stop Logging button pressed"
    Add-Log "Stopping camera recording..."
    try {
        ssh USER_PROFILE@██████ "pkill rpicas-vid" 2>&1 | Out-Null
        Add-Log "Camera recording stopped"
    } catch {
        Add-Log "Error stopping camera: $($_.Exception.Message)"
    }
    Start-Sleep -Seconds 1
    Add-Log "Stopping GPS logging..."
    try {
        $stopLogOutput = ssh USER_PROFILE@██████ "cd
/home/USER_PROFILE/workspace; source .venv/bin/activate; python3 -m
flying_machine stop_log; deactivate" 2>&1
        $stopLogText = $stopLogOutput -join "`n"
        Add-Log "Stop log response: $stopLogText"
        if ($stopLogText -match "Stopping log" -and $stopLogText -match
"Log stopped") {
            Add-Log "GPS logging stopped successfully"
        } else {
            [System.Windows.Forms.MessageBox]::Show(
                "GPS Subsystem Failed to Exit Logging Loop",
                "Error",
                [System.Windows.Forms.MessageBoxButtons]::OK,
                [System.Windows.Forms.MessageBoxIcon]::Error
            )
            Add-Log "Failed to stop GPS logging"
            return
        }
    } catch {
        [System.Windows.Forms.MessageBox]::Show(
            "GPS Subsystem Failed to Exit Logging Loop",
            "Error",
            [System.Windows.Forms.MessageBoxButtons]::OK,
            [System.Windows.Forms.MessageBoxIcon]::Error
        )
        Add-Log "Error stopping GPS logging: $($_.Exception.Message)"
        return
    }
    $connectSuccess = $false
    $maxTries = 3

```

```

$currentTry = 0
$loggerIP = ""

while ($currentTry -lt $maxTries -and -not $connectSuccess) {
    $currentTry++
    Add-Log "Attempting to connect logger (Attempt $currentTry of $maxTries)..."

    try {
        $connectOutput = ssh USER_PROFILE@[REDACTED] "cd /home/USER_PROFILE/workspace; source .venv/bin/activate; python3 -m flying_machine connect_logger; deactivate" 2>&1
        $connectText = $connectOutput -join "`n"
        Add-Log "Connect logger response: $connectText"
        if ($connectText -match "Logger connected" -and $connectText -match "IP:\s*([\d\.]+)" -and $connectText -match "http://") {
            $loggerIP = $matches[1]
            $connectSuccess = $true
            Add-Log "Logger connected successfully. IP: $loggerIP"
        } elseif ($connectText -match "Failed to connect to WiFi") {
            Add-Log "Failed to connect to WiFi (Attempt $currentTry)"
            if ($currentTry -lt $maxTries) {
                Add-Log "Waiting 10 seconds before retry..."
                Start-Sleep -Seconds 10
            }
        } else {
            Add-Log "Unexpected response from connect_logger (Attempt $currentTry)"
            if ($currentTry -lt $maxTries) {
                Start-Sleep -Seconds 10
            }
        }
    } catch {
        Add-Log "Error connecting logger (Attempt $currentTry): $($_.Exception.Message)"
        if ($currentTry -lt $maxTries) {
            Start-Sleep -Seconds 10
        }
    }
}

if (-not $connectSuccess) {
    [System.Windows.Forms.MessageBox]::Show(
        "GPS Subsystem Network Failure, Perform Manual Connection",
        "Error",
        [System.Windows.Forms.MessageBoxButtons]::OK,
        [System.Windows.Forms.MessageBoxIcon]::Error
    )
    Add-Log "Failed to connect logger after $maxTries attempts"
    return
}

Start-Sleep -Seconds 2
Add-Log "Downloading log..."
try {
    $downloadOutput = ssh USER_PROFILE@[REDACTED] "cd /home/USER_PROFILE/workspace; source .venv/bin/activate; python3 -m flying_machine download_log; deactivate" 2>&1
}

```

```

$downloadText = $downloadOutput -join "`n"
Add-Log "Download log response: $downloadText"
if ($downloadText -match "Download complete:") {
    Add-Log "Log downloaded successfully"
    $downloadSuccess = $true
} elseif ($downloadText -match "Ping attempt" -and $downloadText -
match "failed") {
    Add-Log "Download failed - connection issues detected"
    $downloadSuccess = $false
} else {
    Add-Log "Unexpected download response"
    $downloadSuccess = $false
}
} catch {
    Add-Log "Error downloading log: $($_.Exception.Message)"
    $downloadSuccess = $false
}
}
if (-not $downloadSuccess) {
    Add-Log "Entering failure recovery procedure..."
    Start-Sleep -Seconds 5
    Add-Log "Disconnecting logger..."
    try {
        $disconnectOutput = ssh USER_PROFILE@[REDACTED] "cd
/home/USER_PROFILE/workspace; source .venv/bin/activate; python3 -m
flying_machine disconnect_logger; deactivate" 2>&1
        $disconnectText = $disconnectOutput -join "`n"
        Add-Log "Disconnect response: $disconnectText"

        if ($disconnectText -match "Logger disconnected") {
            Add-Log "Logger disconnected successfully"
        }
    } catch {
        Add-Log "Error disconnecting logger: $($_.Exception.Message)"
    }
    Start-Sleep -Seconds 5
    Add-Log "Reconnecting logger..."
    try {
        $reconnectOutput = ssh USER_PROFILE@[REDACTED] "cd
/home/USER_PROFILE/workspace; source .venv/bin/activate; python3 -m
flying_machine connect_logger; deactivate" 2>&1
        $reconnectText = $reconnectOutput -join "`n"
        Add-Log "Reconnect response: $reconnectText"
        if ($reconnectText -match "Logger connected" -and
$reconnectText -match "IP:" -and $reconnectText -match "http://") {
            Add-Log "Logger reconnected successfully"
        } elseif ($reconnectText -match "Failed to connect to WiFi") {
            [System.Windows.Forms.MessageBox]::Show(
                "Retrieval system general failure, try manual
extraction",
                "Error",
                [System.Windows.Forms.MessageBoxButtons]::OK,
                [System.Windows.Forms.MessageBoxIcon]::Error
            )
            Add-Log "Failed to reconnect logger"
            return
        } else {
            [System.Windows.Forms.MessageBox]::Show(

```

```

        "Retrieval system general failure, try manual
extraction",
        "Error",
        [System.Windows.Forms.MessageBoxButtons]::OK,
        [System.Windows.Forms.MessageBoxIcon]::Error
    )
    Add-Log "Unexpected reconnect response"
    return
}
} catch {
    [System.Windows.Forms.MessageBox]::Show(
        "Retrieval system general failure, try manual extraction",
        "Error",
        [System.Windows.Forms.MessageBoxButtons]::OK,
        [System.Windows.Forms.MessageBoxIcon]::Error
    )
    Add-Log "Error reconnecting logger: $($_.Exception.Message)"
    return
}
Start-Sleep -Seconds 5
Add-Log "Retrying download..."
try {
    $retryDownloadOutput = ssh USER_PROFILE@[REDACTED] "cd
/home/USER_PROFILE/workspace; source .venv/bin/activate; python3 -m
flying_machine download_log; deactivate" 2>&1
    $retryDownloadText = $retryDownloadOutput -join "`n"
    Add-Log "Retry download response: $retryDownloadText"
    if ($retryDownloadText -match "Download complete:") {
        Add-Log "Log downloaded successfully on retry"
    } elseif ($retryDownloadText -match "Ping attempt" -and
$retryDownloadText -match "failed") {
        [System.Windows.Forms.MessageBox]::Show(
            "Retrieval system general failure, try manual
extraction",
            "Error",
            [System.Windows.Forms.MessageBoxButtons]::OK,
            [System.Windows.Forms.MessageBoxIcon]::Error
        )
        Add-Log "Retry download failed - connection issues"
        return
    } else {
        [System.Windows.Forms.MessageBox]::Show(
            "Retrieval system general failure, try manual
extraction",
            "Error",
            [System.Windows.Forms.MessageBoxButtons]::OK,
            [System.Windows.Forms.MessageBoxIcon]::Error
        )
        Add-Log "Unexpected retry download response"
        return
    }
} catch {
    [System.Windows.Forms.MessageBox]::Show(
        "Retrieval system general failure, try manual extraction",
        "Error",
        [System.Windows.Forms.MessageBoxButtons]::OK,
        [System.Windows.Forms.MessageBoxIcon]::Error
    )
}

```

```

        Add-Log "Error during retry download: $($_.Exception.Message)"
        return
    }
}
Start-Sleep -Seconds 2
Add-Log "Disconnecting logger..."
try {
    $finalDisconnectOutput = ssh USER_PROFILE@  "cd
/home/USER_PROFILE/workspace; source .venv/bin/activate; python3 -m
flying_machine disconnect_logger; deactivate" 2>&1
    $finalDisconnectText = $finalDisconnectOutput -join "`n"
    Add-Log "Final disconnect response: $finalDisconnectText"

    if ($finalDisconnectText -match "Logger disconnected") {
        Add-Log "Logger disconnected successfully"
        Add-Log "Stop logging procedure completed successfully"

        [System.Windows.Forms.MessageBox]::Show(
            "Logging stopped and data retrieved successfully",
            "Success",
            [System.Windows.Forms.MessageBoxButtons]::OK,
            [System.Windows.Forms.MessageBoxIcon]::Information
        )
    }
} catch {
    Add-Log "Error during final disconnect: $($_.Exception.Message)"
}
})
$form.Controls.Add($button3)

$button7 = New-Object System.Windows.Forms.Button
$button7.Location = New-Object System.Drawing.Point(($buttonLeft +
$buttonWidth + $buttonSpacing), ($buttonTopRow + (2 * $verticalSpacing)))
$button7.Size = New-Object System.Drawing.Size($buttonWidth, $buttonHeight)
$button7.Text = 'Launch Live Control'
$button7.Font = New-Object System.Drawing.Font("Segoe UI", 10)
$button7.Add_Click({
    Add-Log "Launch Data Explorer - Opening tool"
    $firefoxPath = "C:\Program Files\Firefox Nightly\firefox.exe"
    $url = "http:// :5000/"
    $firefoxArgs = "-private-window", $url

    try {
        $process = Start-Process -FilePath $firefoxPath -ArgumentList
$firefoxArgs -PassThru -ErrorAction Stop
        Start-Sleep -Milliseconds 500
        $wshell = New-Object -ComObject wscript.shell
        $wshell.AppActivate($process.Id)

        Add-Log "Firefox launcher started in private mode (PID:
$($process.Id))"
        $script:explorerMonitorJob = Start-Job -ScriptBlock {
            param($launcherPid)
            Start-Sleep -Seconds 2
            $firefoxWindowProc = $null

```

```

$timeout = 30
$elapsed = 0

while (-not $firefoxWindowProc -and $elapsed -lt $timeout) {
    $firefoxWindowProc = Get-Process firefox -ErrorAction
SilentlyContinue |
        Where-Object { $_.MainWindowHandle -ne 0 }
    Start-Sleep -Milliseconds 500
    $elapsed++
}

if ($firefoxWindowProc) {
    $realPid = $firefoxWindowProc.Id
    Write-Output "FIREFOX_PID:$realPid"
    $firefoxWindowProc.WaitForExit()
    Write-Output "FIREFOX_CLOSED:$realPid"
} else {
    Write-Output "FIREFOX_TIMEOUT"
}
} -ArgumentList $process.Id
$script:explorerMonitorTimer = New-Object
System.Windows.Forms.Timer
$script:explorerMonitorTimer.Interval = 500
$script:explorerMonitorTimer.Add_Tick({
    if ($script:explorerMonitorJob -and
$script:explorerMonitorJob.State -eq 'Running') {
        $jobOutput = Receive-Job -Job $script:explorerMonitorJob -
ErrorAction SilentlyContinue

        foreach ($line in $jobOutput) {
            if ($line -match "^FIREFOX_PID:(\d+)$") {
                $realPid = $matches[1]
                Add-Log "Data Explorer started (PID: $realPid)"
            }
            elseif ($line -match "^FIREFOX_CLOSED:(\d+)$") {
                $closedPid = $matches[1]
                Add-Log "Data Explorer closed gracefully (PID:
$closedPid)"
            }
            elseif ($line -eq "FIREFOX_TIMEOUT") {
                Add-Log "Warning: Firefox window detection timed
out"
            }
        }
    }
    elseif ($script:explorerMonitorJob -and
$script:explorerMonitorJob.State -ne 'Running') {
        $jobOutput = Receive-Job -Job $script:explorerMonitorJob -
ErrorAction SilentlyContinue
        foreach ($line in $jobOutput) {
            if ($line -match "^FIREFOX_CLOSED:(\d+)$") {
                $closedPid = $matches[1]
                Add-Log "Data Explorer closed gracefully (PID:
$closedPid)"
            }
        }
    }

    $script:explorerMonitorTimer.Stop()

```

```

        $script:explorerMonitorTimer.Dispose()
        $script:explorerMonitorTimer = $null

        Remove-Job -Job $script:explorerMonitorJob -Force -
ErrorAction SilentlyContinue
        $script:explorerMonitorJob = $null
    }
    })
    $script:explorerMonitorTimer.Start()

} catch {
    Add-Log "Error launching Firefox: $_"
}
})

$form.Controls.Add($button7)
$button4 = New-Object System.Windows.Forms.Button
$button4.Location = New-Object System.Drawing.Point($buttonLeft,
($buttonTopRow + (3 * $verticalSpacing)))
$button4.Size = New-Object System.Drawing.Size($buttonWidth, $buttonHeight)
$button4.Text = 'Transfer Data'
$button4.Font = New-Object System.Drawing.Font("Segoe UI", 10)
$button4.ForeColor = [System.Drawing.Color]::Red
$button4.Enabled = $false
$button4.Add_Click({
    Add-Log "Data transfer sequence initiated"

    try {
        $sysStatPath = Join-Path $scriptPath "sys_stat.json"

        if (-not (Test-Path $sysStatPath)) {
            Add-Log "ERROR: sys_stat.json not found"
            [System.Windows.Forms.MessageBox]::Show("sys_stat.json file not
found", "Error", [System.Windows.Forms.MessageBoxButtons]::OK,
[System.Windows.Forms.MessageBoxIcon]::Error)
            return
        }

        Add-Log "Reading sys_stat.json..."
        $sysStatContent = Get-Content $sysStatPath -Raw | ConvertFrom-Json
        $logID = [int]$sysStatContent.log_number
        Add-Log "Log ID retrieved: $logID"
        Add-Log "Transferring CSV files from remote..."
        scp
USER_PROFILE@[REDACTED]:/home/USER_PROFILE/workspace/data_dump/*.csv .
        Add-Log "CSV transfer complete"
        Add-Log "Transferring H264 files from remote..."
        scp
USER_PROFILE@[REDACTED]:/home/USER_PROFILE/workspace/data_dump/*.h264 .
        Add-Log "H264 transfer complete"

