ELEX 7890: Capstone Project Completion

Final Report

Vine 'Em And Dine 'Em: VineBot

May 13th, 2019

Abstract

The Canadian agriculture industry has seen sustained growth in the wine industry with global exports totaling $145 million dollars in 2017. The industry has seen labour shortages and many vineyards are turning to automation to fill the need. Current solutions are expensive and imprecise. The harvesting of wine grapes at small to medium sized vineyards requires precision and care to avoid loss of product. These factors support an automated solution that allows for precision picking of grapes. This capstone project report outlines our development of an autonomous robotic picking vehicle aided by computer vision and a suite of sensors for navigation.

Prepared for: Dr. Jeff Bloemink, Dr. Craig Hennessey

Prepared by:	Prep Date
Terry Calderbank	May 19th 2019
Navraj Kambo	May 19th 2019
Brayden Aimar	May 19th 2019

Table of Contents

Table of Figures

Table of Tables

Table of Equations

Acknowledgments

We, the VineBot capstone project team, would like to express our deepest gratitude towards our project mentor, Dr. Craig Hennessey, and our project coordinator, Dr. Jeff Bloemink, for their valuable and constructive suggestions during the planning and development of this project, and their patient guidance. Their willingness to give up time so generously has contributed significantly to our success.

Likewise, we would also like to thank the faculty of electrical engineering at BCIT for providing the knowledge and tools to complete our project and being lenient and understanding with regards to deadlines and course work this term.

Finally, we'd like to thank our loved ones for their support and encouragement throughout our studies and development of this project.

VineBot Capstone Team: Navraj Kambo (left), Braydon Aimar (center), Terry Calderbank (right)

Executive Summary

This report describes the need, business case, and implementation plan for an autonomous vineyard crop harvesting robot capstone project, VineBot.

Background

Our team is a composed of senior electrical engineering students studying at the British Columbia Institute of Technology. Our combined past experiences include working with industrial robots, machine vision, mechanical design and fabrication, and rapid prototyping.

This document is mainly intended for our stakeholders including our capstone mentor Dr. Craig Hennessey, our capstone coordinator Dr. Jeff Bloemink, BCITs School of Energy research committee, and external industry partners who would be our effective end-users.

Several key forces drove this project including the lack of available agricultural labour and demand for automation, the growing Canadian wine export market and potential growth of the agriculture sector, and the absence of precision grape picking systems on the market.

Canada is home to 720 wineries and vineyards [1]. A total of $7.2 billion worth of wine was sold in Canada in 2012, with a gain of 3.1% from the previous year. Canadian winery global exports totaled $145 million dollars in 2017 The forecasted market value for 2018 is projected to be $10.28 billion.

"In 1921, agriculture was the single most common occupation," and throughout the years "farms have become more specialized and the average farm size has increased [2]." However, with the projected market growth in agriculture, labour shortages have become of great concern. According to the California Farm Bureau Federation survey, "55 percent of responding farmers had experienced employee shortages." [3] That study also outlined that "[problems] have been more acute among farmers whose crops require the most intensive hand labour, such as tree fruits and grapes." When asked what actions they have taken in response to employee shortages, one- third of respondents said they used mechanization if available; another 29% attempted or investigated mechanization.

According to Agriculture and Agri-Food Canada, most studies of the impact of climate change suggest that most regions of Canada are projected to warm during the next 60 years. Despite the known challenges that come with climate change such as drought and storm intensity, there are opportunities for the Canadian agriculture industry. Canada is a high latitude country, which brings more pronounced warming than the global average. This could lead to expansion of the growing season to go along with milder and shorter winters like those seen in Arizona and California [4]. Without the available labour pool, this potential for climate-change induced growth in Canadian agriculture could be squandered.

Project Objectives & Deliverables

Working in consultation with local wine producers (Okanagan Valley, Lower Mainland) was an objective of this project. Their engagement with our engineering team and feedback on a minimum viable product was and still proves to be very valuable.

In terms of more technical requirements, the main objective for our group was to design and implement a proof of concept autonomous vehicle capable of navigating a vineyard to pick grapes while also avoiding obstacles. We were able to deliver this objective.

Another objective was to become proficient was use of machine learning for image processing, 3D cameras, GPU processing, LiDAR processing, and robot motion and path planning.

By the end of the project, the following proposed deliverables were completed:

1. A proof of concept autonomous grape picking vehicle

2. A final report documenting our project by the end of the project term
3. An organized git repository with extended project documentation
4. A live demonstration of the project for the technology expo including signage and literature

High-level Overview

Our project is meant to be an autonomous grape harvesting system. To create the autonomous system, we decided to split the task into three separate systems:

1. The harvesting system handles everything involving grape detection and picking, including the vision and robotics required
2. The navigation system handles everything involving mapping and position estimation, and also path planning to route from point A to point B around a vineyard
3. The mission planning system handles the vehicle control, as well as allowing and disallowing when the vehicle is able to enter harvesting and mobile states

Planning

To aid in the development of the project, as well as regulate the amount of work being carried out, a set of milestone goals were created. The project was divided up into 4 milestones:

Milestone 1 (January 29th) – Visual Camera, Machine Learning Test, and LiDAR Processing Complete

Milestone 2 (April 25th) – 3D Camera, NVIDIA Board Integration, and SLAM Complete

Milestone 3 (May 7th) – Path Following, Stem Cutting, Integration, Verification, and Tech Expo Preparations Complete

Milestone 4 (May 19th) – Documentation Complete

Project Implementation and Results

The design and implementation of our project was split into sections according to the project plan. Such sections include development into the vision system utilizing the NVIDIA Jetson TX2, working on robotic control for the project, creating mechanical parts and assemblies for the project and establishing algorithms for position estimation and mapping for navigational purposes.

From a high-level view, the different parts of the project were all connected via a software motion planning module which integrated the navigation systems and the harvesting systems. These modules communicated using an inter-process communication standard made up of string-based commands like "ready" and "allow."

Investigation into effective vision systems for grape detection showed that training a machine learning model using the Darknet framework provided superior and robust results compared to a traditional computer vision-based approach. This trained model in combination with the NVIDIA Jetson TX2 development board provided an effective solution that could complete 7 inferences a second when detecting grape clusters.

When developing the robotic control systems using a uArm Swift Pro hobbyist robot arm, various flaws including invalid movement ranges and failure under dynamic load were uncovered and managed appropriately by assigning boundaries and limiting the payload. The result was an effective 2.5D visual servoing feedback control system that allowed the arm to home in on a target for picking.

Various mechanical design tasks were completed including development of a lightweight camera mount, a cuttable and disposable PLA stem system, and several iterations of stem cutting robot end effectors.

Cutting mechanisms using solenoids or the existing grip strength of the robot gripper were rejected in favour of a separate servomotor powered cutter. This mechanism in combination with the gripping position of the grape cluster proved to be an effective harvesting system.

Several goals were achieved when developing the navigation systems including: processing and handling of LIDAR data, statistical identification of vineyard features, persistent map generation using the EKF SLAM library, and some explorations into model predictive control for motion planning and path following. The majority of these tests were successful but the precision of the path following system was not great enough to position the vehicle adequately for harvesting. As such, a simple odometry based approach was used for demonstrations.

Conclusions and Recommendations

The overwhelming majority of the project goals and have been achieved and the results of testing are encouraging. There are several paths that may be pursued by our team at this point.

1. We have the option to continue on with developing a larger scale prototype to be tested in an Okanagan vineyard.
2. There is still further opportunity in developing markets like cannabis where significant work in automation has yet to happen.

Either path involves generating interest and raising capital to invest in full scale prototypes using industrial quality components. Thankfully, this project has produced a significant amount of media that may be used in this manner.

1 Introduction

This report describes the design and potential market for an autonomous wine crop harvesting robot, VineBot. It is addressed towards the staff of the BCIT Electrical Engineering faculty, the BCIT School of Energy and our project mentor, Craig Hennessey.

Wine grape harvesting can be a rigorous, time-consuming, and labour-intensive task which has already begun to be automated. There is also a social push towards organic and sustainably produced products and produce, which could be an agricultural automation opportunity. Our senior engineering capstone group aimed to improve on current systems by developing a robotic vehicle that can harvest grape crops autonomously and selectively. This robotic vehicle is equipped to navigate a vineyard without the need for human intervention, work during all times of the day, and successfully detect and harvest wine grape crops. The robotic vehicle was able to accomplish these tasks by using cutting-edge vision technologies, robotics, and various engineering methods we've picked up at BCIT.

This final report will outline what we were able to accomplish with our project, and how we carried out the proposed idea. We will explore technical aspects of the project in-depth and will discuss what did and didn't work for us, in the hopes that future capstone groups can learn from our successes and failures.

In the following sections of this report, topics such as the motivation for our project will be introduced, including end-users and stakeholders, the application and need for our project and key literature items which we feel provide an important insight into the project. Some background information based on the agricultural industry will then be discussed, followed by and the state of the art with respect to wine grape harvesting. Previous work completed on the project will also be introduced and explored in more detail. The scope of our work for this project including objectives, deliverables, and project execution follow, with short detour regarding the external consultations we've received. Then, in the final sections of our report, we will include some project specifics which encompass a high-level overview of the project, a record pertaining to the design and implementation of the project, and the final results including testing and verification information. A summary will then conclude the report, including a short discussion of the results, unintended applications of our technology, and future work to be done.

2 Background and Motivation

Briefly put, the main objective of our project was to develop an autonomous vehicle capable of navigating a vineyard and picking grapes. This was a worthwhile objective because grape picking is labour-intensive, time consuming, costly, and we have identified a good business case. To start, we will identify end-users and stakeholders in the project. The application and need for VineBot are also addressed, including information regarding the wine industry in Canada, crop migration, and agricultural labour.

2.1 End-Users and Stakeholders

The end users and stakeholders of this project can be divided into two categories: internal and external. The internal stakeholders are comprised of ourselves and various BCIT faculty. Our external stakeholders are the end users and industry contacts who have been consulted for this project.

2.1.1 Internal Stakeholders

This capstone project was a continuation of a BCIT School of Energy (SoE) seed-funded project intended to investigate robotic agriculture management. Our previous achievements and the success of the capstone project will benefit the SoE in several ways. The more we achieve with this project, the more value we bring to the SoE investment. Due to its physically tangible nature, our project can be easily used for marketing purposes and promotion of the technical activities engaged in at BCIT.

We considered our mentor Dr. Craig Hennessey to be one of our key internal stakeholders. Craig had been instrumental in the proposal of the original SoE project and had agreed to be our mentor under the condition that we do our best to successfully achieve our design objectives. Craig has also been a proponent for a robotics-based senior design elective for the B.Eng. Electrical program, which we understand to be implemented next year. It is hopeful that the technology we explored may be helpful in implementing the course.

Our capstone coordinator, Dr. Jeff Bloemink was another internal stakeholder. His goal was to provide us with the knowledge and tools to execute a successful capstone project, which he did to the best of his ability.

Finally, the three members of the capstone team were significant stakeholders in this project. We were poised to gain tremendous amounts of technical knowledge and design practice. Our project and teamwork in some cases was leveraged in our collective seek for employment. Note, more information about our capstone team can be found in the Appendix D: Resumes section of this report.

2.1.2 External Stakeholders

Our first intended end users are employees of small and medium sized vineyards. Larger vineyards can absorb the cost of product lost due to the violent and indiscriminate techniques of current mechanized grape picking. We are hoping to create a proof of concept for what can be an accurate and efficient method of picking for smaller enterprises.

We have several industry contacts we consulted and are currently in contact with. The first was Graeme Duncan of Naramata BC. Graeme owns and produces around 6 acres of Chardonnay grapes on his family owned property in the Okanagan. We hope to scale up our capstone project, and test on his personal vineyard with his permission.

The second was Kim Hoath, also of Naramata. Kim is a member of the International Sommeliers Guild (ISG), and an employee of BC VQA in Penticton, BC. The International Sommelier Guild is the world's leading licensed provider of Sommelier education. Established in 1982, the ISG has grown from its North

American roots to become a truly global provider of Sommelier education and certification. BC VQA stands for "British Columbia Vintners Quality Alliance". The BC VQA program is an "appellation of origin" system, like the AOC and DOC systems utilized in France and Italy respectively. The BC VQA system guarantees origin and ensures that qualifying wines meet certain minimum quality requirements.

2.2 Application and Need

Agriculture is a large part of the Canadian and American Economy. With the increasing opportunities in Canada for agriculture due to pending climate change induced crop migration in the near future, new methods for harvesting and monitoring crops will need to be developed [5].

2.2.1 Wine Industry

Canada is home to 720 wineries and vineyards [1]. The wine industry in Canada occupies a large share of in the agricultural market and continues to grow. A total of $7.2 billion worth of wine was sold in Canada in 2012, with a gain of 3.1% from the previous year and almost all provinces reporting gains [6]. $992 million of this was Canadian domestic shipments, and $61.8 million was exports [7]. This reflects well on the demand for Canadian wine. As for the current market, Figure 1 shows the projected growth of wine industry [8].

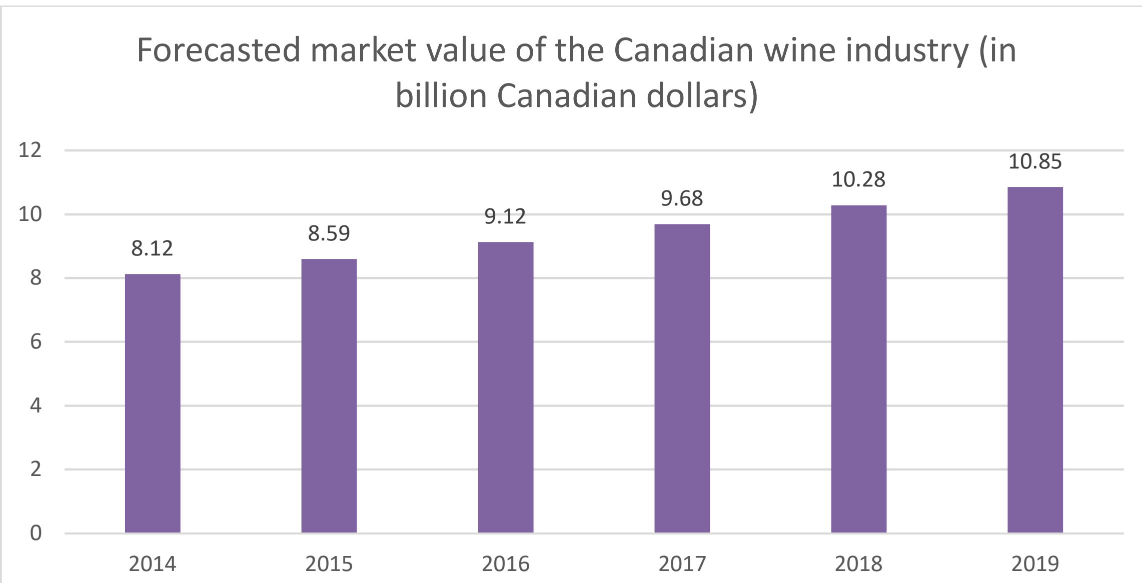

Figure 1 – Projected Market Value of Canadian Wine Industry (Courtesy of Statista)

Canadian winery global exports totaled $145 million dollars in 2017, with growth in the years leading up [9]. Wine exports are currently less than wine imports, but this could one day due to crop migration, and increased growth in the local wine industry.

As British Columbians, we have a vested interest in our provincial successes. Research of the industry from 2012 indicates that most wine-making establishments operate in British Columbia and Ontario. At the time, the industry employed over 3,700 people [10]. As the wine industry continues to grow, it is expected that employment will also increase. Automated solutions could be the answer to filling these new roles.

2.2.2 Labour

"In 1921, agriculture was the single most common occupation," and throughout the years "farms have become more specialized and the average farm size has increased [2]." However, with the projected market growth in agriculture, labour shortages have become of great concern. "In 2014, 26,400 jobs went unfilled in Canada's agriculture sector [11]." The sector lost $1.5 billion in revenue which continues to impact the sector today [11]. Future growth of the industry is stunted, and expansion plans have been delayed due to growing concerns that labour will not be available as readily. These shortages exist even with the Canadian government's Temporary Foreign Worker (TFW) program that brings workers from Mexico for periods of 8 months [12]. Many wineries take advantage of this program but are required to pay for round-trip travel and provide accommodation for the workers. These are costs that could be reduced with automated picking robots.

The labour shortage isn't a problem only in Canada. According to the California Farm Bureau Federation survey, "55 percent of responding farmers had experienced employee shortages [3]." That study also outlined that "[problems] have been more acute among farmers whose crops require the most intensive hand labour, such as tree fruits and grapes." When asked what actions they have taken in response to employee shortages, one- third of respondents said they used mechanization if available; another 29 percent attempted or investigated mechanization. This provides ample opportunity for automation to fill the need in other sectors of agriculture, not just wine.

2.2.3 Crop Migration

Crop migration refers to the drift in growing location of crops due to either an increase or decrease in suitability of growing in that region. The US has already seen northern migration of some crops such as soybeans and corn. This brings the question of whether agricultural regions could move North into Canada. According to Agriculture and Agri-Food Canada, most studies of the impact of climate change suggest that most regions of Canada are projected to warm during the next 60 years.

Despite the known challenges that come with climate change such as drought and storm intensity, there are opportunities for the Canadian agriculture industry. Canada is a high latitude country, which brings more pronounced warming than the global average. This could lead to expansion of the growing season to go along with milder and shorter winters like those seen in Arizona and California [4]. Many areas of the country that are typically thought of as frigid and inhospitable for growing could soon turn into usable agricultural land [13]. Climate change may also improve soil quality by enhancing carbon sequestration.

Without the available labour pool, this potential for climate-change induced growth in Canadian agriculture could be squandered. This is another opportunity for agricultural automation to thrive.

2.3 Key Literature Identified

This section summarizes key resources that make up the research supporting the idea of creating a grape-picking robot system. This research involved looking into areas like wine industry statistics, consultation with outside partners, current solutions, similar solutions, previous patents, safety standards, technological resources, and applicability of previous work we have completed in vineyard automation. The short answer is that a precision grape picking system does not exist yet. This section will show how our research supports our initiative to create one.

2.3.1 Climate Change and Wine Industry Challenges

Some of the inspiration for this project comes from the opportunities and challenges faced due to climate change and wine industry trends. Opportunities include increase in useable land and large exports of Canadian wine. One of the many challenges faced is labour shortage even with government programs.

Our report from Agriculture and Agri-food Canada [4] outlines the possible changes Canada could see in the future with regard to climate change induced increase in farmable land. This report describes the increase in farmable Canadian land that could accompany global warming due to shorter winter seasons. It also details the drought conditions and high storm intensity that we could face if warming increases even further.

Statistics Canada has a report [14] which provides insight into the Canadian wine industry, which includes information regarding exports and local sales. Similarly, an excerpt from a Statistics Canada report yields the sales of alcohol in different categories (Wine, Beer, Spritzers) [6]. Wine makes up 32% of the alcohol market in Canada and grew 3.2% from 2017.

According to a report from the Canadian Agricultural Human Resource Council, in 2014, 26,400 jobs went unfilled in Canada's agriculture sector [11]." The sector lost $1.5 billion in revenue which continues to impact the sector today [11]. Future growth of the industry is stunted, and expansion plans have been delayed due to growing concerns that labour will not be available as readily. These shortages exist even with the Canadian government's Temporary Foreign Worker (TFW) program that brings workers from Mexico for periods of 8 months [12]. Many wineries take advantage of this program but are required to pay for round-trip travel and provide accommodation for the workers. These are costs that could be reduced with automated picking robots.

