WSUAG-Arduino App

Final Report

AgriData Intelligence

Members: Aidan Bachart, Forest Cook, Shreyas Jagalur

CptS 423 Software Design Project II

Spring 2022

TABLE OF CONTENTS

I. Introduction

In the Palouse region, precision agriculture is vital in maximizing crop production. The Palouse region is known for its high crop yields but has challenges like diseases, pests, droughts, and nutrient issues. That is where the project comes in.

At its core, the project has a clear mission to modernize crop monitoring in the Palouse region by utilizing an Internet of Things (IoT) system based on Raspberry Pi sensors to capture wheat development for a growing season in the Palouse region. Thousands of images were collected. In this project, the driving force behind this endeavor is equally straightforward: to automate data analysis, simplify image processing, and efficiently handle substantial data volumes to explore crops responding to diverse factors by implementing advanced computing solutions. The ultimate aim is to provide local farmers with real-time insights, democratizing precision agriculture and ushering in more implementations of heightened crop productivity.

The Project is structured around a rule-based system, with a focus on radiometric correction and plot segmentation. Radiometric correction involves adjusting image data to ensure consistency and accuracy, compensating for variations in light and color. Plot segmentation, on the other hand, entails the identification and separation of individual plots within crop images, facilitating precise monitoring and differentiation of various plant breeds. The overarching goal is to democratize precision agriculture, providing farmers with advanced tools for crop monitoring and decision-making.

In our precision agriculture project, we have two primary clients: Professor Sindhuja Sankaran and Dr. Worasit Sangjan. Both aim to leverage machine learning for automating repetitive tasks in agriculture, emphasizing the need for accuracy and efficiency to enhance their research efforts.

This document serves as a comprehensive progress report, outlining the technical details and advancements achieved during the alpha-prototype phase of the Project. It aims to provide a clear overview of the implemented subsystems, their functionalities, and the corresponding testing efforts. As a summary of the project's evolution, this document facilitates a seamless transition into subsequent development phases, offering insights into the team's engineering efforts and the project's trajectory.

II. Team Members & Bios

Aidan Bachart:

Aidan Bachart, a Computer Science student from Pasco, WA, excels in Operating Systems, Data Mining, and Databases at Washington State University. Experience includes being a Data Acquisition Subteam Lead for Wazzu Racing as well as TA for Intro to Data Structures. Additionally, he is skilled in C, C++, and Python. In our project, Aidan is responsible for testing components, ensuring the robustness and reliability of our system.

Forest Cook:

Forest Cook, from Stanwood, WA, is a Computer Science student at Washington State University with expertise in Operating Systems, Data Mining, and Databases. Forest has practical experience as a Software Engineering Intern at Schweitzer Engineering Labs. Additionally, he is skilled in C, C++, Python, Arm Assembly (AARCH64). Forest leads the implementation of image correction components in our project, ensuring accurate and consistent data processing.

Shreyas Jagalur:

Shreyas Jagalur, a Computer Science major from Sammamish, WA, studies at Washington State University, focusing on Machine Learning, Data Science, and Algorithm Design. Shreyas developed a full-stack web portal and contributed to a data pipeline for ML algorithms predicting snow forecasts. Proficient in C/C++, Java, and Python, Shreyas gained IT internship experience at Boeing. Shreyas plays a key role in our project, focusing on model development for components such as plot identification.

III. Project Requirements Specification

Radiometric Correction:

The radiometric correction component is a fundamental aspect of the project, aiming to ensure consistency and accuracy in image data captured by sensors. This involves adjusting images to account for variations in lighting and color. The system should effectively correct for these variations to provide reliable data for subsequent analysis.

Plot Segmentation:

The plot segmentation functionality is designed to identify and segment individual plots within crop images. This involves distinguishing different pre-labeled breeds of crops and enabling the monitoring of plant development over time. The system's ability to accurately perform plot segmentation is crucial for providing valuable insights to local farmers.