        Add-Log "Clearing Onboard Cache"
        ssh USER_PROFILE@[REDACTED] "cd /home/USER_PROFILE/workspace/data_dump;
rm -vrf *.h264 *.csv"
        Add-Log "Onboard Cache Clean Completed"
        $csvFile = Get-ChildItem -Filter "*.csv" | Sort-Object
LastWriteTime -Descending | Select-Object -First 1
        if ($csvFile) {

```



```

        $sysStatContent.log_filename = $csvFile.Name
        Add-Log "CSV file found: $($csvFile.Name)"
    } else {
        Add-Log "WARNING: No CSV file found in current directory"
    }
    $h264File = Get-ChildItem -Filter "*.h264" | Sort-Object
LastWriteTime -Descending | Select-Object -First 1
    if ($h264File) {
        $sysStatContent.h264_filename = $h264File.Name
        Add-Log "H264 file found: $($h264File.Name)"
    } else {
        Add-Log "WARNING: No H264 file found in current directory"
    }
    $sysStatContent | ConvertTo-Json | Set-Content $sysStatPath
    Add-Log "sys_stat.json updated successfully"
    Add-Log "Calling Process-LatestCsv with LogID: $logID"
    Process-LatestCsv -LogID $logID

    Add-Log "Transfer Data operation completed successfully"
    [System.Windows.Forms.MessageBox]::Show("Data transfer and
processing completed successfully", "Success",
[System.Windows.Forms.MessageBoxButtons]::OK,
[System.Windows.Forms.MessageBoxIcon]::Information)

    } catch {
        Add-Log "ERROR during Transfer Data: $($_.Exception.Message)"
        [System.Windows.Forms.MessageBox]::Show("Error during data
transfer: $($_.Exception.Message)", "Error",
[System.Windows.Forms.MessageBoxButtons]::OK,
[System.Windows.Forms.MessageBoxIcon]::Error)
    }
}

$button4

$button8 = New-Object System.Windows.Forms.Button
$button8.Location = New-Object System.Drawing.Point(($buttonLeft +
$buttonWidth + $buttonSpacing), ($buttonTopRow + (3 * $verticalSpacing)))
$button8.Size = New-Object System.Drawing.Size($buttonWidth, $buttonHeight)
$button8.Text = 'View Processed Data'
$button8.Font = New-Object System.Drawing.Font("Segoe UI", 10)
$button8.Add_Click({
    start
    "C:\Users\USER_PROFILE\Documents\Uni\FYDP\2_EEE499C\C_things\Processed_Data
"
    Add-Log "Processed Data Open Request"
})
$button8
$form.Controls.Add($button8)
$form.Add_FormClosing({
    param($sender, $e)
    $mapPath = Join-Path $scriptPath 'map_visualization.png'
    if (Test-Path $mapPath) {
        $result = [System.Windows.Forms.MessageBox]::Show(
            "Do you want to clear temporary DataViz cache?",
            "Clean Cache",
            [System.Windows.Forms.MessageBoxButtons]::YesNo,

```

```

        [System.Windows.Forms.MessageBoxIcon]::Question
    )

    if ($result -eq [System.Windows.Forms.DialogResult]::Yes) {
        if ($imageBox.Image) {
            $imageBox.Image.Dispose()
        }
        try {
            Remove-Item -Path $mapPath -Force
            Add-Log "Cache cleaned - map_visualization.png deleted"
            Remove-Item -Path "$env:USERPROFILE\Desktop\*.csv"
            Add-Log "Cleaned Temporary CSV Files"
        }
        catch {
            [System.Windows.Forms.MessageBox]::Show(
                "Failed to delete cache file:
                $($_.Exception.Message)",
                "Error",
                [System.Windows.Forms.MessageBoxButtons]::OK,
                [System.Windows.Forms.MessageBoxIcon]::Error
            )
        }
    }

    if ($script:monitorTimer) {
        $script:monitorTimer.Stop()
        $script:monitorTimer.Dispose()
    }

    if ($script:monitorJob) {
        Stop-Job -Job $script:monitorJob -ErrorAction SilentlyContinue
        Remove-Job -Job $script:monitorJob -Force -ErrorAction
        SilentlyContinue
    }
    if ($imageBox.Image) {
        $imageBox.Image.Dispose()
    }
}

Add-Log "DroneLAB initialized - Ready for operation"
if ($script:usingCachedMap) {
    Add-Log "Loaded cached map from previous session
    (map_visualization.png)"
}
[void]$form.ShowDialog()

```

Map DVT Toolchain

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8" />
  <title>Basic Map Data Verification Utility</title>
  <meta name="viewport" content="width=device-width, initial-scale=1" />
  <link href="style.css" rel="stylesheet" />
</head>
<body>
  <div id="appRoot" class="container">
    <header class="header">
      <h1>Map Data Verification Tool by FYDP Group 6</h1>
      <p>Some features are stripped for field use</p>
    </header>
    <main class="main-content">
      <section class="config-panel">
        <div class="card">
          <div class="card__header">
            <h3>Configuration</h3>
          </div>
          <div class="card__body">
            <div class="file-input-group">
              <input id="csvFile" type="file" accept=".csv" class="form-
control" />
              <button id="loadSampleBtn" class="btn btn--secondary">Load
Sample</button>
            </div>
            <div class="config-grid">
              <div>
                <label class="form-label" for="serverBase">Tiles Server
Base</label>
                <input id="serverBase" type="text" class="form-control"
value="http://[REDACTED]:8080" />
              </div>
              <div>
                <label class="form-label" for="zoom">Tiles Zoom
(int)</label>
                <input id="zoom" type="number" class="form-control"
value="22" min="8" max="23"/>
              </div>
              <div>
                <label class="form-label" for="centerLat">Center
Lat</label>
                <input id="centerLat" type="number" step="0.000001"
class="form-control" value="23.773974"/>
              </div>
              <div>
                <label class="form-label" for="centerLon">Center
Lon</label>
                <input id="centerLon" type="number" step="0.000001"
class="form-control" value="90.403619"/>
              </div>
              <div>
                <label class="form-label" for="width">Image Width
(px)</label>
```

```

        <input id="width" type="number" class="form-control"
value="1500" step="100" min="500" max="2000"/>
    </div>
    <div>
        <label class="form-label" for="height">Image Height
(px)</label>
        <input id="height" type="number" class="form-control"
value="1500" step="100" min="500" max="2000"/>
    </div>
</div>
<div class="button-group">
    <button id="processBtn" class="btn btn--primary"
disabled>Process & Render</button>
    <button id="rerenderBtn" class="btn btn--outline"
disabled>Re-render</button>
    <button id="resetBtn" class="btn btn--secondary"
disabled>Reset App</button>
</div>
</div>
<div class="card__footer">
    <span id="mapStatus" class="status status--info">Waiting for
data...</span>
</div>
</div>
</section>
<section class="visualization-area">
    <div class="map-container">
        <div class="canvas-wrapper">
            <canvas id="mapCanvas" class="map-canvas" width="2000"
height="2000"></canvas>
            <div id="pointPopup" class="point-popup hidden">
                <div class="popup-content">
                    <h4 id="popupClass">Class</h4>
                    <div class="popup-details">
                        <div class="detail-item">
                            <span class="detail-label">Lat</span>
                            <span id="popupLat" class="detail-value">-</span>
                        </div>
                        <div class="detail-item">
                            <span class="detail-label">Lon</span>
                            <span id="popupLon" class="detail-value">-</span>
                        </div>
                    </div>
                    <div class="popup-arrow"></div>
                </div>
            </div>
        </div>
        <div class="map-controls">
            <div class="control-buttons">
                <button id="downloadBtn" class="btn btn--primary"
disabled>Download Image</button>
                <button id="resetZoomBtn" class="btn btn--outline"
disabled>Reset Zoom</button>
            </div>
            <div class="map-info">
                <span class="status status--info">Tip: Scroll to zoom, drag
to pan.</span>
            </div>
        </div>
    </div>
</section>

```

```

        </div>
    </div>
</section>
<aside class="legend-panel">
    <div class="card">
        <div class="card__header">
            <h3>Legend</h3>
            <div class="legend-controls">
                <button id="selectAllBtn" class="btn btn--sm btn--outline"
disabled>Select All</button>
                <button id="deselectAllBtn" class="btn btn--sm btn--outline"
disabled>Deselect All</button>
            </div>
        </div>
        <div class="card__body">
            <div id="legendContent" class="legend-content">
                <p class="legend-placeholder">Load data to show legend</p>
            </div>
            <div id="summaryContent" class="summary-content">
                <h4>Dataset</h4>
            </div>
        </div>
    </div>
</div>
</aside>
</main>
</div>
<script src="app.js"></script>
</body>
</html>

```

MapDVT Style CSS

```

:root {
    --color-white: rgba(255, 255, 255, 1);
    --color-black: rgba(0, 0, 0, 1);
    --color-cream-50: rgba(252, 252, 249, 1);
    --color-cream-100: rgba(255, 255, 253, 1);
    --color-gray-200: rgba(245, 245, 245, 1);
    --color-gray-300: rgba(167, 169, 169, 1);
    --color-gray-400: rgba(119, 124, 124, 1);
    --color-slate-500: rgba(98, 108, 113, 1);
    --color-brown-600: rgba(94, 82, 64, 1);
    --color-charcoal-700: rgba(31, 33, 33, 1);
    --color-charcoal-800: rgba(38, 40, 40, 1);
    --color-slate-900: rgba(19, 52, 59, 1);
    --color-teal-300: rgba(50, 184, 198, 1);
    --color-teal-400: rgba(45, 166, 178, 1);
    --color-teal-500: rgba(33, 128, 141, 1);
    --color-teal-600: rgba(29, 116, 128, 1);
    --color-teal-700: rgba(26, 104, 115, 1);
    --color-teal-800: rgba(41, 150, 161, 1);
    --color-red-400: rgba(255, 84, 89, 1);
    --color-red-500: rgba(192, 21, 47, 1);
    --color-orange-400: rgba(230, 129, 97, 1);
    --color-orange-500: rgba(168, 75, 47, 1);

```

```

--color-brown-600-rgb: 94, 82, 64;
--color-teal-500-rgb: 33, 128, 141;
--color-slate-900-rgb: 19, 52, 59;
--color-slate-500-rgb: 98, 108, 113;
--color-red-500-rgb: 192, 21, 47;
--color-red-400-rgb: 255, 84, 89;
--color-orange-500-rgb: 168, 75, 47;
--color-orange-400-rgb: 230, 129, 97;
--color-bg-1: rgba(59, 130, 246, 0.08);
--color-bg-2: rgba(245, 158, 11, 0.08);
--color-bg-3: rgba(34, 197, 94, 0.08);
--color-bg-4: rgba(239, 68, 68, 0.08);
--color-bg-5: rgba(147, 51, 234, 0.08);
--color-bg-6: rgba(249, 115, 22, 0.08);
--color-bg-7: rgba(236, 72, 153, 0.08);
--color-bg-8: rgba(6, 182, 212, 0.08);
--color-background: var(--color-cream-50);
--color-surface: var(--color-cream-100);
--color-text: var(--color-slate-900);
--color-text-secondary: var(--color-slate-500);
--color-primary: var(--color-teal-500);
--color-primary-hover: var(--color-teal-600);
--color-primary-active: var(--color-teal-700);
--color-secondary: rgba(var(--color-brown-600-rgb), 0.12);
--color-secondary-hover: rgba(var(--color-brown-600-rgb), 0.2);
--color-secondary-active: rgba(var(--color-brown-600-rgb), 0.25);
--color-border: rgba(var(--color-brown-600-rgb), 0.2);
--color-btn-primary-text: var(--color-cream-50);
--color-card-border: rgba(var(--color-brown-600-rgb), 0.12);
--color-card-border-inner: rgba(var(--color-brown-600-rgb), 0.12);
--color-error: var(--color-red-500);
--color-success: var(--color-teal-500);
--color-warning: var(--color-orange-500);
--color-info: var(--color-slate-500);
--color-focus-ring: rgba(var(--color-teal-500-rgb), 0.4);
--color-select-caret: rgba(var(--color-slate-900-rgb), 0.8);
--focus-ring: 0 0 0 3px var(--color-focus-ring);
--focus-outline: 2px solid var(--color-primary);
--status-bg-opacity: 0.15;
--status-border-opacity: 0.25;
--select-caret-light: url("data:image/svg+xml,%3Csvg
xmlns='http://[REDACTED]/2000/svg' width='16' height='16' viewBox='0 0 24 24'
fill='none' stroke='%23134252' stroke-width='2' stroke-linecap='round'
stroke-linejoin='round'%3E%3Cpolyline points='6 9 12 15 18
9'%3E%3C/polyline%3E%3C/svg%3E");
--select-caret-dark: url("data:image/svg+xml,%3Csvg
xmlns='http://[REDACTED]/2000/svg' width='16' height='16' viewBox='0 0 24 24'
fill='none' stroke='%23f5f5f5' stroke-width='2' stroke-linecap='round'
stroke-linejoin='round'%3E%3Cpolyline points='6 9 12 15 18
9'%3E%3C/polyline%3E%3C/svg%3E");
--color-success-rgb: 33, 128, 141;
--color-error-rgb: 192, 21, 47;
--color-warning-rgb: 168, 75, 47;
--color-info-rgb: 98, 108, 113;
--font-family-base: Arial, "Helvetica Neue", Helvetica, sans-serif;
--font-family-mono: "Courier New", Courier, monospace;
--font-size-xs: 11px;
--font-size-sm: 12px;

