COATL Millimeter-wave RADAR
Assessing the Quality of Coffee Beans

Capstone Team 4

Portland State University

Maseeh College of Engineering & Computer Science
Department of Electrical & Computer Engineering

Industry Sponsor:
Joshua Mendez, joshmen@pdx.edu

Faculty Advisor:
Mark Martin, marmart2@pdx.edu

Engineering Team:
Wallace McKenzie, mwallace@pdx.edu
Chris Kane-Pardy, chrisk@pdx.edu
Henry Sanders, hensan@pdx.edu
Kamal Smith, kamal@pdx.edu

June 12, 2025

Maseeh College of Engineering and Computer Science
PORTLAND STATE UNIVERSITY

ECE 412/413
Senior Project Development I/II

Contents

List of Figures

1 Executive Summary

The *COATL Millimeter-wave RADAR for Assessing the Quality of Green Coffee* project set out to explore a new, non-destructive method for measuring the moisture content of green coffee beans—an essential factor in maintaining flavor and quality during storage and roasting. Existing methods, such as oven drying or capacitive sensors, are often destructive, expensive, or impractical for small-scale producers. Our project aimed to create a scalable, low-cost alternative using advanced radar technology.

Dr. Joshua Mendez sponsored this work through the Coffee And Telesensing Lab (COATL), an organization focused on technology and education that supports community-driven innovation. With Dr. Mendez's guidance, we designed and tested a radar-based sensing system using the Acconeer XM125 (60 GHz) millimeter-wave radar module. This module is particularly sensitive to water due to how water absorbs energy at this frequency.

Our primary deliverable was a functional system capable of distinguishing between three different types of green coffee beans with varying moisture content levels. To achieve this, we developed a Python-based testbed to extract complex I/Q signal data from the radar module and trained a machine learning classifier using PyTorch to analyze these signals. While we originally aimed to estimate the permittivity of each bean sample, we shifted our focus toward classification accuracy based on amplitude differences, still influenced by dielectric properties like permittivity.

The final system was unable to distinguish between the different types of green coffee beans reliably, but was able to capture distinct features for roasted coffee beans. The main inconsistencies for the green beans come from the heterogeneous bean geometries and poor radar resolution. While we removed our initial requirement of calculating permittivity values, the project still has potential for further experimentation, using a millimeter-wave radar with a different wavelength. With some changes, the combination of radar sensing and machine learning could offer a practical path forward for a reliable moisture analysis tool.

2 Background

Sponsor

The project is sponsored by COATL, a research group focused on applying advanced sensing technologies to improve agricultural processes and product quality. COATL specializes in using state-of-the-art remote sensing techniques, including radar and microwave systems, to address challenges faced by coffee producers and processors. By combining expertise in electrical engineering, signal processing, and agricultural science, COATL aims to develop scalable, cost-effective solutions to enhance coffee quality monitoring and control.

Technology Domain and Context

The technology domain of this project is millimeter-wave radar sensing for agricultural product quality assessment. Specifically, this system uses the XM125 60 GHz pulsed coherent radar module to measure the moisture content of green coffee beans. Moisture content is critical in coffee's long-term storage stability, roasting profile, and overall quality. Traditional moisture measurement methods include oven drying (destructive and time-consuming) or capacitive sensors (non-destructive but expensive). The millimeter-wave radar system offers a proposed alternative by leveraging water's strong absorption characteristics at 60 GHz. This approach has already been used for monitoring moisture in other agricultural products, such as drying leaves.

System Functionality

The radar-based moisture sensor works on a time-of-flight principle. The radar transmits a series of electromagnetic pulses and measures the delay and amplitude of reflected waves. When green coffee beans are placed in a sample holder, the dielectric properties of the beans, directly related to their water content, affect the reflected signal. By extracting the dielectric constant, the system estimates the moisture level in the sample. The relationship between reflectivity γ and permittivity is given by:

$$\gamma = \left(\frac{\sqrt{\epsilon_1} - \sqrt{\epsilon_2}}{\sqrt{\epsilon_1} + \sqrt{\epsilon_2}} \right)^2$$

where ϵ_1 and ϵ_2 are the relative permeabilities on either side of the coffee-air boundary.

A conceptual design of the proposed RADAR-based moisture sensor is illustrated in Figure 1 below. The system includes: (1) the XM125 radar module, (2) a sample holder designed to contain coffee beans, and (3) a corner reflector. When the sample holder is empty, the millimeter-wave pulses emitted by the radar are reflected back with high amplitude from the corner reflector. To minimize interference with the measurements, the sample holder must be made from materials transparent to 60 GHz waves. When filled with green coffee beans, the pulses are attenuated proportionally to the moisture content in the beans. The raw radar data from the XM125 module is processed using a computer or microcontroller to calculate the dielectric constant of the sample. The systems performance will be validated by comparing its moisture content estimates to those obtained using a commercial capacitive moisture sensor.

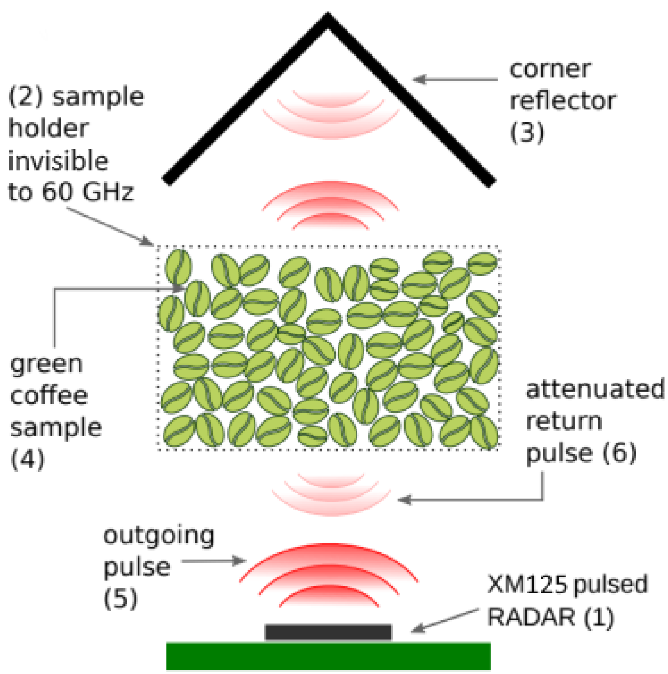

Figure 1: Conceptual model of material sensing with 60 GHz RADAR for coffee

2.1 Research

2.1.1 Commercial Off-the-Shelf Products

Dr. Joshua Mendez provided the Acconeer XE125 EVK (evaluation board for the XM125 PCR module, utilizing the A121 radar sensor). Additionally, he provided access to the *RoastRite Coffee Moisture and Density Meter RM-800* to act as a known test module/baseline for comparison to our test results.

2.1.2 Open Source Projects

- Acconeer Exploration Tool (version A121)
- Python 3.11.9
- Spyder 6
- PyTorch

2.1.3 Patents, Papers, White Papers, Articles, Conference Proceedings

Patents & White Papers:
Our project proposal and Product Design Specification (PDS) have undergone a thorough review to ensure they do not infringe on any existing patents or intellectual property. We have conducted due diligence to verify that all designs, methods, and materials used are either original or fall within the public domain.

Papers: The two papers below illustrate studies that use the XM125.

A 60 GHz pulsed coherent radar for online monitoring of the withering condition of leaves, ScienceDirect, Jun. 16, 2022.
`https://www.sciencedirect.com/science/article/pii/S0924424722003314`.

T. Albing, R. Nelander, and Acconeer AB, Material Classification of Recyclable Containers Using 60 GHz Radar, arxiv.org, 2023.
`https://arxiv.org/pdf/2312.14539`

3 Requirements

3.1 Original Requirements

- **Must**:
  - Be able to extract the dielectric constant of a fixed-volume sample placed in the Radar's beam path for materials with dielectric constants ranging from 5-20.
  - Differentiate between at least 3 coffees with different moisture contents (e.g. 8%, 10%, and 12%)
  - Detect when the sample holder is empty.
- **Should**:
  - Be able to determine the moisture content of coffees in the range of 7%-15%.
  - Have the ability to determine the moisture content of green coffee with a resolution of 1%.
  - Provide estimates of moisture content that are within 20% of the value reported by a professional capacitive sensor moisture meter.
- **May**:
  - Have the ability to determine the moisture content of green coffee with a resolution of 0.1%.
  - Provide estimates of moisture content that are within 10% of the value reported by a professional capacitive sensor moisture meter.

After failing to extract the permittivity from green coffee beans, a discussion was had with Dr. Mendez about relating the I/Q amplitude data received by the radar module to moisture content within the beans, essentially skipping the step of extracting the permittivity. He agreed that this approach would streamline the success of the project. Below are the project's final requirements. Please note the only difference is the removal of the first requirement from the "Must" section of the original requirements.

3.2 Final Requirements

- **Must**:
  - Differentiate between at least 3 coffees with different moisture contents (e.g. 8%, 10%, and 12%)
  - Detect when the sample holder is empty.
- **Should**:
  - Be able to determine the moisture content of coffees in the range of 7%-15%.
  - Have the ability to determine the moisture content of green coffee with a resolution of 1%.
  - Provide estimates of moisture content that are within 20% of the value reported by a professional capacitive sensor moisture meter.
- **May**:
  - Have the ability to determine the moisture content of green coffee with a resolution of 0.1%.
  - Provide estimates of moisture content that are within 10% of the value reported by a professional capacitive sensor moisture meter.

4 Objectives and Deliverables

4.1 Objectives

The original objective of this project was to develop a non-destructive, radar-based sensor system capable of determining the moisture content of green coffee beans using the Acconeer XM125 60 GHz radar module. The goal was to provide a low-cost, scalable alternative to traditional moisture analysis methods, particularly to the RoastRite RM800, which can be cost-prohibitive for small-scale producers.

Industry Sponsor Deliverables

Based on the original objective, we initially set out to accomplish the **Must** requirements from the initial project description:

- Extract the dielectric constant of green coffee beans.
- Differentiate between at least three moisture levels (e.g., 8%, 10%, and 12%)
- Detect when the sample holder was empty

However, after encountering significant challenges in extracting accurate permittivity values due to geometric inconsistencies and radar resolution limits, the project's direction was revised in collaboration with the sponsor. As a result, the goal of permittivity extraction was removed, and the focus shifted to classifying moisture content directly from I/Q amplitude data.

This approach also proved challenging, as variations in bean geometry continued to impact data consistency. While initial results were somewhat promising, our machine learning program failed to identify consistent patterns, even with a large dataset. Ultimately, it became clear that a different radar module would be necessary to meet the original goals.