Design Considerations:

The design considerations emphasize key principles that shape the project, including scalability, real-time data processing, efficient automation, and robust data processing. These considerations guide the development team in ensuring the system's capability to handle a large volume of data generated by multiple sensors.

IV. Software Design

The solution approach for the Data Processing Project articulates the envisaged design and components, aligning with the outlined requirements. The project is conceptualized as a sophisticated software system, emphasizing two core components: Radiometric Correction and Plot Segmentation. Radiometric Correction addresses variations in light and color, ensuring consistent and accurate image data for further analysis. Plot Segmentation involves identifying and segmenting individual plots within crop images, enabling precise monitoring of plant development and crop breeds.

The solution approach outlines the implementation details for radiometric correction and plot segmentation components. YOLOv8 models, algorithms, and interfaces constitute the subsystems. The project's current stage doesn't necessitate extensive data manipulation, but potential future considerations include raw data storage and event logs.

The user interface design prioritizes a rule-driven system, foregoing a graphical interface at present. Testing strategies encompass unit testing, integration testing, and system testing, with milestones defined for key testing phases. Environment requirements specify compatibility with a Windows computer running Python 3.9.0, and YOLOv8

V. Test Case Specifications and Results

V.1. Testing Overview

V.1.1. Unit Testing

Due to the nature of our project, automated unit tests that ensure the system is 100% accurate will not be possible to produce. Our project relies on machine learning which is not 100% accurate. We can, however, gain statistics on how accurate the model is likely to be. For part 1 (plate identification), we will have a set of test images we run it against to check the accuracy. The test set will be images that the model has not seen before. This should give us a good idea of what results we can expect.

For part 2 (Image processing) we will run it on a series of test images, and compare the output files to existing outputs. Given that we are using a different method than was previously used and that different parameters may be used, we will allow for some variation. We will also run unit tests with a high zoom out parameter to allow the user to see that the individual boxes have been correctly identified.

For part 3 (Plot identification) we will likely be using a deep learning model, thus it will suffer from the same uncertainty as any other model. Given this we will use a test set of images to get statistical information about its accuracy. The test data will be a separate dataset that has not been seen before as with part 1.

V.1.2. Integration Testing

For integration testing we will have two phases, first, we will run it on a set of test images where it outlines the plot and boxes, allowing us to visually verify that all components are outputting the correct data. Then we will run the whole system and verify that it is producing output similar to the previous results. Given the different methods employed we expect some variation, but this should allow us to detect problems with the system.

V.1.3. System Testing

System testing is a type of black box testing that tests all the components together, seen as a single system to identify faults with respect to the scenarios from the overall requirements specifications. Entire system is tested as per the requirements. The key component that will be stressed and tested in this step is the pipelining system between the separate model components, i.e., the piece that allows the entire system to act as a cohesive unit.

V.1.4. Functional Testing

In the context of our project, functional testing will validate that our software performs the functions specified in the project requirements. We will develop test cases based on the functional requirements outlined in our project documents, including radiometric correction and plot segmentation. These test cases will encompass various scenarios, such as image correction, plot identification, and plot normalization. The software will be executed with the defined test cases, simulating real-world use cases. The outcomes will be compared to the expected results to identify any discrepancies.

V.1.4. Performance Testing

Performance testing for our project is vital to ensure it meets nonfunctional requirements and design goals. We will focus on load testing, stress testing, and performance under typical operational conditions.

Details:

- Load testing will evaluate the software's performance under typical operating conditions, such as processing a substantial number of images.
- Stress testing will examine how the system behaves when pushed beyond its specifications, helping us identify its failure points.

We will monitor and record response times, throughput, and resource utilization as well and the performance data will be analyzed against predefined performance goals and nonfunctional requirements to ensure our system performs optimally under various conditions.

V.1.4. User-Acceptance Testing

In the context of our project, user acceptance testing (UAT) ensures that our system aligns with the project agreement and is ready for operational use. This phase involves end-users who will assess the system's functionality and performance.

Details:

- End-users, potentially with assistance from developers, will compare the system's functionality and behavior against the initial requirements and project agreement.
- We will design test scenarios to reflect real-world usage of our software, including radiometric correction and plot segmentation.
- End-users will actively interact with the system to validate its performance and functionality.