```

```

--font-size-base: 14px;
--font-size-md: 14px;
--font-size-lg: 16px;
--font-size-xl: 18px;
--font-size-2xl: 20px;
--font-size-3xl: 24px;
--font-size-4xl: 30px;
--font-weight-normal: 400;
--font-weight-medium: 500;
--font-weight-semibold: 550;
--font-weight-bold: 600;
--line-height-tight: 1.2;
--line-height-normal: 1.5;
--letter-spacing-tight: -0.01em;
--space-0: 0;
--space-1: 1px;
--space-2: 2px;
--space-4: 4px;
--space-6: 6px;
--space-8: 8px;
--space-10: 10px;
--space-12: 12px;
--space-16: 16px;
--space-20: 20px;
--space-24: 24px;
--space-32: 32px;
--radius-sm: 6px;
--radius-base: 8px;
--radius-md: 10px;
--radius-lg: 12px;
--radius-full: 9999px;
--shadow-xs: 0 1px 2px rgba(0, 0, 0, 0.02);
--shadow-sm: 0 1px 3px rgba(0, 0, 0, 0.04), 0 1px 2px rgba(0, 0, 0,
0.02);
--shadow-md: 0 4px 6px -1px rgba(0, 0, 0, 0.04), 0 2px 4px -1px rgba(0,
0, 0, 0.02);
--shadow-lg: 0 10px 15px -3px rgba(0, 0, 0, 0.04), 0 4px 6px -2px
rgba(0, 0, 0, 0.02);
--shadow-inset-sm: inset 0 1px 0 rgba(255, 255, 255, 0.15), inset 0 -
1px 0 rgba(0, 0, 0, 0.03);
--duration-fast: 150ms;
--duration-normal: 250ms;
--ease-standard: cubic-bezier(0.16, 1, 0.3, 1);
--container-sm: 640px;
--container-md: 768px;
--container-lg: 1024px;
--container-xl: 1280px;
}

@media (prefers-color-scheme: dark) {
:root {
--color-gray-400-rgb: 119, 124, 124;
--color-teal-300-rgb: 50, 184, 198;
--color-gray-300-rgb: 167, 169, 169;
--color-gray-200-rgb: 245, 245, 245;
--color-bg-1: rgba(29, 78, 216, 0.15);
--color-bg-2: rgba(180, 83, 9, 0.15);
--color-bg-3: rgba(21, 128, 61, 0.15);

```

```

--color-bg-4: rgba(185, 28, 28, 0.15);
--color-bg-5: rgba(107, 33, 168, 0.15);
--color-bg-6: rgba(194, 65, 12, 0.15);
--color-bg-7: rgba(190, 24, 93, 0.15);
--color-bg-8: rgba(8, 145, 178, 0.15);
--color-background: var(--color-charcoal-700);
--color-surface: var(--color-charcoal-800);
--color-text: var(--color-gray-200);
--color-text-secondary: rgba(var(--color-gray-300-rgb), 0.7);
--color-primary: var(--color-teal-300);
--color-primary-hover: var(--color-teal-400);
--color-primary-active: var(--color-teal-800);
--color-secondary: rgba(var(--color-gray-400-rgb), 0.15);
--color-secondary-hover: rgba(var(--color-gray-400-rgb), 0.25);
--color-secondary-active: rgba(var(--color-gray-400-rgb), 0.3);
--color-border: rgba(var(--color-gray-400-rgb), 0.3);
--color-error: var(--color-red-400);
--color-success: var(--color-teal-300);
--color-warning: var(--color-orange-400);
--color-info: var(--color-gray-300);
--color-focus-ring: rgba(var(--color-teal-300-rgb), 0.4);
--color-btn-primary-text: var(--color-slate-900);
--color-card-border: rgba(var(--color-gray-400-rgb), 0.2);
--color-card-border-inner: rgba(var(--color-gray-400-rgb), 0.15);
--shadow-inset-sm: inset 0 1px 0 rgba(255, 255, 255, 0.1), inset 0
-1px 0 rgba(0, 0, 0, 0.15);
--button-border-secondary: rgba(var(--color-gray-400-rgb), 0.2);
--color-border-secondary: rgba(var(--color-gray-400-rgb), 0.2);
--color-select-caret: rgba(var(--color-gray-200-rgb), 0.8);
--color-success-rgb: var(--color-teal-300-rgb);
--color-error-rgb: var(--color-red-400-rgb);
--color-warning-rgb: var(--color-orange-400-rgb);
--color-info-rgb: var(--color-gray-300-rgb);
}
}

[data-color-scheme="dark"] {
--color-gray-400-rgb: 119, 124, 124;
--color-teal-300-rgb: 50, 184, 198;
--color-gray-300-rgb: 167, 169, 169;
--color-gray-200-rgb: 245, 245, 245;
--color-bg-1: rgba(29, 78, 216, 0.15);
--color-bg-2: rgba(180, 83, 9, 0.15);
--color-bg-3: rgba(21, 128, 61, 0.15);
--color-bg-4: rgba(185, 28, 28, 0.15);
--color-bg-5: rgba(107, 33, 168, 0.15);
--color-bg-6: rgba(194, 65, 12, 0.15);
--color-bg-7: rgba(190, 24, 93, 0.15);
--color-bg-8: rgba(8, 145, 178, 0.15);
--color-background: var(--color-charcoal-700);
--color-surface: var(--color-charcoal-800);
--color-text: var(--color-gray-200);
--color-text-secondary: rgba(var(--color-gray-300-rgb), 0.7);
--color-primary: var(--color-teal-300);
--color-primary-hover: var(--color-teal-400);
--color-primary-active: var(--color-teal-800);
--color-secondary: rgba(var(--color-gray-400-rgb), 0.15);
--color-secondary-hover: rgba(var(--color-gray-400-rgb), 0.25);

```



```

--color-secondary-active: rgba(var(--color-gray-400-rgb), 0.3);
--color-border: rgba(var(--color-gray-400-rgb), 0.3);
--color-error: var(--color-red-400);
--color-success: var(--color-teal-300);
--color-warning: var(--color-orange-400);
--color-info: var(--color-gray-300);
--color-focus-ring: rgba(var(--color-teal-300-rgb), 0.4);
--color-btn-primary-text: var(--color-slate-900);
--color-card-border: rgba(var(--color-gray-400-rgb), 0.15);
--color-card-border-inner: rgba(var(--color-gray-400-rgb), 0.15);
--shadow-inset-sm: inset 0 1px 0 rgba(255, 255, 255, 0.1), inset 0 -1px
0 rgba(0, 0, 0, 0.15);
--color-border-secondary: rgba(var(--color-gray-400-rgb), 0.2);
--color-select-caret: rgba(var(--color-gray-200-rgb), 0.8);
--color-success-rgb: var(--color-teal-300-rgb);
--color-error-rgb: var(--color-red-400-rgb);
--color-warning-rgb: var(--color-orange-400-rgb);
--color-info-rgb: var(--color-gray-300-rgb);
}

[data-color-scheme="light"] {
--color-brown-600-rgb: 94, 82, 64;
--color-teal-500-rgb: 33, 128, 141;
--color-slate-900-rgb: 19, 52, 59;
--color-background: var(--color-cream-50);
--color-surface: var(--color-cream-100);
--color-text: var(--color-slate-900);
--color-text-secondary: var(--color-slate-500);
--color-primary: var(--color-teal-500);
--color-primary-hover: var(--color-teal-600);
--color-primary-active: var(--color-teal-700);
--color-secondary: rgba(var(--color-brown-600-rgb), 0.12);
--color-secondary-hover: rgba(var(--color-brown-600-rgb), 0.2);
--color-secondary-active: rgba(var(--color-brown-600-rgb), 0.25);
--color-border: rgba(var(--color-brown-600-rgb), 0.2);
--color-btn-primary-text: var(--color-cream-50);
--color-card-border: rgba(var(--color-brown-600-rgb), 0.12);
--color-card-border-inner: rgba(var(--color-brown-600-rgb), 0.12);
--color-error: var(--color-red-500);
--color-success: var(--color-teal-500);
--color-warning: var(--color-orange-500);
--color-info: var(--color-slate-500);
--color-focus-ring: rgba(var(--color-teal-500-rgb), 0.4);
--color-success-rgb: var(--color-teal-500-rgb);
--color-error-rgb: var(--color-red-500-rgb);
--color-warning-rgb: var(--color-orange-500-rgb);
--color-info-rgb: var(--color-slate-500-rgb);
}

html {
font-size: var(--font-size-base);
font-family: var(--font-family-base);
line-height: var(--line-height-normal);
color: var(--color-text);
background-color: var(--color-background);
-webkit-font-smoothing: antialiased;
box-sizing: border-box;
}

```

```

body {
  margin: 0;
  padding: 0;
}

*,
*::before,
*::after {
  box-sizing: inherit;
}

h1, h2, h3, h4, h5, h6 {
  margin: 0;
  font-weight: var(--font-weight-semibold);
  line-height: var(--line-height-tight);
  color: var(--color-text);
  letter-spacing: var(--letter-spacing-tight);
}

h1 {
  font-size: var(--font-size-4xl);
}

h2 {
  font-size: var(--font-size-3xl);
}

h3 {
  font-size: var(--font-size-2xl);
}

h4 {
  font-size: var(--font-size-xl);
}

h5 {
  font-size: var(--font-size-lg);
}

h6 {
  font-size: var(--font-size-md);
}

p {
  margin: 0 0 var(--space-16) 0;
}

a {
  color: var(--color-primary);
  text-decoration: none;
  transition: color var(--duration-fast) var(--ease-standard);
}

a:hover {
  color: var(--color-primary-hover);
}

```

```

code, pre {
  font-family: var(--font-family-mono);
  font-size: calc(var(--font-size-base) * 0.95);
  background-color: var(--color-secondary);
  border-radius: var(--radius-sm);
}

code {
  padding: var(--space-1) var(--space-4);
}

pre {
  padding: var(--space-16);
  margin: var(--space-16) 0;
  overflow: auto;
  border: 1px solid var(--color-border);
}

pre code {
  background: none;
  padding: 0;
}

.btn {
  display: inline-flex;
  align-items: center;
  justify-content: center;
  padding: var(--space-8) var(--space-16);
  border-radius: var(--radius-base);
  font-size: var(--font-size-base);
  font-weight: 500;
  line-height: 1.5;
  cursor: pointer;
  transition: all var(--duration-normal) var(--ease-standard);
  border: none;
  text-decoration: none;
  position: relative;
}

.btn:focus-visible {
  outline: none;
  box-shadow: var(--focus-ring);
}

.btn--primary {
  background: var(--color-primary);
  color: var(--color-btn-primary-text);
}

.btn--primary:hover {
  background: var(--color-primary-hover);
}

.btn--primary:active {
  background: var(--color-primary-active);
}

.btn--secondary {

```

```

    background: var(--color-secondary);
    color: var(--color-text);
}

.btn--secondary:hover {
    background: var(--color-secondary-hover);
}

.btn--secondary:active {
    background: var(--color-secondary-active);
}

.btn--outline {
    background: transparent;
    border: 1px solid var(--color-border);
    color: var(--color-text);
}

.btn--outline:hover {
    background: var(--color-secondary);
}

.btn--sm {
    padding: var(--space-4) var(--space-12);
    font-size: var(--font-size-sm);
    border-radius: var(--radius-sm);
}

.btn--lg {
    padding: var(--space-10) var(--space-20);
    font-size: var(--font-size-lg);
    border-radius: var(--radius-md);
}

.btn--full-width {
    width: 100%;
}

.btn:disabled {
    opacity: 0.5;
    cursor: not-allowed;
}

.form-control {
    display: block;
    width: 100%;
    padding: var(--space-8) var(--space-12);
    font-size: var(--font-size-md);
    line-height: 1.5;
    color: var(--color-text);
    background-color: var(--color-surface);
    border: 1px solid var(--color-border);
    border-radius: var(--radius-base);
    transition: border-color var(--duration-fast) var(--ease-standard),
    box-shadow var(--duration-fast) var(--ease-standard);
}

textarea.form-control {

```

```

    font-family: var(--font-family-base);
    font-size: var(--font-size-base);
}

select.form-control {
    padding: var(--space-8) var(--space-12);
    -webkit-appearance: none;
    -moz-appearance: none;
    appearance: none;
    background-image: var(--select-caret-light);
    background-repeat: no-repeat;
    background-position: right var(--space-12) center;
    background-size: 16px;
    padding-right: var(--space-32);
}

@media (prefers-color-scheme: dark) {
    select.form-control {
        background-image: var(--select-caret-dark);
    }
}

[data-color-scheme="dark"] select.form-control {
    background-image: var(--select-caret-dark);
}

[data-color-scheme="light"] select.form-control {
    background-image: var(--select-caret-light);
}

.form-control:focus {
    border-color: var(--color-primary);
    outline: var(--focus-outline);
}

.form-label {
    display: block;
    margin-bottom: var(--space-8);
    font-weight: var(--font-weight-medium);
    font-size: var(--font-size-sm);
}

.form-group {
    margin-bottom: var(--space-16);
}

.card {
    background-color: var(--color-surface);
    border-radius: var(--radius-lg);
    border: 1px solid var(--color-card-border);
    box-shadow: var(--shadow-sm);
    overflow: hidden;
    transition: box-shadow var(--duration-normal) var(--ease-standard);
}

.card:hover {
    box-shadow: var(--shadow-md);
}

```

```

.card__body {
  padding: var(--space-16);
}

.card__header, .card__footer {
  padding: var(--space-16);
  border-bottom: 1px solid var(--color-card-border-inner);
}

.status {
  display: inline-flex;
  align-items: center;
  padding: var(--space-6) var(--space-12);
  border-radius: var(--radius-full);
  font-weight: var(--font-weight-medium);
  font-size: var(--font-size-sm);
}

.status--success {
  background-color: rgba(var(--color-success-rgb, 33, 128, 141), var(--status-bg-opacity));
  color: var(--color-success);
  border: 1px solid rgba(var(--color-success-rgb, 33, 128, 141), var(--status-border-opacity));
}

.status--error {
  background-color: rgba(var(--color-error-rgb, 192, 21, 47), var(--status-bg-opacity));
  color: var(--color-error);
  border: 1px solid rgba(var(--color-error-rgb, 192, 21, 47), var(--status-border-opacity));
}

.status--warning {
  background-color: rgba(var(--color-warning-rgb, 168, 75, 47), var(--status-bg-opacity));
  color: var(--color-warning);
  border: 1px solid rgba(var(--color-warning-rgb, 168, 75, 47), var(--status-border-opacity));
}

.status--info {
  background-color: rgba(var(--color-info-rgb, 98, 108, 113), var(--status-bg-opacity));
  color: var(--color-info);
  border: 1px solid rgba(var(--color-info-rgb, 98, 108, 113), var(--status-border-opacity));
}

.container {
  width: 100%;
  margin-right: auto;
  margin-left: auto;
  padding-right: var(--space-16);
  padding-left: var(--space-16);
}

```

```

@media (min-width: 640px) {
  .container {
    max-width: var(--container-sm);
  }
}

@media (min-width: 768px) {
  .container {
    max-width: var(--container-md);
  }
}

@media (min-width: 1024px) {
  .container {
    max-width: var(--container-lg);
  }
}

@media (min-width: 1280px) {
  .container {
    max-width: var(--container-xl);
  }
}

.flex {
  display: flex;
}

.flex-col {
  flex-direction: column;
}

.items-center {
  align-items: center;
}

.justify-center {
  justify-content: center;
}

.justify-between {
  justify-content: space-between;
}

.gap-4 {
  gap: var(--space-4);
}

.gap-8 {
  gap: var(--space-8);
}

.gap-16 {
  gap: var(--space-16);
}

.m-0 {