After this pivot toward maximizing project viability, the agreed-upon deliverables include:

- A system demonstrating the preliminary ability to distinguish between beans with differing moisture content using machine learning
- Supporting the sponsor in compiling comprehensive documentation and test data to assist in the refinement of the project scope and requirements for prospective future teams

Course Deliverables

In addition to sponsor deliverables, the team also fulfilled the following course-specific requirements:

- Project Proposal / PDS
- Weekly Progress Reports
- Test Plan
- Design & Ethics Considerations
- Final Report
- User's Manual
- ECE Capstone Poster
- Version-controlled documentation for prospective future teams

5 Approach

The initial direction of this project was inspired by the research presented in "*A 60 GHz Pulsed Coherent Radar for Online Monitoring of the Withering Condition of Leaves*" (ScienceDirect, June 16, 2022) and "*Material Classification of Recyclable Containers Using 60 GHz Radar*" by Albing, Nelander, and Acconeer AB (arXiv, 2023). These studies offered some insight into radar interactions with various materials and moisture content. However, their methodologies and the materials they tested did not fully align with the specific requirements of our system and offered limited guidance for testing coffee beans.

Our initial steps involved selecting a programming language and conducting in-depth research into the radar system and the Acconeer Exploration Tool. We evaluated both Python and C, initially favoring C due to our plan to interface the XE125 EVK with an ESP32 via the ESP-IDF extension in VSCode. However, we ultimately chose Python as our primary development language, prioritizing the projects core objectives and taking advantage of Python's robust libraries for data visualization, signal processing, and machine learning. Development was conducted using Spyder 6, which served as our standard IDE for the remainder of the project unless otherwise noted.

During the research phase, we divided key focus areas among team members to efficiently cover the technical breadth of the project. We met twice weekly to share findings and coordinate progress. Over approximately three weeks, we developed an initial project plan, acquired necessary tools, and addressed knowledge gaps through targeted self-education. With a solid foundation established, we transitioned to the design phase.

5.1 Initial Design and 3D Printing Exploration

The design process began with the goal of developing a container for green coffee beans that was radar-transparent (at 60GHz), capable of rotation, and suitable for controlled testing. With no existing designs to reference, we started from scratch and conducted extensive testing using the Acconeer Exploration Tool and researching both container design principles and the capabilities of the Acconeer XM125 Pulsed Coherent Radar (PCR).

Our initial concept relied on rotating the container using a stepper motor, with the intent of improving data collection by capturing radar reflections from multiple angles. We modeled the design in Fusion 360, incorporating a geared stepper motor at the base for controlled rotation and a circular track to hold the beans, positioning the radar near the edge to optimize the return signal amplitude capture.

Before printing, we conducted a comparative analysis of several filament materials, including PETG, Nylon, Polycarbonate, and others, primarily evaluating their dielectric properties and mechanical durability. PETG and resin emerged as the most promising due to their relatively low permittivities. Two iterations of the container were printed, each integrating the stepper motor mechanism.

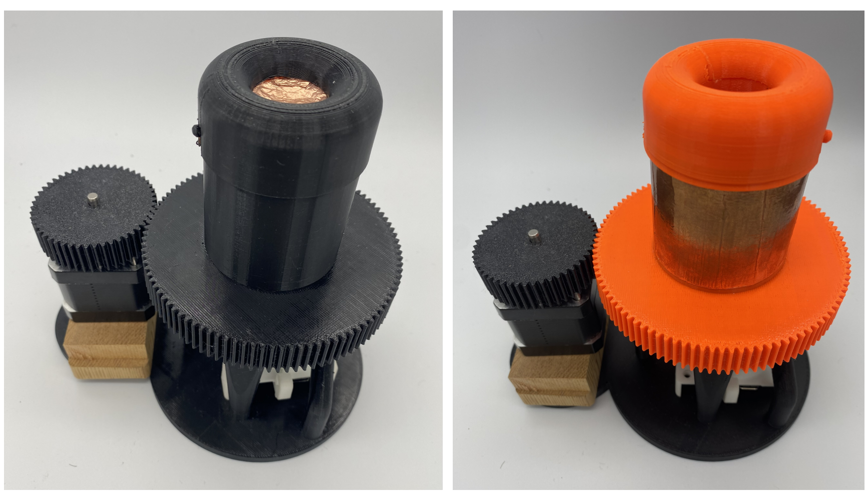

Figure 2: 3D Printed Bean Enclosures

Using the Acconeer Exploration Tool, we tested the rotational functionality and observed that motor-induced vibrations introduced significant noise into the radar signal. To mitigate this, we switched to manual rotation between scans, eliminating mechanical vibration as a noise source.

However, further testing revealed that the 3D-printed enclosures themselves introduced consistent interference into our readings as well. We hypothesized that the non-uniform layer structure and internal mesh patterns of the prints were responsible. To validate this, we conducted experiments in multiple environments across campus, including high-noise areas with active lab equipment. Regardless of location, repeatable patterns of interference were detected in scans of the empty container, supporting our hypothesis.

To address this, we lined both the interior and exterior surfaces of the prototype with copper tape to shield the design from exterior noise and increase reflectivity within the device, leaving only the platform supporting the beans uncovered. While this provided some shielding, the inconsistent dielectric profile of the 3D-printed material supporting the beans continued to distort return amplitude values, making it difficult to isolate changes attributable to the beans themselves. These recurring issues ultimately led us to reconsider our approach to enclosure design.

5.2 Transition to Aluminum Tubing and Reflectors

Before developing a new prototype, we repurposed a previously 3D-printed funnel from the last design phase. We coated the funnel in copper tape and positioned it directly over the radar module, slotting it onto the lens holder cover. Beans were placed inside the funnel, and while viewing the exploration tool, obvious changes could be seen. This promising outcome reinforced our hypothesis and led us to conclude that using a uniform material would yield better results from the following prototype.

We chose to use aluminum tubing for the next prototype and calculated the optimal dimensions for both the corner reflector and tube length. The design was modeled in Fusion 360 and then fabricated using a 1.5-inch aluminum tube, which was readily available in the machine shop. Dr. Mendez assisted in fabricating the tube and parabolic corner reflector based on our specifications.

Figure 3: Original aluminum tube design

Upon receiving the prototype, we first tested it empty in various environments. The results were encouraging, as there was no observable interference from external signals or material properties. We then divided testing tasks to fine-tune the settings within the Acconeer Exploration Tool and determine the optimal quantity of beans for consistent measurement. To control for variability in bean size and water content, we standardized all tests by using a fixed weight of beans.

Before testing, recording, and comparing bean data, we developed several versions of the program. Our initial goal was to calculate the dielectric constant of the coffee beans. To do this, we wrote several iterations of programs to extract permittivity values. However, we encountered several challenges. The primary issue was that the I/Q data provided by Acconeer is a poorly documented, unitless amplitude measurement, making it difficult to interpret directly. According to Acconeer, I/Q data is a complex representation of the “in-phase” and “quadrature” components of the return signal. Essentially, the in-phase part of the signal represents the amplitude (real) and the quadrature part represents the phase (imaginary). Attempts to convert the I/Q data into permittivity proved challenging. Despite foregoing this requirement, we initially assumed that relative consistency in permittivity values would be sufficient for our analysis. This assumption was challenged when we observed significant amplitude variability, even when scanning the same bean sample under seemingly identical conditions. To mitigate this, we introduced a new methodology: averaging the results from multiple scans of the same sample, manually shaking the beans between each scan to introduce randomization and reduce position-based inconsistencies. This new methodology meant the initial requirement of finding permittivity would be obsolete. After consulting with Dr. Mendez, we received confirmation that shifting to an amplitude-only analysis was appropriate.

We then entered the next phase of iterative coding, focusing on improving the consistency of our amplitude averages. After developing and refining several versions of the program, we began extensive testing to achieve repeatable results using identical bean samples. While the revised approach yielded results that were more consistent than earlier attempts, significant issues remained. Most notably, the amplitude readings for the high moisture, low moisture, and intermediate moisture bean samples showed considerable overlap, making it difficult to reliably distinguish between the different moisture levels.

We investigated the cause of the overlapping and inconsistent readings and concluded that the issue was likely related to the ∼5mm wavelength of the radar signal closely matching the average width of a coffee bean. This wavelength-to-target size ratio resulted in poor resolution, making it difficult to resolve individual bean features. Dr. Mendez theorized that the non-uniform geometries most certainly resulted in inconsistent diffraction of our radar between tests as well. As a result, minor geometric imperfections in the beans disproportionately impacted the readings produced by the XM125.

After presenting our findings to Dr. Mark Martin, he agreed with our assessment and confirmed that the wavelength-to-bean size relationship was likely the root of the issue. He recommended revising the prototype to incorporate a narrower tube, just wide enough to fit a single bean per vertical layer, while filling the entire tube. This approach would effectively average the radar readings across multiple geometries in each scan, eliminating the need for shaking between measurements. Based on this guidance, we proceeded with the development of our next prototype.

5.3 Refining the Testing Methodology

We fabricated the revised, narrower prototype using an 8mm aluminum tube, adhering to the same design methodologies as before. To ensure precise alignment with the radar beam, we also 3D-modeled a custom holder, resembling a Christmas tree stand, that allowed for fine positional adjustments. Consistent with previous trials, we followed the same testing procedures for the bean samples. However, we quickly encountered a critical issue: even at the radar's highest power setting, the return signal was fully attenuated when the tube was filled with beans. We incrementally removed beans until a measurable signal was restored, which occurred at a reduced sample size of approximately 5-6 beans, depending on their dimensions.

Figure 4: 8mm aluminum tube design

Once we had settled on this reduced sample size, we conducted extensive testing. However, this approach proved impractical as our weight-based standardization method could no longer be applied due to the extreme attenuation at higher fill levels. Furthermore, the results continued to exhibit significant inconsistency, and the signal-to-noise ratio remained too low for reliable classification.

In hopes of adhering to Dr. Martin's hypothesis while maintaining feasibility for consistent testing, we transitioned to a slightly larger third iteration of the prototype, fabricated using a 3/8" diameter aluminum tube. This version showed the most promise, as it accommodated beans of all geometries reliably and allowed for consistent randomization when shaken. We conducted another round of extensive testing to determine optimal radar parameters, and with a stable setup established, we proceeded to the machine learning phase.

6 Design

Our final design features the "medium" 3/8" aluminum tube paired with a custom corner reflector, the "aluminum tube stand", and the finalized version of our data collection program. The last phase of the project centered on developing a machine learning model to classify beans based on a large dataset. We implemented a deep learning architecture with five hidden layers: the first contained 256 neurons, and the final hidden layer contained 16. The model accepted a 100-point wide input vector, which was comprised of 50 points of real amplitudes and 50 points of imaginary amplitudes across our measurement distance, and produced one of three potential outputs, which corresponded to one of the three sample categories used during testing.