- Feedback and observations will be collected and analyzed to determine whether the system is ready for operational use.

V.2. Environment Requirements

Necessary
- PC with windows 10
- Python >= 3.9
- Python libraries (listed below)

Preferred
- Strong GPU
- Cuda

V.3. Test Results

This section still requires finalization and will be included in Draft 3.

VI. Projects and Tools used

Tool/library/framework	Quick note on what it was for
Cuda	Used to train the model.
Matplotlib	Graphics creation and data visualization, alongside seaborn.
NumPy	Efficient statistics and matrix math.
Pandas	Efficient manipulation of large datasets.
PyTorch	Used to train the model.
Roboflow	Object detection and image segmentation models.
Seaborn	Graphics creation and data visualization, alongside matplotlib.
Ultralytics YOLOv8	Object detection and image segmentation models.

Languages Used in Project

Python	Markdown

VII. Description of Final Prototype

VII.1. Subsystems

The prototype is split into 3 distinct sections. Subsystem 1 is Plate Identification. Subsystem 2 is Plot Segmentation. Subsystem 3 is Normalized Difference Vegetation Index (NDVI) Growth Curve Plotting.

VII.1.1. Plate Identification

The primary piece in this subsystem is the model utilizing YOLOv8 that has been trained to locate a reference panel in an image. Currently, this subsystem operates with the following functionalities.

- **Data Collection:** Implemented a module to collect raw data from sensors capturing crop-related images.
- **Preprocessing:** Conducted basic preprocessing on the captured images to enhance the quality of data for subsequent analysis.
- **Machine Learning Model:** The Plate Identification model has been designed and implemented using YOLOv8. This model aims to identify and locate the radiometric correction reference panels within the crop images.
- **Training Interface:** Developed an interface for training the Plate Identification model using labeled datasets.

VII.1.2. Image Processing

The Image Processing subsystem focuses on adjusting image colors based on radiometric correction reference panels. The following functionalities have been implemented:

- **Pixel Information Retrieval:** The algorithm retrieves from the identified reference panels.
- **Image Adjustment:** The Image Adjustment Algorithm has been designed to adjust image colors using the pixel information obtained previously.

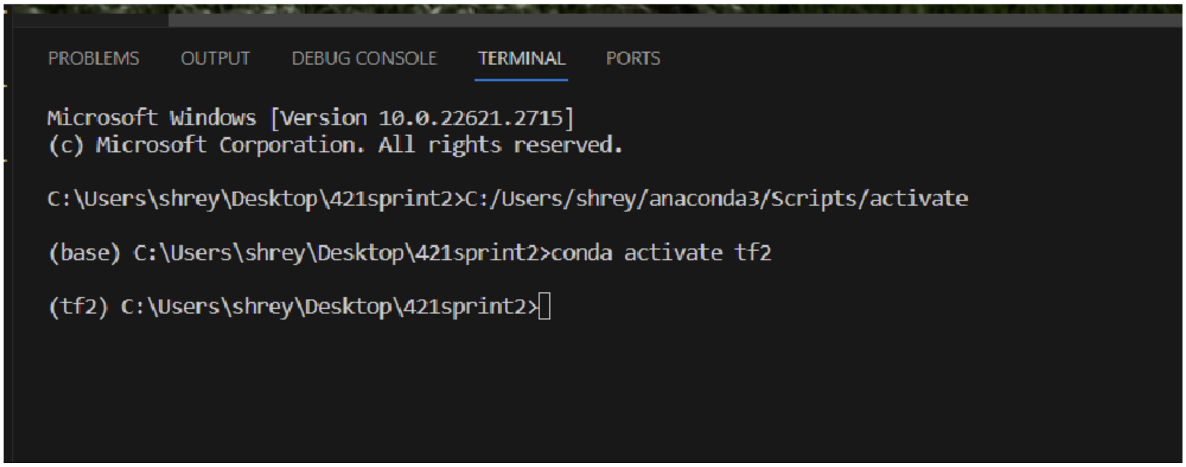
VII.1.3. NDVI Growth Curve

This subsystem gathers base values from the reference plate, adjusts the dataset being analyzed using those values and finds the resulting NDVI values from there. Features include:

- **Dataset Analyzation:** The subsystem will take data from the previous two subsystems and find NDVI values from them.
- **Growth Curve Plotting:** The subsystem will take these values and display them graphically.