```

```

    margin: 0;
}

.mt-8 {
    margin-top: var(--space-8);
}

.mb-8 {
    margin-bottom: var(--space-8);
}

.mx-8 {
    margin-left: var(--space-8);
    margin-right: var(--space-8);
}

.my-8 {
    margin-top: var(--space-8);
    margin-bottom: var(--space-8);
}

.p-0 {
    padding: 0;
}

.py-8 {
    padding-top: var(--space-8);
    padding-bottom: var(--space-8);
}

.px-8 {
    padding-left: var(--space-8);
    padding-right: var(--space-8);
}

.py-16 {
    padding-top: var(--space-16);
    padding-bottom: var(--space-16);
}

.px-16 {
    padding-left: var(--space-16);
    padding-right: var(--space-16);
}

.block {
    display: block;
}

.hidden {
    display: none;
}

.sr-only {
    position: absolute;
    width: 1px;
    height: 1px;
    padding: 0;
}

```



```

    margin: -1px;
    overflow: hidden;
    clip: rect(0, 0, 0, 0);
    white-space: nowrap;
    border-width: 0;
}

:focus-visible {
    outline: var(--focus-outline);
    outline-offset: 2px;
}

[data-color-scheme="dark"] .btn--outline {
    border: 1px solid var(--color-border-secondary);
}

.header {
    text-align: center;
    margin-bottom: var(--space-32);
    padding: var(--space-24) 0;
    border-bottom: 1px solid var(--color-border);
}

.header h1 {
    margin-bottom: var(--space-8);
    color: var(--color-text);
}

.header p {
    color: var(--color-text-secondary);
    margin-bottom: 0;
}

.main-content {
    display: grid;
    grid-template-columns: 1.3fr 2fr 1fr;
    gap: var(--space-24);
    align-items: start;
}

.config-panel .card {
    position: sticky;
    top: var(--space-20);
}

.file-input-group {
    display: flex;
    gap: var(--space-8);
    align-items: center;
}

.file-input-group input[type="file"] {
    flex: 1;
}

.config-grid {
    display: grid;
    grid-template-columns: 1fr 1fr;

```

```

    gap: var(--space-16);
    margin: var(--space-20) 0;
}

.button-group {
    display: flex;
    gap: var(--space-8);
    margin-top: var(--space-20);
}

.button-group .btn {
    flex: 1;
}

.visualization-area {
    display: flex;
    flex-direction: column;
    gap: var(--space-16);
}

.map-container {
    background: var(--color-surface);
    border-radius: var(--radius-lg);
    border: 1px solid var(--color-card-border);
    overflow: hidden;
}

.canvas-wrapper {
    position: relative;
    background: var(--color-bg-1);
    overflow: hidden;
}

.map-canvas {
    display: block;
    cursor: grab;
    transition: cursor var(--duration-fast) var(--ease-standard);
}

.map-canvas.panning {
    cursor: grabbing;
}

.map-canvas.zooming {
    cursor: zoom-in;
}

.map-controls {
    display: flex;
    justify-content: space-between;
    align-items: center;
    padding: var(--space-16);
    background: var(--color-surface);
    border-top: 1px solid var(--color-card-border-inner);
}

.control-buttons {
    display: flex;

```

```

    gap: var(--space-8);
}

.map-info {
  display: flex;
  align-items: center;
}

.point-popup {
  position: absolute;
  background: var(--color-surface);
  border: 1px solid var(--color-card-border);
  border-radius: var(--radius-base);
  box-shadow: var(--shadow-lg);
  padding: var(--space-12);
  z-index: 1000;
  min-width: 200px;
  pointer-events: none;
}

.point-popup.hidden {
  display: none;
}

.popup-arrow {
  position: absolute;
  width: 0;
  height: 0;
  border: 6px solid transparent;
  border-top-color: var(--color-surface);
  bottom: -12px;
  left: 50%;
  transform: translateX(-50%);
}

.popup-arrow::before {
  content: '';
  position: absolute;
  width: 0;
  height: 0;
  border: 7px solid transparent;
  border-top-color: var(--color-card-border);
  bottom: 1px;
  left: -7px;
}

.popup-content h4 {
  margin: 0 0 var(--space-8) 0;
  font-size: var(--font-size-md);
  color: var(--color-text);
}

.popup-details {
  display: flex;
  flex-direction: column;
  gap: var(--space-4);
}

```

```

.detail-item {
  display: flex;
  justify-content: space-between;
  align-items: center;
}

.detail-label {
  font-weight: var(--font-weight-medium);
  color: var(--color-text-secondary);
  font-size: var(--font-size-sm);
}

.detail-value {
  font-weight: var(--font-weight-semibold);
  color: var(--color-text);
  font-size: var(--font-size-sm);
}

.legend-panel {
  position: sticky;
  top: var(--space-20);
}

.legend-panel .card__header {
  display: flex;
  justify-content: space-between;
  align-items: center;
}

.legend-controls {
  display: flex;
  gap: var(--space-4);
}

.legend-content {
  margin-bottom: var(--space-16);
}

.legend-placeholder {
  color: var(--color-text-secondary);
  font-style: italic;
  margin: 0;
  text-align: center;
  padding: var(--space-16);
}

.legend-item {
  display: flex;
  align-items: center;
  gap: var(--space-8);
  padding: var(--space-6) 0;
  border-bottom: 1px solid var(--color-card-border-inner);
}

.legend-item:last-child {
  border-bottom: none;
}

```

```

.legend-checkbox {
  margin: 0;
  cursor: pointer;
}

.legend-swatch {
  width: 16px;
  height: 16px;
  border-radius: var(--radius-sm);
  border: 1px solid var(--color-border);
  flex-shrink: 0;
}

.legend-label {
  flex: 1;
  font-size: var(--font-size-sm);
  color: var(--color-text);
  cursor: pointer;
}

.legend-count {
  font-size: var(--font-size-xs);
  color: var(--color-text-secondary);
  font-weight: var(--font-weight-medium);
}

.summary-content {
  border-top: 1px solid var(--color-card-border-inner);
  padding-top: var(--space-16);
  margin-top: var(--space-16);
}

.summary-content h3 {
  margin: 0 0 var(--space-12) 0;
  font-size: var(--font-size-lg);
  color: var(--color-text);
}

.summary-content h4 {
  margin: 0 0 var(--space-8) 0;
  font-size: var(--font-size-md);
  color: var(--color-text);
}

.summary-content p {
  margin: var(--space-4) 0;
  font-size: var(--font-size-sm);
}

.summary-stats {
  display: flex;
  flex-direction: column;
  gap: var(--space-4);
}

.summary-item {
  display: flex;
  justify-content: space-between;
}

```

```

    font-size: var(--font-size-sm);
}

.summary-label {
    color: var(--color-text-secondary);
}

.summary-value {
    font-weight: var(--font-weight-semibold);
    color: var(--color-text);
}

@media (max-width: 1200px) {
    .main-content {
        grid-template-columns: 1fr 1.5fr 280px;
    }
}

@media (max-width: 1024px) {
    .main-content {
        grid-template-columns: 1fr;
        gap: var(--space-20);
    }

    .config-panel .card,
    .legend-panel {
        position: static;
    }

    .visualization-area {
        order: -1;
    }

    .config-grid {
        grid-template-columns: 1fr 1fr 1fr;
    }
}

@media (max-width: 768px) {
    .container {
        padding: var(--space-12);
    }

    .config-grid {
        grid-template-columns: 1fr 1fr;
    }

    .file-input-group {
        flex-direction: column;
        align-items: stretch;
    }

    .button-group {
        flex-direction: column;
    }

    .map-controls {
        flex-direction: column;
    }
}

```

```

        gap: var(--space-8);
        align-items: stretch;
    }

    .control-buttons {
        justify-content: center;
    }

    .legend-controls {
        flex-direction: column;
        width: 100%;
    }

    .legend-panel .card__header {
        flex-direction: column;
        align-items: stretch;
        gap: var(--space-8);
    }
}

@media (max-width: 480px) {
    .config-grid {
        grid-template-columns: 1fr;
    }

    .control-buttons {
        flex-direction: column;
    }
}

.loading {
    opacity: 0.6;
    pointer-events: none;
}

.loading .btn {
    position: relative;
}

.loading .btn::after {
    content: '';
    position: absolute;
    width: 16px;
    height: 16px;
    margin: auto;
    border: 2px solid transparent;
    border-top-color: currentColor;
    border-radius: 50%;
    animation: spin 1s linear infinite;
    top: 0;
    left: 0;
    bottom: 0;
    right: 0;
}

@keyframes spin {
    0% { transform: rotate(0deg); }
    100% { transform: rotate(360deg); }
}

```

```

}

.error {
  color: var(--color-error);
  background-color: rgba(var(--color-error-rgb), 0.1);
  border-color: var(--color-error);
}

.success {
  color: var(--color-success);
  background-color: rgba(var(--color-success-rgb), 0.1);
  border-color: var(--color-success);
}

.interactive-point {
  cursor: pointer;
  transition: all var(--duration-fast) var(--ease-standard);
}

.interactive-point:hover {
  transform: scale(1.2);
}

.legend-checkbox:focus {
  outline: var(--focus-outline);
  outline-offset: 2px;
}

.legend-label:focus {
  outline: var(--focus-outline);
  outline-offset: 2px;
}

.legend-content {
  max-height: 300px;
  overflow-y: auto;
}

.legend-content::-webkit-scrollbar {
  width: 6px;
}

.legend-content::-webkit-scrollbar-track {
  background: var(--color-secondary);
  border-radius: var(--radius-sm);
}

.legend-content::-webkit-scrollbar-thumb {
  background: var(--color-text-secondary);
  border-radius: var(--radius-sm);
}

.legend-content::-webkit-scrollbar-thumb:hover {
  background: var(--color-text);
}

```


MapDVT Javascript Application

```
class MapVisualization {
  constructor() {
    this.canvas = null;
    this.ctx = null;
    this.data = [];
    this.originalData = [];
    this.selectedClasses = new Set([0]);
    this.mapImage = null;
    this.mapBlobUrl = null;
    this.savedFileName = 'raw_map.png';
    this.savedFilePath = null;

    this.config = {
      serverBase: "http://[REDACTED]:8080",
      styleId: "basic-preview/static",
      centerLat: [REDACTED],
      centerLon: [REDACTED],
      zoom: 17,
      width: 1000,
      height: 1000
    };

    this.isDragging = false;
    this.lastMousePos = { x: 0, y: 0 };
    this.viewOffset = { x: 0, y: 0 };
    this.zoomLevel = 1;
    this.minZoom = 0.5;
    this.maxZoom = 3;

    this.initialView = {
      zoomLevel: this.zoomLevel,
      viewOffset: { ...this.viewOffset }
    };

    this._pointHits = [];
    this._hitRadius = 7;
    this._activePoint = null;

    this.colorMap = {
      0: { r: 255, g: 0, b: 0, name: "Red" }
    };

    this.legendLabels = {
      0: "Mesh Points"
    };

    this.sampleData = [{ class: 0, lat: [REDACTED], lon: [REDACTED] }, { class:
0, lat: [REDACTED], lon: [REDACTED] }, { class: 0, lat: [REDACTED], lon: [REDACTED] }];

    this.overlay = {
      padding: 12,
      cornerRadius: 10,
      bgColor: 'rgba(255,255,255,0.85)',
      borderColor: 'rgba(0,0,0,0.15)',
      borderWidth: 1,
      titleFont: 'bold 18px Arial',

```

```

        textFont: '14px Arial',
        textColor: '#111',
        smallFont: '12px Arial',
        watermarkFont: 'bold 16px Arial',
        watermarkColor: 'rgba(0,0,0,0.75)',
        shadow: '0 2px 6px rgba(0,0,0,0.15)'
    };

    this.init();
}

init() {
    this.setupElements();
    this.bindEvents();
    this.loadConfigFromForm();
    this.initialView = {
        zoomLevel: 1,
        viewOffset: { x: 0, y: 0 }
    };
    this.updateStatus('Ready - Load data to begin');
}

setupElements() {
    this.canvas = document.getElementById('mapCanvas');
    this.ctx = this.canvas.getContext('2d');
    this.resizeCanvas();

    this.elements = {
        csvFile: document.getElementById('csvFile'),
        loadSampleBtn: document.getElementById('loadSampleBtn'),
        processBtn: document.getElementById('processBtn'),
        resetBtn: document.getElementById('resetBtn'),
        rerenderBtn: document.getElementById('rerenderBtn'),
        downloadBtn: document.getElementById('downloadBtn'),
        resetZoomBtn: document.getElementById('resetZoomBtn'),
        selectAllBtn: document.getElementById('selectAllBtn'),
        deselectAllBtn: document.getElementById('deselectAllBtn'),
        legendContent: document.getElementById('legendContent'),
        summaryContent: document.getElementById('summaryContent'),
        mapStatus: document.getElementById('mapStatus'),
        pointPopup: document.getElementById('pointPopup'),
        popupClass: document.getElementById('popupClass'),
        popupLat: document.getElementById('popupLat'),
        popupLon: document.getElementById('popupLon'),
        serverBase: document.getElementById('serverBase'),
        centerLat: document.getElementById('centerLat'),
        centerLon: document.getElementById('centerLon'),
        zoom: document.getElementById('zoom'),
        width: document.getElementById('width'),
        height: document.getElementById('height')
    };
}

bindEvents() {
    this.elements.csvFile.addEventListener('change', (e) =>
this.handleFileUpload(e));
    this.elements.loadSampleBtn.addEventListener('click', () =>
this.loadSampleData());
}

```

```

        this.elements.processBtn.addEventListener('click', () =>
this.processData());
        this.elements.resetBtn.addEventListener('click', () =>
this.resetApplication());
        this.elements.rerenderBtn.addEventListener('click', () =>
this.rerender());
        this.elements.downloadBtn.addEventListener('click', () =>
this.downloadImage());
        this.elements.selectAllBtn.addEventListener('click', () =>
this.selectAllClasses());
        this.elements.deselectAllBtn.addEventListener('click', () =>
this.deselectAllClasses());

        if (this.elements.resetZoomBtn) {
            this.elements.resetZoomBtn.addEventListener('click', () =>
this.resetZoom());
        }

        this.canvas.addEventListener('mousedown', (e) =>
this.handleMouseDown(e));
        this.canvas.addEventListener('mousemove', (e) =>
this.handleMouseMove(e));
        this.canvas.addEventListener('mouseup', () =>
this.handleMouseUp());
        this.canvas.addEventListener('wheel', (e) => this.handleWheel(e));
        this.canvas.addEventListener('click', (e) => this.handleClick(e));
        this.canvas.addEventListener('contextmenu', (e) =>
e.preventDefault());

        document.addEventListener('click', (e) => {
            if (!this.canvas.contains(e.target)
&& !this.elements.pointPopup.contains(e.target)) {
                this.hidePopup();
            }
        });
    }

    loadConfigFromForm() {
        if (this.elements.serverBase && this.elements.serverBase.value) {
            this.config.serverBase = this.elements.serverBase.value;
        }
        if (this.elements.centerLat && this.elements.centerLat.value) {
            this.config.centerLat =
parseFloat(this.elements.centerLat.value);
        }
        if (this.elements.centerLon && this.elements.centerLon.value) {
            this.config.centerLon =
parseFloat(this.elements.centerLon.value);
        }
        if (this.elements.zoom && this.elements.zoom.value) {
            this.config.zoom = parseInt(this.elements.zoom.value);
        }
        if (this.elements.width && this.elements.width.value) {
            this.config.width = parseInt(this.elements.width.value);
        }
        if (this.elements.height && this.elements.height.value) {
            this.config.height = parseInt(this.elements.height.value);
        }
    }