For our final testing phase, we originally planned for Kamal, Henry, and Chris to conduct 750 tests each, using the same averaging program and intermittent shaking method employed in the previous aluminum prototypes. The three varieties of beans we used were *Sweet Maria's Guatemala Antigua Hunapu* (7.8%), *Rulindo Tumba* (9.7%), and *Yemen Mokha Sanani* (11.4%), to provide a wide range of moisture content across the samples. Each person tested a different type of bean, which would have yielded a dataset of 2,475,000 input points. However, testing proved significantly more time-consuming than anticipated, and by the end of the first week, approximately 20% of the planned data was collected.

We consolidated the gathered results into a single dataset containing 330,000 input vectors and used it to train the model. Multiple variations of the model were evaluated, adjusting parameters such as learning rate, number of epochs, and the number of hidden layers to optimize classification accuracy.

Although the model reported high training accuracy, reaching up to approximately 95%, its performance on live, shaken average inputs was much lower, with actual classification accuracy closer to 25-30%. The model consistently favored the dampest beans in its predictions. We attribute this discrepancy largely to the over-

Figure 5: Final design

lap issues observed earlier. While human analysis showed that drier beans generally produced higher amplitude peaks and moister beans resulted in greater attenuation, the overlap in signal characteristics made it difficult to distinguish between the three bean types.

Moreover, persistent geometric inconsistencies introduced unpredictable diffraction effects and signal path lengths that varied from scan to scan. These inconsistencies led to unreliable radar reflections, ultimately preventing the model from identifying consistent, learnable patterns in the data.

To further investigate the limitations of our original setup, we collaborated with Dr. Joshua Mendez to test an alternative prototype of his design. This system employed the same radar module but operated in open air rather than within a shielded aluminum tube. The radar was mounted upside down and angled toward a 3D-printed dish containing the beans, with a trihedral corner reflector wrapped in copper tape placed beneath. Dr. Mendez conducted tests using two bean varieties—*Ethiopian* and *Peaberry*, across four roast levels: Light, Medium Light, Medium Dark, and Dark.

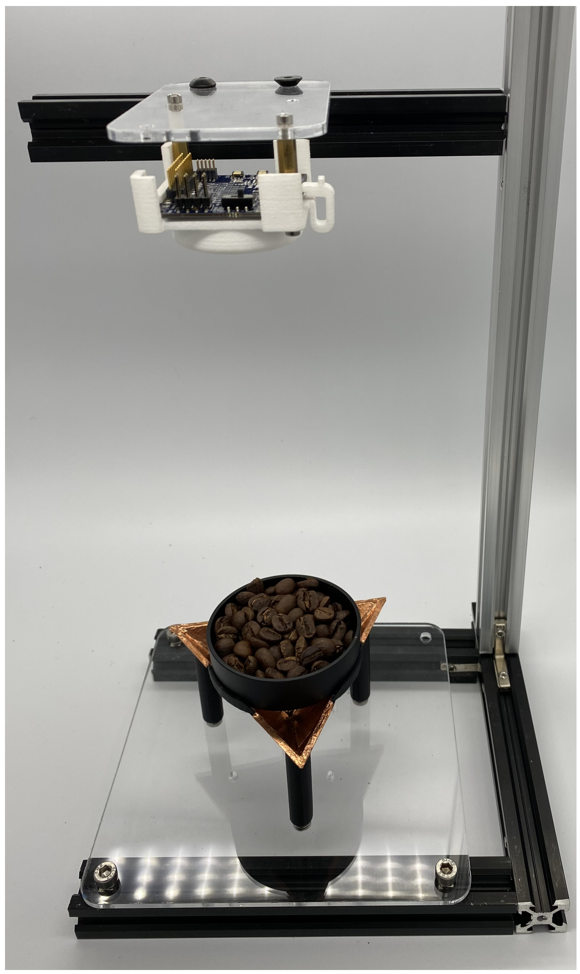

Figure 6: Dr. Mendez's design

Although the open-air configuration lacked the electromagnetic shielding of the aluminum tube, it reduced geometric inconsistencies and internal multipath effects by limiting reflections from scattered radar waves, factors that likely contributed to unreliable measurements and model misclassification in our original design. By simplifying the wave propagation environment and reducing the impact of unpredictable diffraction and attenuation, this setup enabled more consistent amplitude variations across roast levels.

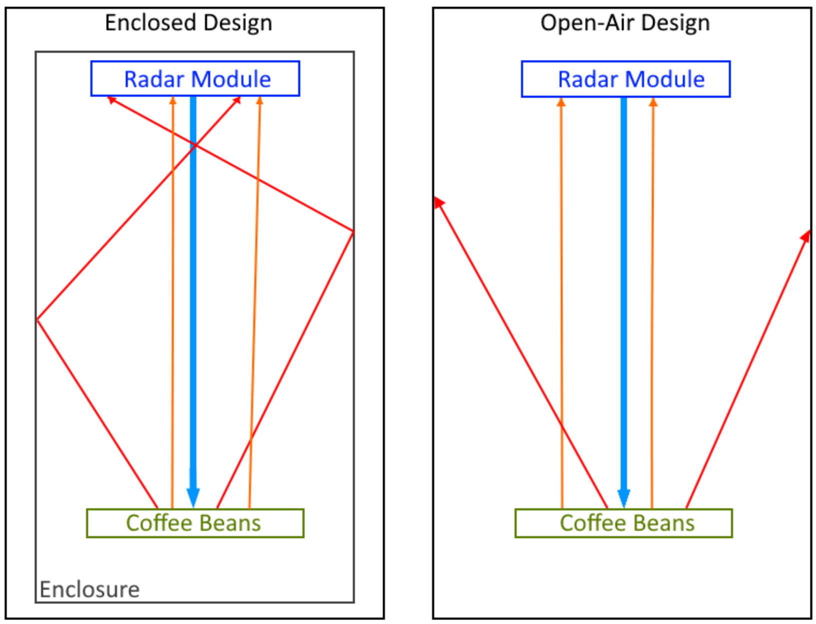

Figure 7: Scattering Effect Comparison

The preliminary results suggest that the primary sources of error in our models performance were variations in bean geometry and internal moisture content. Roasting significantly reduced water content and minimized the moisture differential between samples, thereby mitigating the effects of moisture-induced attenuation. As a result, the alternative setup and bean samples produced more stable radar reflections, greatly increasing the radar's ability to reliably classify beans.

7 Test Plans

7.1 Top Down Test Plan v1.1

7.1.1 Purpose

The purpose of this test plan is to evaluate the complete COATL mmWave RADAR system designed for differentiating between green coffee bean varieties using I/Q data collected via the XM125 radar. This includes verification of both hardware and software functionality, reliability of the data collection pipeline, and effectiveness of the machine learning model in a live testing environment.

The testing procedure begins with structured data collection using a 3/8" diameter aluminum tube and a guided scanning protocol. Data is collected from three green coffee varieties: *Sweet Maria's Guatemala Antigua Hunapu* (7.8%), *Rulindo Tumba* (9.7%), and *Yemen Mokha Sanani* (11.4%). A total of 100 series of scans per variety are performed, with 3,300 I/Q measurements per series, producing a dataset of approximately 330,000 data points.

Following data collection, the test plan outlines steps to train and validate the machine learning model using the compiled dataset. The model is then used in live testing to predict the variety of beans placed into the device based on real-time radar data. The goal is to assess the systems accuracy, repeatability, and responsiveness in a realistic classification scenario. This test plan ensures all components of the project, from data collection to machine learning inference, are functional and aligned with the system's performance objectives.

7.1.2 Equipment Needed and Pre-Test Setup

Refer to the Developer's Manual in Section 12 for a complete list of all required hardware, tools, and software packages. Follow the instructions for hardware assembly and software environment setup prior to initiating any data collection or machine learning operations.

7.1.3 Top-down Test Steps for Data Collection

1. Read the Developer's Manual and User's Manual in Section 12 to understand hardware assembly and software operation.
2. Choose one of the three green coffee bean varieties.
3. Use a metric scale to weigh out 3g of beans.
4. Run the "FullAverageScan (Green Coffee).py" script.
5. After calibration, carefully insert the 3g bean sample into the tube.
6. Place the corner reflector on top of the tube.
7. Follow all printed prompts.
   - Important: After each scan, pick up the full assembly and shake it a few times to vary bean orientation.
8. Complete all 10 scans, then remove and store the beans in a separate container.
9. Repeat until 100 runs of the program are completed per variety.
10. The I/Q data will be exported to a CSV file within the same directory as the program file. Successive runs will append the I/Q data to the same CSV file automatically.

7.1.4 Machine Learning Model with Live Testing

After all data collection is complete, the user must prepare the dataset for the machine learning model. If data was collected in three separate CSV files (one per bean type), the files must be combined into a single CSV with matching formatting.

- Reference formatting can be found here: CombinedBeanDataAll3(100).csv

- The formatting of the combined CSV must match the above file exactly for the following script to function correctly.
- Open “Final Machine Learning Program with Live Testing.py” in your IDE.
- Modify line 16 to reflect the path and filename of your combined CSV dataset.

7.1.5 Top-down Test Steps for ML Model Testing

1. Ensure the hardware is connected and the system is assembled as described in earlier sections.
2. Run “Final Machine Learning Program with Live Testing.py” with the tube empty to allow the system to calibrate.
3. Once calibration is complete, add 3g of one of the three coffee bean types used in the data collection step into the tube.
4. Place the corner reflector on top.
5. Follow all on-screen prompts. The system will perform a scan and attempt to predict the variety of beans based on the training data.

7.1.6 Test Plan Conclusions / Discussion

This full-system test was developed to validate the functionality of the COATL RADAR device, from data collection to live machine learning inference. The target dataset of 330,000 data points, derived from 100 runs per bean variety, was intended to serve as a comprehensive foundation for training and evaluating a classification model.

A key objective was to ensure consistency and repeatability in the data acquisition process. The inclusion of physical variation between scans, by shaking the tube, was intended to introduce enough diversity in bean geometry to promote a generalized model response.

However, despite the volume of data collected, current results indicate that the machine learning model is only accurately predicting the correct variety of bean approximately 30% of the time. We hypothesize that this underperformance is due to the following factors: inconsistencies in the data collection method, the fact that the radar's wavelength is comparable to the size of an individual coffee bean (creating complex scattering behavior), significant overlap in I/Q data between the different bean varieties, and possible improper tuning of the machine learning models parameters (e.g., learning rate, number of epochs, or network architecture).

A more detailed analysis of these potential issues, along with proposed modifications and next steps, is presented in Section 8 and Section 9.