2.3.2 Robot Safety

A relevant standard that didn't apply to our project due to application specifics is CAN/CSA-Z434-14 Industrial robots and robot systems. While we didn't have access to this standard, the Canadian adoption of ISO 10218, it covers safety requirements for automated robot systems in a variety of industries. We were able to deduce that this didn't include autonomous agriculture, which may be due to the industry being too new to have completely developed safety standards. Some of the main safety features outlined in the standard include barriers for safe operation such that human interaction with machinery be very limited, and in the case of interaction, safety measures be in place to prevent injury.

Since what we created is a mobile robotic platform, stationary robotic standards were not applicable. Also, since our robotics were slightly more comparable to commercial use than industrial use, different considerations were made with respect to safety and safe operation. It has been difficult to locate safety standards for our specific case; however, we did our best to extract relevant information from the CAN/CSA-Z434-14 industrial standard, in combination with our own set standards to ensure safe operation and use.

2.3.3 Grape Detection Work

Since most grape picking systems are lacking in "intelligence", a smarter method is sought after. Using computers, many have been able to come up with different approaches to identify grapes using image processing.

One method for detecting grape clusters, from a paper titled *"Yield Estimation in Vineyards by Visual Grape Detection"* [15], outlines the use of a complex algorithm for detection in addition to a sideways camera and a lighting source. The algorithm detects "potential [grape] locations with a radial symmetry transform" to start, then identifies "the potential locations that have similar appearance to grape berries", and then finally "[groups] neighboring [grapes] into clusters." The problem with this algorithm is that it is very un-intuitive and requires a semi-controlled lighting environment to work effectively.

Another method for grape detection comes from a paper titled *"Robust Grape Cluster Detection in a Vineyard by Combining the AdaBoost Framework and Multiple Color Components"* [16], which describes the use of a cheap camera, advanced image processing techniques, and machine learning for grape detection. The method uses a dataset along with an extraction of the "effective color components for grape clusters", and then "[constructs a] linear classification model". This is a good method since it's use of the AdaBoost machine learning framework makes it robust, however the use of color components and morphological filtering are not very clearly described.

The last method that will be discussed for grape detection, from a paper titled *"Automatic detection of bunches of grapes in natural environment from color images"* [17], is a method which claims to use a simple color thresholding methodology for detection. The method is impressively able to succeed in detection with "97% and 91% correct classifications for red and white grapes, respectively", and even is able to "[calculate] the location of the bunch stem [for a group of grapes.]" However, where this method falls short is that it is heavily dependent on color, is only able to detect one bunch of grapes per image and is to be used at night to "avoid any spurious reflections or bad illumination during sunny day time".

Of all the papers discussed above, a common problem arises. The complex image processing required for almost all of the methods above would be very difficult to run on an embedded platform. And if they were to run, real time detection would not be likely or would require advanced hardware.

Darknet/YOLO

A technology for state-of-the art object recognition and detection, dubbed YOLO (you only look once) by its creator, Joseph Redmon of the University of Washington, is a classification and localization method in which a neural network is applied to a single, full sized image, instead of prior methods which apply the network to an image at different scales and locations. To implement YOLO, Darknet [18], "an open source neural network [software] written in C and CUDA" is used. For our purposes, we used Darkflow, a wrapper that works on Google's TensorFlow open source software library. TensorFlow is an application used commonly for training neural networks.

The use of Darkflow/Darknet with YOLO over other technologies was advantageous because it was created specifically for real time object detection and has a large community. Furthermore, YOLO is constantly being developed to increase detection speeds and accuracy.

Google Open Images V4

To create accurate models in machine learning, a large dataset of images and annotations is required. Most research papers we've come across that have use machine learning to identify objects have used custom datasets, which requires time to manually annotate. These custom datasets may also be bias towards what is being tested (e.g. similar lighting conditions, environment, etc.). We decided that it would be the best use of our time and resources to use a pre-annotated dataset.

The Google Open Images database was a resource which we used to train our grape detection model on, due to its size and variation in images. The dataset can be downloaded, which includes a plethora of classifiers, however we were only interested in the grape classifier. This dataset was beneficial to the robustness of our work because the successful use of it to train a model that detects our simulated vineyard grapes was an indication of high probability that the model will also detect grapes in other environments.

In terms of the dataset annotation, formatting of bounding boxes is of importance. Since the program in which we planned to use the dataset required a certain annotation format, we needed to modify the set. The Google Open Images dataset annotation format is shown below in Figure 2, while the YOLO annotation format is shown in Figure 3.

<Classifier [String]> <Xmin [Abs Float]> <Xmax [Abs Float]> <Ymin [Abs Float]> <Ymax [Abs Float]>

Figure 2 – Google Open Images Annotation Format

<Classifier [Integer]> <Xcenter [Rel Float]> <Ycenter [Rel Float]> <Width [Rel Float]> <Height [Rel Float]>

Figure 3 - YOLO Annotation Format

2.3.4 Navigation and Mobility

Any robotic system intended for use in a vineyard must be able to traverse the property and overcome challenges like ruts, holes, ditches, loose branches, and mud. Many of the common driving platforms used in vineyard automation research are well outside the budget of a capstone project. Thankfully, lessons can be learned from these platforms and implemented at a smaller scale that better fits the Capstone restraints.

Once a driving platform has been chosen, it still needs to be guided through the vineyard by some sort of navigation system. This section includes a description of some previous research completed for BCITs School of Energy that is applicable to the topic of navigation.

Husky UGV

For projects similar to ours, others have used ready-made autonomous vehicles such as the Husky UGV, from Clearpath Robotics [19]. It is a versatile platform that is built with ruggedness in mind and has been used for agricultural applications previously. In fact, researchers at University of California used the Husky UGV for "a robotic system to help vintners manage their water systems, and implement precision irrigation across the vineyard" [20]. The downside to this platform is that it is very expensive, especially with our budget in consideration. However, if our budget was not a concern, and we were going to take our project to market, then this would be a platform to build around. One defining characteristic of this vehicle is that it has a separate electric motor for each wheel. This gives the vehicle four-wheel-drive capability and the ability to make tight turns in confined spaces.

Figure 4 – Husky UGV agriculture research platform

Autonomous Robot Navigation Algorithms

A repository of robotics algorithms for autonomous navigation posted by Atsushi Sakai, an autonomous navigation system engineer from Japan. He was nice enough to include documentation, which clearly communicates the basic idea of each algorithm and provides visualizations to demonstrate their use [21]. The repository outlines methods and test code regarding a plethora of methods for autonomous navigation; Some of which we are interested in using for our project.

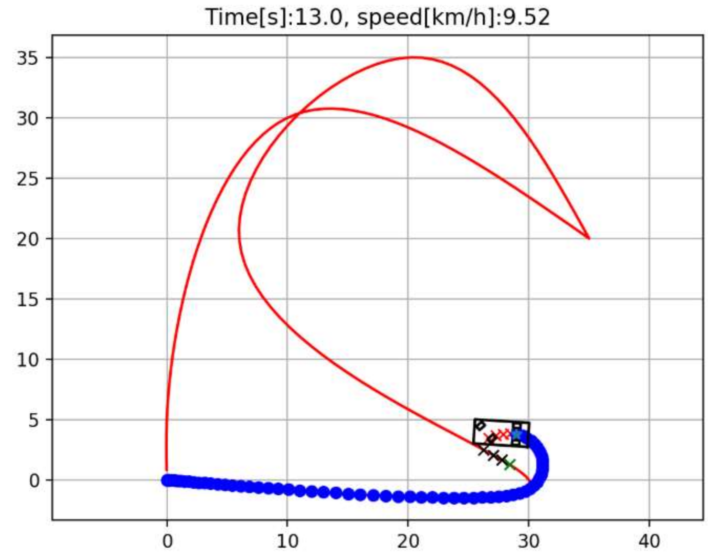

Figure 5 – Model Predictive Speed and Steering Control from the Atsushi Sakai Repository

2.4 State of the Art

This section of the report outlines the current state of the art for grape picking of both table and wine varieties. It has been spread up into current grape picking systems, current precision fruit picking systems, and fruit picking patents.

2.4.1 Current Grape Picking Systems

For some vineyards, hand picking of grapes is still preferred over mechanized systems due to the higher picking quality. Some larger vineyards have been experiencing labour shortages and have moved to mechanized systems. One example is a Washington vineyard's [22] use of the Optimum system (Figure 6). The machine is built by the Pellenc Group, a company that has been manufacturing agricultural tools since 1973.

Figure 6 – Pellenc Group Optimum Harvester (400,000 USD)

This $400,000 USD harvesting system indiscriminately scrapes a series of arms across the plants and knocks the berries into a sorting system that also filters out stems and other materials other than grapes (MOG). As seen in the above picture, there is plenty of potential for damage to the grapes and they must be used quickly. The yield is high, but the quality is low.

2.4.2 Current Precision Fruit Picking Systems

Some of the common precision picked fruits include bell peppers, strawberries, and apples. A commonality can be seen here in that these are all general brightly colored fruits that are distinct from the canopy and leaves around them making them easier to detect than green grapes on a green canopy.

SWEEPER

One solution that stood out in our research, is the Sweeper robot (Figure 7). This is a multi-institute project funded by the EU Horizon 2020 research and innovation program [23].
The project is the result of 4 million EUR being allocated to various universities and businesses for development of a picking solution tailored for use specifically in greenhouses. It uses depth cameras and detection models trained specifically for the pepper plants to identify ripe peppers. The robot is confined to tracks that run between rows of plants.

The Horizon 2020 project website contains a repository of all the papers submitted by each agency, and our selected paper outlines development and testing conditions used for their proof of concept. This paper gave us the idea of training a machine learning model to detect the grape clusters.

Figure 7 – EU Funded SWEEPER Bell Pepper Picking Robot (4,000,000 EUR)

AGROBOT

Strawberry harvesting can currently be completed by the AGROBOT, a $100,000 USD robot that requires uniform row spacing to conduct picking operations (Figure 8). The vehicle may be equipped with up to 24 small robot arms that pick individual berries.

Figure 8 – AGROBOT Strawberry Picker

Abundant Robotics

Google Ventures has taken the lead in investing in Silicon Valley based start-up, Abundant Robotics, to the tune of $10,000,000 USD [24]. This budding company is demonstrating their suction-based apple picking system. The system (Figure 9), which is essentially a horizontal mounted delta robot, can pick apples from the trees with minimal bruising or other damage.

Figure 9 – Abundant Robotics Apple Harvesting System

2.4.3 Fruit Picking Patents

A 2005 patent awarded to Vision Robotics outlines a Robot Mechanical Picker System and Method [25]. The patent describes an autonomous robot that makes its way through crops, maps fruit locations, and plans a picking strategy for its harvesting counterpart. The patent outlines the fact that harvesting with the same vehicle that does the mapping may be too computationally intensive. Also, machine learning was not as accessible for object training and recognition as it is today due to the lack of knowledge, and hardware acceleration. 14 years since the patent has been filed, technology has developed to the point where it is now feasible for this to be done.

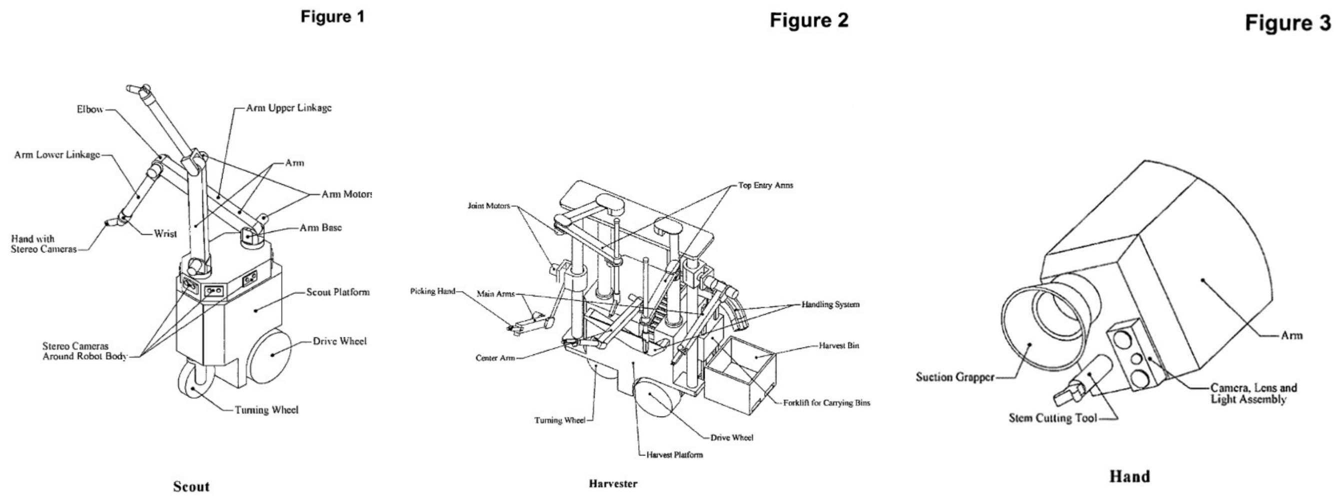

Figure 10 – Figures from Vision Robotics 2005 Patent

One of the technologies described in this patent involves the use of a stereoscopic camera. Most of the stereoscopic 3D cameras on the market right now do not contain enough precision to locate the stem of the grape cluster accurately enough to grasp it.

2.5 SOE Previous Work

Another vineyard management project was proposed originally from the School of Energy (SoE). A main objective of the SoE project was to streamline data collection from the field, including the use of remote imaging. This imaging could be used to estimate crop yields and assess plant health without human intervention to increase the operational efficiency of a vineyard.

2.5.1 Actobotics Nomad

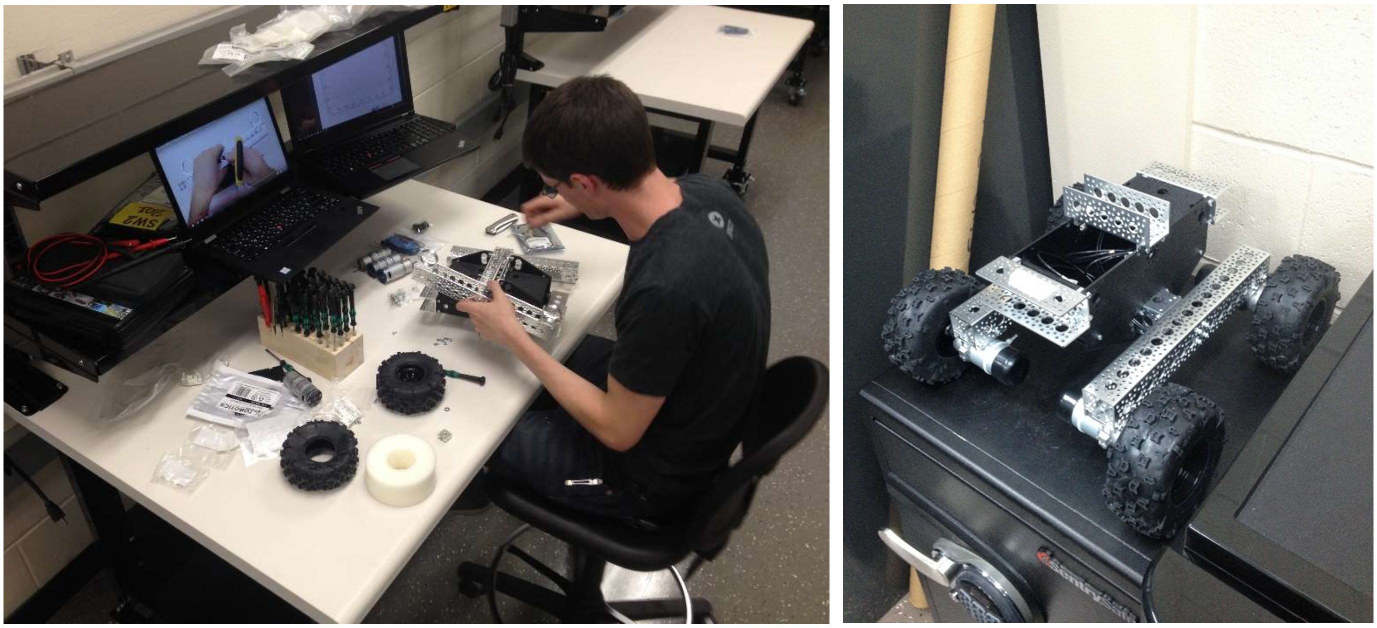

Figure 11 – Nomad Kit-Based Robotics Platform

Many of the mobility requirements of a grape picking solution are actually solved by the platform chosen in the SoE project. Due to the software-rooted nature of the original project, a kit-built vehicle platform was utilized to minimize time spent on mechanical design. The Actobotics Nomad platform (Figure 11) was chosen for its cost effective yet rugged nature. The vehicle dimensions are 15" long, 9" high, with a wheelbase of 13".

Figure 12 – SoE VineBot Vehicle Prototype

The vehicle structure is constructed from a rigid aluminum channels, planetary gear hand drill motors, a chassis with cut-outs for various accessories, and an internal cavity for circuitry. The vehicle is equipped with rubber tires intended for outdoor all-terrain use. Just like the Husky UGV described earlier, this vehicle has a dedicated motor attached to each wheel.

2.5.2 LiDAR

Navigation of the vineyard involves driving down rows of grape plants and knowing where trellis posts and other objects are so that vehicle can avoid obstacles and mitigate the risk of collision. To aid in navigation, a rotary LiDAR sensor was used along with an inertial measurement unit (IMU).

A two-dimensional rotary Light Detection and Ranging (LiDAR) unit is used to survey the immediate area around the vehicle for obstacles. This produces a point-cloud map (Figure 13) with the origin defined as the location of the sensor. The point-cloud by itself is simply a series of data-points in a cartesian coordinate system.

To retrieve meaningful information from this data, associative clustering algorithms are applied to divide the dataset into groups that follow the natural structure of the data [26]. Through this process, the data-points for each observed physical object such as surrounding structures, vine trunks, and trellis posts, are divided into individual clusters. By applying a series of statistical analysis algorithms to each cluster, clusters can be classified as erroneous data-points, surrounding structures, and vineyard structures. Objects other than the grape vines and related trellis posts can be removed from the data-set to simplify further processing [27].

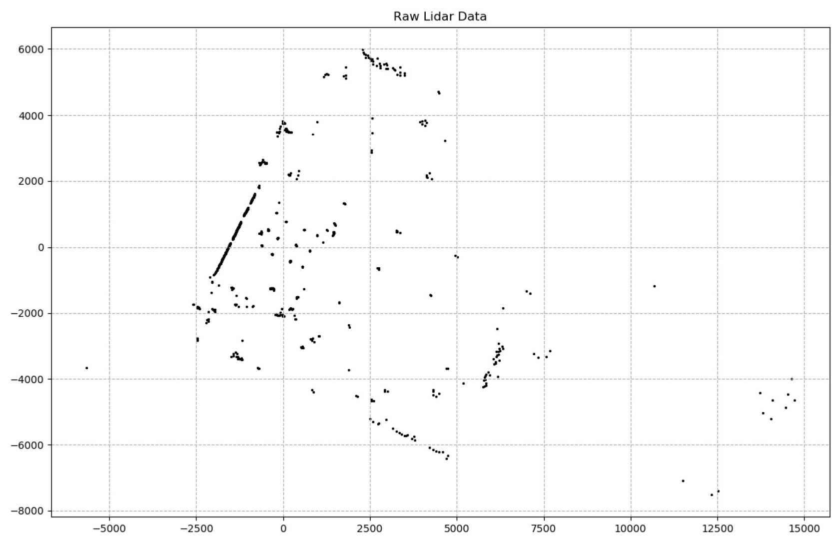

Figure 13 – Raw Lidar Data

A reasonable estimate of the location and orientation of the vine rows around the vehicle can be produced through further statistical analysis of the clusters. By analyzing the distances and angles between each cluster representing a grape vine or trellis post and applying a series of probability distribution functions (Figure 14) to these metrics, an educated guess can be made as to the position of the vine rows. In addition, a probability of accuracy can be associated with each resulting vine row, as seen in Figure 15.