```

```

    }

    resizeCanvas() {
      this.canvas.width = this.config.width;
      this.canvas.height = this.config.height;
      this.canvas.style.maxWidth = '100%';
      this.canvas.style.height = 'auto';
      this.canvas.style.border = '1px solid #ccc';
    }

    buildRemoteMapUrl() {
      const { serverBase, styleId, centerLat, centerLon, zoom, width,
height } = this.config;
      return
`${serverBase}/styles/${styleId}/${centerLon},${centerLat},${zoom}/${width}
x${height}.png`;
    }

    async fetchAndSaveMapImage() {
      const url = this.buildRemoteMapUrl();
      this.updateStatus('Downloading base map image...', 'info');

      const controller = new AbortController();
      const timeoutId = setTimeout(() => controller.abort(), 15000);

      try {
        const res = await fetch(url, {
          mode: 'cors',
          signal: controller.signal
        });

        if (!res.ok) {
          throw new Error(`Map fetch failed: ${res.status}
${res.statusText}`);
        }

        const blob = await res.blob();

        if (this.mapBlobUrl) URL.revokeObjectURL(this.mapBlobUrl);
        this.mapBlobUrl = URL.createObjectURL(blob);

        const a = document.createElement('a');
        a.href = this.mapBlobUrl;
        a.download = this.savedFileName;
        document.body.appendChild(a);
        a.click();
        document.body.removeChild(a);

        this.updateStatus('Base map downloaded as raw_map.png and ready
for rendering', 'success');
        return blob;
      } finally {
        clearTimeout(timeoutId);
      }
    }

    async loadMapFromLocal(blobOrUrl) {
      return new Promise((resolve, reject) => {

```

```

        const img = new Image();
        img.onload = () => {
            this.mapImage = img;
            resolve();
        };
        img.onerror = () => {
            this.mapImage = null;
            reject(new Error('Failed to load local map image'));
        };
        img.src = typeof blobOrUrl === 'string' ? blobOrUrl :
URL.createObjectURL(blobOrUrl);
    });
}

async processData() {
    if (this.data.length === 0) {
        this.updateStatus('No data to process - please load data
first', 'error');
        return;
    }

    this.loadConfigFromForm();
    this.resizeCanvas();
    this.updateStatus('Creating map visualization...', 'info');
    this.setLoading(true);

    try {
        const blob = await this.fetchAndSaveMapImage();
        await this.loadMapFromLocal(this.mapBlobUrl);

        this.renderMap();
        this.createLegend();
        this.updateSummary();
        this.enableControls();

        if (this.mapImage) {
            this.updateStatus('Map visualization ready', 'success');
        } else {
            this.updateStatus('Map visualization ready (using fallback
background)', 'warning');
        }
    } catch (error) {
        console.error(error);
        this.updateStatus(`Error: ${error.message}`, 'error');

        this.mapImage = null;
        this.renderMap();
        this.createLegend();
        this.updateSummary();
    } finally {
        this.setLoading(false);
    }
}

lonLatToPixel(lon, lat) {
    const { centerLon, centerLat, zoom, width, height } = this.config;
    const tileSize = 256;
    const scale = tileSize * Math.pow(2, zoom);

```

```

function project(lonDeg, latDeg) {
    const x = (lonDeg + 180) / 360 * scale;
    const sinLat = Math.sin(latDeg * Math.PI / 180);
    const y = (0.5 - Math.log((1 + sinLat) / (1 - sinLat))) / (4 *
Math.PI)) * scale;
    return { x, y };
}

const centerProj = project(centerLon, centerLat);
const pointProj = project(lon, lat);
const dx = pointProj.x - centerProj.x;
const dy = pointProj.y - centerProj.y;

return {
    x: width / 2 + dx,
    y: height / 2 + dy
};
}

renderMap() {
    this.ctx.clearRect(0, 0, this.canvas.width, this.canvas.height);
    this.ctx.save();
    this.ctx.scale(this.zoomLevel, this.zoomLevel);
    this.ctx.translate(this.viewOffset.x / this.zoomLevel,
this.viewOffset.y / this.zoomLevel);

    if (this.mapImage) {
        this.ctx.drawImage(this.mapImage, 0, 0, this.canvas.width,
this.canvas.height);
    } else {
        this.createMapBackground();
    }

    this._pointHits = [];
    this.drawDataPoints();

    this.drawSummaryBoxOnCanvas();
    this.drawLegendOnCanvas();
    this.drawWatermark();

    this.ctx.restore();
}

createMapBackground() {
    this.ctx.fillStyle = '#f0f8ff';
    this.ctx.fillRect(0, 0, this.canvas.width, this.canvas.height);

    this.ctx.strokeStyle = '#e0e0e0';
    this.ctx.lineWidth = 1;
    const gridSize = 50;

    for (let x = 0; x <= this.canvas.width; x += gridSize) {
        this.ctx.beginPath();
        this.ctx.moveTo(x, 0);
        this.ctx.lineTo(x, this.canvas.height);
        this.ctx.stroke();
    }
}

```

```

for (let y = 0; y <= this.canvas.height; y += gridSize) {
  this.ctx.beginPath();
  this.ctx.moveTo(0, y);
  this.ctx.lineTo(this.canvas.width, y);
  this.ctx.stroke();
}

this.ctx.fillStyle = '#666';
this.ctx.font = '16px Arial';
this.ctx.textAlign = 'center';
this.ctx.fillText('Map not loaded - using fallback grid',
this.canvas.width / 2, this.canvas.height / 2);
}

drawDataPoints() {
  for (const point of this.data) {
    if (!this.selectedClasses.has(point.class)) continue;

    const { r, g, b } = this.colorMap[point.class] || { r: 255, g:
0, b: 0 };
    const pixel = this.lonLatToPixel(point.lon, point.lat);

    this.ctx.fillStyle = `rgb(${r},${g},${b})`;
    this.ctx.beginPath();
    this.ctx.arc(pixel.x, pixel.y, 5, 0, Math.PI * 2);
    this.ctx.fill();

    this._pointHits.push({
      x: pixel.x,
      y: pixel.y,
      r: this._hitRadius,
      data: point
    });
  }
}

drawWatermark() {
  const pad = 16;
  const text = 'BSEEE FYDP Group 6, 2025';
  this.ctx.save();
  this.ctx.font = this.overlay.watermarkFont;
  this.ctx.textBaseline = 'alphabetic';
  this.ctx.textAlign = 'left';

  const metrics = this.ctx.measureText(text);
  const height = 22;
  const width = metrics.width + 12;
  const x = pad - 6;
  const y = this.canvas.height - pad - height;

  this.roundRect(x, y, width, height + 8, 8, this.overlay.bgColor,
this.overlay.borderColor);

  this.ctx.fillStyle = this.overlay.watermarkColor;
  this.ctx.fillText(text, pad, this.canvas.height - pad);

  this.ctx.restore();
}

```

```

}

drawLegendOnCanvas() {
  const pad = this.overlay.padding;
  const swatch = 14;
  const gap = 8;

  this.ctx.save();
  this.ctx.font = this.overlay.titleFont;
  const title = 'Mesh Points';
  const titleW = this.ctx.measureText(title).width;

  this.ctx.font = this.overlay.textFont;
  const count = this.data.filter(p => p.class === 0 &&
this.selectedClasses.has(0)).length;
  const text = `Detections: ${count}`;
  const textW = this.ctx.measureText(text).width + swatch + gap;

  const lineH = 20;
  const headerH = 26;
  const boxW = pad * 2 + Math.max(titleW, textW);
  const boxH = pad * 2 + headerH + lineH;

  const x = this.canvas.width - boxW - pad;
  const y = pad;

  this.roundRect(x, y, boxW, boxH, this.overlay.cornerRadius,
this.overlay.bgColor, this.overlay.borderColor);

  this.ctx.fillStyle = this.overlay.textColor;
  this.ctx.font = this.overlay.titleFont;
  this.ctx.textAlign = 'left';
  this.ctx.textBaseline = 'top';
  this.ctx.fillText(title, x + pad, y + pad);

  this.ctx.font = this.overlay.textFont;
  const ly = y + pad + headerH - 2;

  const { r, g, b } = this.colorMap[0];
  this.ctx.fillStyle = `rgb(${r},${g},${b})`;
  this.ctx.fillRect(x + pad, ly + 3, swatch, swatch);
  this.ctx.strokeStyle = 'rgba(0,0,0,0.25)';
  this.ctx.strokeRect(x + pad, ly + 3, swatch, swatch);

  this.ctx.fillStyle = this.overlay.textColor;
  this.ctx.fillText(text, x + pad + swatch + gap, ly);

  this.ctx.restore();
}

drawSummaryBoxOnCanvas() {
  const pad = this.overlay.padding;

  this.ctx.save();
  const title = 'Path Tracer Data Points';
  this.ctx.font = this.overlay.titleFont;
  const titleW = this.ctx.measureText(title).width;

```



```

    this.ctx.font = this.overlay.textFont;
    const count = this.data.filter(p => p.class === 0).length;
    const line = `Total: ${count}`;
    const lineW = this.ctx.measureText(line).width;

    const maxW = Math.max(titleW, lineW);
    const lineH = 20;
    const headerH = 26;
    const boxW = pad * 2 + maxW;
    const boxH = pad * 2 + headerH + lineH;

    const x = pad;
    const y = pad;

    this.roundRect(x, y, boxW, boxH, this.overlay.cornerRadius,
this.overlay.bgColor, this.overlay.borderColor);

    this.ctx.fillStyle = this.overlay.textColor;
    this.ctx.font = this.overlay.titleFont;
    this.ctx.textAlign = 'left';
    this.ctx.textBaseline = 'top';
    this.ctx.fillText(title, x + pad, y + pad);

    this.ctx.font = this.overlay.textFont;
    const ly = y + pad + headerH - 2;
    this.ctx.fillText(line, x + pad, ly);

    this.ctx.restore();
}

roundRect(x, y, w, h, r, fillStyle, strokeStyle) {
    const ctx = this.ctx;
    ctx.save();
    ctx.beginPath();
    const radius = Math.min(r, Math.abs(w) / 2, Math.abs(h) / 2);
    ctx.moveTo(x + radius, y);
    ctx.arcTo(x + w, y, x + w, y + h, radius);
    ctx.arcTo(x + w, y + h, x, y + h, radius);
    ctx.arcTo(x, y + h, x, y, radius);
    ctx.arcTo(x, y, x + w, y, radius);
    ctx.closePath();

    if (fillStyle) {
        ctx.fillStyle = fillStyle;
        ctx.fill();
    }
    if (strokeStyle) {
        ctx.strokeStyle = strokeStyle;
        ctx.lineWidth = this.overlay.borderWidth;
        ctx.stroke();
    }
    ctx.restore();
}

createLegend() {
    const container = this.elements.legendContent;
    container.innerHTML = '';

```

```

const div = document.createElement('label');
div.className = 'legend-item';
div.style.cursor = 'pointer';

const checkbox = document.createElement('input');
checkbox.type = 'checkbox';
checkbox.className = 'legend-checkbox';
checkbox.checked = this.selectedClasses.has(0);
checkbox.addEventListener('change', () => {
  if (checkbox.checked) this.selectedClasses.add(0);
  else this.selectedClasses.delete(0);
  this.renderMap();
  this.updateSummary();
});

const swatch = document.createElement('span');
swatch.className = 'legend-swatch';
swatch.style.backgroundColor = `rgb(255,0,0)`;

const label = document.createElement('span');
label.className = 'legend-label';
label.textContent = 'Mesh Points';

div.appendChild(checkbox);
div.appendChild(swatch);
div.appendChild(label);
container.appendChild(div);
}

selectAllClasses() {
  this.selectedClasses = new Set([0]);
  this.createLegend();
  this.renderMap();
  this.updateSummary();
}

deselectAllClasses() {
  this.selectedClasses.clear();
  this.createLegend();
  this.renderMap();
  this.updateSummary();
}

haversineDistance(lat1, lon1, lat2, lon2) {
  const R = 6371000;
  const phi1 = lat1 * Math.PI / 180;
  const phi2 = lat2 * Math.PI / 180;
  const dphi = (lat2 - lat1) * Math.PI / 180;
  const dlambd = (lon2 - lon1) * Math.PI / 180;

  const a = Math.sin(dphi / 2) * Math.sin(dphi / 2) +
    Math.cos(phi1) * Math.cos(phi2) *
    Math.sin(dlambd / 2) * Math.sin(dlambd / 2);
  const c = 2 * Math.atan2(Math.sqrt(a), Math.sqrt(1 - a));

  return R * c;
}

```

```

    updateSummary() {
      const totalPoints = this.data.length;

      let summaryHTML = '<h3>Dataset Statistics</h3>';

      if (totalPoints === 0) {
        summaryHTML += '<p>No data loaded</p>';
      } else if (totalPoints === 1) {
        summaryHTML += '<p><strong>Total Detections:</strong>
${totalPoints}</p>`;
        summaryHTML += '<p><em>Need at least 2 points for distance
calculations</em></p>`;
      } else {
        const distances = [];
        for (let i = 0; i < totalPoints - 1; i++) {
          const dist = this.haversineDistance(
            this.data[i].lat, this.data[i].lon,
            this.data[i + 1].lat, this.data[i + 1].lon
          );
          distances.push(dist);
        }

        const avgDistance = distances.reduce((a, b) => a + b, 0) /
distances.length;
        const pathDistance = distances.reduce((sum, dist) => sum +
dist, 0);

        const deviations = distances.map(d => Math.abs(d -
avgDistance));
        const avgDeviation = deviations.reduce((a, b) => a + b, 0) /
deviations.length;

        summaryHTML += '<p><strong>Total Detections:</strong>
${totalPoints}</p>`;
        summaryHTML += '<p><strong>Avg Distance Between
Points:</strong> ${avgDistance.toFixed(2)} m</p>`;

        if (pathDistance > 1000) {
          summaryHTML += '<p><strong>First to Last Distance:</strong>
${(pathDistance / 1000).toFixed(3)} km</p>`;
        } else {
          summaryHTML += '<p><strong>First to Last Distance:</strong>
${pathDistance.toFixed(2)} m</p>`;
        }

        summaryHTML += '<p><strong>Avg Deviation:</strong>
${avgDeviation.toFixed(2)} m</p>`;
      }

      this.elements.summaryContent.innerHTML = summaryHTML;
    }

    handleFileUpload(event) {
      const file = event.target.files[0];
      if (!file) return;
      const reader = new FileReader();
      reader.onload = (e) => {
        try {