7.2 Top Down Test Plan V1.2 (Roast Comparison)

7.2.1 Purpose

The purpose of this test plan is to evaluate the complete COATL mmWave RADAR system designed for differentiating between different roasts from the same coffee bean using I/Q data collected via the XM125 radar. This includes verification of both hardware and software functionality and reliability of the data collection pipeline. The testing procedure begins with structured data collection using our new 3D printed testing apparatus. Data is collected using *Yemen Mokha Sanani* green coffee beans roasted to three different levels: light roast, medium roast, and dark roast. A total of 50 series of scans per variety are performed, with 3,300 I/Q measurements per series, producing a dataset of approximately 165,000 data points.

7.2.2 Equipment Needed and Pre-Test Setup

Refer to the Developers Manual in Section 12 for a complete list of all required hardware, tools, and software packages. Follow the instructions for hardware assembly and software environment setup before initiating any data collection or machine learning operations.

7.2.3 Top-down Test Steps for Data Collection

1. Read the Developers Manual and Users Manual in Section 12 to understand hardware assembly and software operation.

2. Choose one of the three coffee roast varieties.

3. Fill and level the provided testing container.

4. Run the FullAverageScan (Coffee Roast).py script.

5. Follow all printed prompts while turning to the set measured steps on the bottom of the testing container in between each prompt.
   Important: At the start of each test, place the testing container with the star as the start point. This will ensure that you finish the required number of turns for the test with the same positioning each time.

6. Complete all 50 scans, then remove and store the beans in a separate container.

7. Repeat until 5 runs of the program are completed per roast.

8. The I/Q data will be exported to a CSV file within the same directory as the program file. Successive runs will append the I/Q data to the same CSV file automatically.

8 Results

The primary objective of this study was to develop a reliable method for classifying green coffee beans based on their moisture content using the 60 GHz pulsed coherent radar (Acconeer XM125). The base hypothesis for this study was that variations in moisture content would produce distinct radar amplitude readings, enabling consistent differentiation between bean samples. However, the results revealed significant challenges due to the radar's wavelength (∼5 mm) closely matching the average width of a coffee bean (∼5 mm), placing the system within the Mie scattering regime. This section presents the outcomes of the primary experiment, supported by a brief mathematical analysis rooted in Mie scattering theory, and concludes with insights from a secondary experiment using an alternative hardware configuration.

The primary experiment involved classifying three coffee bean varieties: *Sweet Maria's Guatemala Antigua Hunapu* (7.8% moisture), *Rwanda Rulindo Tumba* (9.7% moisture), and *Yemen Mokha Sanani* (11.4% moisture), using a deep learning model trained on radar I/Q amplitude data. The setup utilized a 3/8" diameter aluminum tube, a custom corner reflector, and a 3D-printed holder for precise radar alignment. Data collection involved averaging multiple scans with manual shaking to randomize bean orientation, mitigating position-based inconsistencies. The dataset comprised 330,000 input vectors, each containing 50 real and 50 imaginary amplitude points.

The deep learning model, with five hidden layers (256 to 16 neurons), reported a training accuracy of approximately 95%. However, live testing on shaken samples yielded a classification accuracy of only ∼33%, consistently favoring the highest-moisture beans (*Yemen Mokha Sanani*). Analysis of the amplitudes revealed significant overlap across the three moisture levels. Drier beans generally exhibited higher amplitude peaks across the sweep due to lower signal attenuation, while moister beans showed greater attenuation. However, the standard deviation of amplitude measurements for a single sample type often exceeded the differences between samples of varying moisture contents, rendering classification unreliable.

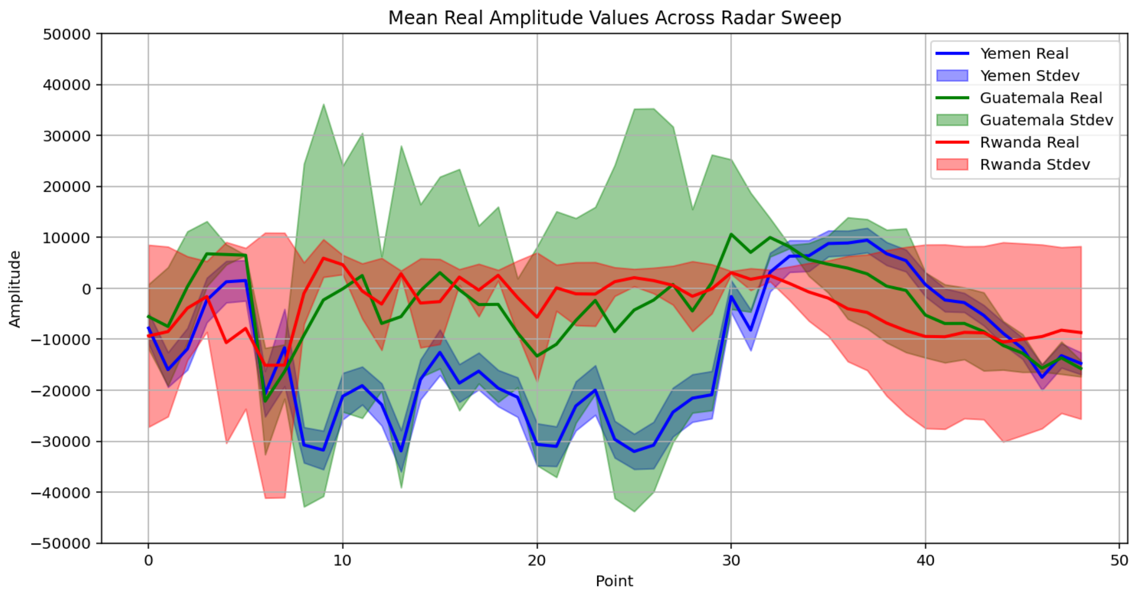

Figure 8: Mean Real Amplitude Comparison in 3/8" Diameter Aluminum Tube

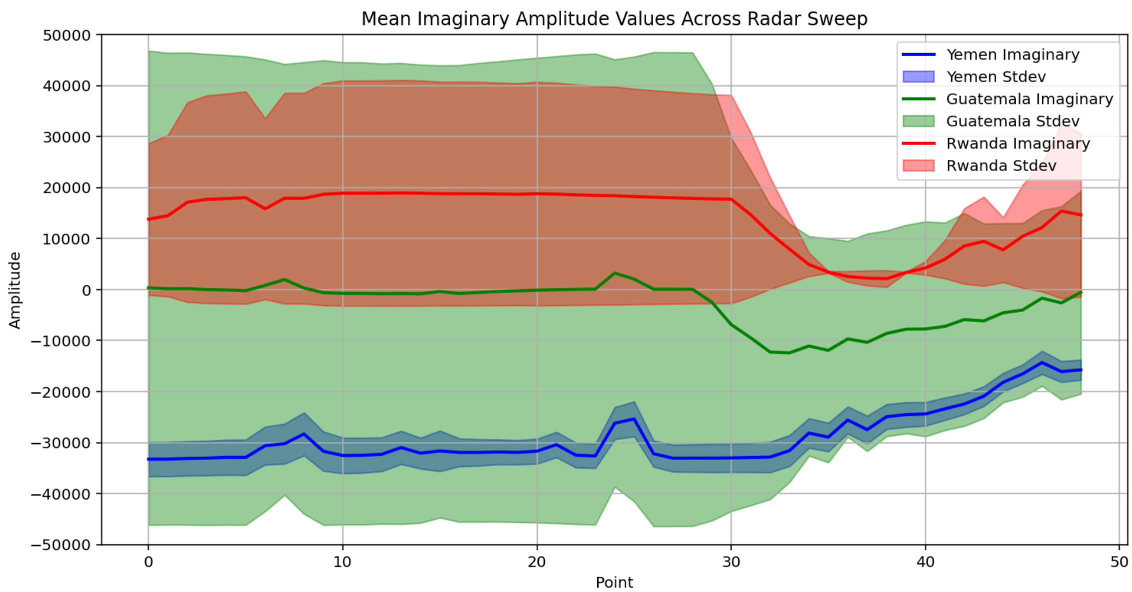

Figure 9: Mean Imaginary Amplitude Comparison in 3/8" Diameter Aluminum Tube

The primary cause of this inconsistency lies in the scattering behavior governed by the size parameter, $x = \frac{2\pi r}{\lambda}$, where r is the characteristic dimension of the coffee beans ($\sim$2.5 mm radius) and λ is the radar wavelength (5 mm). This yields $x \approx 0.8\pi\,(\approx 2.513)$, placing the system in the Mie scattering regime ($x \approx 1$ or greater). In Mie scattering, relevant for particles comparable to the wavelength, complex, angle-dependent scattering patterns are produced with significant interference effects. For our system, the size parameter indicates Mie scattering dominance, leading to multiple scattering events within the confined 3/8" ($\sim$9.5 mm) diameter tube.

The scattered electric field in the Mie regime can be approximated as:

$$E_{\text{scattered}} = E_0 \sum_{n=1}^{\infty} \left(a_n \pi_n(\cos\theta) + b_n \tau_n(\cos\theta)\right),$$

where E_0 is the incident field, a_n and b_n are Mie scattering coefficients dependent on the size parameter and refractive index, and π_n and τ_n are angular functions. These coefficients are highly sensitive to small variations in bean geometry and orientation, causing unpredictable interference patterns, which is the likely cause for our inconsistent amplitude readings. The confined tube geometry exacerbated this by limiting bean separation, leading to overlapping scattering paths and constructive/destructive interference, as observed in the high variance of I/Q amplitudes.

Rayleigh scattering ($x << 1$), while believed to be less dominant, also contributes to background scattering for smaller bean features or irregularities. The Rayleigh scattering cross-section is given by:

$$\sigma_{\text{Rayleigh}} \propto \left(\frac{2\pi}{\lambda}\right)^4 |\epsilon_r - 1|^2,$$

where ϵ_r is the relative permittivity of the beans (estimated at 2.5–3.5 for green beans). However, the Mie regime's complex scattering almost certainly overwhelmed these contributions within the confines of our aluminum tube, as evidenced by amplitude variations that were uncorrelated with moisture content (which was unchanged) after shaking.

To further investigate performance limitations and mitigate the effects of both Rayleighs and Mie scattering, we collaborated with Dr. Joshua Mendez using an open-air radar setup with roasted *Ethiopian* and *Peaberry* beans across four roast levels with a tighter moisture content range (Dark $< 1.0\%$ (DD), Medium-Dark $1.2\% \pm 0.2\%$ (MD), Medium-Light $2.0\% \pm 0.2\%$ (ML), Light $2.4\% \pm 0.1\%$ (LL)). By removing aluminum tube-induced scattering amplification and reducing internal moisture through roasting, the setup minimized multipath interference and dielectric variability, resulting in more consistent reflections driven primarily by surface geometry.