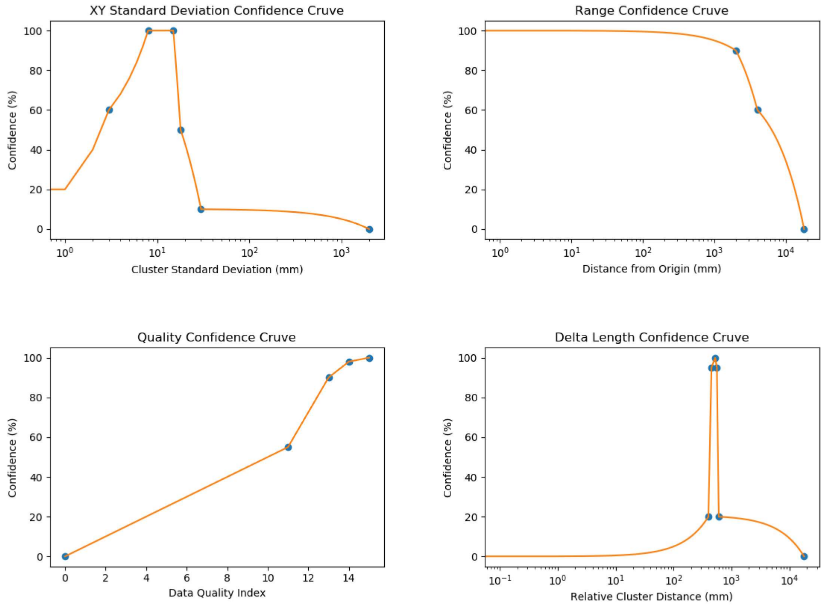

Figure 14 – Confidence Curves for Statistical Analysis

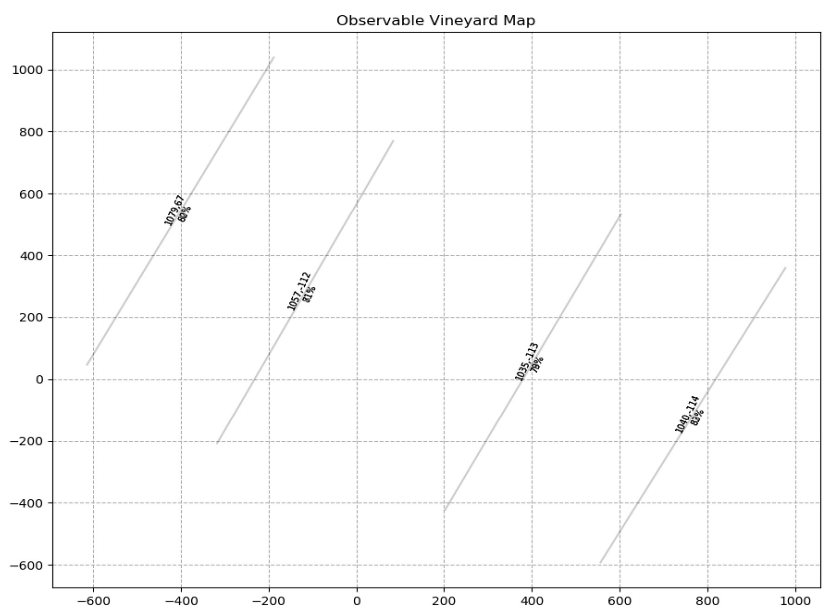

Figure 15 – Computed Vineyard Rows

From the data shown in Figure 13, it appears that no meaningful patterns can be extracted. To give meaning to data points, clustering techniques found from a third-party source were used with modification for our purposes [26]. In terms of statistical quantification, confidence intervals show in Figure 14 provide insight into how linear rows were filtered from the data set [26].

2.5.3 Inertial Measurement Unit

The SoE vehicle contains an Inertial Measurement Unit (IMU), comprised of a gyroscope, accelerometer, and magnetometer. In order to leave the navigation system with the possibility of GPS use in the future, we have developed an IMU server on the device that can be accessed by any software being developed. This server allows access to the vehicle's compass bearing after adjustment for local variation (declination), as well as access to the acceleration forces being felt by the vehicle with respect to North as seen on a GPS. The server also provides information as to how far the vehicle is tilted in any direction and could be used to avoid situations where the vehicle could overturn.

3 Scope of Work

To better understand how we plan on providing value to vineyard owners, the scope of our work needs to be defined. Some objectives and goals associated with this project will be introduced, including a more general strategic plan which will help outline steer our group in the right direction, and an operational plan which will cover the implementation of our goals. The plan can be broken up into key design activities.

3.1 Recommended Long Term Objectives

Based on feedback and objectives outlined by the SoE, and through discussions with industry partnerships, some recommended objectives have been drawn up. These objectives include both general goals for the project, but also delve into more technical ambitions.

Listed below are some long-term objectives of the project, with more information provided below in the respective sections.

1. Support Canadian agriculture, and facilitate competition with other large agricultural regions
2. Help Facilitate in creating a robotics design elective for B. Eng. students at BCIT
3. Implement methods to ensure adequate vehicular operation in low-light conditions
4. Use or modification of mechanical end-effector for selectively picking grape clusters
5. Develop a minimum viable product (MVP)

3.1.1 General Goals

A basic goal of this project is to support Canadian agriculture. The Canadian agricultural industry can benefit greatly from the innovative works of engineering students, by ways of a creative-destruction culture. This also ties in another goal, which is to facilitate competition with California, Europe and Japan agriculturally, since British Columbia is an agricultural competitor of these regions.

Lastly, having this project help facilitate in creating a robotics design elective for B.Eng. students at BCIT is also an outcome we hope to achieve. It is hopeful that the technology we explore may be helpful in implementing the course.

3.1.2 Technical Objectives

Various vision and lighting methods with the help of image processing should make the vehicle adequate for operation in low-light conditions, making it a relevant technical objective. Another technical objective for the vehicle is to be able to make use of a mechanical end-effector for selectively picking entire grape clusters, without the need for human intervention.

3.1.3 Future Vision

For the VineBot project, we hope to expand on previous work to develop an MVP VineBot all the way up until the end of our project term in May. If we finish the MVP VineBot early, there are some optional objectives which we plan on accomplishing.

If time permits, and a product created by our team is deemed to be adequate, The DARPA Grand Challenge Winter may be a competition that our project could be entered in. Another optional objective would be to experiment with lighting. Since a large chunk of our project involves image recognition, further research into lighting for computer vision, and image processing techniques that can be used to help reduce the effects of bad lighting could also be explored. Due to the modular design of the project, we may also want to investigate selectively harvesting other types of produce which are hard to mechanically pick, including blueberries and cannabis. The modular design philosophy may require partial redesign of some components but would bring added value.

3.2 Capstone Project Objectives

Working in consultation with local wine producers (Okanagan Valley, Lower Mainland) is an objective of this project. Their engagement with our engineering team and feedback on the MVP would be very valuable.

In terms of more technical requirements, the main objective for our group is to design and implement a proof of concept autonomous vehicle capable of navigating a vineyard to pick grapes while also avoiding obstacles.

Another objective is to become proficient in use of machine learning for image processing, 3D cameras, GPU processing, LiDAR processing, and robot motion and path planning.

A final objective is to have a professional demonstration of our project ready for the year end project exposition, with the goal of showcasing our project to students, public, faculty, and prospective employers.

3.3 Project Execution and Deliverables

At the start of the project, we would've liked to have the following deliverables by the end of the project:

1. A proof of concept autonomous grape picking vehicle
2. A final report documenting our project by the end of the project term
3. An organized git repository with extended project documentation
4. A live demonstration of the project for the technology expo including signage and literature

However, by the end of the project, we realized how easy it was to stray away from the project plan in terms of execution. We were able to complete the following deliverables:

1. A proof of concept autonomous grape picking vehicle capable of picking artificial grapes off an artificial grape vine
2. A final report documenting our project including changes from the initial proposal, and methodologies which did and didn't work out
3. An organized set of git repositories with instructions on operation of the modules they contain
4. A live demonstration of the project for the technology expo, and a recorded demonstration to be used by our mentor Craig Hennessey for his new robotics course
5. A working system that can repurposed for a B. Eng. robotics design elective at BCIT

We were able to create a proof of concept vehicular grape picking robot, as well as some artificial grape vines, and some artificial stems to go along. We needed to slightly alter the original work plan to complete these tasks, since creating artificial stems wasn't originally included in the project scope.

We also were able to document our methods throughout the project, making the task of writing this report easier. This is somewhat due to the work we put into milestone meeting reports, which were helpful for keeping us on track. We did however deviate from the original plan of having five milestone meetings, by only having three.

A pair of GitHub repositories were created at the start of the term and were used to backup any information we felt was important to keep, regarding the project. This includes raw data which we used and a large code base that contains all of the code for our project, including some unused code which can be used in the future for tasks such as GPS integration.

At the technology exposition, we were able to supply a captivating presentation on our accomplishments this term using a poster and an embedded video, with a live demonstration of grapes being picking off an artificial vine. After the expo, we filmed a demonstration of the project for future purposes and included it in the video we developed for the exposition.

Lastly, we were able to develop a system that could be easily repurposed, by our mentor Craig Hennessey, for use in an engineering course. Since he is running the course for the first time, we are able to provide him with most of the materials he would need to run the course and have done a lot of the heavy lifting for him, so to speak.

Its worthy of noting that the we needed to alter the fourth deliverable by demonstrating only partial operation (stationary grape picking) of the project at the technology expo. This is due to the fact that we were behind schedule, resulting in an untreated bug (a significant lag between picking and moving conditions). After the exposition, we were able to patch this bug and verify correct operation of our project.

3.4 Collaboration and External Input to Project

While consultation with industry contacts including BC VQA and local Okanagan vineyard owners are ongoing, there have been opportunities to pitch both the business case and the production feasibility of a grape picking robot to outside consultants including BCIT business students and multidisciplinary research staff with the Center for Applied Research and Innovation's MAKE+ team. We also were able to communicate with other capstone groups regarding technical issues throughout the term and were given advice for future growth at the technology exposition, held on May 10th, 2019.

3.4.1 BBA

We gave students from the Bachelor of Business Administration (BBA) program a short abstract of our project, as well as a short summary of our objectives related to this project. In return, they gave us a business analysis of our project. We found it useful to note that trying to become a price leader during commercialization would result in the start and increase of demand for VineBot. The BBA students pointed out how our lack of internal marketing and advertising specialists, could potentially lead to problems since advertising is an important aspect of any business.

The BBA students informed us of opportunities including the continuously growing industry and the relatively low competition regarding small-to-medium-sized customers. This confirmed our intuition and was useful as verification.

3.4.2 Make+

The consultation at the BCIT MAKE+ center allowed us to form some new relationships between visiting researchers, which can be used further in the project. General take-aways from the consultation included networking with a manufacturing expert who suggested we use vacuum molding to build an attractive outer shell for the VineBot before the capstone exposition. We also gained the contact info of an SFU mechatronics graduate student, Garrett Kryt, who offered to serve as a helpful resource for any of our complicated robotics questions. Overall, the research staff concluded that our project can be feasibly created based on the ideas we presented.

3.4.3 Capstone Groups

Although we did initially consult students from BCIT's BBA program during the proposal period, we didn't further pursue outside student consultation. We did however collaborate with other electrical engineering students at BCIT, including members of the Black Line project. They were helpful during the mechanical design phase, giving advice on 3D printing. They also allowed us use to use some of their materials for prototyping, including a small solenoid which we used for testing a solenoid cutting mechanism.

In return, members of our team were more than happy to help as mechanical design consultants, pitching solutions to some of the design problems they experienced.

3.4.4 BCIT Capstone Technology Exposition

During the technology expo, we received advice from attendees, including the possibility of commercializing our project, and applying the technology we've both created and compiled to other crops such as cannabis. Although our mentor Craig Hennessey has previously floated the idea, we found it reassuring to know that others were able to envision our technological success in automating the harvesting of another crop.

4 Project Implementation and Results

We introduce a high-level description of the project, including how it was split up into sub systems for simplicity. We then discuss the design and implementation of our project, which is comprised of a description of each element in the design and the actual design process. Problems we've encountered will also be included in the design and implementation section, in addition to problems caused by design choices.

4.1 High-Level Description

To create the autonomous system, we decided to split the task into three separate systems:

4. The harvesting system handles everything involving grape detection and picking, including the vision and robotics required
5. The navigation system handles everything involving mapping and position estimation, and also path planning to route from point A to point B around a vineyard
6. The mission planning system handles the vehicle control, as well as allowing and disallowing when the vehicle is able to enter harvesting and mobile states

Shown below in is a high-level diagram showing the interaction between the 3 segmented systems Figure 16.

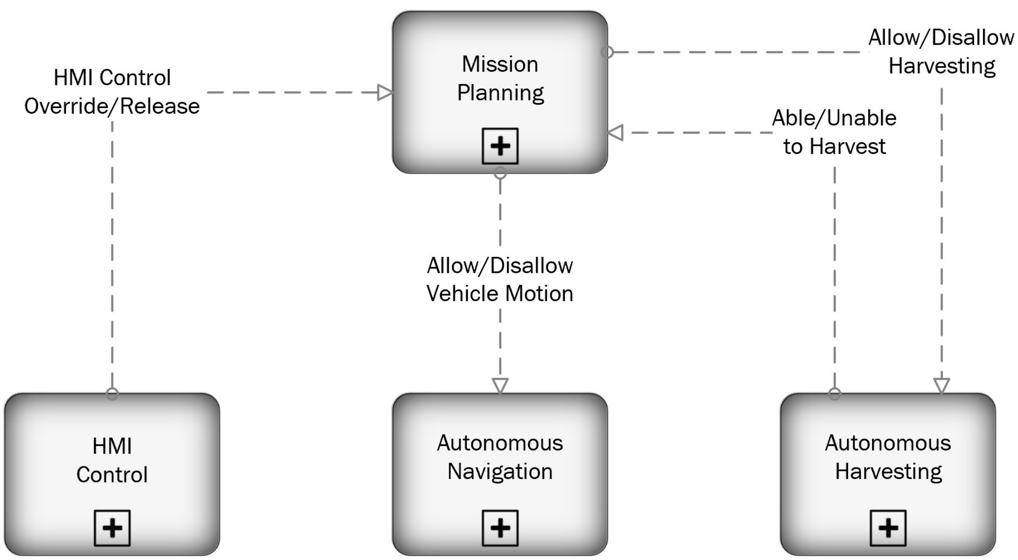

Figure 16 – Integrated Systems Block Diagram

Together, the systems are bridged with a mission planning system, which determines the operational order. A more in-depth diagram of the interaction between the systems is also provided below in Figure 17.

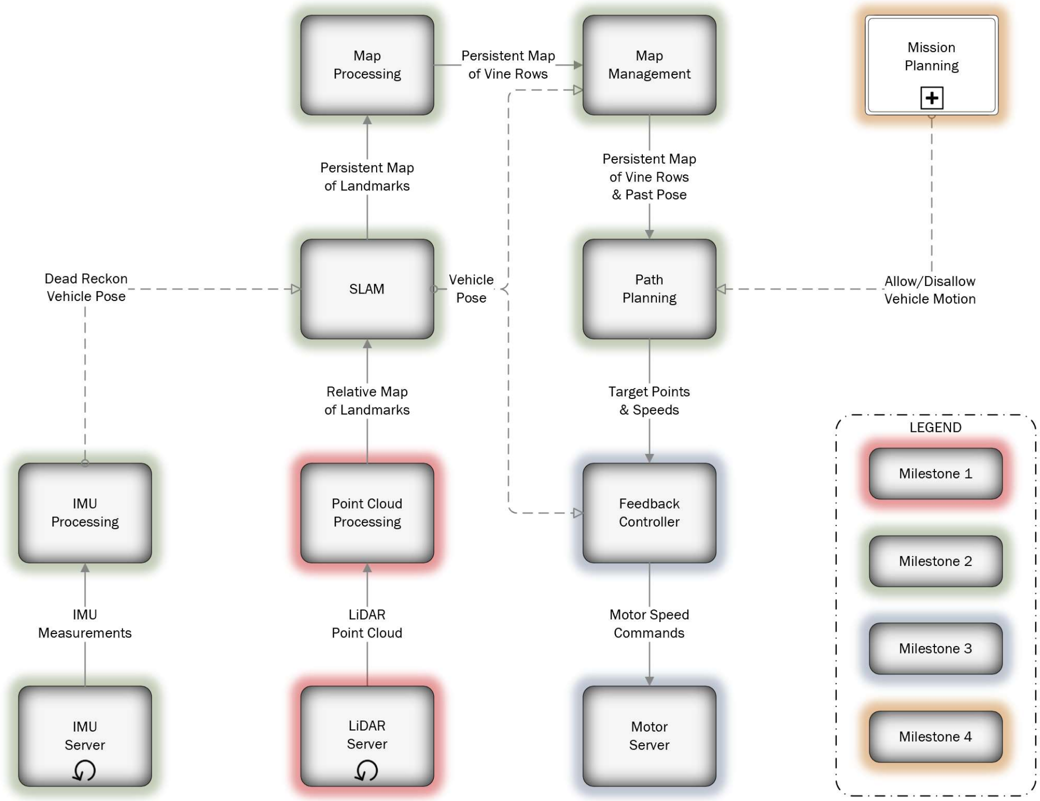

Figure 17 – Autonomous Navigation System

4.2 Design and Implementation

The design and implementation of our project was split into sections according to the project plan. Such sections include development into the vision system, utilizing the NVIDIA Jetson TX2, working on robotic control for the project, creating mechanical parts and assemblies for the project and establishing algorithms for position estimate and mapping for navigational purposes. This design and implementation section of the report will also cover estimated costs for each section of the project.

4.2.1 Vision

The primary goal of the vision systems was to detect grape clusters in 3D space and provide feedback for the robotic picking system. To implement the vision system, we used two cameras, the PMD Pico Flexx depth camera and Logitech C922x HD webcam as the primary sensors for detection. We also used the NVIDIA Jetson TX2 computer in the vision system to run our neural network models.

Pure OpenCV Based Vision Approach

Early work using a purely OpenCV based approach to grape cluster detection provided some mixed results. While the image detection algorithm based on the Hough Circle Detector in combination with density-based clustering of detected grapes provided good results at low frame rates and fixed camera positions, the system proved noisy and unreliable when subjected to motion.

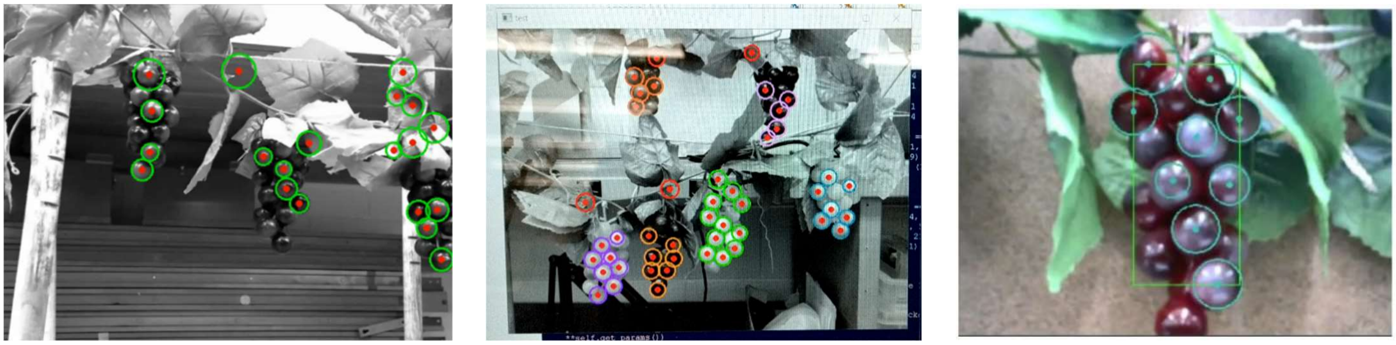

Figure 18 – Hough Circle Detector (left), Density-Based Clustering (middle), Final Algorithm (right)

Based on the early results, concurrent development into a more robust neural network-based system was initiated. As detailed in the next section, results were more favorable and work into the OpenCV system was discontinued.