```

```

        const content = e.target.result;
        this.parseCSV(content);
        this.updateStatus(`Loaded ${this.data.length} data points
from file`, 'success');
        this.updateSummary();
        this.elements.processBtn.disabled = false;
        this.elements.resetBtn.disabled = false;
    } catch (error) {
        this.updateStatus(`Error loading file: ${error.message}`,
'error');
    }
    };
    reader.readAsText(file);
}

parseCSV(content) {
    const lines = content.trim().split('\n');
    const data = [];

    for (let i = 1; i < lines.length; i++) {
        const line = lines[i].trim();
        if (!line) continue;

        const parts = line.split(',');
        if (parts.length >= 3) {
            const classVal = parseInt(parts[0]);
            const lat = parseFloat(parts[1]);
            const lon = parseFloat(parts[2]);

            if (!isNaN(classVal) &&& !isNaN(lat)
&&& !isNaN(lon)) {
                if (classVal === 0) {
                    data.push({ class: 0, lat, lon });
                }
            }
        }
    }

    this.data = data;
    this.originalData = [...data];

    if (this.data.length > 0) {
        const avgLat = this.data.reduce((sum, p) => sum + p.lat, 0) /
this.data.length;
        const avgLon = this.data.reduce((sum, p) => sum + p.lon, 0) /
this.data.length;
        this.config.centerLat = avgLat;
        this.config.centerLon = avgLon;

        this.elements.centerLat.value = avgLat.toFixed(6);
        this.elements.centerLon.value = avgLon.toFixed(6);
    }
}

loadSampleData() {
    this.data = [...this.sampleData];
}

```

```

    this.originalData = [...this.sampleData];
    this.updateStatus('Loaded sample data', 'success');
    this.updateSummary();
    this.elements.processBtn.disabled = false;
    this.elements.resetBtn.disabled = false;
  }

  rerender() {
    if (this.data.length === 0) {
      this.updateStatus('No data to render', 'warning');
      return;
    }
    this.renderMap();
    this.createLegend();
    this.updateSummary();
    this.updateStatus('Map re-rendered', 'success');
  }

  resetApplication() {
    this.data = [];
    this.originalData = [];
    this.selectedClasses = new Set([0]);
    this.mapImage = null;

    if (this.mapBlobUrl) {
      URL.revokeObjectURL(this.mapBlobUrl);
      this.mapBlobUrl = null;
    }

    this.viewOffset = { x: 0, y: 0 };
    this.zoomLevel = 1;
    this.initialView = {
      zoomLevel: 1,
      viewOffset: { x: 0, y: 0 }
    };

    this.elements.csvFile.value = '';

    this.ctx.clearRect(0, 0, this.canvas.width, this.canvas.height);
    this.elements.legendContent.innerHTML = '<p class="legend-
placeholder">Load data to show legend</p>';
    this.elements.summaryContent.innerHTML = '<p>No data loaded</p>';

    this.disableControls();
    this.updateStatus('Application reset', 'info');
  }

  downloadImage() {
    if (!this.mapImage && this.data.length === 0) {
      this.updateStatus('Nothing to download', 'warning');
      return;
    }

    try {
      const link = document.createElement('a');
      link.download = 'map_visualization.png';

```

```

        link.href = this.canvas.toDataURL('image/png');
        link.click();
        this.updateStatus('Image downloaded', 'success');
    } catch (error) {
        this.updateStatus(`Download failed: ${error.message}`,
'error');
    }
}

resetZoom() {
    this.zoomLevel = this.initialView.zoomLevel;
    this.viewOffset = { ...this.initialView.viewOffset };
    this.renderMap();
    this.updateStatus('Zoom reset', 'info');
}

handleMouseDown(e) {
    const rect = this.canvas.getBoundingClientRect();
    this.isDragging = true;
    this.lastMousePos = {
        x: e.clientX - rect.left,
        y: e.clientY - rect.top
    };
    this.canvas.style.cursor = 'grabbing';
}

handleMouseMove(e) {
    if (!this.isDragging) return;

    const rect = this.canvas.getBoundingClientRect();
    const currentX = e.clientX - rect.left;
    const currentY = e.clientY - rect.top;

    const dx = currentX - this.lastMousePos.x;
    const dy = currentY - this.lastMousePos.y;

    this.viewOffset.x += dx;
    this.viewOffset.y += dy;

    this.lastMousePos = { x: currentX, y: currentY };
    this.renderMap();
}

handleMouseUp() {
    this.isDragging = false;
    this.canvas.style.cursor = 'grab';
}

handleWheel(e) {
    e.preventDefault();

    const rect = this.canvas.getBoundingClientRect();
    const mouseX = e.clientX - rect.left;
    const mouseY = e.clientY - rect.top;

```

```

    const delta = e.deltaY > 0 ? 0.9 : 1.1;
    const newZoom = Math.max(this.minZoom, Math.min(this.maxZoom,
this.zoomLevel * delta));

    const scale = newZoom / this.zoomLevel;
    this.viewOffset.x = mouseX - (mouseX - this.viewOffset.x) * scale;
    this.viewOffset.y = mouseY - (mouseY - this.viewOffset.y) * scale;

    this.zoomLevel = newZoom;
    this.renderMap();
}

handleClick(e) {
    const rect = this.canvas.getBoundingClientRect();
    const clickX = (e.clientX - rect.left - this.viewOffset.x) /
this.zoomLevel;
    const clickY = (e.clientY - rect.top - this.viewOffset.y) /
this.zoomLevel;

    for (const hit of this._pointHits) {
        const dx = clickX - hit.x;
        const dy = clickY - hit.y;
        const dist = Math.sqrt(dx * dx + dy * dy);

        if (dist <= hit.r) {
            this.showPopup(hit.data, e.clientX, e.clientY);
            return;
        }
    }

    this.hidePopup();
}

showPopup(point, x, y) {
    this._activePoint = point;
    this.elements.popupClass.textContent =
this.legendLabels[point.class] || `Class ${point.class}`;
    this.elements.popupLat.textContent = point.lat.toFixed(6);
    this.elements.popupLon.textContent = point.lon.toFixed(6);

    this.elements.pointPopup.style.display = 'block';
    this.elements.pointPopup.style.left = `${x + 10}px`;
    this.elements.pointPopup.style.top = `${y + 10}px`;
}

hidePopup() {
    this.elements.pointPopup.style.display = 'none';
    this._activePoint = null;
}

enableControls() {
    this.elements.processBtn.disabled = false;
    this.elements.rerenderBtn.disabled = false;
    this.elements.downloadBtn.disabled = false;
    this.elements.resetZoomBtn.disabled = false;
    this.elements.selectAllBtn.disabled = false;
}

```

```

        this.elements.deselectAllBtn.disabled = false;
    }

    disableControls() {
        this.elements.rerenderBtn.disabled = true;
        this.elements.downloadBtn.disabled = true;
        this.elements.resetZoomBtn.disabled = true;
    }

    setLoading(isLoading) {
        this.elements.processBtn.disabled = isLoading;
        if (isLoading) {
            this.elements.processBtn.textContent = 'Processing...';
        } else {
            this.elements.processBtn.textContent = 'Process & Render
Map';
        }
    }

    updateStatus(message, type = 'info') {
        const status = this.elements.mapStatus;
        status.textContent = message;
        status.className = `status status-${type}`;
    }
}

document.addEventListener('DOMContentLoaded', () => {
    window.mapViz = new MapVisualization();
});

```


Deep Exploration Utility

```
import os
import csv
import math
import base64
import requests
from pathlib import Path
from PIL import Image, ImageDraw, ImageFont
import pandas as pd
import streamlit as st
import folium
from streamlit_folium import folium_static
from folium.features import DivIcon
from io import BytesIO

SMP_CSV_PATH =
"C:\\Users\\[REDACTED]\\FYDP\\2_EEE499C\\C_things\\DeepUtil\\path.csv"
DEBRIS_CSV_PATH =
"C:\\Users\\[REDACTED]\\FYDP\\2_EEE499C\\C_things\\DeepUtil\\locdat.csv"

LAT_FIELD_1 = "lat"
LON_FIELD_1 = "lon"
CLASS_FIELD_1 = "class"

LAT_FIELD_2 = "lat"
LON_FIELD_2 = "lon"
CLASS_FIELD_2 = "class"

CENTER_LON = [REDACTED]
CENTER_LAT = [REDACTED]
ZOOM = 18

COLOR_DEBRIS = (255, 215, 0, 255)
RED_POINT = (255, 0, 0, 255)

DEBRIS_LABEL = "Debris"
DEBRIS_CLASS = 0

LOCAL_BASE_TILES = "http://[REDACTED]:8080/styles/basic-preview/{z}/{x}/{y}.png"

IMAGE_ROOT = "C:\\Users\\[REDACTED]\\FYDP\\2_EEE499C\\C_things\\DeepUtil\\"

def rgba_to_hex(rgba):
    r, g, b, a = rgba
    return f"#{r:02x}{g:02x}{b:02x}"

def image_to_base64(image_path, target_size=(400, 400)):
    try:
        if not os.path.exists(image_path):
            return None

        img = Image.open(image_path)
        img = img.resize(target_size, Image.Resampling.LANCZOS)

        buffered = BytesIO()
        img.save(buffered, format="PNG")
        img_str = base64.b64encode(buffered.getvalue()).decode()
```

```

        return img_str
    except Exception as e:
        print(f"Error converting image: {e}")
        return None

def load_dataframe(csv_path, lat_field, lon_field, class_field=None):
    df = pd.read_csv(csv_path)
    df = df.rename(
        columns={
            lat_field: "LAT",
            lon_field: "LON",
            class_field: "CLASS" if class_field else "CLASS",
        }
    )
    df["LAT"] = pd.to_numeric(df["LAT"], errors="coerce")
    df["LON"] = pd.to_numeric(df["LON"], errors="coerce")
    if "CLASS" in df.columns:
        df["CLASS"] = pd.to_numeric(df["CLASS"],
errors="coerce").astype("Int64")
    df = df.dropna(subset=["LAT", "LON"])
    if "CLASS" not in df.columns:
        df["CLASS"] = 0
    return df

def make_folium_map(center_lat: float, center_lon: float, zoom_start: int =
15) -> folium.Map:
    m = folium.Map(
        location=[center_lat, center_lon],
        zoom_start=zoom_start,
        control_scale=True,
        tiles=None,
    )

    base_tiles = folium.TileLayer(
        tiles=LOCAL_BASE_TILES,
        attr="Local Base Tiles",
        name="Local Base",
        overlay=False,
        control=True,
        min_zoom=0,
        max_zoom=22,
    )
    base_tiles.add_to(m)

    folium.LayerControl(position="topright", collapsed=False).add_to(m)

    return m

def add_small_red_dots_layer(m: folium.Map, df: pd.DataFrame):
    for _, row in df.iterrows():
        lat = float(row["LAT"])
        lon = float(row["LON"])
        folium.CircleMarker(
            location=(lat, lon),
            radius=4,
            color="#ff0000",
            fill=True,
            fill_color="#ff0000",

```

```

        fill_opacity=0.9,
        opacity=0.9,
    ).add_to(m)

def add_clickable_debris_layer(m: folium.Map, df: pd.DataFrame,
    debris_class=DEBRIS_CLASS):
    debris = df[df["CLASS"] == debris_class].copy().reset_index(drop=True)

    for idx, row in debris.iterrows():
        lat = float(row["LAT"])
        lon = float(row["LON"])
        cls = int(row["CLASS"])
        label = "Debris"

        point_number = idx + 1
        image_path = os.path.join(IMAGE_ROOT, f"{point_number}.png")

        img_base64 = image_to_base64(image_path)

        if img_base64:
            popup_html = f"""
            <div style="font-family:Arial; font-size:13px; width:420px;">
                <div style="margin-bottom:10px;">
                    <b>Class:</b> {label}<br/>
                    <b>Class ID:</b> {cls}<br/>
                    <b>Latitude:</b> {lat:.6f}<br/>
                    <b>Longitude:</b> {lon:.6f}
                </div>
                <div style="text-align:center;">
                    
                </div>
            </div>
            """
        else:
            popup_html = f"""
            <div style="font-family:Arial; font-size:13px;">
                <b>Class:</b> {label}<br/>
                <b>Class ID:</b> {cls}<br/>
                <b>Latitude:</b> {lat:.6f}<br/>
                <b>Longitude:</b> {lon:.6f}<br/>
                <br/>
                <span style="color:red;">Image not found:
{point_number}.png</span>
            </div>
            """

        color_hex = rgba_to_hex(COLOR_DEBRIS)
        folium.CircleMarker(
            location=(lat, lon),
            radius=7,
            color="#000000",
            weight=1,
            fill=True,
            fill_color=color_hex,
            fill_opacity=0.95,

```

```

        popup=folium.Popup(popup_html, max_width=450),
    ).add_to(m)

def add_overlay_legend_debris(m: folium.Map):
    legend_html = f"""
    <div id="overlay-legend" style="
        position: absolute;
        top: 60px;
        right: 10px;
        z-index: 1000;
        background: rgba(255,255,255,0.95);
        border: 2px solid #666;
        padding: 8px 12px;
        border-radius: 6px;
        box-shadow: 0 2px 6px rgba(0,0,0,0.4);
        font-family: Arial, sans-serif;
    ">
        <div style="display:flex; align-items:center; gap: 6px;">
            <div style="width:14px; height:14px; background-
color:#{COLOR_DEBRIS[0]:02x}{COLOR_DEBRIS[1]:02x}{COLOR_DEBRIS[2]:02x};
border:1px solid #000; border-radius:50%;"></div>
            <span style="font-size:12px; font-weight:bold;">Debris (Class
{DEBRIS_CLASS})</span>
        </div>
    </div>
    """
    m.get_root().html.add_child(folium.Element(legend_html))

def sidebar_analytics(df_smp: pd.DataFrame, df_debris: pd.DataFrame):
    st.sidebar.header("Analytics Options")

    total_path_points = len(df_smp)
    total_detections = len(df_debris[df_debris["CLASS"] == DEBRIS_CLASS])

    st.sidebar.subheader("Statistics")
    st.sidebar.markdown(f"**Total Path Points:** {total_path_points}")
    st.sidebar.markdown(f"**Total Detections:** {total_detections}")

    st.sidebar.markdown("---")

    st.sidebar.subheader("Map Settings")
    center_lat = st.sidebar.number_input("Center LAT", value=CENTER_LAT,
format="%.6f")
    center_lon = st.sidebar.number_input("Center LON", value=CENTER_LON,
format="%.6f")
    zoom_start = st.sidebar.slider("Zoom", min_value=3, max_value=20,
value=ZOOM)

    return center_lat, center_lon, zoom_start

def build_and_render():
    if not os.path.exists(SMP_CSV_PATH):
        st.error(f"First CSV not found: {SMP_CSV_PATH}")
        return
    if not os.path.exists(DEBRIS_CSV_PATH):
        st.error(f"Second CSV not found: {DEBRIS_CSV_PATH}")
        return