VII.2. Interface

As the client has not specified a need of an interface, the project is projected to be ran autonomously and is currently using terminal commands to perform actions like below:



VIII. Product Delivery Status

This section still requires finalization and will as we deliver a more final product to the client.

IX. Conclusions and Future Work

IX.1. Limitations and Recommendations

The primary limitation we encountered was the model not being ideal, as no machine learning model is ideal. We continued to improve it every sprint, however it could still use more improvement.

IX.2. Future Work

This section still requires finalization and will as we deliver a more final product to the client.

X. Acknowledgements

Thank you to Sindhuja Sankaran for providing us with the opportunity to work on this project, as well as the Washington State University Plant Phenomics Facilities.

XI. Glossary

Edge Computing - A method where computers do jobs right where they are needed, in order to make things faster and more efficient.

Image Correction - Correcting the images in order for them to all be based on the same base reference values.

IoT - Internet of Things (IoT) devices are everyday devices that are connected to the internet and to each other.

Phenotype - The observable characteristics of an organism.

Plot Segmentation - The division of images of crops into separate plots of different breeds of plant.

YOLOv8 - A machine learning library that lends itself well to this kind of work

XII. References

[1] A. Paszke *et al.*, "PyTorch: An Imperative Style, High-Performance Deep Learning Library," 2019. Available: https://proceedings.neurips.cc/paper_files/paper/2019/file/bdbca288fee7f92f2bfa9f701272774 0-Paper.pdf

[2] Gitter, "Building Online Communities: Keras.io," *We've moved to freeCodeCamp.org/news*, Oct. 11, 2016. https://medium.com/free-code-camp/building-online-communities-keras-io-be7256abe9c5 (accessed Sep. 22, 2023).

[3] "Image segmentation | TensorFlow Core," *TensorFlow*. https://www.tensorflow.org/tutorials/images/segmentation

[4] J. Terra, "Keras vs Tensorflow vs Pytorch: Popular Deep Learning Frameworks," *Simplilearn.com*, Jul. 27, 2020. https://www.simplilearn.com/keras-vs-tensorflow-vs-pytorch-article

[5] "Object Detection | TensorFlow Hub," *TensorFlow*. https://www.tensorflow.org/hub/tutorials/object_detection

[6] C. Hansen, "Winter wheat production in Northwest Region up 59 percent from last year," *Morning Ag Clips*, Oct. 04, 2022. https://www.morningagclips.com/winter-wheat-production-in-northwest-region-up-59-percent-f rom-last-year/#:~:text=acres%20from%202021.- (accessed Sep. 29, 2023).

[7] "Study: Climate change a growing threat to wheat | World Grain," *www.world-grain.com*. https://www.world-grain.com/articles/18599-study-climate-change-a-growing-threat-to-wheat#: ~:text=The%20study%20noted%20that%20high (accessed Sep. 29, 2023).

[8] W. Sangjan, A. H. Carter, M. O. Pumphrey, V. Jitkov, and S. Sankaran, "Development of a Raspberry Pi- Based Sensor System for Automated In-Field Monitoring to Support Crop Breeding Programs," *Inventions*, vol. 6, no. 2, p. 42, Jun. 2021, doi: https://doi.org/10.3390/inventions6020042.

XIII. Appendix A – Team Information

Team Members:

Aidan Bachart

Forest Cook

Shreyas Jagalur

[Team photo pending]

XIV. Appendix B - Example Testing Strategy Reporting

This section still requires finalization and will be included in Draft 3.

XV. Appendix C - Project Management

This section still requires finalization and will be included in Draft 3.