Machine Learning

To detect grape clusters, we used machine learning object recognition. The specific machine learning framework we decided casts a single neural network on every image at their original sizes, instead of scaling and re-applying the network. This is beneficial because it allows faster detection of objects when compared to more traditional neural network frameworks. More information on machine learning can be found in the machine learning section of Appendix A: Background Information. Shown below are the results of the algorithm on some artificial grapes.

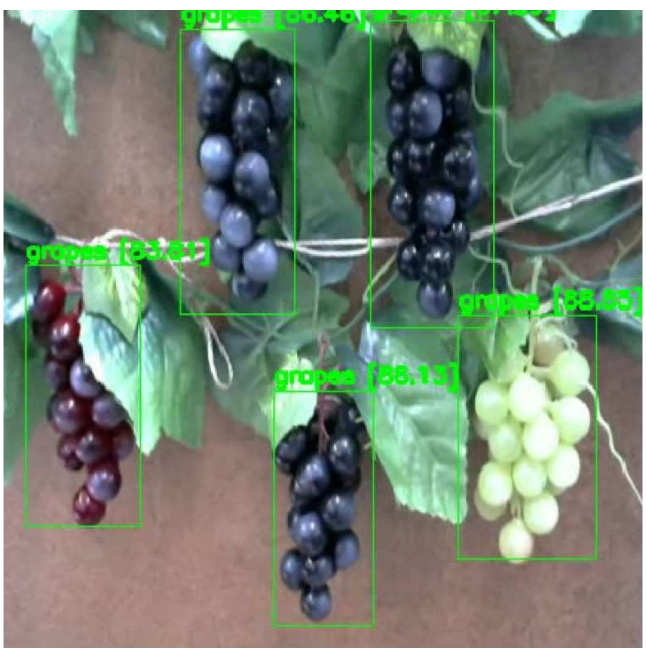

Figure 19 – YOLOV2 Based Detection Algorithm

Initially, we trained the system on the Google Open Images V4 grape dataset, which contained about 750 pre-annotated images. Starting with a pre-annotated dataset reduced development time significantly. Various utilities were found for hand annotating data but were not used due to the pre-existing dataset.

When testing out a machine learning model, we noticed that camera rotation, grape colour, and image exposure influenced probability of detections. In order to train a grape detection model independent of these variables traditionally, the diversity of the data set would need to be expanded to accommodate the encountered variation. Thankfully dataset augmentation when implemented through YOLO is trivial since there are configuration parameters built-in, which allow variation in angle, hue, saturation, and exposure of duplicated training images. For our training, the augmentation parameters and their associated values are shown below, in Table 1.

Table 1 – Dataset Augmentation Parameters

Parameter	Associated Value
Angle	15°
Hue	0.1
Saturation	1.5
Exposure	1.5

Shown below in Figure 20, is a curve which represents the both the mean average precision (mAP) and the loss function for the machine learning process.

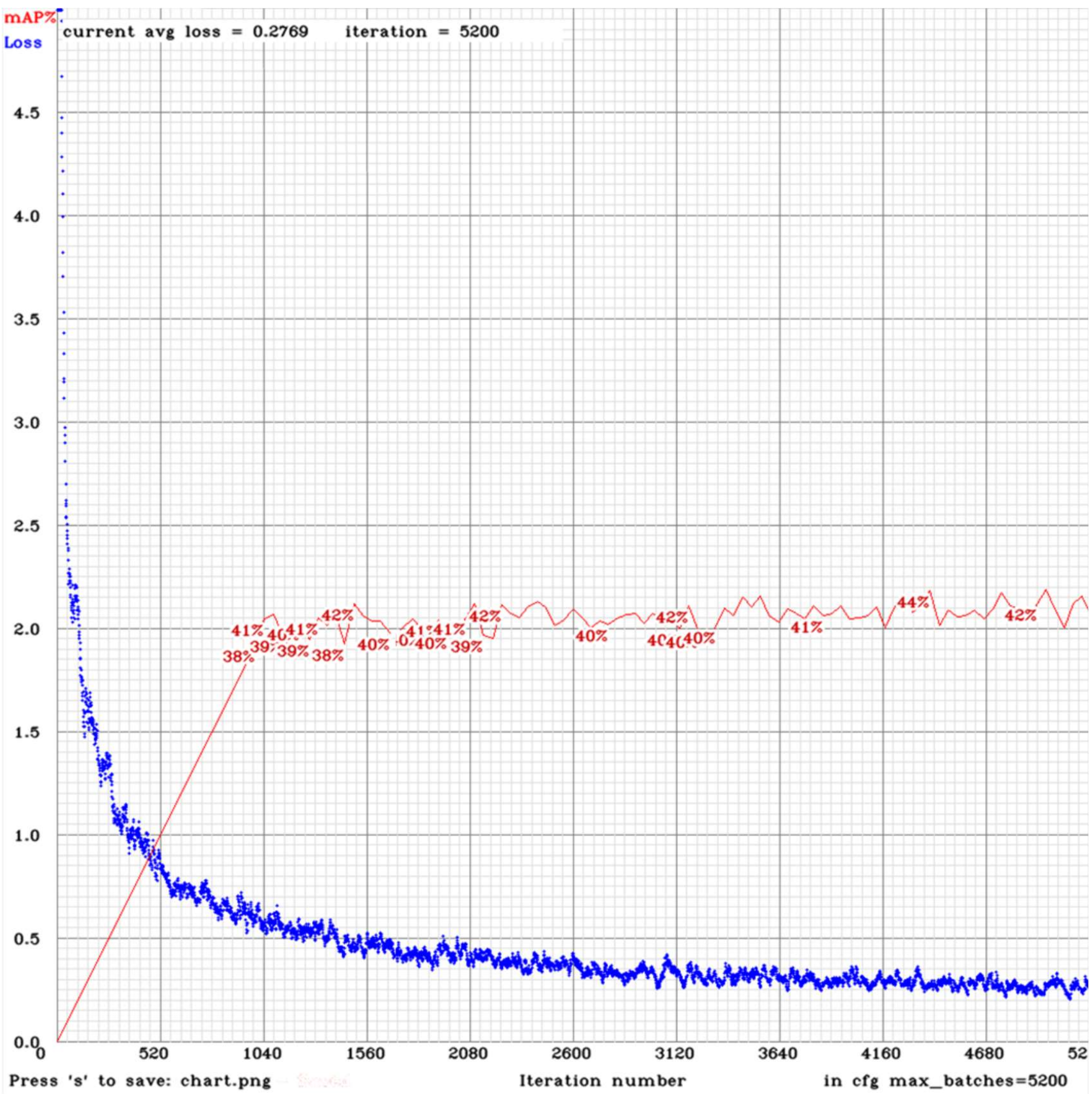

Figure 20 – mAP-Loss Training Curve

From training only on the Google Open Images dataset grape classifier, and training with dataset augmentation, we were able to reduce losses to less than 0.2769 and achieve a mAP of about 43%. It is recommended that for a reliable machine learning model, the mAP is greater than or equal to 50%, however from our field tests, we were able to determine that the initial model was sufficient for the initial development of our capstone project

Reinforced Machine Learning

To improve the machine learning results, we realized that we needed to increase our dataset size. The first option that came to mind was to pay for a grape dataset. This option was expensive, and did not guarantee new data, since we were using the Google Open Images V4 dataset, which was a compilation of images from Google. The second option, which was suggested by our project mentor, Craig Hennessey, was to take pictures of our grapes that we were planning to use, and then use a third-party annotation service. This option like the previous option would cost money, but the price would depend on the turnaround time and amount of data needing to be annotated. This option would also make the machine learning model slightly biased towards our artificial grapes. The last option which we decided to purse, was to use the current machine learning model to annotate data for us. This involved creating an algorithm which would only save data if all the detections in the frame were above a certain threshold. The method is shown below in Figure 21.

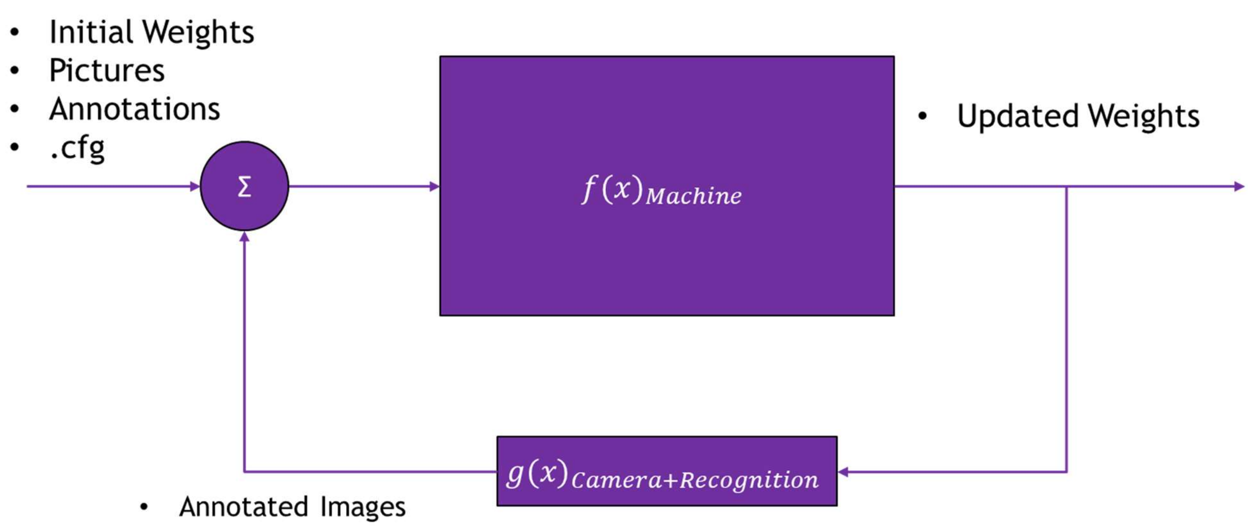

Figure 21 – Closed-loop Machine Learning

This method was good in the sense that it was cost effective and required a relatively minimal time investment. One potential pitfall for this method was that the annotated images fed-back into the machine are susceptible to false negatives.

Using the closed-loop method, we were able to double our dataset size from 750 training images to about 1500 training images. Training the model to 5200 iterations with the same parameters as the previous model, we were able to bring mAP from 43% to about 63%.

Tiny YOLO

Using the NVIDIA GeForce GTX-1080, we were able to run our machine learning detection models at 60 frames per second (FPS) on a desktop computer. However, we knew that at some point we were going to need to run the model on an embedded system, which would have a fraction of the processing power. We decided that it would be in our best interest to invest some time in experimenting with Tiny-YOLO, a version of the YOLO algorithm meant for higher inference speeds.

The revised algorithm operates identical to the normal YOLO algorithm, but differs in the fact that there are only 9 convolutional layers as opposed to 24, which increases the detection speed while decreasing the detection accuracy. In particular, we noticed that detections would disappear and reappear in subsequent frames, and a prevalence of false positives. However, it was also noted that there was a dramatic increase in frame rate (14 FPS) from the switch between YOLO (7 FPS) and Tiny YOLO (20 FPS), when running on the Jetson TX2.

To solve the detection noise problem, we then tried to increase the resolution of our detection model from 416x416 to 608x608. During the training process, this seemed to boost the mAP by 20% compared to the previous model, however after testing, we realized that there was increased noise. Also, increasing the resolution to 608x608 made the processing time increase, resulting in the frame rate dropping to around 12 FPS. Ultimately, we decided that the increase in frame rate wasn't as beneficial to us as detection accuracy, and therefore sidelined the Tiny-YOLO model. A possible yet unexplored solution to increasing the frame rate while keeping the detection accuracy could be to convert to the current YOLO model to a TensorRT model, which is an optimized neural network format for the NVIDIA Jetson TX2.

False Positive Keyboard Detection Problem

In the previous model, we were able to accurately determine what was and wasn't a grape, however one object which was constantly mis-classified was a standard computer keyboard. When tilted at a certain angle in addition to lighting reflected from the keys, the keys of the keyboard resemble what look like grapes in a cluster.

To solve this problem, we decided to take pictures of keyboards which were in the capstone lab. In total, about 100 pictures of keyboards were added to the machine learning training data. To use these images as training data, we needed to also include a text file for each image, which would be empty since an empty file is indicative of a negative image.

From tests, we were able to determine that the updated neural network had stopped misidentifying keyboards as grapes, as it had previously.

Depth Measurement

Building off the existing 2D vision system, the next order of business was to expand our 2D machine learning system to a 3D machine learning system. We planned on accomplishing this task by utilizing the PMD Pico Flexx depth camera.

Figure 22 – PMD Pico Flexx

Unlike a stereoscopic depth camera, which compares two images to infer distance, the Pico Flexx falls into the category of time-of-flight (ToF) depth cameras and operates by flashing an infrared light source and using the phase difference of the returning light to infer distance. Images taken with the camera are composed of pixels representing depth (in millimeters) instead of grayscale color intensities. Shown below in Figure 23 is the output of the Pico Flexx camera in colourmap form, where depth range is mapped to a hue value.

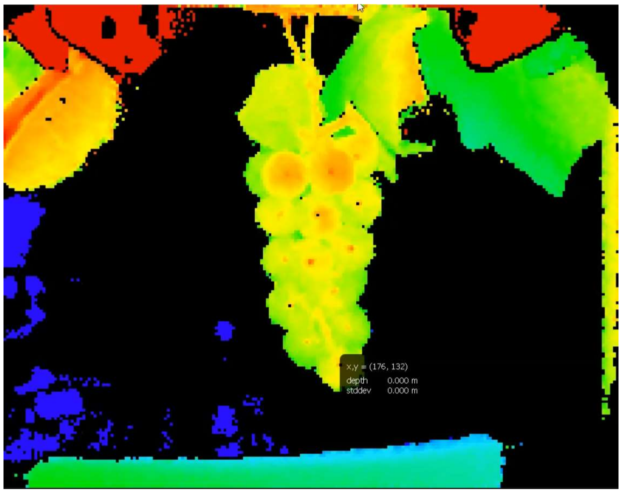

Figure 23 – Depth Measurement Heatmap

Our initial intention was to train a machine learning model on 3D point cloud data of grapes. However, due to the lack of 3D datasets readily available and our specific application, point cloud detection of a grape seemed like a difficult task. Instead what we ended up doing was using the existing 2D detection model, and then applying a linear coordinate transformation between the 2D camera and the depth camera to evaluate the distance to the center of the detected cluster. The caveat with this depth measurement method is the requirement to calibrate the differences in lens position between the two cameras. Calibration between the two cameras proved to be a simple, visually aided process involving a translation and scaling between the two aspect ratios of each camera. Shown below in Figure 24 are the detection models for both the 2D and depth images.

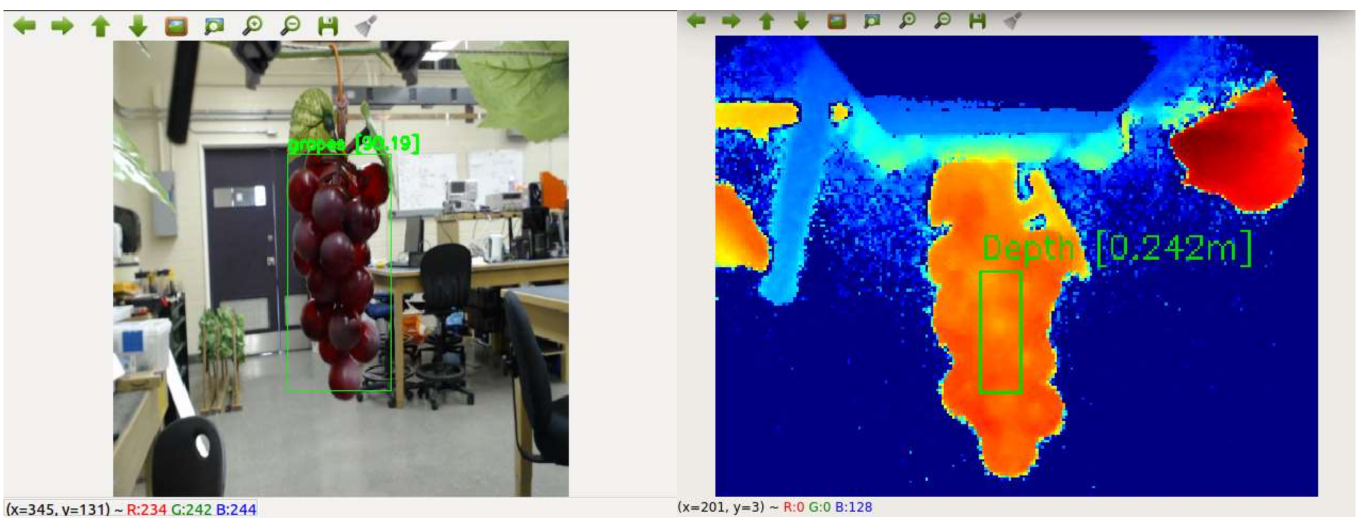

Figure 24 – Coordinate Transformation between Color and Depth Cameras

It's important to note that the bounding box for the depth image has been taken in by a factor of 0.25 in both directions to filter out the distance of the background filling the corners and sides of the detection bounding box.

One issue presented when trying to determine depth measurements for grape clusters was inconsistency of measurements due to the color of the artificial grapes being tested. This problem predominantly effected the dark purple grapes and was most likely due to the grapes absorbing the infrared light being emitted by the depth camera instead of reflecting it back for phase measurement. Since authentic grapes have a light hazy film on the exterior compared to their shiny black artificial counterparts, we can assume that they may not absorb light to the extent of the artificial grapes we have tested on. That being said, we decided to use red grapes for testing consistency. It is doubtful that this problem would arise when using a stereoscopic camera instead of an infrared ToF camera.

Another problem we ran into when trying to integrate the depth cameras was that there was a significant lag associated with the frames provided by the depth camera. We only noticed this issue when testing the vision system on the vehicle battery instead of plugging in to a wall outlet. After some consideration, we realized that this was due to the fact that the depth camera sample code was reading in frames to a FIFO buffer, and when the queue was not being read, it would build up a substantial backlog. This issue was easily fixed by removing the buffer and only reading the most recently received frame.

4.2.2 NVIDIA Jetson TX2

An NVIDIA Jetson TX2 was utilized to aid in the processing intensive tasks associated with machine vision and neural network object detection. The hardware acceleration provided by the Jetson's 256 CUDA cores help bring the machine learning detection models close to real-time, while consuming little power.

Setup

Setting up the Jetson was easier than expected. It required creating an Ubuntu machine (partition on a Windows PC) which was used to download the Jetson JetPack SDK, and then transfer it to the Jetson via SSH and USB. Other pieces of important software which were installed on the Jetson include the OpenCV, DarkFlow, and TensorFlow. The setup took days as opposed to the weeks allocated in the project plan.

Vision

After installing all the required packages, we decided to test our detection model. Some important findings included an inherent USB camera buffer lag, lack of camera parameter control in OpenCV, and higher than expected YOLO inference time. Overall, we were able to achieve around 3 FPS on the Jetson, which was below the 60 FPS performance seen on the desktop computer with the NVIDIA GTX-1080.

To solve the USB camera buffer lag, we updated the way we processed and captured frames. By multi-threading our capture and processing, we were able to grab the most recent frames from the camera undependably of the processing done to the frames (which was the slowest process in the loop).