```

```

    df_smp = load_dataframe(SMP_CSV_PATH, LAT_FIELD_1, LON_FIELD_1,
CLASS_FIELD_1)
    df_debris = load_dataframe(DEBRIS_CSV_PATH, LAT_FIELD_2, LON_FIELD_2,
CLASS_FIELD_2)

    center_lat, center_lon, zoom_start = sidebar_analytics(df_smp,
df_debris)

    m = make_folium_map(center_lat, center_lon, zoom_start=zoom_start)

    add_small_red_dots_layer(m, df_smp)
    add_clickable_debris_layer(m, df_debris, debris_class=DEBRIS_CLASS)
    add_overlay_legend_debris(m)

    folium_static(m, width=None, height=1000)

def main():
    st.set_page_config(
        page_title="MapLab Group 6",
        layout="wide",
        initial_sidebar_state="expanded",
    )

    st.markdown(
        """
        <style>
        header[data-testid="stHeader"] {
            display: none;
        }

        .stDeployButton {
            display: none;
        }

        .block-container {
            padding-top: 0.5rem;
            padding-bottom: 0rem;
            padding-left: 2rem;
            padding-right: 2rem;
            max-width: 100%;
        }

        h1 {
            margin-top: 0rem;
            margin-bottom: 0.3rem;
            font-size: 1.5rem;
            padding-bottom: 0.3rem;
        }

        [data-testid="stSidebar"] {
            min-width: 240px;
            max-width: 280px;
        }

        [data-testid="stSidebar"] > div:first-child {
            padding-top: 0.5rem;
        }
        """
    )

```

```

        .element-container {
            margin-bottom: 0rem;
        }

        .sidebar .sidebar-content h2 {
            font-size: 1rem;
            margin-top: 0.3rem;
            margin-bottom: 0.3rem;
        }

        .sidebar .sidebar-content h3 {
            font-size: 0.95rem;
            margin-top: 0.3rem;
            margin-bottom: 0.3rem;
        }

        .sidebar .markdown-text-container {
            font-size: 0.85rem;
        }

        iframe {
            height: 1020px !important;
            border: none;
        }

        footer {
            display: none;
        }

        .stNumberInput, .stSlider {
            margin-bottom: 0.3rem;
        }

        .main .block-container {
            padding-top: 0.5rem;
        }

        @-moz-document url-prefix() {
            .block-container {
                padding-top: 0.3rem;
            }
        }
    </style>
    """
    unsafe_allow_html=True,
)

st.title("Map Lab by FYDP Group 6")

build_and_render()

if __name__ == "__main__":
    main()

```

FYDP (P) Spring 2025 Summary of Team Log Book/Journal

Project Title: Computer Vision Powered Autonomous Aerial Survey Platform

Final Year Design Project (P) Spring 2025			
Student Details	Name & ID	Email	Phone
Member 1	Md. Kowshik (20121020)	md.kowshik@g.bracu.ac.bd	01302692460
Member 2	Sultanul Arefin Nabil (20221030)	sultanul.arefin.nabil@g.bracu.ac.bd	01634243774
Member 3	Mushfeka Anika Bushra (21321070)	mushfeka.anika.bushra@g.bracu.ac.bd	01753472820
Member 4	Al Hossain Shawn (22121073)	al.hossain.shawn@g.bracu.ac.bd	01537237192
ATC Details			
ATC 6	Name	Email	Phone
Chair	Dr. A.S. Nazmul Huda	nazmul.huda@bracu.ac.bd	N/A
Member 1	Nahid Hossain Taz	nahid.hossain@bracu.ac.bd	N/A
Member 2	Sharif Mohammad Shams	sharif.mohd@bracu.ac.bd	N/A

General Notes:

1. In addition to detail journal/logbook fill out the summary/key steps and progress of your work
2. Reflect planning assignments, who has what responsibilities.
3. The logbook should contain all activities performed by the team members (Individual and team activities).

FYDP (P) Spring 2025 Summary of Team Log Book/Journal

Date/Time/ Place	Attendee	Summary of Meeting Minutes	Responsible	Comment by ATC
13.2.2025 (EEE499P Class 1)	Students: 1. Shawn 2. Bushra 3. Nabil Faculty Members: Mohaimenul Islam	Introductory Class of EEE499P		
16.2.2025 (Group Meeting 1)	Students: 1. Shawn 2. Bushra 3. Nabil 4. Kowshik	1. Discussion on and identification of project scope 2. Analyse related fields to find suitable problems and propose solutions	Task 1: Everyone Task 2: Everyone Progress: Task 1: Project scope narrowed down to three specific areas (Environmental, Agricultural, Automation) Task 2: Patterns found among listed projects both in BRACU Institutional Repository	
18.2.2025 (Group Meeting 2)	Students: 1. Shawn 2. Bushra 3. Nabil 4. Kowshik	1. Brainstorming some project ideas within the predetermined scope 2. Further study on selected field	Task 1: Everyone Task 2: Everyone Progress: Task 1: A few ideas were discussed and noted down along with their biblical references and previous work	

FYDP (P) Spring 2025 Summary of Team Log Book/Journal

			history (one main candidate is chosen) Task 2: Updated and classified over 120 projects from university repository	
20.2.2025 (EEE499P Class 2)	Students: 1. Shawn 2. Bushra 3. Nabil 4. Kowshik Faculty Members: Dr. Arshad Mahmud Chowdhury	Identification of Complex Engineering Problems		
20.2.2025 (ATC Meeting 1)	Students: 1. Shawn 2. Bushra 3. Nabil 4. Kowshik Faculty Members: 1. Dr. A.S. Nazmul Huda 2. Sharif Mohammad Shams	1. Shortlist four topics based on group preferences 2. Identify if they satisfy all the CO's and PO's 3. Create a comparison chart between selected project idea and previous work from BRACU	Task 1: Everyone Task 2: Bushra Task 3: Shawn, Kowshik Progress: Task 1: Started Task 2: Started Task 3: Started	
22.2.2025 (Group Meeting 3)	Students: 1. Shawn 2. Bushra 3. Nabil 4. Kowshik	1. Shortlist topics based on CO and PO 2. Compare with previous works	Task 1: Bushra, Nabil Task 2: Shawn, Kowshik Progress: Task 1: Partially Completed Task 2: Partially	

FYDP (P) Spring 2025 Summary of Team Log Book/Journal

			Completed	
23.2.2025 (ATC Meeting 2)	Students: 1. Kowshik 2. Shawn Faculty Members: Nahid Hossain Taz	1. Prepare a broad roadmap/goals that are associated with the project candidate		General Suggestions About Choosing a Feasible Project
25.2.2025 (Group Meeting 4)	Students: 1. Shawn 2. Bushra 3. Nabil 4. Kowshik	1. Prepare a broad roadmap/list of features/goals 2. Aggregation of findings from the shortlisted project topics 3. Prepare a final shortlist with the project candidate and pick the preferred project 4. Create a comparison chart for the final project candidate	Task 1: Kowshik Task 2: Nabil Task 3: Shawn Task 4: Bushra Progress: Task 1: Completed Task 2: Completed Task 3: Completed Task 4: Completed	
27.2.2025 (EEE499P Class 3)	Students: 1. Shawn 2. Bushra 3. Nabil 4. Kowshik Faculty Members: Dr. Md. Mosaddequr Rahman	Introduction to Engineering Design Process		
3.3.2025 (Group Meeting 5)	Students: 1. Shawn 2. Bushra 3. Nabil 4. Kowshik	1. Preparation of the draft concept note	Task 1: Everyone Progress: Task 1: Incomplete	
6.3.2025 (EEE499P Class 4)	Students: 1. Shawn 2. Bushra	Case Study: How to Identify a Complex Engineering Design Project and Fulfill CO Criteria		

FYDP (P) Spring 2025 Summary of Team Log Book/Journal

	3. Nabil 4. Kowshik Faculty Members: Dr. Abu S.M. Mohsin			
6.3.2025 (Group Meeting 6)	Students: 1. Shawn 2. Bushra 3. Nabil 4. Kowshik	1. Final compilation and submission of draft concept note	Task 1: Everyone Progress: Task 1: Complete	
7.3.2025 (Group Meeting 7)	Students: 1. Shawn 2. Bushra 3. Nabil 4. Kowshik	1. Problem statement, Objective 2. Requirements, specification, Constrains 3. Design Approach 1 4. Design Approach 2	Task 1: Shawn Task 2: Kowshik Task 3: Nabil Task 4: Bushra Progress: Task 1: Completed Task 2: Partially Completed Task 3: Completed Task 4: Partially Completed	
10.3.2025 (Group Meeting 8)	Students: 1. Shawn 2. Bushra 3. Nabil 4. Kowshik	1. Requirements, Specification and Constraints 2. Design Approach 2	Task 1: Kowshik Task 2: Bushra Progress: Task 1: Completed Task 2: Partially Completed	
16.3.2025 (Group Meeting 9)	Students: 1. Shawn 2. Bushra 3. Nabil 4. Kowshik	1. Work on multiple design approach 2. Specification, Requirements and Constraint	Task 1: Shawn Task 2: Nabil Progress: Task 1: Partially Completed Task 2: Completed	

FYDP (P) Spring 2025 Summary of Team Log Book/Journal

27.3.2025 (Group Meeting 10)	Students: 1. Shawn 2. Bushra 3. Nabil 4. Kowshik	1. Find applicable standards and codes	Task 1: Kowshik Progress: Task 1: Completed	
5.4.2025 (Group Meeting 11)	Students: 1. Shawn 2. Bushra 3. Kowshik	1. Work on multiple design approach 2. Work on design approach 2 3. Review on overall concept note	Task 1: Shawn Task 2: Bushra Task 3: Everyone Progress: Task 1: Completed Task 2: Completed Task 3: Completed	
10.4.2025 (EEE499P Class 5)	Students: 1. Shawn 2. Bushra 3. Nabil Faculty Members: Mohaimenul Islam	Project Safety, Sustainability and Environmental Impact in Engineering Design		
11.4.2025 (Group Meeting 12)	Students: 1. Shawn 2. Bushra 3. Nabil 4. Kowshik	1. Start working on project proposal 2. Background Research 3. Methodology	Task 1: Shawn Task 2: Kowshik Task 3: Nabil, Bushra	
15.4.2025 (Group Meeting 13)	Students: 1. Shawn 2. Bushra 3. Nabil 4. Kowshik	1. Start working on project proposal 2. Background Research 3. Methodology	Task 1: Shawn Task 2: Kowshik Task 3: Nabil, Bushra Progress: Task 1: Started Task 2: Partially Completed Task 3: Partially Completed	

FYDP (P) Spring 2025 Summary of Team Log Book/Journal

17.4.2025 (EEE499P Class 6)	Students: 1. Shawn 2. Bushra 3. Nabil Faculty Members: Dr. A.S. Nazmul Huda	Review of Engineering Ethics and Professional Practices		
19.4.2025 (ATC Meeting 3)	Students: 1. Shawn 2. Bushra 3. Nabil 4. Kowshik Faculty Members: Nahid Hossain Taz	Risk Assessment and Sustainability		Remove arbitrary values from risk assessment. Sustainability analysis must be verbose and logical discussion.
22.4.2025 (Group Meeting 14)	Students: 1. Shawn 2. Bushra 3. Nabil 4. Kowshik	1. Risk Mitigation 2. Safety Consideration 3. Background Research 4. Methodology	Task 1: Bushra Task 2: Nabil Progress: Task 1: Started Task 2: Started Task 3: Completed Task 4: Completed	
24.4.2025 (EEE499P Class 7)	Students: 1. Shawn 2. Bushra 3. Nabil 4. Kowshik Faculty Members: Taiyeb Hasan Sakib	Review of Project Proposal Preparation, Report Writing and Presentation Techniques		
3.5.2025 (Group Meeting 15)	Students: 1. Shawn 2. Bushra 3. Nabil 4. Kowshik	1. Prepare for Final Presentation 2. Final Project Proposal 3. Risk Mitigation 4. Safety Consideration	Task 1: Shawn Task 2: Kowshik Progress: Task 1: Started	

FYDP (P) Spring 2025 Summary of Team Log Book/Journal

			Task 2: Started Task 3: Completed Task 4: Completed	
5.5.2025 (ATC Meeting 5)	Students: 1. Shawn 2. Bushra 3. Nabil 4. Kowshik Faculty Members: Sharif Mohd Shams	Problem statement refinement, power system, initial investment estimation		Redefine and clarify problem statement, further background research
6.5.2025 (Group Meeting 16)	Students: 1. Shawn 2. Bushra 3. Nabil 4. Kowshik	1. Mock Presentation 2. Prepare Slide 3. Prepare for Final Presentation 4. Final Project Proposal	Task 1: Everyone Task 2: Shawn Progress: Task 1: Completed Task 2: Completed Task 3: Completed Task 4: Completed	
8.5.2025 (EEE499P Final Presentation)	Students: 1. Shawn 2. Bushra 3. Nabil 4. Kowshik			
15.5.2025 (Group Meeting 17)	Students: 1. Shawn 2. Bushra 3. Nabil 4. Kowshik	1. Final Project Proposal Report Submission	Task 1: Everyone Progress: Task 1: Complete	

FYDP (D) Summer 2025 Summary of Team Log Book/Journal

Project Title: Autonomous Aerial Survey Platform

Final Year Design Project (D) Summer 2025			
Student Details	Name & ID	Email	Phone
Member 1	Md. Kowshik (20121020)	md.kowshik@g.bracu.ac.bd	01302692460
Member 2	Sultanul Arefin Nabil (20221030)	sultanul.arefin.nabil@g.bracu.ac.bd	01634243774
Member 3	Mushfeka Anika Bushra (21321070)	mushfeka.anika.bushra@g.bracu.ac.bd	01753472820
Member 4	Al Hossain Shawn (22121073)	al.hossain.shawn@g.bracu.ac.bd	01537237192
ATC Details			
ATC 6	Name	Email	Phone
Chair	Dr. A.S. Nazmul Huda	nazmul.huda@bracu.ac.bd	N/A
Member 1	Nahid Hossain Taz	nahid.hossain@bracu.ac.bd	N/A
Member 2	Sharif Mohammad Shams	sharif.mohd@bracu.ac.bd	N/A

General Notes:

1. In addition to detail journal/logbook fill out the summary/key steps and progress of your work
2. Reflect planning assignments, who has what responsibilities.
3. The logbook should contain all activities performed by the team members (Individual and team activities).