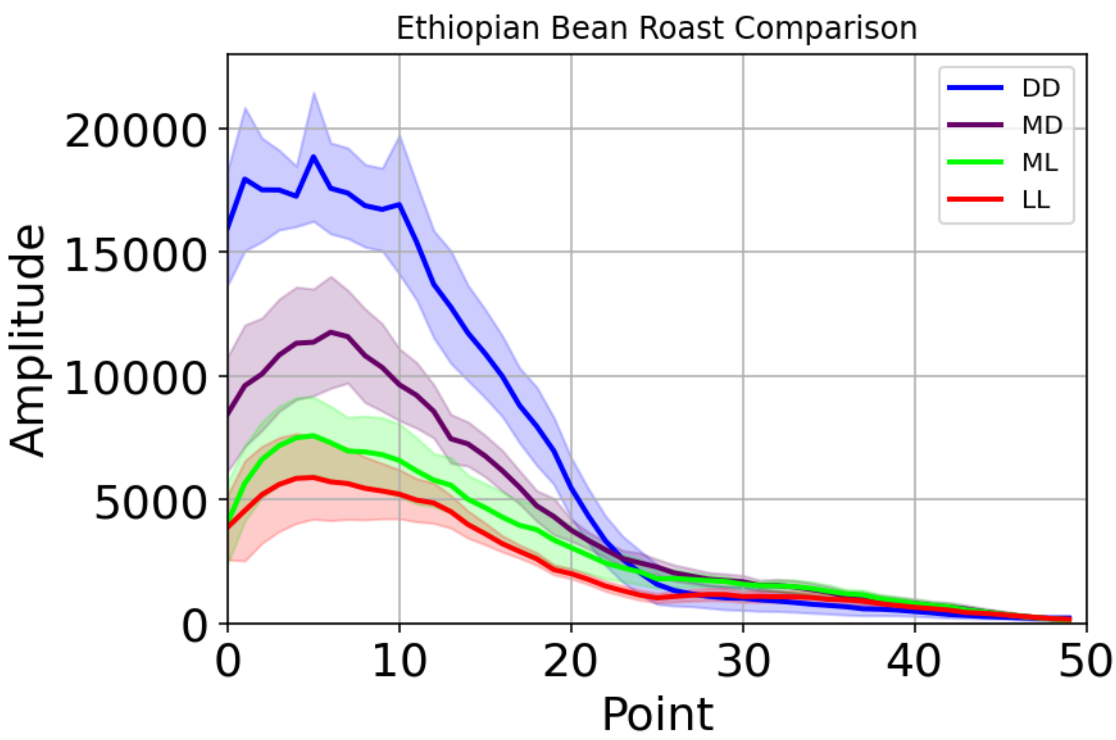

Figure 10: Roasted Peaberry Bean Amplitude Comparison in Open Air

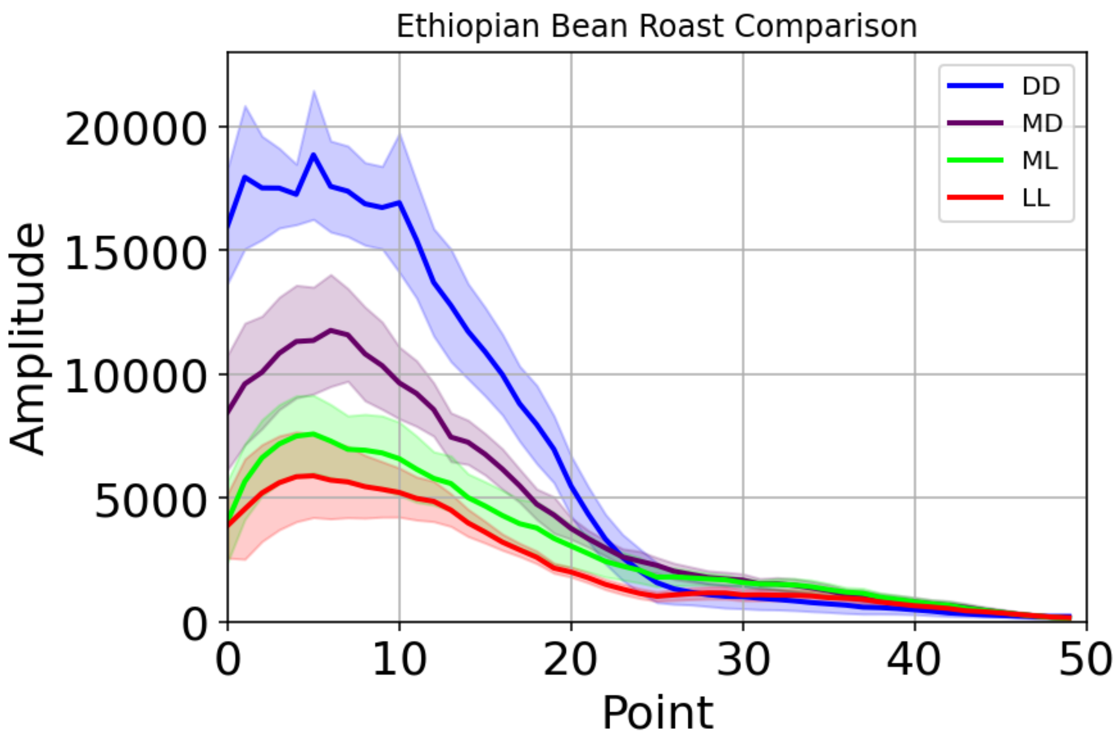

Figure 11: Roasted Ethiopian Bean Amplitude Comparison in Open Air

These conditions led to more consistent radar reflections across roast levels and reduced variance in both real and imaginary amplitude components. In contrast to the high variability observed in the green bean trials, the roasted samples produced tighter amplitude distributions, with distinguishable trends across roast levels. To further evaluate the impact of this improved setup, we also tested it using our original green bean samples. While this configuration did reduce overall standard deviations compared to the aluminum tube setup, suggesting a modest improvement in measurement consistency, the resulting variance remained significantly higher than that observed with the roasted beans. This demonstrated that even with reduced amplification of Mie scattering from geometric conditions, the internal moisture variability in green beans due to Rayleigh scattering dominated the signal response, overwhelming the classification model's ability to extract consistent features.

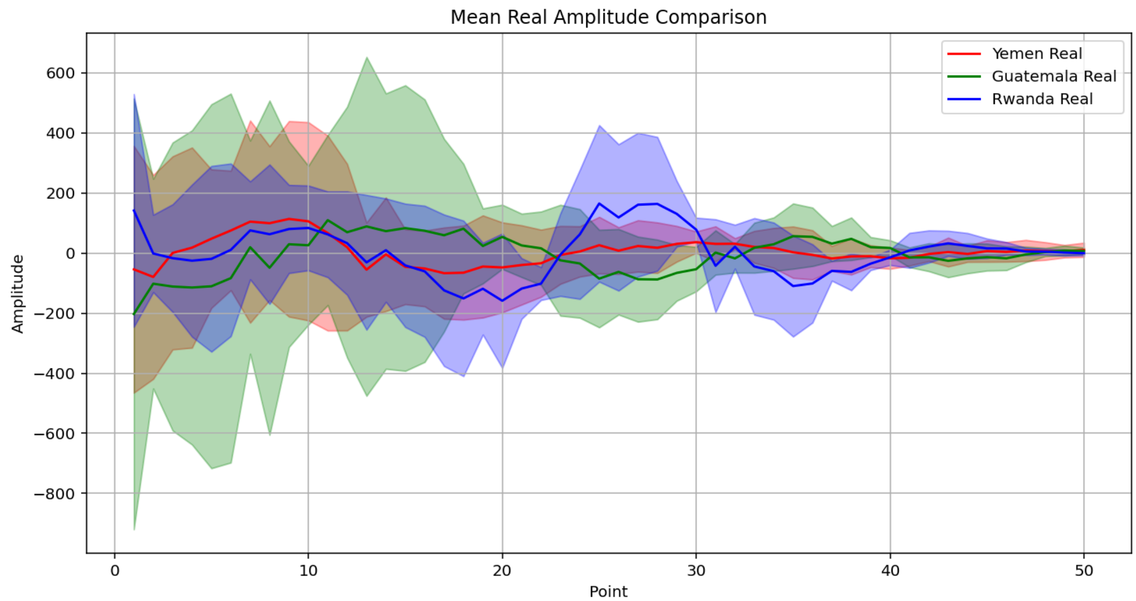

Figure 12: Mean Real Amplitude Comparison in Open Air

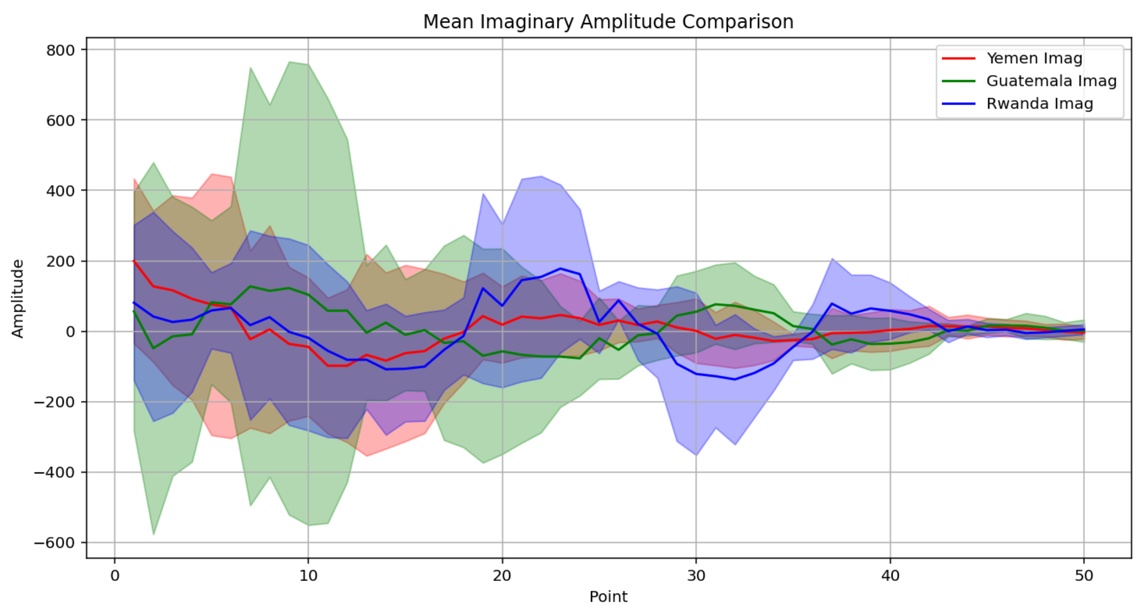

Figure 13: Mean Imaginary Amplitude Comparison in Open Air

These results support the conclusion that the poor performance of our original model was not due to deficiencies in the machine learning architecture, but rather to compounded signal variability stemming from moisture content, bean geometry, and the confined scattering environment. Moisture plays a particularly disruptive role in radar sensing due to waters high dielectric constant, which introduces substantial attenuation and unpredictable phase shifts. Variations in water content within green beans can lead to large differences in effective permittivity, drastically altering radar wave behavior and obscuring any normally repeatable patterns. Roasting the beans reduced this internal moisture, stabilizing dielectric properties and shifting the signal response toward surface geometry rather than internal composition, mitigating Rayleigh scattering. Combined with the open-air configuration that minimized multipath interference caused by Mie scattering, the secondary experiment enabled more reliable and interpretable radar measurements.