OpenCV currently only allows up to 30 FPS capture rates using USB webcams. In order to ensure that we were always operating on the most recent information, a capture rate closer to 60 FPS would have been more ideal. A Python 3.x binding for Linux's Video For Linux (V4L) allowed us to set the webcam capture rate to 60 FPS.

Even with these improvements, inference time was still high. After browsing through Darknet GitHub issues, we decided to clone a GitHub user's (AlexeyAB) custom Darknet repository fork, which included an update to the neural network hidden layers which were previously causing problems. With all of these optimizations, we were able to bring the frame rate of the machine learning detection to about 7 FPS.

Grape Server

With the machine learning model able to detection grape clusters position in 2D, and the depth camera being able to provide a depth measurement, the combined vision system was in a position to provide 3D (technically 2.5D) grape detection. To aid in the integration of the robotics and vision systems, we designed a system which would be able to coordinate communication between the motion controller in the robotics system. The grape server communication system communicates over TCP sockets, and provides information regarding the center of the cluster in the frame, as well as the distance of the grape being tracked from the depth camera.

4.2.3 Robotic Control

We were able to complete an early implementation of robotic picking by integrating the vision system with a robot arm, the uArm Swift Pro. Data was sent over TCP sockets and used to move the robot accordingly. The current picking algorithm works as follows.

1. Detect grape clusters in frame
2. Center left most grape cluster in frame
3. Move in to pick
4. Re-center left most cluster if needed
5. At picking depth, engage grippers

For this initial test system, appropriate proportional gains in the x and y direction were determined experimentally by increasing the gain to the point of oscillation, and then backing it off until the system was stable. These gain parameters are heavily dependent on update rate of the feedback loop. If any changes are made to the sampling rate, these parameters will have to be revisited.

The proportional gain of the depth feedback loop was divided into two stages: far gain, and close gain. The far gain was used to move the gripper quickly to the proximity of the grape cluster, while the close gain was used to fine tune the position before picking.

Motion Controller

We initially tested the uArm Swift desktop robot arm with a simple camera feedback system. A camera was mounted to the arm and used to generate a feedback signal in a simple "eye-in-hand" visual servoing control technique. The setpoint for the two-dimensional tracking system, shown in Figure 25, is keeping the target in the center of the camera's field of view. Any motion of the target enters the system as a disturbance in the output.

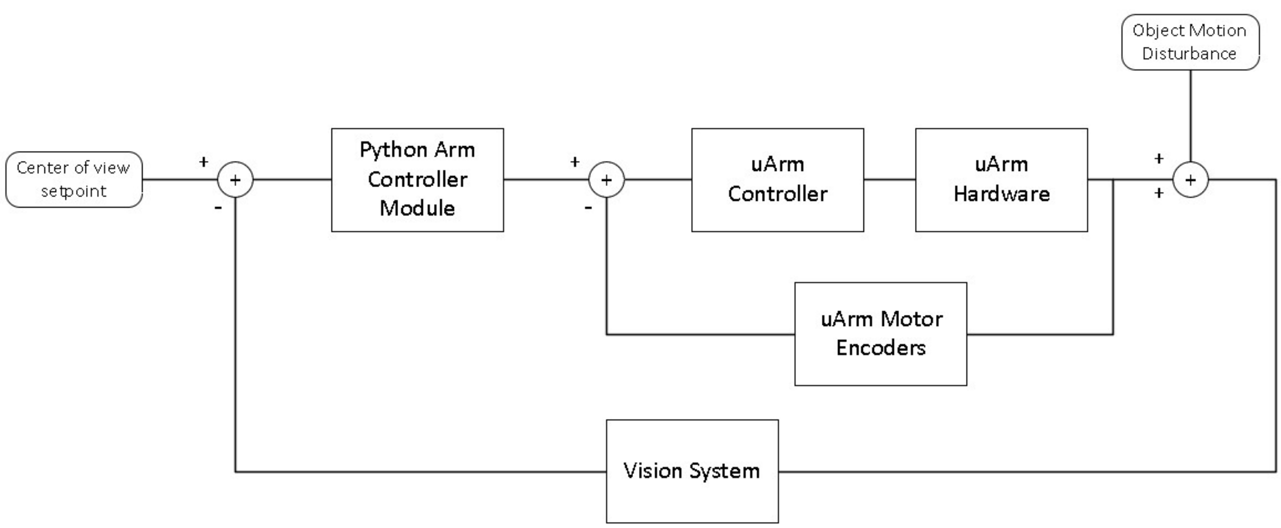

Figure 25 – 2D Camera Tracking Robot Feedback System

The vision system in the diagram can be replaced with any grape tracking algorithm desired. Initial testing utilized a simple circle detector (Figure 26) to generate a feedback signal from the position of a cardboard box[28].

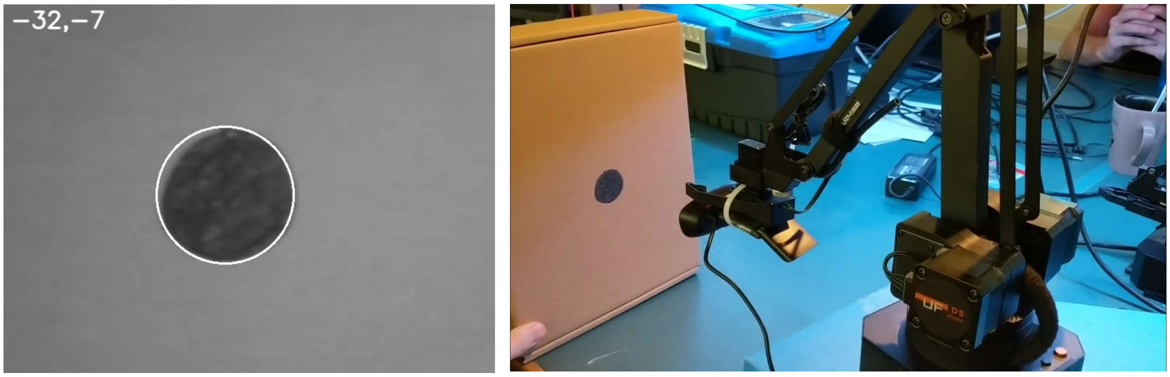

Figure 26 – Simple Object Tracking Vision System

For this initial test system, appropriate proportional gains in the x and y direction were determined experimentally by increasing the gain to the point of oscillation, and then backing it off until the system was stable. These gain parameters are heavily dependent on update rate of the feedback loop, and since we needed to make changes to incorporate the slower sampling rate of the detection system, these parameters needed to be revisited.

4.2.4 Mechanical Design

Camera Mounts

For the 3D grape detection to work well, we needed a way to keep the two camera sensors as lined up as close to each other as possible, while also having the cameras somehow mount to the robot end-effector. It was also a requirement during testing, that the camera mount be easily removable from the robot. Using these requirements, we were able to create the camera mount shown below in Figure 27.

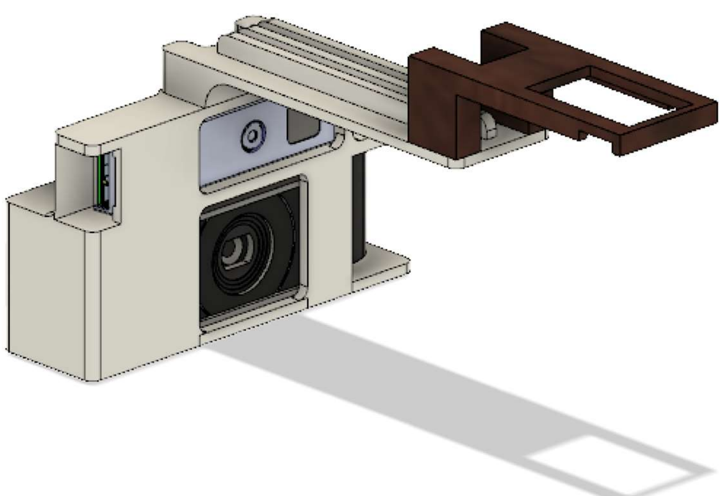

Figure 27 – Camera Mount

The camera mount is a 2-piece design, which houses both cameras, with the PMD Pico Flexx above the Logitech C922x. To allow the two pieces to successfully fit together without any adhesive or fasteners, we decided to compression-fit the two parts together using a jack-plug design.

Before landing on the most recent revision, changes were made to the camera mount assembly to incorporate proper camera fitting, suitable distance between cameras and the picking mechanism, strengthening of mechanically weak areas, and general aesthetics.

Gripper Extensions

Another set of parts we designed were extensions for the existing uFactory uArm Gripper. The extensions help loosen the constraints on the robot motion and accuracy of the depth camera.

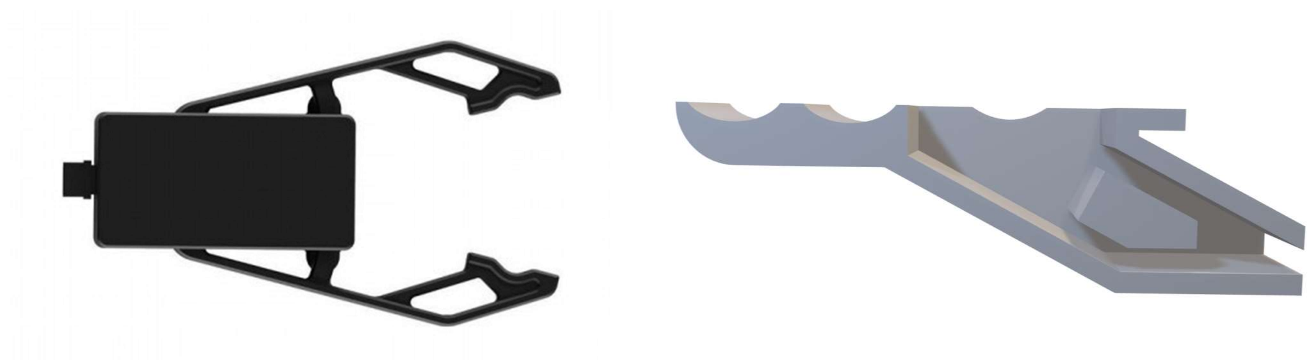

Figure 28 – uArm Gripper and Gripper Extension

The gripper extensions fit over the gripper and increase the capturing threshold to 2 centimeters. The gripper extensions are also very light and do not hinder performance of the robot in terms of static and dynamic loading.

Stems

Since we decided to try an make our setup as realistic as possible, we decided to create artificial stems which we could cut using a cutting mechanism of some sort. The stems would need to be easily cut using a razor blade and would need to hook onto the existing artificial grapes and grape vines.

We started off by printing stems that were two layers of PLA thick, thinking that they would be brittle enough for easy cutting. However, from experimentation we realised that the length of these stems also played an important part. An observation made early into the stem development was that the longer stems we printed ended up being less brittle and were less prone to being cut consistently. Thus, we decided to create shorter, 3 cm long stems that were two layers of PLA thick (0.44 mm) and carefully lifted off the 3D printer bed. The exact dimensions of the PLA stem are provided below in Table 2.

The stems themselves also needed to be able to hook onto an existing artificial grape bunch's loop and to the artificial vine we were using. To solve this problem, we decided to add a diamond/arrow shape solid to the bottom of the stem, which could be easily slipped in a given grape bunch's loop, and a T-shaped extrusion at the top of the stem, which could be easily attached to a hook on the artificial vine. The stem is shown below in Figure 29, and the stem hook, which we've dubbed "Vine-Grip" is shown in Figure 30.

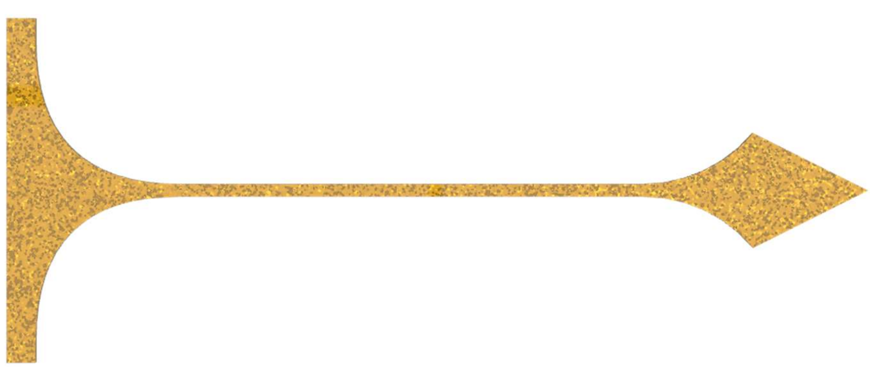

Figure 29 – PLA Stem

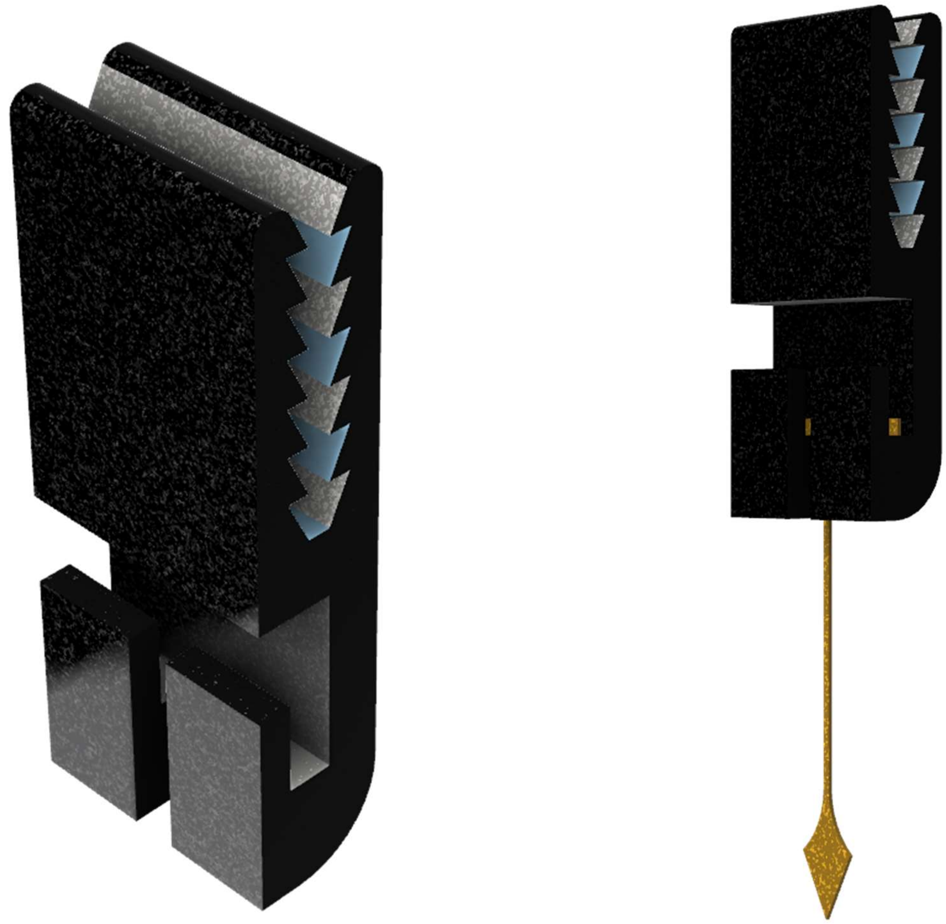

Figure 30 – Vine-Grip Stem Hook, and Vine-Grip with Stem

Table 2 – Stem Dimensions

Section	Dimensions
Overall Length	30 mm
Width of T-Bar	12 mm
Stem Width	0.44 mm
Max Barb Width	4 mm

Stem Cutter

A stretch-goal of this project was to be able to come up with a semi-realistic way to harvest grape bunches using our robot, which could be scaled up for testing on real crop. Since the end-effector for the robot had an existing gripping system, we decided to possibly utilize the force provided by the grippers to cut stems. The cutting system would've needed to incorporate a razor blade, since we had about a dozen sharp razors readily accessible for use and are cheap to purchase. Furthermore, we were able to find CAD files related to the razor blades which we decided to use, that could be used to save time when modelling.

To properly model the motion of the grippers, we needed to determine the operating radius (center of rotation) of our grippers. This was done by measuring the distances of the ends of the grippers in the open-position and drawing two concentric circles with one being a gripper length larger in radius than the other. We then added chords to the circles which represented the distances we initially measured and allowed our CAD program to solve for the correct radius of each circle based on the constraints. This is visually depicted below in Figure 31.

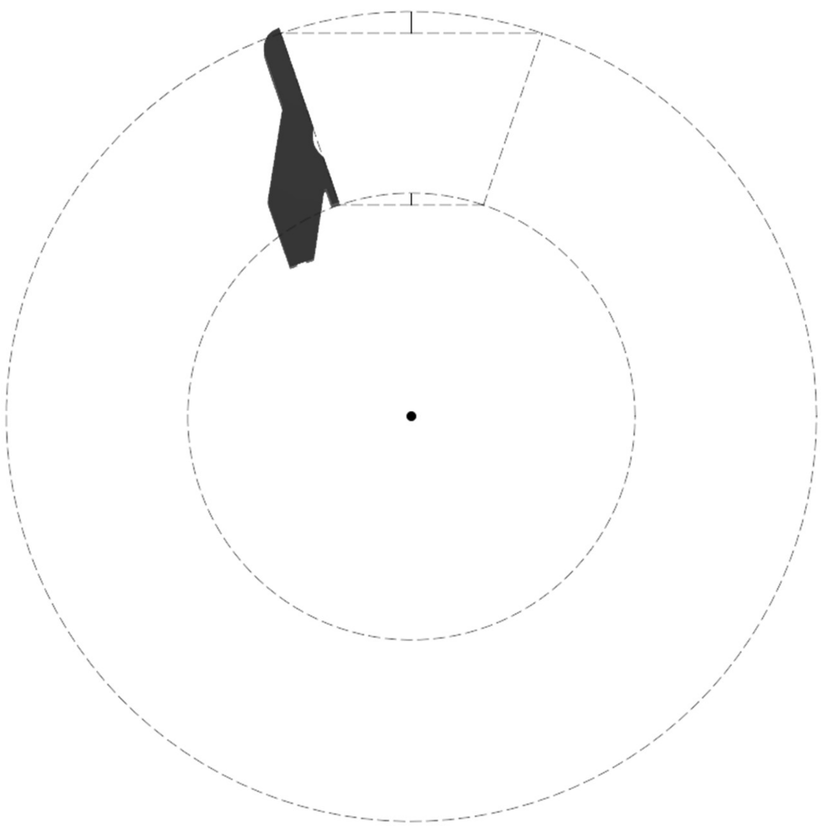

Figure 31 – Gripper Motion Sketch

When looking into the grip-cutting method, we realised that the force provided by the grippers wasn't enough to accurately cut artificial stems every time. The uArm gripper documentation advertises that the grippers can provide up to 800g of force [29], however we came to the conclusion that the force provided was much less. This was due to a limit switch built into the gripper circuitry and would stop the grippers from moving past a force (current) threshold. Shown below is a design for a grip-cutting mechanism, which would cut but moving a razor blade into an anvil. This design required adding, and possibly gluing down extra pieces on to the gripper extension pieces.