FYDP (D) Summer 2025 Summary of Team Log Book/Journal

Date/Time/ Place	Attendee	Summary of Meeting Minutes	Responsible	Comment by ATC
26.06.2025 (EEE499D Committee Class 1)	Students: 1. Shawn 2. Bushra 3. Nabil 4. Kowshik Faculty Members: Atib Mohammad Oni	Introductory Class EEE499(D)		
28.06.2025 (Group Meeting 1)	Students: 1. Shawn 2. Bushra 3. Nabil 4. Kowshik	1. Dataset selection 2. Dataset pre-processing 3. Test image classification performance on a base CNN model	Task 1: Bushra, Nabil Task 2: Kowshik Task 3: Shawn Progress Task 1: Completed Task 2: Completed Task 3: Completed	
30.06.2025 (Group Meeting 2)	Students: 1. Shawn 2. Bushra 3. Nabil	1. Dataset annotation for the object detection model	Task 1: Nabil, Bushra Progress Task 1: Dataset Split into two large sections and divided among the two team members	
07.07.2025 (ATC Meeting 1)	Students: 1. Shawn 2. Bushra 3. Nabil 4. Kowshik Faculty Members:	1. Update the task distribution among the group members.	Task 1: Kowshik Progress Task 1: Completed	1. The log book must be updated with each meetings detail

FYDP (D) Summer 2025 Summary of Team Log Book/Journal

	Sharif Mohd Shams			
08.07.2025 (Group Meeting 3)	Students: 1. Shawn 2. Bushra 3. Nabil 4. Kowshik	1. Drone simulation environment selection and preparation 2. Dataset annotation for the object detection model	Task 1: Shawn Task 2: Nabil, Bushra Progress: Task 1: Partially Completed Task 2: Partially Completed	
11.07.2025 (ATC Meeting 2)	Students: 1. Shawn 2. Bushra 3. Nabil 4. Kowshik Faculty Members: Sharif Mohd Shams	1. Conduct in depth research on drone flight paths, standard design and components 2. Research flight pattern and flight time	Task 1: Shawn Task 2: Nabil, Bushra Progress: Task 1: Some research areas were identified as well as limitations were taken into consideration Task 2: According to the discussed limitations the research is conducted	1. Gather data on standard design of drones that are deployed in the field 2. Research commercial product functionality 3. Find out the limits, the standard specs of the drone i.e. coverage area, flight time, max altitude etc
13.07.2025 (Group Meeting 4)	Students: 1. Shawn 2. Bushra 3. Nabil 4. Kowshik	1. Conduct in depth research on drone flight paths, standard design and components 2. Research flight pattern and flight time 3. Dataset annotation for the object detection model 4. Test image classification performance on a base	Task 1: Shawn Task 2: Nabil, Bushra Task 3: Nabil, Bushra Task 4: Kowshik Progress: Task 1: Completed	

FYDP (D) Summer 2025 Summary of Team Log Book/Journal

		CNN model	Task 2: Completed Task 3: Partially Completed Task 4: Partially Completed	
22.07.2025 (Group Meeting 5)	Students: 1. Bushra 2. Nabil 3. Kowshik	1. Dataset annotation for the object detection model 2. Test image classification performance on a base CNN model	Task 1: Nabil, Bushra Task 2: Kowshik Progress: Task 1: Completed Task 2: Completed	
24.07.2025 (ATC Meeting 3)	Students: 1. Shawn 2. Bushra 3. Nabil 4. Kowshik Faculty Members: Sharif Mohd Shams	Midterm Progress Presentation		1. Presentation performance was not satisfactory 2. Slides must be made more concise
01.8.2025 (Group Meeting 6)	Students: 1. Shawn 2. Bushra 3. Nabil 4. Kowshik	1. Discussion and sharing results and planning for final meeting with ATC 2. Machine learning model performance analysis	Task 1: Nabil, Bushra Task 1: Kowshik Progress: Task 1: Completed Task 2: Completed	
15.08.2025 (ATC Meeting 4)	Students: 1. Shawn 2. Bushra 3. Nabil 4. Kowshik Faculty	1. Functional demonstration of the remaining parts of the project.		1. All required functionality has been shown 2. Must start work on final presentation slides and design report

FYDP (D) Summer 2025 Summary of Team Log Book/Journal

	Members: Sharif Mohd Shams			
21.08.2025 (Group Meeting 6)	Students: 1. Shawn 2. Bushra 3. Nabil 4. Kowshik	1. Optimize slides for final presentation	Task 1: Shawn Progress: Task 1: Partially Completed	
27.08.2025 (Group Meeting 7)	Students: 1. Shawn 2. Bushra 3. Nabil 4. Kowshik	1. Optimize slides for final presentation 2. Practice Presentation	Task 1: Shawn Task 2: Everyone Progress: Task 1: Completed Task 2: Completed	
28.08.2025 (EEE499D Final Presentation)	Students: 1. Shawn 2. Bushra 3. Nabil 4. Kowshik			
29.08.2025 (Group Meeting 8)	Students: 1. Shawn 2. Bushra 3. Nabil 4. Kowshik	1. Report Compilation	Task 1: Everyone Progress: Task 1: Partially Completed	
31.08.2025 (Group Meeting 9)	Students: 1. Shawn 2. Bushra 3. Nabil 4. Kowshik	1. Report Compilation	Task 1: Everyone Progress: Task 1: Partially Completed	
04.09.2025 (Group Meeting 10)	Students: 1. Shawn 2. Bushra 3. Nabil 4. Kowshik	1. Report Compilation	Task 1: Everyone Progress: Task 1: Partially Completed	
05.09.2025 (Group Meeting 11)	Students: 1. Shawn 2. Bushra 3. Nabil	1. Report Compilation	Task 1: Everyone Progress: Task 1: Partially	

FYDP (D) Summer 2025 Summary of Team Log Book/Journal

	4. Kowshik		Completed	
06.09.2025 (Group Meeting 12)	Students: 1. Shawn 2. Bushra 3. Nabil 4. Kowshik	1. Report Compilation	Task 1: Everyone Progress: Task 1: Partially Completed	
07.09.2025 (Group Meeting 13)	Students: 1. Shawn 2. Bushra 3. Nabil 4. Kowshik	1. Report Compilation	Task 1: Everyone Progress: Task 1: Partially Completed	
08.09.2025 (Group Meeting 14)	Students: 1. Shawn 2. Bushra 3. Nabil 4. Kowshik	1. Report Compilation	Task 1: Everyone Progress: Task 1: Partially Completed	
09.09.2025 (Group Meeting 15)	Students: 1. Shawn 2. Bushra 3. Nabil 4. Kowshik	1. Report Compilation	Task 1: Everyone Progress: Task 1: Partially Completed	
10.9.2025 (ATC Meeting 5)	Students: 1. Kowshik Faculty Members: Sharif Mohd Shams	Draft Submission		
11.09.2025 (Group Meeting 16)	Students: 1. Shawn 2. Bushra 3. Nabil 4. Kowshik	Final Design Report Submission	Task 1: Everyone Progress: Task 1: Completed	

FYDP (C) Fall 2025 Summary of Team Log Book/Journal

Project Title: Autonomous Aerial Survey Platform

Final Year Design Project (C) Summer 2025			
Student Details	Name & ID	Email	Phone
Member 1	Md. Kowshik (20121020)	md.kowshik@g.bracu.ac.bd	01302692460
Member 2	Sultanul Arefin Nabil (20221030)	sultanul.arefin.nabil@g.bracu.ac.bd	01634243774
Member 3	Mushfeka Anika Bushra (21321070)	mushfeka.anika.bushra@g.bracu.ac.bd	01753472820
Member 4	Al Hossain Shawn (22121073)	al.hossain.shawn@g.bracu.ac.bd	01537237192
ATC Details			
ATC 6	Name	Email	Phone
Chair	Dr. A.S. Nazmul Huda	nazmul.huda@bracu.ac.bd	N/A
Member 1	Nahid Hossain Taz	nahid.hossain@bracu.ac.bd	N/A
Member 2	Sharif Mohammad Shams	sharif.mohd@bracu.ac.bd	N/A

General Notes:

1. In addition to detail journal/logbook fill out the summary/key steps and progress of your work
2. Reflect planning assignments, who has what responsibilities.
3. The logbook should contain all activities performed by the team members (Individual and team activities).

FYDP (C) Fall 2025 Summary of Team Log Book/Journal

Date/Time/Place	Attendee	Summary of Meeting Minutes	Responsible	Comment by ATC
04.10.2025 (Group Meeting 1)	Students: 1. Shawn 2. Bushra 3. Nabil 4. Kowshik	1. Start Market Scouting 2. Identification of literature gaps 3. Report formatting	Task 1: Kowshik, Shawn Task 2: Nabil Task 3: Bushra Progress Task 1: Completed Task 2: Completed Task 3: Completed	
12.10.2025 (Group Meeting 2)	Students: 1. Shawn 2. Bushra 3. Nabil	1. Collect Components	Task 1: Kowshik, Shawn, Bushra, Nabil Progress Task 1: Partially Completed	
16.10.2025 (EEE499C Committee Class 1)	Students: 1. Shawn 2. Bushra 3. Nabil 4. Kowshik Faculty Members: Mirza Rasheduzzaman	Introductory Class EEE499(C)		
19.10.2025 (Group Meeting 3)	Students: 1. Shawn 2. Nabil 3. Kowshik	1. Collect Components	Task 1: Kowshik, Shawn, Bushra, Nabil Progress Task 1: Completed	
23.10.2025 (Group Meeting 4)	Students: 1. Shawn	1. Build the bare bones airframe	Task 1: Kowshik, Shawn, Nabil	

FYDP (C) Fall 2025 Summary of Team Log Book/Journal

Meeting 4)	2. Bushra 3. Nabil 4. Kowshik		<u>Progress</u> Task 1: Completed	
29.10.2025 (ATC Meeting 1)	Students: 1. Shawn 2. Kowshik Faculty Members: Sharif Mohd Shams	Progress update overview		Test each subsystem before starting integration process
01.11.2025 (Group Meeting 5)	Students: 1. Shawn 2. Bushra 3. Nabil 4. Kowshik	1. Test every subsystem before overall integration 2. First hover test	<u>Task 1:</u> Kowshik, Bushra, Nabil <u>Task 2:</u> Shawn <u>Progress</u> Task 1: Partially Completed Task 2: Completed	
05.11.2025 (ATC Meeting 2)	Students: 1. Shawn 2. Bushra 3. Nabil 4. Kowshik Faculty Members: Sharif Mohd Shams	Progress update overview		Test the camera system as battery life is limited
12.11.2025 (Group Meeting 6)	Students: 1. Shawn 2. Bushra 3. Nabil 4. Kowshik	1. Test every subsystem before overall integration	<u>Task 1:</u> Kowshik, Bushra, Nabil <u>Progress</u> Task 1: Partially Completed	
27.11.2025 (Group Meeting 7)	Students: 1. Shawn 2. Bushra 3. Nabil	1. Test every subsystem before overall integration	<u>Task 1:</u> Kowshik, Bushra, Nabil <u>Progress</u> Task 1: Completed	
6.12.2025	Students:	1. Fully automatic flight	<u>Task 1:</u> Nabil	

FYDP (C) Fall 2025 Summary of Team Log Book/Journal

(Group Meeting 8)	1. Shawn 2. Nabil 3. Kowshik		<u>Progress</u> Task 1: Completed	
10.12.2025 (ATC Meeting 3)	Students: 1. Shawn 2. Bushra 3. Nabil 4. Kowshik Faculty Members: Sharif Mohd Shams	Progress Presentation		Clearly mention the challenges being faced
17.12.2025 (Group Meeting 9)	Students: 1. Shawn 2. Bushra 3. Nabil 4. Kowshik	1. Sensor denoising 2. Camera calibration 3. Report writing 4. Documentation	<u>Task 1:</u> Kowshik <u>Task 2:</u> Nabil <u>Task 3:</u> Shawn <u>Task 4:</u> Kowshik <u>Progress</u> Task 1: Partially Completed Task 2: Completed Task 3: Partially Completed Task 4: Partially Completed	
22.12.2025 (Group Meeting 10)	Students: 1. Shawn 2. Bushra 3. Nabil 4. Kowshik	1. Sensor denoising 2. Camera testing 3. Report writing 4. Documentation	<u>Task 1:</u> Kowshik <u>Task 2:</u> Nabil <u>Task 3:</u> Shawn <u>Task 4:</u> Kowshik <u>Progress</u> Task 1: Partially Completed Task 2: Partially Completed Task 3: Partially Completed Task 4: Partially Completed	
28.12.2025 (Group Meeting 11)	Students: 1. Shawn	1. Sensor denoising 2. Camera testing	<u>Task 1:</u> Kowshik <u>Task 2:</u> Nabil	

FYDP (C) Fall 2025 Summary of Team Log Book/Journal

Meeting 11)	2. Bushra 3. Nabil 4. Kowshik	3. Report writing 4. Documentation	<u>Task 3:</u> Shawn <u>Task 4:</u> Kowshik <u>Progress</u> Task 1: Completed Task 2: Completed Task 3: Partially Completed Task 4: Partially Completed	
29.12.2025 (Group Meeting 12)	Students: 1. Shawn 2. Bushra 3. Nabil 4. Kowshik	1. Report Writing	<u>Task 1:</u> Kowshik, Shawn, Bushra, Nabil <u>Progress</u> Task 1: Partially Completed	
30.12.2025 (Group Meeting 13)	Students: 1. Shawn 2. Bushra 3. Nabil 4. Kowshik	1. Report writing	<u>Task 1:</u> Kowshik, Shawn, Bushra, Nabil <u>Progress</u> Task 1: Completed	
01.01.2026 (Group Meeting 14)	Students: 1. Shawn 2. Bushra 3. Nabil	1. Video editing 2. Poster design	<u>Task 1:</u> Kowshik, Shawn, Nabil <u>Task 1:</u> Bushra <u>Progress</u> Task 1: Completed Task 1: Completed	
05.01.2026 (Project Showcasing)	Students: 1. Shawn 2. Bushra 3. Nabil 4. Kowshik	Project Showcasing		
09.09.2026 (Group Meeting 15)	Students: 1. Shawn 2. Bushra 3. Nabil 4. Kowshik	Draft Report Submission		