9 Post-Mortem

9.1 What worked

One of the most successful aspects of the project was our ability to develop a design that maintained stability in the presence of external interference. Through several iterations, we arrived at a physical setup using an aluminum tube and custom corner reflector that effectively mitigated signal noise. This hardware configuration enabled us to consistently gather clean radar returns in various environments, validating the physical design's resilience against ambient interference.

Another key success was the effectiveness of our data acquisition methodology. By averaging scans and incorporating randomized shaking between samples, we improved consistency and created a large dataset—over 330,000 input vectors—that supported extensive machine learning training and testing. This data acquisition-to-machine learning training pipeline functioned reliably across testing sessions and produced a well-structured dataset suitable for future work.

Additionally, we found encouraging results in a secondary experiment conducted late in the project, where we mimicked a study previously performed by Dr. Mendez involving roasted coffee beans of varying roast levels: light, medium, and dark. Although this experiment was not a primary requirement of our original project scope, it yielded promising results. The radar data for these roasted samples showed clearer and more consistent distinctions between roast levels than what we observed with green coffee beans of varying moisture content. This suggests that the system may be better suited for applications where the physical and dielectric properties of the sample differ more significantly. These results validate the radar's capability to detect meaningful differences under more favorable conditions and offer a potential future application path for the system.

9.2 What didn't work

The most major issue we had was rooted in the physical limitations of the sensing modality. The 60 GHz radar module used in this project, which corresponds to a wavelength of approximately 5 mm, proved poorly matched for detecting moisture variations in coffee beans whose average size is also around 5 mm. In radar applications, when the wavelength is close to the size of the target, scattering effects such as resonance, diffraction, and multi-path interference become highly pronounced. This leads to inconsistent returns and signal degradation, which we observed even with careful data collection protocols.

Attempts to work around this limitation—including reducing sample size, redesigning enclosures, and averaging across multiple beans—were only partially successful. The fundamental issue persisted: the radars resolution and the scattering behavior of similar-sized targets created high variance in readings and undermined classification accuracy.

9.3 What we would do differently

If given the opportunity to revisit this project, our team would have prioritized a radar system with a more appropriate wavelength from the start. Although 60 GHz radar is highly sensitive to water due to absorption properties, the close match between the radar wavelength and bean dimensions severely impacted accuracy. A different mmWave radar module, perhaps in the 24-40 GHz ($\lambda = 12.5\text{mm} - 7.5\text{mm}$) range, would maintain high moisture sensitivity while increasing the wavelength relative to the object size, thus reducing scattering and improving overall signal coherence.

9.4 What we wish we had known

We wish we had fully understood the implications of target size relative to radar wavelength at the beginning of the project. Although we knew that 60 GHz radar is highly sensitive to water content due to dielectric interactions, we underestimated how problematic it would be to resolve meaningful signals when the radars wavelength (∼5 mm) was nearly identical to the average dimensions of the coffee beans. This wavelength-to-target size ratio produced complex diffraction and scattering behavior that made signal interpretation difficult and inconsistent from the start.

Additionally, we wish we had more thoroughly evaluated the feasibility of extracting permittivity from I/Q data early in the project timeline. While it is theoretically possible to determine dielectric properties from reflected radar signals, our attempts were significantly hampered by two key issues: the lack of formal documentation from Acconeer regarding how their I/Q amplitude values are scaled or interpreted, and the broader absence of publicly available methods specific to the XM125 module. As a result, a substantial portion of the project timeline was spent exploring how to convert I/Q data into relative permittivity value—an effort that ultimately proved too time-consuming and uncertain for our scope.

In hindsight, initiating a more critical discussion around abandoning the permittivity extraction goal earlier on would have been beneficial. Although we eventually pivoted to a more direct amplitude-based classification approach, doing so sooner would have allowed us to focus earlier on improving data quality, refining the test apparatus, and optimizing the machine learning model.

9.5 What we would do with more time

With additional time, we would conduct a broader hardware evaluation to compare radar modules across multiple frequencies. A series of controlled experiments using radars with varying wavelengths could help identify an optimal range for moisture detection in organic samples of similar dimensions. We would also refine the ML model architecture to account for non-linearities in scattering behavior, potentially incorporating phase data or other signal processing techniques to boost our model's classification ability.

9.6 Recommendations for the next team

The next team should begin by selecting a radar with a different center frequency, still within the mmWave domain, but with a longer wavelength relative to the coffee beans. Modules operating in a different range could yield cleaner returns and more consistent attenuation patterns. The next team should also explore advanced signal conditioning techniques, such as averaging across phased scans or extracting higher-order features from the complex I/Q data. If feasible, collaboration with RF signal processing experts could provide new insights into interpreting radar returns for complex, organic materials.

Finally, the existing data collection pipeline, enclosure design, and documentation should provide a solid foundation. Building upon these tools with refined hardware and deeper theoretical insight into wave-material interaction will likely lead to better model performance and project success.

10 Project Resources

GitHub: COATL-RADAR Git Repository

- Current data can be found in Software/Data Collection/Datasets
- Current machine learning programs can be found in Software/Machine Learning
- All STL files are found in 3D Modeling

Google Drive: ECE 412/413 Google Drive

- Primarily contains:
  - Meeting Notes
  - Weekly Progress Reports
  - Project Schedule

Tools:

- **Python** (version 3.8, 3.9, 3.10, 3.11, or 3.12)
  - Download: https://www.python.org/downloads/
  - During the installation, please be sure to check the boxes for "*Use admin privileges when installing py.exe*", as well as "*Add python.exe to PATH*". Then click "*Install now*".
- **PyTorch**
  - Within your terminal, type the following:
    `pip3 install torch torchvision torchaudio --extra-index-url https://download.pytorch.org/whl/cu118`
  - or (depending on your setup):
  - `pip install torch torchvision torchaudio --extra-index-url https://download.pytorch.org/whl/cu118`

- **Scikit-learn**
  - Within your terminal, type the following:
    `python -m venv sklearn-env sklearn-env\Scripts\activate # activate`
    `pip install -U scikit-learn`
- **Acconeer Exploration Tool (AET)**
  - Download: Open your terminal of choice, and type:
    `python -m pip install --upgrade acconeer-exptool[app]`
  - This should begin installing the most up-to-date version of the exploration tool. There are a few reasons that this command may fail to run:
    1. Python doesnt have access to your current directory
       * To fix this issue, enter "`where python`" into your terminal, which should provide you with the location of your local Python installation. Copy the file path displayed.
       * Next, enter the following command into your terminal:
         `cd C:\FolderContainingPython\python.exe`
       * Then attempt to rerun:
         `python -m pip install --upgrade acconeer-exptool[app]`
    2. Python install is referenced as `py` rather than `python`.
       * Attempt to rerun the command as:
         `py -m pip install --upgrade acconeer-exptool[app]`
    3. Insufficient PATH access or admin privileges
       * Freshly reinstall Python, ensuring to give it PATH access and admin privileges.
  - To run the AET, enter the following command in your terminal:
    `python -m acconeer.exptool.app`
  - The AET is primarily used for live visualization of the A121's return signal values.

  - It has many settings that you can adjust. Hovering over the name of each setting will display a pop-up window with a description of what each setting does. More detailed descriptions of our particular configuration can be found in Section 12.

- **Spyder 6 IDE**
  - Downloadable either directly or through Anaconda distribution package (recommended)
  - Direct: Spyder IDE for Windows 10
  - Anaconda: Anaconda Distribution

- **Fusion 360**
  - Download: Autodesk Fusion for Education
  - Primary tool used for 3D-modeling of original prototypes and aluminum tube stand

Other resources:

- Datasheets, research articles, and additional guides for software installation can all be found in the "Resources" directory in the Google drive

11 Conclusion

This project set out to develop a non-destructive testing system using a 60 GHz millimeter-wave radar to classify green coffee bean varieties based on internal moisture characteristics. While the original objective proved more complex than initially anticipated—primarily due to the wavelength of the radar being comparable to the physical size of the beans—we successfully developed a complete system that integrates radar sensing, data acquisition, and machine learning classification.

Despite encountering significant challenges related to wave-object interaction and radar signal consistency, the project produced several important technical achievements. Chief among them was the development of a stable hardware configuration capable of mitigating environmental interference and producing clean, repeatable radar returns. Our data acquisition strategy resulted in a large and well-structured dataset suitable for training and evaluating machine learning models.

Although we ultimately chose to forgo attempts to determine permittivity from the I/Q data—due to insufficient vendor documentation and the complexities of interpreting unitless amplitude measurements—we pivoted toward a data-driven classification approach that demonstrated modest performance gains. Additionally, our late-stage experimentation with roasted coffee beans showed clear distinctions in radar return profiles across roast levels, revealing a potential alternative application for the system outside its original scope.

This capstone effort ultimately demonstrated both the promise and limitations of using high-frequency radar for fine-grained classification of organic material. While we were unable to use the Acconeer XM125 to distinguish classes of green coffee beans, we were able to differentiate roasted beans. As such, this approach could serve to monitor the degree of roast, allowing roasters to achieve more consistent products. With refined hardware, an adjusted radar frequency, and further exploration of signal interpretation methods, we believe this technology can still be a viable tool for non-invasive agricultural product analysis for green coffee beans as well.

12 Appendices

12.1 Tooling

All tools can be found listed in Section 10. Specific configuration settings for the final data collection and machine learning programs are discussed below:

12.1.1 Data Collection Parameters

- **Step Length** (`step_length = 1`)
  This parameter defines the spacing between adjacent range points (or bins) in the radar sweep. A value of 1 yields the finest available resolution (approximately 2.5 mm per step), allowing for detailed spatial detection of reflection profiles. This granularity is essential for resolving small variations in signal caused by differences in moisture content.

- **Start Point** (`start_point = 40`)
  This defines the index at which the sensor begins sampling distance. Each step represents approximately 2.5 mm, so a start point of 40 corresponds to a physical distance of approximately 10 cm. This value was chosen based on a measured distance between the radar and where the beans were positioned in the device.