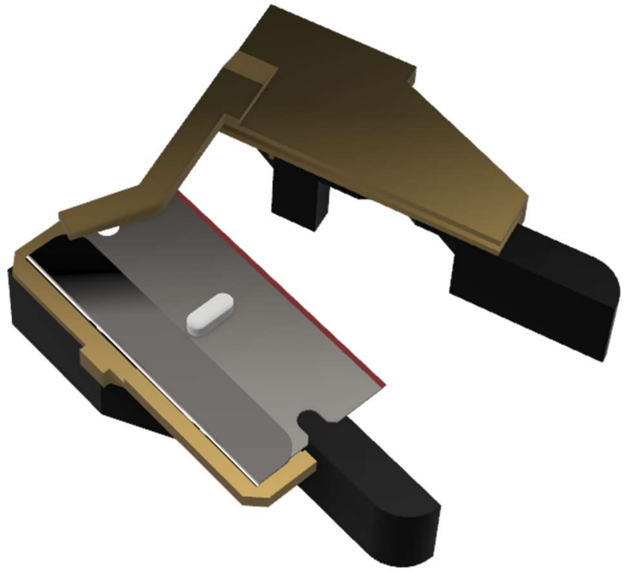

Figure 32 – Gripper Stem-Cutting Mechanism

We also looked into using a solenoid-powered cutting system, that would use the force provided by a linear actuator or solenoid to drive a razor blade into an anvil. This design however was deemed infeasible due to the challenges associated with solenoids. For example, the forces provided by low voltage solenoids are considerably low compared to a servo for instance. Also, the stronger solenoids tend to be heavier, which is impractical due to our robot being extremely prone loading failure (The stepper motor's holding force is overcome by loading force).

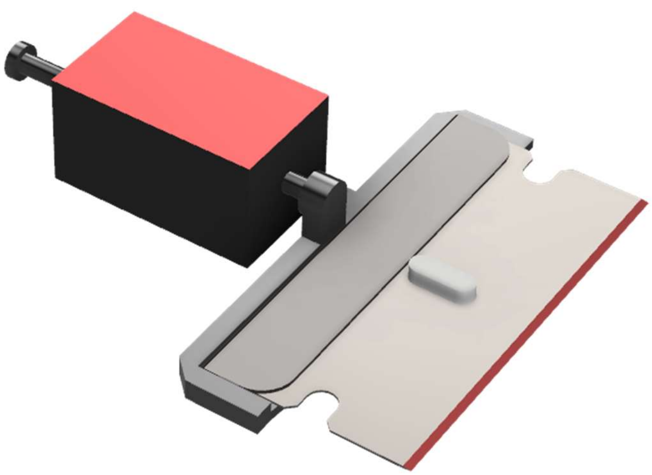

Figure 33 – Solenoid Stem-Cutting Mechanism

The chosen design involved repurposing the wrist joint servo on the robot for cutting purposes and fitting it with a blade. This meant that the robot would lose 1 degree of freedom, however the robotic algorithm didn't utilize the mobility of this joint, so the losses were minimal. From online documentation, we were able to determine that the wrist joint servo was able to provide 2.4 kg·cm of torque, which was determined to be adequate for our purposes. The other benefit to using the servo was that there were existing connections for the servo on the robot, and control of the servo was trivial since it was technically built into the robot motion control code. After testing and tweaking the design, we were able to design the cutting system as shown below in figure, which requires the fabrication of four parts. Three of the parts are fastened to each other, while the blade is secured to the assembly by a compression fit of the third part.

Figure 34 – Servo Stem-Cutting Mechanism

With a more powerful cutting system, we would be in a position to consistently cut stems regardless of the stem size. Another possible design choice for a cutting tool would be a rotating blade, such as a saw. A small saw blade attached to the end-effector of the robot would be able to spin up and cut stems, or even a pair of scissor-like blades. However, due to the loading constraints on our robot and power constraints of the end-effector, we decided to utilize the most minimalistic design which would incorporate a single razor blade.

4.2.5 Pose Estimation and Mapping

LiDAR Point Processing

The primary goal of the LiDAR sensor and subsequent point processing is to determine the location and orientation of the vine rows with respect to that of the vehicle. This information will be used to determine how the vehicle should move to bring the harvesting system within picking range of grapes.

Figure 35, below, shows the data from a single LiDAR scan consisting of around 700 radial distance measurements. This set of data points is referred to as a point cloud. At this resolution, the LiDAR sensor produces around six scans per second.

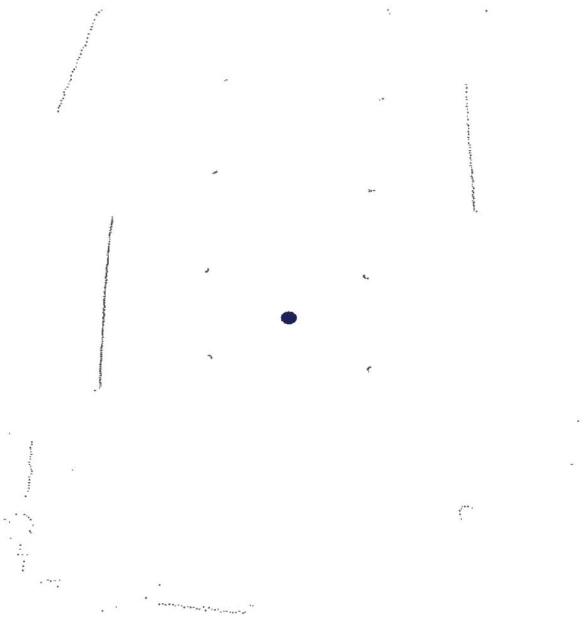

Figure 35 – LiDAR Point Cloud

By applying standard cluster analysis techniques to the point cloud, the data points for each distinct object are grouped together. By determining the physical size of each group, a probability factor of a group being a vine post is assigned. All groups with a probability factor lower than the threshold are ignored. Figure 36, below, shows the result of the cluster analysis with the groups to be ignored colored grey.

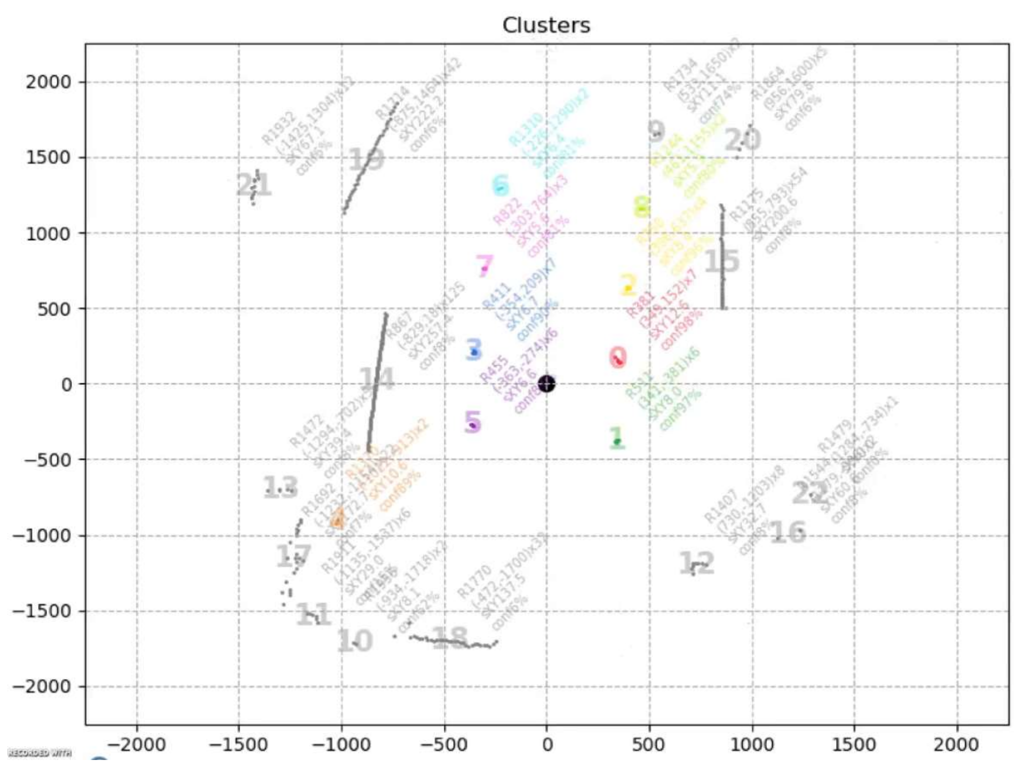

Figure 36 – Cluster Analysis of LiDAR Scan Data

The orientation of each vine row is found by determining the vectors between each group suspected to be a vine post, as seen in Figure 37, and looking for sets of vectors with similar length and orientation.

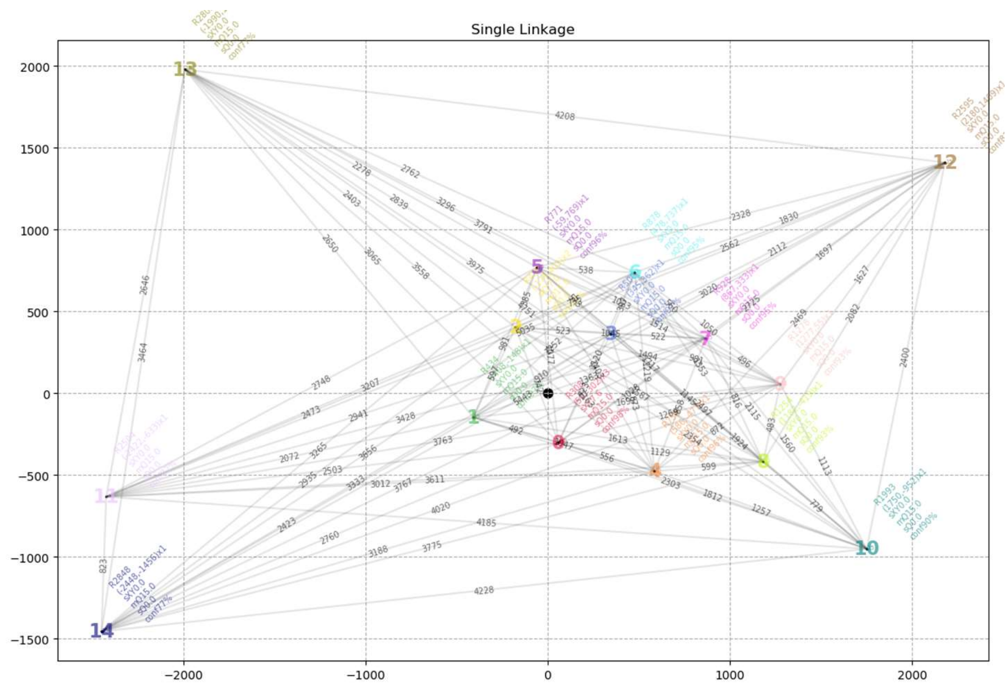

Figure 37 – Distance Between Clusters

The result of the above analysis is a set of vectors describing the location and orientation of the vine rows with respect to the vehicle, as seen in Figure 38.

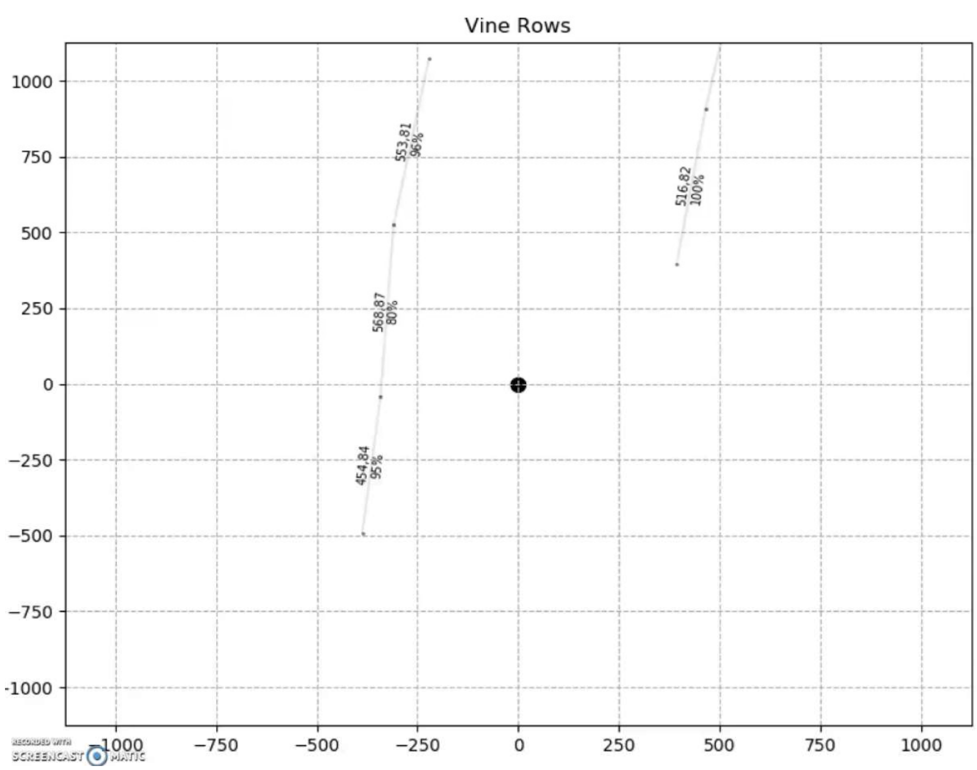

Figure 38 – Plot of Grape Vines

SLAM

The next step taken was to run this filtered data (without vine rows lines) through the Python EKF SLAM program, to see if the data provided was enough to build an accurate persistent map from the instantaneous scans using a SLAM algorithm. However, what ended up happening was the vine posts were being translated wrong in the map, causing the pose estimation to be incorrect. This is shown below in Figure 39, with the red path representing the estimated path of the robot and the green crosses representing possible landmarks.

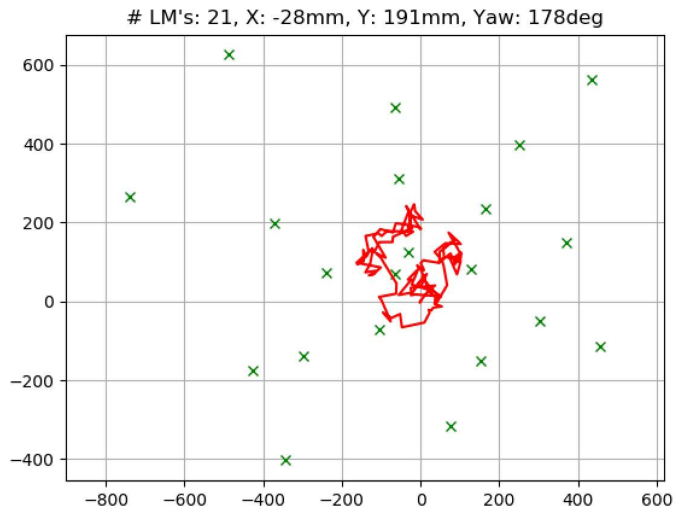

Figure 39 – EKF SLAM Initial Pose Estimation

Since the problems we experienced with SLAM were buried deep in a sea of covariance settings and kinematics models, it was decided that we would try and run the original data through the MATLAB SLAM algorithm, provided in the Robotics Systems Toolbox. Shown below in Figure 40 and Figure 41 are the results of the data from the algorithm.

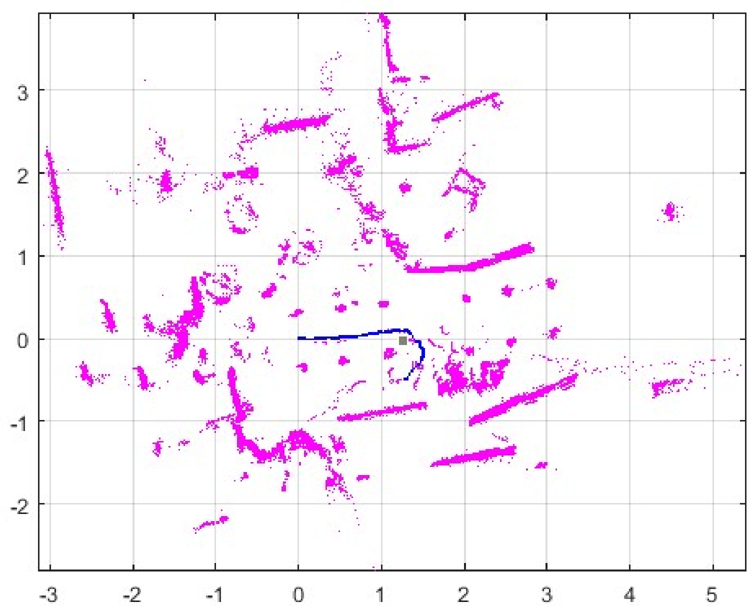

Figure 40 – MATLAB SLAM Algorithm (Curved Path)

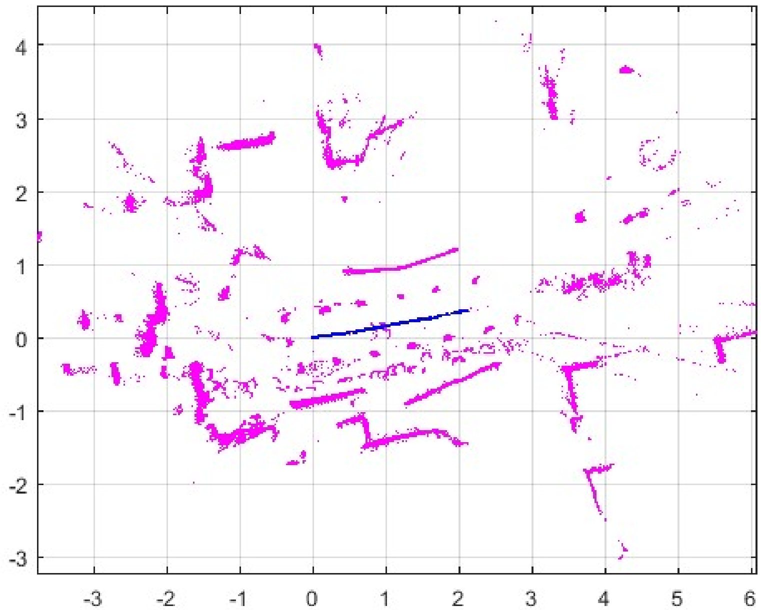

Figure 41 – MATLAB SLAM Algorithm (Straight Path)

The MATLAB SLAM algorithm does provide an accurate representation of the data, however, lacks in some areas. For instance, we noticed the algorithm runs slowly compared to the EKF SLAM algorithm. It also requires a Simulink connection between our mobile hardware and a computer running MATLAB. Lastly, the data shown above cannot be exported in it's given state. Instead, the map needs to be converted into an Occupancy Grid, which in the process may filter out the vine post clusters.

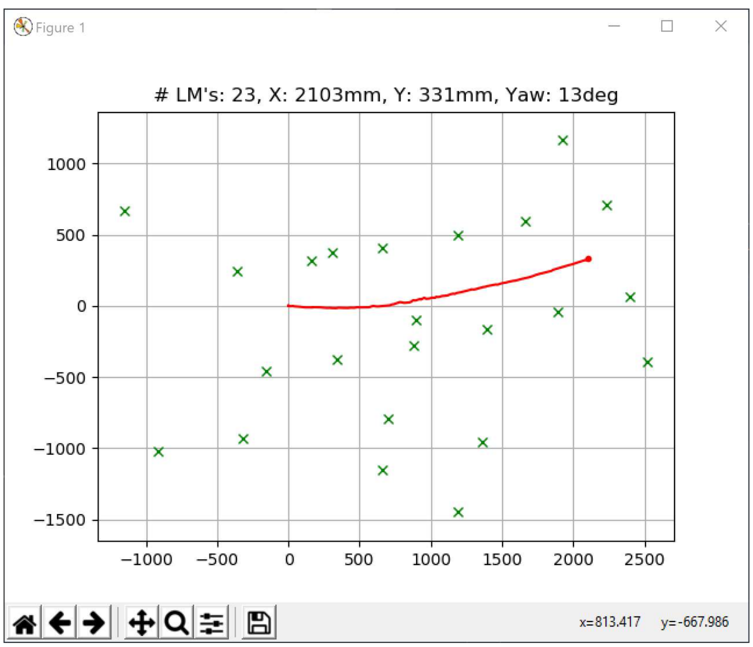

Figure 42 – Python EKF SLAM (Straight Path)

Based on the issues discovered through our testing of MATLAB SLAM, we decided to return to Python EKF SLAM. This time we were able to solve the issues related to the kinematic and covariance models of the system, and we got vineyard mapping and vehicle localization working in real-time, as seen in Figure 42. Through testing we determined that the pose estimation had a typical deviation of ±20mm and ±5°. This is largely due to the fact that the lidar can only see the face of the vine post nearest to it, so as the vehicle drives by, the position of the vine post is measured from a different side, therefore introducing an error relative to the size of the vine post. In our scaled down vineyard, this pose deviation is significant. However, since this deviation would be almost identical in an actual vineyard, it would be insignificant with respect to the width of the rows and size of the vehicle and harvesting arm.