- **Number of Points** (`num_points = 50`)
  Specifies the number of range bins sampled per sweep. Combined with the start point, this covers a sensing window of approximately 10 to 22.25 cm. This range was chosen to encompass the expected position and volume of the green coffee bean sample, ensuring that all relevant reflections were captured.

- **Sweeps per Frame** (`sweeps_per_frame = 1`)
  Indicates how many radar sweeps are collected for each output frame. A value of 1 means each frame is based on a single sweep, providing real-time responsiveness. This is suitable for applications where high temporal resolution is

not required and the goal is to extract statistical or average values over many discrete scans.

- **HWAAS Hardware Accelerated Average Samples** (`hwaas = 500`)
  This setting controls the number of radar pulses averaged internally by the sensor for each range bin in a sweep. A high HWAAS value significantly improves the signal-to-noise ratio (SNR), which is critical when detecting subtle reflection changes caused by varying dielectric properties (e.g., moisture). However, it also increases power consumption and slows down the frame rate, a trade-off acceptable in this static measurement context.

- **Profile** (`profile = PROFILE_1`)
  The selected radar profile determines pulse bandwidth and signal processing characteristics. `PROFILE_1` provides the highest depth resolution and is optimal for near-field, fine-grain analysis. This is particularly beneficial when measuring small shifts in reflection patterns from similar-looking materials, such as varying moisture levels in beans. Profile should be adjusted according to particular test bench set ups.

- **Pulse Repetition Frequency** (`prf = 19.5 MHz`)
  The PRF defines the rate at which radar pulses are emitted. A high PRF (19.5 MHz being the maximum supported) allows for dense pulse averaging under high HWAAS, improving measurement stability and enabling subtle reflection analysis.

- **Receiver Gain** (`receiver_gain = 19`)
  Sets the analog gain applied to the received radar signal. A relatively high gain value was used to enhance weak reflections, which are typical when radar interacts with low-reflectivity, non-metallic targets such as organic materials. Care was taken to avoid signal saturation or excessive noise amplification. Receiver gain should be adjusted according to particular test bench set ups to avoid excessive noise.

- **Phase Enhancement** (`phase_enhancement = True`)
  Enables firmware-level processing to unwrap and stabilize phase data. This setting is crucial for accurate phase tracking and is especially important when estimating material properties like relative permittivity. We enabled phase enhancement in the hopes of producing more precise analysis of the bean's dielectric properties.

12.1.2 Machine Learning Parameters

Model Architecture (Dense (Fully Connected) Neural Network)

- **Input Features** (`in_features=100`)
  Each line of our CSV file contains 100 inputs. These inputs are comprised of a point of real data and a point of imaginary data at each of the 50 distance points in our radar scan.

- **Hidden Layers** (`h1=256, h2=128, h3=64, h4=32, h5=16`)
  ```
  self.fc1 = nn.Linear(in_features, h1)
  self.fc2 = nn.Linear(h1, h2)
  self.fc3 = nn.Linear(h2,h3)
  self.fc4 = nn.Linear(h3,h4)
  self.fc5 = nn.Linear(h4,h5)
  self.out = nn.Linear(h5, out_features)
  ```

- **Output Features** (`out_features=3`)
  One output for each of the beans in our tests (Yemen, Rwanda, Guatemala), only one of which can be asserted for any given input vector.

- **Activation Function** `x = F.relu(self.fcn(x))`
  "Applies the rectified linear unit function element-wise."

- **Loss Function** (`criterion = nn.CrossEntropyLoss()`)
  Used for multi-class classification problems. It calculates the difference between predicted class probabilities and the actual class labels.

- **Optimizer** (`optimizer = torch.optim.Adam(model.parameters(), lr=1e-6)`)
  Adam optimizer is used to update model weights. It combines the benefits of *AdaGrad* and *RMSProp*.

- **Learning Rate** (`lr = 1e-6`)
  Specifies the step size at each iteration while moving toward a minimum of the loss function. A very small value, which ensures gradual learning.

- **Epochs** (`epochs = 1250`)
  Number of complete passes through the training dataset. The model will iterate 1250 times over the training data to minimize the loss.

- **Training/Test Split** (`(test_size = 0.2), (train_test_split(..., test_size=0.2, random_state=15)`)
  20% of the dataset is reserved for testing the model. A fixed random seed ensures reproducibility of the split.

- **Random Seed** (`torch.manual_seed(29)`)
  Sets the seed for generating random numbers to ensure reproducible results for model initialization and training.

12.2 Developer's Manual

12.2.1 Hardware

- 3/8" diameter aluminum tube
- accompanying corner reflector
- 3D-printed aluminum tube stand
- 8 M2.5 x 10mm screws
- 4 M2.5 x 16mm screws
- 4 M2.5 stand-offs

- XE125 Module Evaluation Kit
- LH112 Lens Kit and Holder
- USB-A/C to USB-C data cable
- Computer with USB and/or USB-c ports (depending on cable type). Must have correct version of Python, AET, PyTorch, and Sci-kit installed (see Section 10 for installation instructons)

12.2.2 Tools

- PH0 or PH00 Phillips screwdriver
- 2.0mm hex wrench (allen key)
- Small pocket level

Figure 14: Example pre-assembly setup

12.2.3 Assembly

For the following steps, reference Getting Started: A121 Lens Evaluation Kit for greater detail in assembling the A121 Lens Evaluation Kit:

1. Insert the XM125 PCB into the bottom of the LH112. Ensure it is oriented correctly. You should hear a soft click once it's inserted. There is a tab on the LH112 that secures the PCB in place. You may need to pull it back to insert the PCB. See Figures 15 and 16 below for examples of correct orientation.

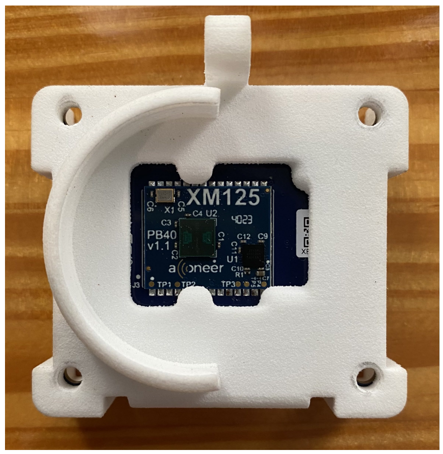

Figure 15: Top-down view

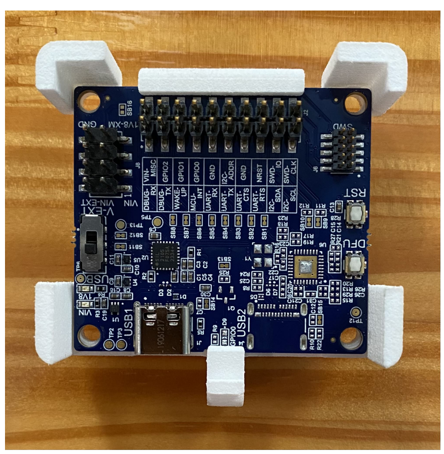

Figure 16: Bottom-up view

2. Decide which lens you'd like to use. Insert it into the lens holder at the depth you'd like, either D1 or D2. See Figures 17 and 18 below for settings.

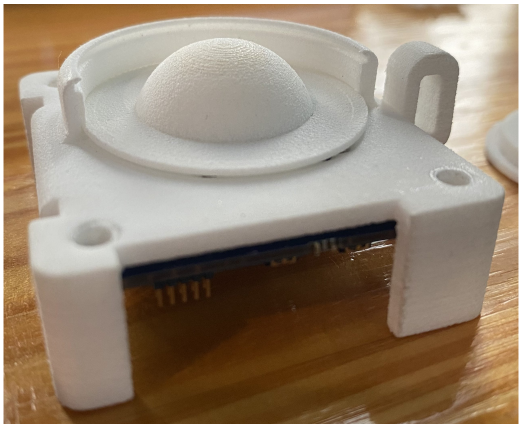

Figure 17: D1 setting

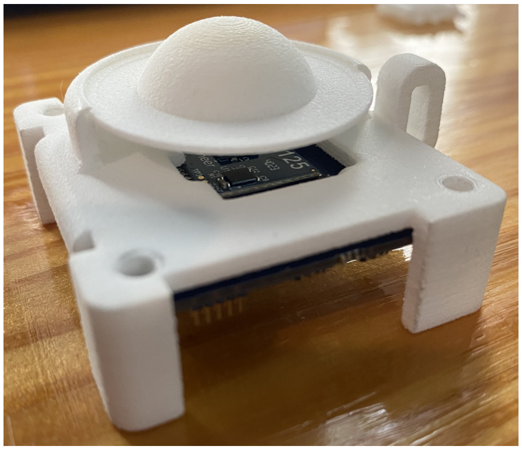

Figure 18: D2 setting

3. Align the stand on top of the lens holder. Rotate it until you find the correct orientation (it only fits in one particular orientation). See Figure 19 below for correct stand orientation.

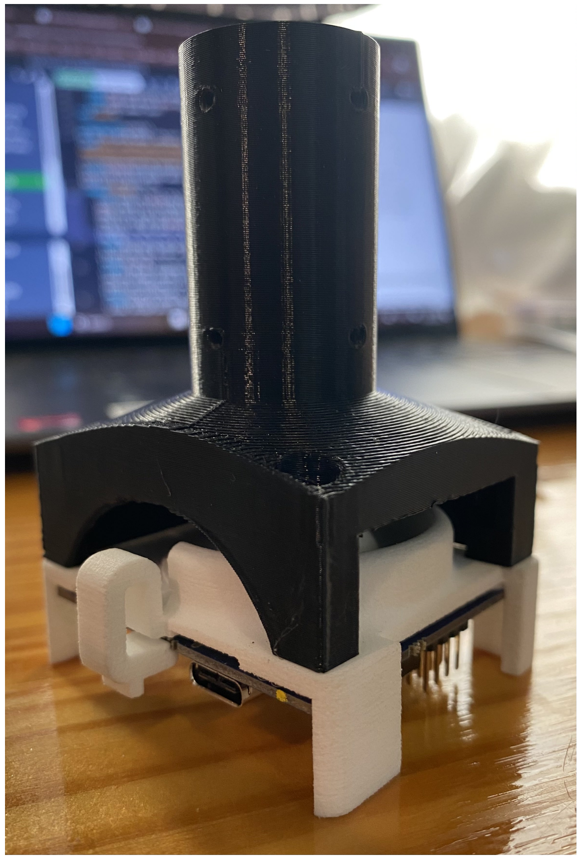

Figure 19: Example of correct stand orientation

4. Carefully insert the four M2.5 x 16mm screws through both the screw holes of the stand and LH112, one at a time. Thread them through the four M2.5 stand-offs. It is easiest to do this by holding the screws in place either by hand or with the appropriate tool (PH0 for Phillips head or 2.0mm allen key for hex head) and turning the stand-offs by hand. Hand-tight threading is fine for this application. See Figure 20 below for how to thread the screws.