Path Following

In order to simplify vehicle motion control, we started by implemented a wheel speed controller. This was an independent system running on an Arduino Nano and receiving motor speed commands via serial. Since the vehicle is a skid-steer, we grouped the motor together, so we only controlled left-side and right-side motor speeds respectively. The motor speed was measured by counting encoder pulses from two of the motors and fed into a PID loop to maintain the set speed.

Upon researching many feedback control solutions, we chose to use Model Predictive Control (MPC) as this provides a more robust motion control solution. Using the pose estimations from SLAM, the vehicle could follow a pre-determined path next to a row of grapes. However, the deviation in the pose estimation turned out to be too great to enable harvesting due to the short reach of the picking arm. Therefore, an odometry-based path following system was implemented for demonstration purposes.

4.2.6 Integration

This section outlines the systems put in place for communication between the main control modules of the autonomous harvesting system.

System Architecture

The inter-process communication (IPC) standard dictates how all elements of the robot communicate with each other. However, coordinate the sequencing between the systems and the system communication, the mission planner is used. The mission planning system is a state machine that dictates permissions of each system to peripherals such as the drive wheels and picking arm, as shown in Figure 43 above.

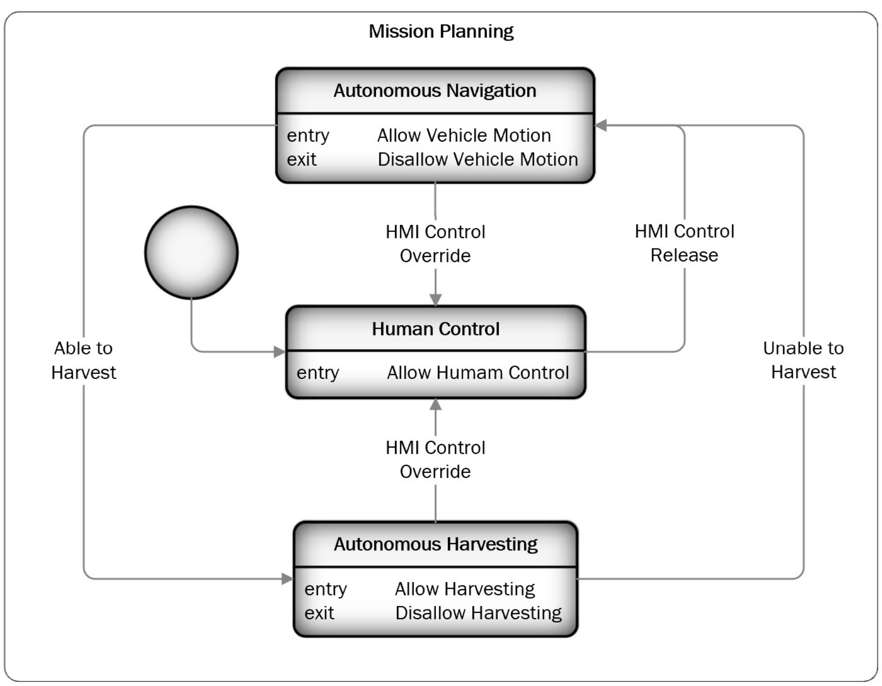

Figure 43 – Mission Planning System State Diagram

As the vehicle is driving along a vine row, the autonomous harvesting system sends a request to pick grapes when a grape is seen by the vision system. The mission planner halts the vehicle navigation and motion, allowing the robotic arm to pick grapes. Once all of the grapes in frame of the vision system have been picked, the autonomous navigation system then continues with route guidance while the vehicle advances, until the next request to pick.

The mission planning state machine also allows human interaction with the system for tasks such as starting and stopping the robotic vehicle.

IPC Standard

We decided to model the IPC standard as a simple handshake interface over python sockets. Commands took the form of strings like "ready," "allow," or "done," and were encoded into json format before being sent over TCP socket between the mission planning module and the harvesting module. The mission planner was considered the master with the harvesting module being the slave.

Table 3 – IPC Command Standards

Task	Requested By:	Command	Acknowledged By:	Command
Picking	Slave	"ready"	Master	"allow"
Finished picking	Slave	"done"	Master	N/A
Move Forward	Slave	"fwd"	Master	"done"
Move Backward	Slave	"rev"	Master	"done"
Move Closer	Slave	"far"	Master	"done"

4.2.7 Costs

Shown below are some estimated costs for the materials used in development of our capstone project. Its worthy to note that there were some materials omitted from the list due to previous development in the project.

Table 4 – Estimated Development Costs

Item	Quantity	Estimated Cost ($ CAD)
NVIDIA Jetson TX2	1	$ 388.70
Raspberry Pi 3B+	1	$ 63.49
Arduino Nano	1	$ 9.99
SLAMTec RPLidar A2	1	$ 429.42
uFactory uArm Swift Pro	1	$ 749.00
Logitech C922x Pro	1	$ 129.99
PMD CamBoard Pico Flexx	1	$ 447.33
Cytron 10A 5-30V Dual Channel DC Motor Driver	2	$ 51.34
12V DC-DC Converter	1	$ 24.00

Nomad Chassis with Motors	1	$ 279.99
Replacement Encoder Motors	2	$ 120.00
Turnigy High Capacity 16000mah 6s 12c Multi-Rotor LiPo Pack W/XT90	1	$ 190.26

4.3 Results, Testing, and Verification

To best understand what was accomplished during this project, it's imperative that the results be brought forward, along with the testing procedure. In this section of the report, we hope to go over the requirements which we had initially set during the proposal period of the project and introduce our results in relation to those requirements. A full list of the requirements for this project can be found in the VineBot requirements document [30]. It's also worthy to note that the two GitHub repositories which hold the code pertaining to our project are provided in Table 5 below, with their use-cases.

Table 5 – GitHub Repositories

Repository Use	Link
Repository for code regarding everything except machine vision and learning (Raspberry Pi)	https://github.com/terrycalderbank/soe-VineBot
Repository for code regarding machine vision and learning (Nvidia Jetson TX2)	https://github.com/terrycalderbank/VineBot_ml

4.3.1 Vision

The vision system had the job of being able to detect grapes, and aid in robotic movement. Some relevant requirements for the vision system were as follows.

1. Identify grape clusters with at least 75% accuracy
2. Identify purple, green, and red grapes
3. Identify a point in space with cameras, using a cartesian coordinate system

To test the first requirement, we made our YOLO model detection threshold 75% and moved our camera assembly around a set of 8 artificial grapes on a white background. Initially, we tested on our first machine learning model which was trained solely off the Google Open Images grape dataset. We then tested on our Tiny-YOLO model, and the normal YOLO model with our reinforced learning method. These tests were repeated, however with the artificial grapes attached to our artificial grape vine. Shown below are the results of those tests, which show the number of clusters continuously detected while moving the camera.

Table 6 – Detection Rate of 8 Clusters with Threshold Of 75%

	Basic Machine Learning	Reinforced Machine Learning	Tiny-YOLO
White Background	75%	100%	50%
Grape Vines in Background	63%	100%	50%

The machine learning curves are also shown below for the originally trained YOLO model, the revised YOLO model (using the closed-loop training method), and the Tiny-YOLO model we attempted to look into.

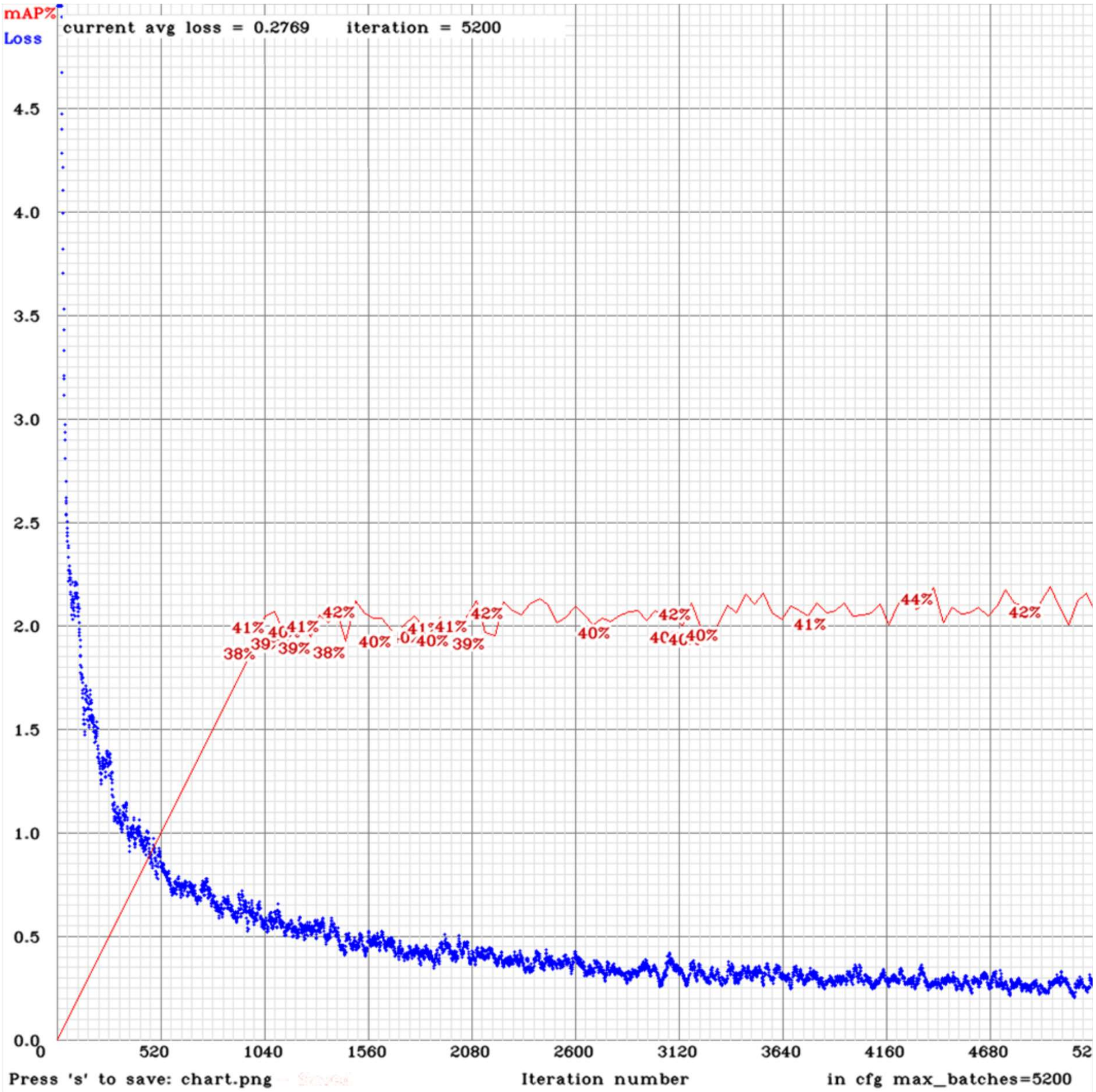

Figure 44 – mAP-Loss Training Curve for YOLO Training

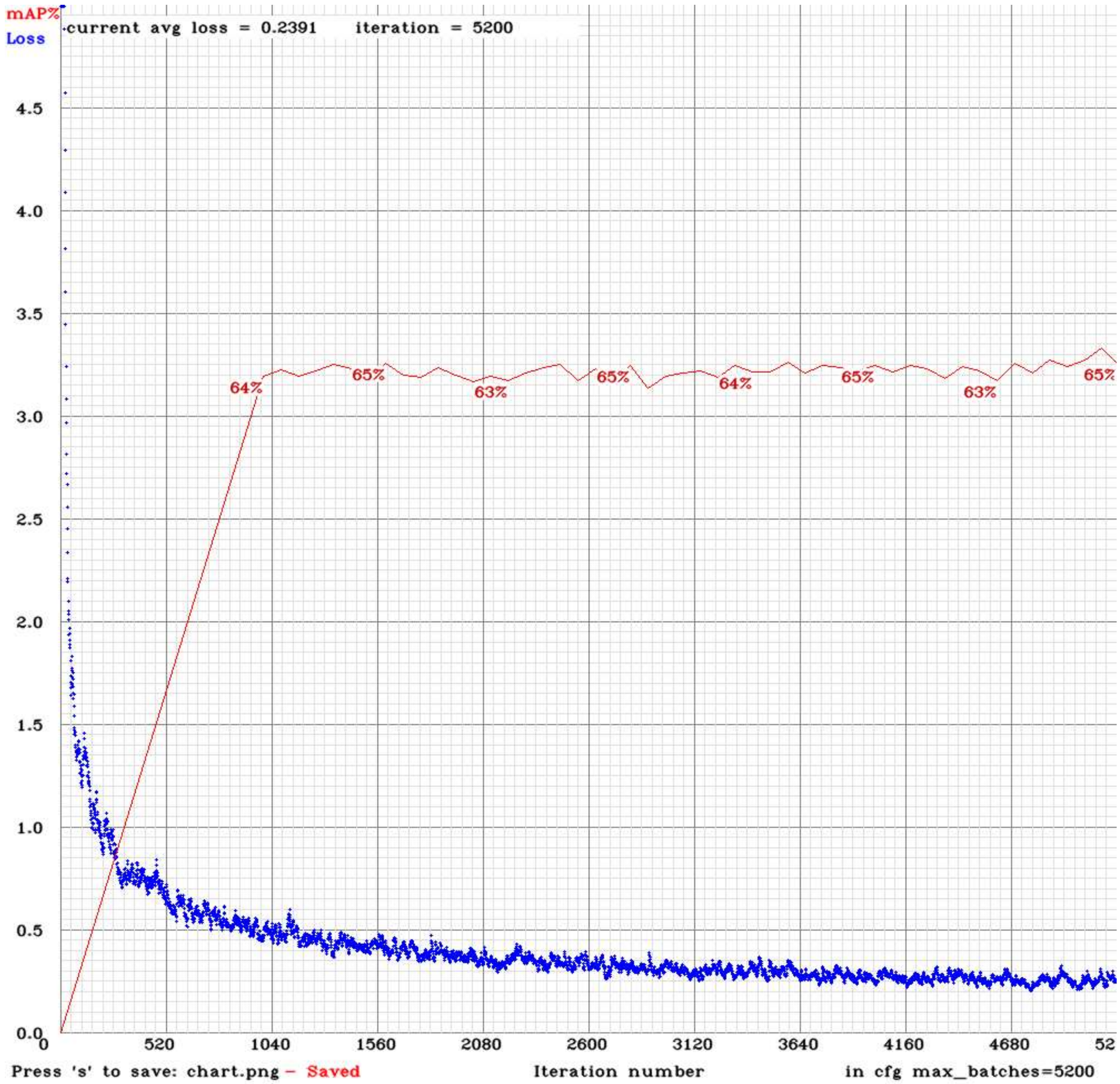

Figure 45 – mAP-Loss Training Curve for Revised YOLO Training

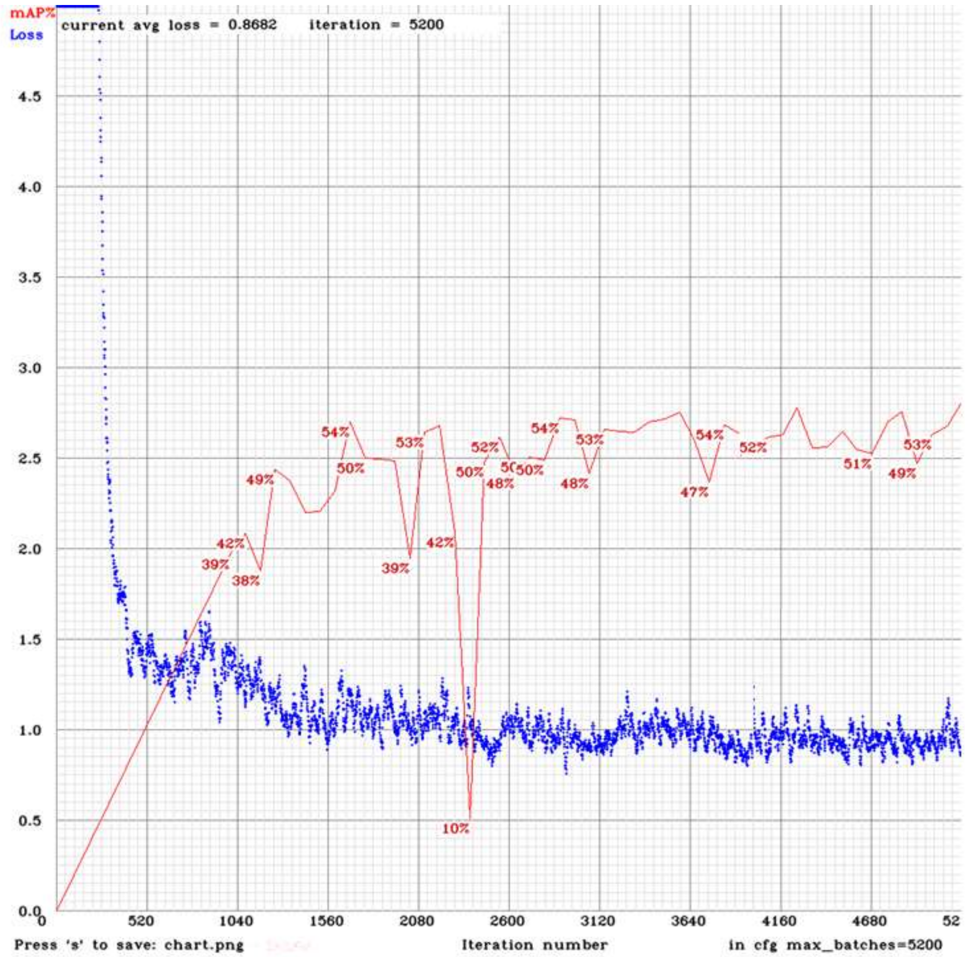

Figure 46 – mAP-Loss Training Curve for Tiny-YOLO Training

For the second requirement, we were able to detect grapes of different colors based on parameters we changed in our machine learning setup. Shown below is a detection frame which proves we are able to detect purple, green, and red grapes.

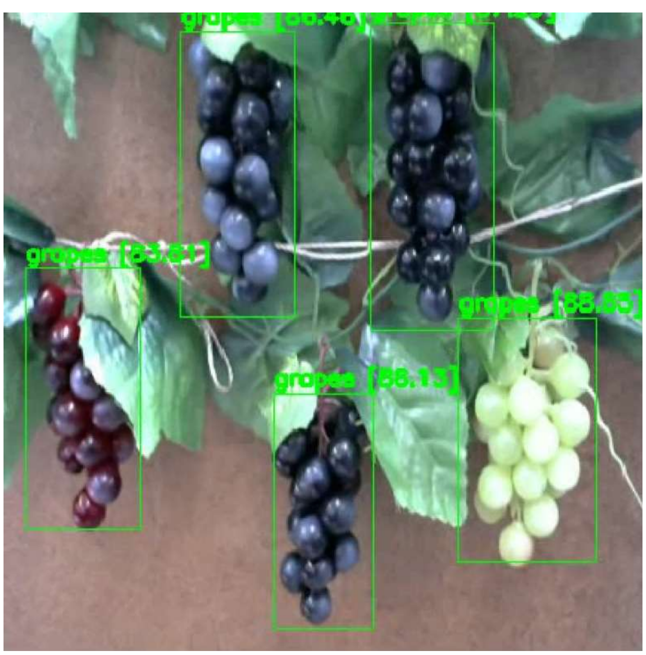

Figure 47 – YOLOV2 Based Detection Algorithm

The last requirement, being able to identify a coordinate in space, we were able to complete using relative coordinates from a 2D image and a relative coordinate for depth from the depth camera. This method allows us to identify a 3D coordinate. The coordinates from the 2D image are in pixels, while the depth measurement is in meters.