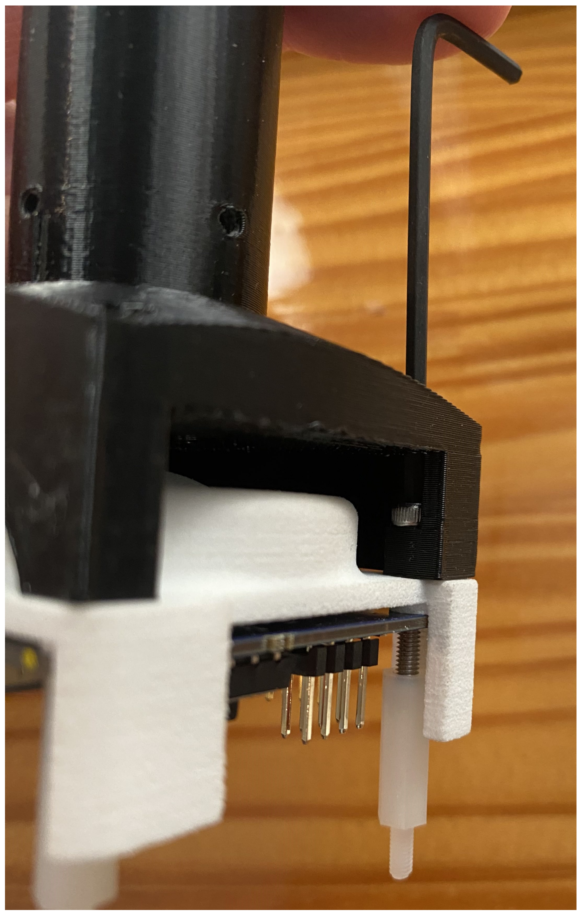

Figure 20: M2.5 x 16mm threading example

5. Start threading the eight M2.5 x 10mm screws ∼1mm into the chimney screw holes. It's easiest to do this using a hex wrench or screwdriver. See Figure 21 below to see what the assembly should look like after this step.

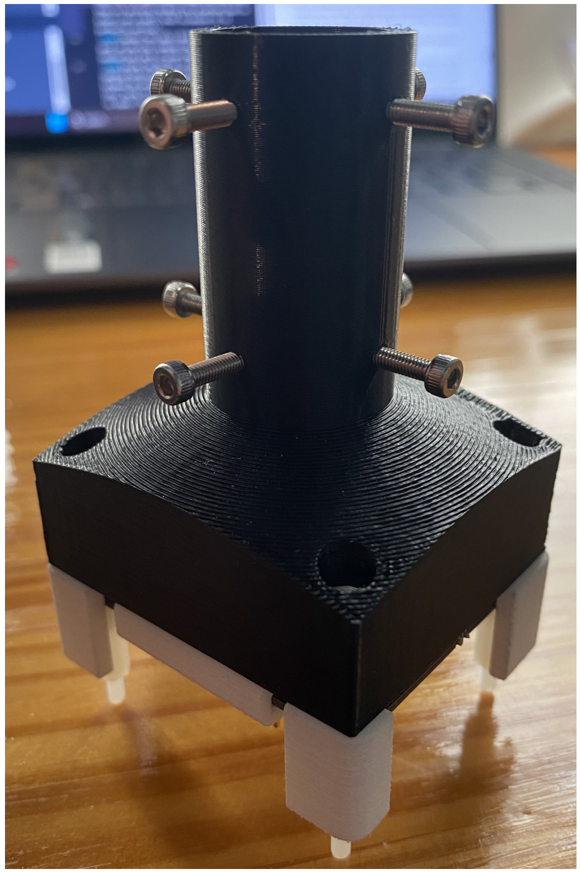

Figure 21: M2.5 x 10mm threading example

6. Insert the 3/8" aluminum tube into the chimney of the stand.

7. Align the aluminum tube such that it's centered directly above the lens (which is centered above the radar). Carefully thread the eight M2.5 x 10mm screws through the such that they are securely holding the aluminum tube in place. Use a small level, if you have one, to ensure it is perfectly vertical. See Figure **??** below for an example of what the finished product should look like.

Figure 22: Complete assembly

8. Next, connect the USB-C data cable to the USB-C port on the PCB. Connect the other USB-A/C to the appropriate port in your computer.

9. Lastly, ensure the device is being recognized by opening Device Manager on your computer. For Windows 10 users, press Win+X, then press M. Open the "Ports" drop down and ensure the COM port is there. The COM port number will vary between users, but there should be one labeled "Enhanced", indicating proper connection between the device and your computer. See Figure 23.

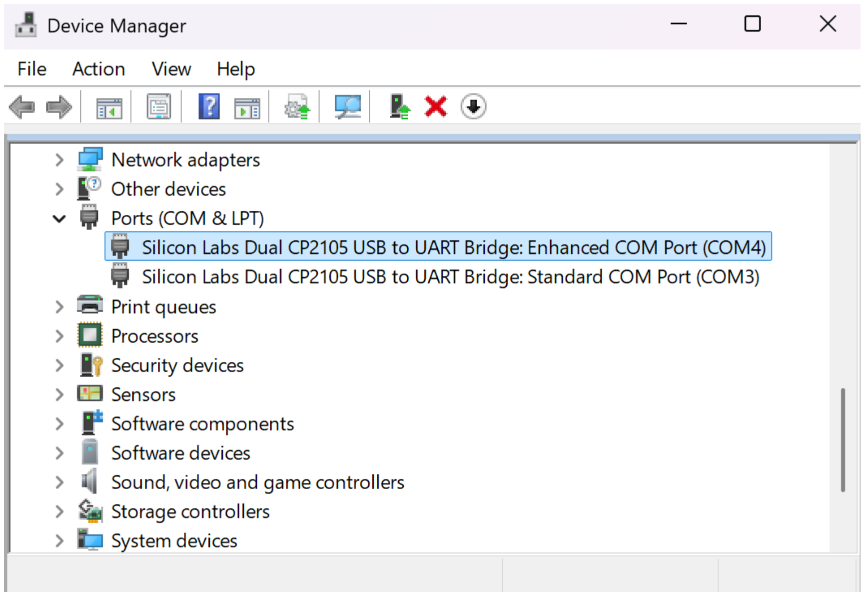

Figure 23: Device Manager highlighting "Enhanced" COM port

12.2.4 Software Guide

If using a brand-new device, the firmware must be updated to enable communication with the radar module. The XE125 EVK should be flashed using the Acconeer Exploration Tool. Installation instructions for the tool can be found in Section 10.

The latest version of the Acconeer firmware file, a121.bin, is also required. It can be downloaded from the official Acconeer developer site: Acconeer A121 Software and Documentation. The documentation also provides direct instructions on how to flash it.

Once the Acconeer Exploration Tool has been installed, our software should be compatible with your device. The only required modification in either the data collection program (FullAverageScan (Green Coffee).py) or the machine learning program (Final Machine Learning Program with Live Testing.py) is to update the serial port name. This can be found via the Device Manager. Locate the following line in the code, `client = a121.Client.open(serial_port="COM8")`, and replace "`COM8`" with the name of your device's enhanced communication port.

12.3 User's Manual

FullAverageScan (Green Coffee).py

- **Exploration Tool Parameters**
  This program is configured for data collection using the 3/8" diameter aluminum tube setup. To modify these settings for a different sensor setup, adjust the following parameters in the script to match the desired values from the Acconeer Exploration Tool: `sensor_config = a121.SensorConfig()`
  ```
  sensor_config.step_length = 1
  sensor_config.start_point = 40
  sensor_config.num_points = 50
  sensor_config.sweeps_per_frame = 1
  sensor_config.hwaas = 500
  sensor_config.profile = et.a121.Profile.PROFILE_1
  sensor_config.prf = 19.5e6
  sensor_config.receiver_gain = 19
  sensor_config.phase_enhancement = True
  ```

- **Scan Settings**
  To modify the scanning procedure, three key parameters in the code can be adjusted. The `CalibrationAmplitudes` and `BeanData` lines each default to averaging 50 scans per group, which can be changed by modifying the `Scans` parameter. Additionally, the total number of scan groups used for bean measurements is controlled by the `ScanGroups` variable. Adjusting these values will impact the resolution and consistency of the final averaged data.

```
CalibrationAmplitudes = MultiScanAverage(client, sensor_config, Scans=50)
BeanData = MultiScanAverage(client, sensor_config, Scans=50)
ScanGroups = 10
```

- **Collecting Data**

  1. To initiate testing, configure the input parameters as outlined in the Developer's Manual, the relevant section of the User's Manual above, and the Tooling appendix.
  2. Place the corner reflector on top of the metal device you wish to use in your test. Ensure the device is empty, place it on a flat surface, and run the program.
  3. After receiving the calibration completion message, insert the bean sample into the device and reposition the corner reflector. Press Enter to start the scan.
  4. Once the scan completion message appears, thoroughly shake the beans within the tube and press Enter again.
  5. Repeat this process until testing is complete. Data will be automatically recorded to a CSV file.

Final Machine Learning Program with Live Testing.py

- **Exploration Tool Parameters**
  To optimize performance, adjust the parameters of the exploration tool to align with the settings used in the data collection program. Ensure these parameters match for best results:

```
sensor_config = a121.SensorConfig()
sensor_config.step_length = 1
sensor_config.start_point = 40
sensor_config.num_points = 50
sensor_config.sweeps_per_frame = 1
sensor_config.hwaas = 500
sensor_config.profile = et.a121.Profile.PROFILE_1
sensor_config.prf = 19.5e6
sensor_config.receiver_gain = 19
sensor_config.phase_enhancement = True
```

- **Scan Settings**
  The program employs the `MultiScanAverage` function from the data collection program, limited to a single scan group for both calibration and data collection scans. Adjust the parameters below to match the data collection program, to modify the number of scans averaged.
  ```
  CalibrationAmplitudes = MultiScanAverage(client, sensor_config, Scans=50)
  BeanData = MultiScanAverage(client, sensor_config, Scans=50)
  ```

- **Training the Model and Testing with Live Inputs**

  1. To train the model, input the CSV file as currently formatted; the program will handle parameter formatting.
  2. Adjust settings per the Developer Manual and Tooling appendices and initiate training.
  3. For testing the trained model, perform a calibration scan with an empty device, insert the bean sample, and press Enter.
  4. The model will output its prediction after a brief wait.

12.4 Additional Testing Documentation

Test Plan v1.0
Testing Processes