Some other relevant results to bring forward include the machine learning detection performance on different devices. Table 7 summarizes the frame rate at various network resolutions and versions of YOLO running on the NVIDIA Jetson TX2 and an Ubuntu computer with the NVIDIA GeForce GTX-1080.

Table 7 – YOLO Performance with respect to Device

	Resolution (px)	Device Performance	
		NVIDIA Jetson TX2 (FPS)	NVIDIA GeForce GTX-1080 (FPS)
YOLO	416x416	7	60
Tiny-YOLO	416x416	30	144
	608x608	12	N/A

From the results above, you can see that the performance is much better on the Ubuntu PC, but it's important to note that it's due the hardware acceleration (NVIDIA GeForce GTX-1080 graphics card).

4.3.2 Arm Motion

Effective grape picking requires adequate operation of the robotic arm. The requirements for the robotic arm system were as follows.

1. Robotic arm should pick grapes from grape vine without damaging grape vine or grapes
2. Robotic arm should be able to move end-effector with 6 degrees of freedom
3. Robotic arm should not oscillate after movement to new position
4. Robotic arm should be able to move end-effector to a location with precision of ±0.25" up and down, ±0.50" side to side and ±0.38" back and forth
5. Robotic arm end-effector should pick grape cluster from stem.
6. Robotic arm assembly should spin no more than one rotation
7. End-effector and gripper assembly should weigh less than or equal to 500g
8. Robotic arm assembly should run off a 12V 5A supply

The robotic arm we chose to use for our project was the uArm Swift Pro. We were able to determine the operating limits of our robot including the mass of the end-effector assembly, shown below in table.

To test the first requirement, we created an artificial stem and stem cutting end-effector. An acceptance test was carried out and prove that the operation was satisfactory, which involved the robotic gripper holding a grape cluster at the stem, and then cutting the 3D printed plastic stem using the cutting mechanism.

For the second requirement, we needed to reduce our requirement to have the robot arm move with 3 degrees of freedom. Therefore, we were not able to meet the hardware requirement of having a robot with 6 degrees of freedom. We were however able to meet the non-oscillatory requirement for the robotic arm.

Due to our choice of robotic arm, we were able move our end-effector with a precision greater than the distances shown in Table 8, and therefore were able to verify acceptance. The related requirement was verified by using the uArm Studio software package to control the robotic arm and move the end-effector in 3 directions.

Table 8 – Precision of Movements for Robotic End-Effector

Direction	Distance
Up/Down	0.25"
Left/Right	0.50"
Forward/Backward	0.38"

With respect to the weight requirement for the end-effector and gripper assembly, we were able to bring the total mass to less than 500g, as stated by the requirement. The mass breakup is shown below in Table 9.

In terms of performance, we noticed that the robot was failing due to excessive dynamic loading. Given certain accelerations required to achieve high speeds, the inertia of the load would overload the staying power of the stepper motors. Keeping the mass of the end-effector to a reasonably low level allowed us to achieve higher tool movement speeds.

Table 9 – End-Effector Mass Breakdown

Item	Mass
Gripper	220 g
Camera Mount	40 g
Cutting Mechanism	60 g
Grape Cluster	20 g
Total	340 g

We were also able to run the robotic arm off a 12V-5A supply with the possibility of up to a 15V supply for increased performance due to the higher staying power within the stepper motors.

It was also noticed in testing that some motion ranges of the arm were protected from overreaching by software limits while others were not. We needed to program additional motion limits into the robot to prevent it from failing. Shown below in table are the hard-coded limits we've programmed in the robot.

Table 10 – Programmed Robot Motion Limits

Tool Height	Reach Limit
> 150mm	System imposed: ~340mm
<= 150mm	330mm

A stretch goal of this project that was not included in the deliverables was to create a cutting mechanism which would emulate a semi-realistic harvest of a grape bunch. A short summary of the different cutting mechanisms we experimented with are shown below in table, with the chosen method being the servo-assisted cutter.

Table 11 – Cutting-Mechanism Force Comparison

	Gripper	Solenoid	Servo
Force (N)	4.9	0.5	11.8
Voltage (V)	5	5	5
Built & Tested	Yes	No	Yes
Adequate Performance	No	No	Yes

4.3.3 Chassis

Some relevant requirements for the vehicle chassis were as follows.

- The robot should be powered by a single 12V rechargeable supply with optional low-voltage supply outputs (e.g. 3.3V, 5V)
- The robot battery supply should be easily changeable.
- The robot supply should power the device for at least an hour on a single charge
- All bare conductors should not be left exposed near metal chassis
- All electronics should be insulated from metal chassis
- Chassis should be redesigned to fit all electronics and insulate them from environment.

For the first set of requirements, we decided to use a 22.2V 6 cell LiPo battery, which we calculated to last us at least 4 hours of operation. The battery was connected to the central supply via xt90 connectors for swapping and was regulated using a 12V automotive buck converter. The requirements were tested using equipment such as the benchtop digital multi-meter and oscilloscope. The output ripple on the converter was found to be <45 mV peak-to-peak.

In terms of the chassis and electronics, we were not able to redesign the chassis to enclose all of the electronics due to time constraints. Instead, we decided to mount the electrical components on the outside of the chassis. We did however insulate the metal chassis from bare conductors. All of the chassis requirements were verified by acceptance tests.

4.3.4 Motion & Navigation

Some relevant requirements regarding the motion and navigation were as follows.

- Control robotics using vision and sensors and communication with a single platform
- Handle processing with single board System on Chip (SoC), or a network of boards
- Processing unit should utilize a single programming language for processing, and limit the use of other languages (Python, C++ or C)
- Processing unit should have 2 or more USB interfaces, and a combined multicore processor for software requirements
- Vehicle should be able to navigate vineyard (without pre-programmed route), avoid obstacles, and stay within 35 cm (from inner tire of vehicle) to vine of interest, all with the help of sensor input

For the processing requirements, we were decided to use the NVIDIA Jetson TX2 and a Raspberry Pi 3B+ single board computer. The NVIDIA Jetson TX2 was used to run the vision system, while the Raspberry Pi was used to control the navigation and robotics systems. However, networking issues we experienced using the Raspbian operating system (OS) made it difficult to run the navigational and robotics tasks reliably. Therefore, we decided to run the effected tasks on a separate PC, while having the NVIDIA Jetson TX2 continue to run the machine vision operations. For all of the scripting in the project, we used Python due to ease of rapid prototyping, it's plethora of code samples and support, and the number of ready-made packages for tasks such as machine learning and robotics control.

For the processing requirements, we determined that the NVIDIA Jetson TX2 runs off of a Dual-Core NVIDIA Denver 2 64-Bit CPU and a Quad-Core ARM® Cortex®-A57 MPCore, while the Raspberry Pi runs the 1.4GHz 64-bit quad-core Broadcom Arm Cortex A53-architecture processor. Between the two devices we were able to meet the requirements for multiple cores in the system (sufficient for multithreading), and multiple USB ports.

Some performance results for the NVIDIA Jetson TX2 with our final configuration are shown below in Table 12.

Table 12 – NVIDIA Jetson TX2 Performance

Metric	Performance
Darknet/YOLO Detection	7 FPS
Depth Camera	25 FPS

5 Conclusions and Recommendations

We hope that the earliest sections have been indicative of a definite need for a precision picking system for grapes with the goal of combating labour shortages in a growing industry that has the potential to continue growing due to external factors like climate change induced crop migration.

Through development, testing, and verification of our project, we have outlined enough detail to show that we were successful in achieving the following deliverables (with some completed fully, and others to a degree):

1. A proof of concept autonomous grape picking vehicle
2. A final report documenting our project
3. An organized git repository with extended project documentation
4. A live demonstration of the project for the capstone expo including signage and literature

From our results, one can deduce that the vision system is fairly reliable. Throughout testing, we were unable to witness a case when grapes were not detected. Running the machine learning model on the embedded computer (Nvidia Jetson TX2) was significantly slower than on a desktop with a graphics card, however this was expected, and could be improved by switching to a more optimized neural network framework (TensorRT).

The robotic arm that we decided to use made it difficult, but not impossible, to meet some of our requirements and objectives. Having less degrees of freedom (DoF) meant that the robot was harder to maneuver, while also simplifying its control. We were also able to determine our robot's working limits with the load of the camera assembly on the end-effector and were able to verify most of the robotic requirements via acceptance testing. A cutting end-effector was implemented and tested.

Given the time constraints, we were unable to re-design the chassis to accommodate all of the electronics internally. This however wasn't a planned deliverable and therefore wasn't considered in the minimum viable product (MVP). The supply for our vehicle was regulated to 12V using a DC-DC convert, and through the results, we were able to determine the output ripple on the converter to be less than 45 mV peak-to-peak. A majority of the requirements were met via acceptance testing.

For the motion and navigational processing requirements, we were unable to meet the requirement of using on-board computers only. This is was due to unreliable network performance we experienced when using the Raspberry Pi with the Raspbian OS. Instead, we decided to use a computer in its place, which we plan on replacing in the near future. We were successful in integrating the two separate systems to work as one cohesive unit.

5.1 Relatable Applications

A possible application which we can relate to our project is the harvesting of cannabis. Since the legalization of cannabis in Canada, there has been an increase in attention and growth in British Columbia's cannabis industry. In 2018, Statistics Canada estimated the size of the cannabis market to be $5.7 billion, rising at an average rate of 1.0% per quarter [31], and according to presenters at a BCIT hosted ISA presentation on cannabis automation, little work has been done with regards to the automation of harvesting cannabis. We could potentially take advantage of this recent agricultural opportunity by modifying our project to detect and pick cannabis buds rather than grape clusters.

Since cannabis grows in greenhouses, a rugged vehicular design wouldn't be needed. Thus, we could retrofit a robotic arm on a conveyer or rail system, that would be able to move across crop rows. We also wouldn't need to be as concerned with environmental conditions, since crops grown in greenhouses are typically well-regulated.

In terms of the technical modifications needed to detect marijuana buds, we would simply need to retrain our neural network model on marijuana buds, instead of grape clusters. The main challenge would be finding a large enough dataset to train our model on. Another possible challenge would be to develop a new end-effector specifically for harvesting buds, based on current methods (clipping).

5.2 Future Work

Since we've developed a scaled down autonomous grape picking vehicle, it would be beneficial to create a full-sized vehicle which can be tested on a real vineyard. Our industry contact Graeme Duncan, of Naramata BC, has been kind-enough to allow us to test a prototype in his personal vineyard. This gives us strong motivation to create a full-sized vehicle capable of dealing with real crops rather than our simulated vineyard. It would also be a good idea to redesign the vehicle chassis so that it is properly equipped for outdoor use. This includes shielding from the elements (Rain, Snow, Dust) and coming up with measures for dealing with performance-hindering natural occurrences (i.e. dust and moisture on camera lenses).

More time could also be spent on the picking mechanism, including replacing the consumer grade robot with something more industrial with a higher payload, designing a custom cut-and-grip end-effector, and improving the machine learning model such that it is more accurate, precise, and fast with regards to crop detection.

Lastly, we would like to explore the possibility of commercialization. We've determined that "robot rental" business model best suits our product and collective goals.

6 References

[1] Government of Canada, "Summary - Canadian Industry Statistics - Wineries - 31213," *Canadian Industry Statistics*, 2016. [Online]. Available: https://www.ic.gc.ca/app/scr/app/cis/summary-sommaire/31213. [Accessed: 07-Dec-2018].

[2] Statistics Canada, "Agriculture," *11-402-X*, 2018. [Online]. Available: https://www150.statcan.gc.ca/n1/pub/11-402-x/2011000/chap/ag/ag-eng.htm. [Accessed: 19-Oct-2018].

[3] California Farm Bureau Federation, "Searching for Solutions : California Farmers Continue to Struggle with Employee Shortages," no. October, 2017.

[4] Agriculture Canada, "The Impact of climate change on Canadian Agriculture," *Agriculture and Agri-Food Canada*, 2015. [Online]. Available: http://www.agr.gc.ca/eng/science-and-innovation/agricultural-practices/climate-change-and-agriculture/future-outlook/impact-of-climate-change-on-canadian-agriculture/?id=1329321987305. [Accessed: 19-Oct-2018].

[5] C. Arsenault, "In Canada, climate change could open new farmland to the plow," *Thomson Reuters*, 2017. [Online]. Available: https://www.reuters.com/article/us-heatwave-canada-farming/in-canada-climate-change-could-open-new-farmland-to-the-plow-idUSKCN1BZ075. [Accessed: 19-Oct-2018].

[6] The Daily, "Control and sale of alcoholic beverages , for the year ending March 31 , 2012," pp. 7–12, 2018.

[7] Agriculture and Agri-Food Canada, "Canada's Wine Industry," *Canadian agri-food sector intelligence*, 2016. [Online]. Available: http://www.agr.gc.ca/eng/industry-markets-and-trade/canadian-agri-food-sector-intelligence/processed-food-and-beverages/profiles-of-processed-food-and-beverages-industries/canada-s-wine-industry/?id=1449859691976. [Accessed: 07-Dec-2018].

[8] MarketLine, "Forecasted market value of the wine industry in Canada from 2014 to 2019 (in billion Canadian dollars)," 2019.

[9] Statistics Canada, "Canadian total exports," *NAICS 31213 - wineries*, 2017. [Online]. Available: https://www.ic.gc.ca/app/scr/tdst/tdo/crtr.html?lang=eng&hSelectedCodes=31213&searchType=BL&productType=NAICS&currency=CDN&runReport=true&grouped=GROUPED&toFromCountry=CDN&naArea=9999&timePeriod=5%7CComplete%2BYears&reportType=TE&countryList=ALL. [Accessed: 19-Oct-2018].

[10] Statistics Canada, "Canada's Wine Industry," *NAICS 31213*, 2016. [Online]. Available: http://www23.statcan.gc.ca/imdb/p3VD.pl?Function=getVD&TVD=307532&CVD=307548&CPV=312130&CST=01012017&CLV=5&MLV=5.

[11] Canadian Agricultural Human Resource Council, "Agriculture 2025 : How the Sector's Labour Challendes Will Shape It's Future," 2016.

[12] Government of Canada, "Hire a temporary worker through the Seasonal Agricultural Worker Program - Overview," *Employment and Social Development Canada*, 2018. [Online]. Available: https://www.canada.ca/en/employment-social-development/services/foreign-workers/agricultural/seasonal-agricultural.html. [Accessed: 07-Dec-2018].

[13] C. Arsenault, "In Canada, climate change could open new farmland to the plow," *Reuters*, 2017. [Online]. Available: https://www.reuters.com/article/us-heatwave-canada-farming/in-canada-climate-change-could-open-new-farmland-to-the-plow-idUSKCN1BZ075. [Accessed: 19-Oct-2018].

[14] Statistics Canada, "Canada's Wine Industry," *NAICS 31213*, 2016. [Online]. Available:

http://www.agr.gc.ca/eng/industry-markets-and-trade/canadian-agri-food-sector-intelligence/processed-food-and-beverages/profiles-of-processed-food-and-beverages-industries/canada-s-wine-industry/?id=1449859691976. [Accessed: 19-Oct-2018].

[15] S. Nuske, S. Achar, T. Bates, S. Narasimhan, and S. Singh, "Yield Estimation in Vineyards by Visual Grape Detection (in process of publication)," *2011 IEEE Conf. Intell. Robot. Syst.*, 2011.

[16] L. Luo, Y. Tang, X. Zou, C. Wang, P. Zhang, and W. Feng, "Robust Grape Cluster Detection in a Vineyard by Combining the AdaBoost Framework and Multiple Color Components," *Sensors*, vol. 16, no. 12, p. 2098, Dec. 2016.

[17] M. J. C. S. Reis *et al.*, "Automatic detection of bunches of grapes in natural environment from color images," *J. Appl. Log.*, vol. 10, no. 4, pp. 285–290, Dec. 2012.

[18] J. Redmon, "Darknet: Open Source Neural Networks in C." [Online]. Available: https://pjreddie.com/darknet/. [Accessed: 11-Feb-2019].

[19] C. Robotics, "HUSKY UGV - Outdoor Research Field Robot by Clearpath."

[20] Anna, "Husky UGV helps US grape growers conserve water," 2018.

[21] A. Sakai, "Python sample codes for robotics algorithms." [Online]. Available: https://atsushisakai.github.io/PythonRobotics/. [Accessed: 07-Dec-2018].

[22] M. Hansen and T. Mullinax, "New grape harvester leaves MOG in the field," *Good Fruit Grower*, 2015. [Online]. Available: https://www.goodfruit.com/new-grape-harvester-leaves-mog-in-the-field/. [Accessed: 19-Oct-2018].

[23] A. R. Barth, J. Hemming, P. Kurtser, Y. Edan, O. Ben-shahar, and J. Hemming, "Sweet Pepper Harvesting Robot GRANT AGREEMENT Number 644313 Deliverable title : Report on test scenarios and definition performance measures WP number : 7 Lead beneficiary : DLO Nature : R Dissemination level : PU," 2016.

[24] E. Ackerman, "SRI Spin-off Abundant Robotics Developing Autonomous Apple Vacuum," *IEEE Spectrum*, 2016. [Online]. Available: https://www.intorobotics.com/fruit-harvesting-robots/. [Accessed: 19-Oct-2018].

[25] A. A. Yavuz, B. Glas, M. Ihle, H. Hacioglu, and K. Wehefritz, "(12) Patent Application Publication (10) Pub. No.: US 2014/0270163 A1 - ROBOT MECHANICAL PICKER SYSTEM AND METHOD," 2005.

[26] T. Pang-Ning, S. Michael, K. Anuj, and K. Vipin, "Cluster Analysis: Basic Concepts and Algorithms," in *Introduction to Data Mining (Second Edition)*, Pearson, 2018.

[27] D. Gatziolis and H.-E. Andersen, "A Guide to LIDAR Data Acquisition and Processing for the Forests of the Pacific Northwest.," no. July, 2008.

[28] OpenCV, "OpenCV Houghcircles()." [Online]. Available: https://docs.opencv.org/3.1.0/da/d53/tutorial_py_houghcircles.html. [Accessed: 19-Mar-2019].

[29] uFactory, "uArm Gripper - store.ufactory.cc," 2019. .

[30] T. Calderbank, N. Kambo, and B. Aimar, "VineBot Technical Design Requirements." British Columbia Institute of Technology, Burnaby, 2018.

[31] Statistics Canada, "Cannabis statistics," *Statistics Canada*, 2018. .

7 Appendices

7.1 Appendix A: Background Information

7.1.1 Machine Learning

One of the key findings that we realized when attempting to train for machine learning was that there are several formats for image annotation. We also were able to experience the difference hardware acceleration (graphics card) plays in terms of machine learning. Also, lighting may not be important enough to worry about, based on our testing with the machine learning model.

Convolutional Neural Network

A neural network is a method for detecting features within an image. A typical neural network consists of an input layer, an output layer, and some hidden layers in-between which are used for manipulation. A basic neural network diagram is shown in Figure 48, where a single hidden layer is sandwiched between the input layer and output layer.

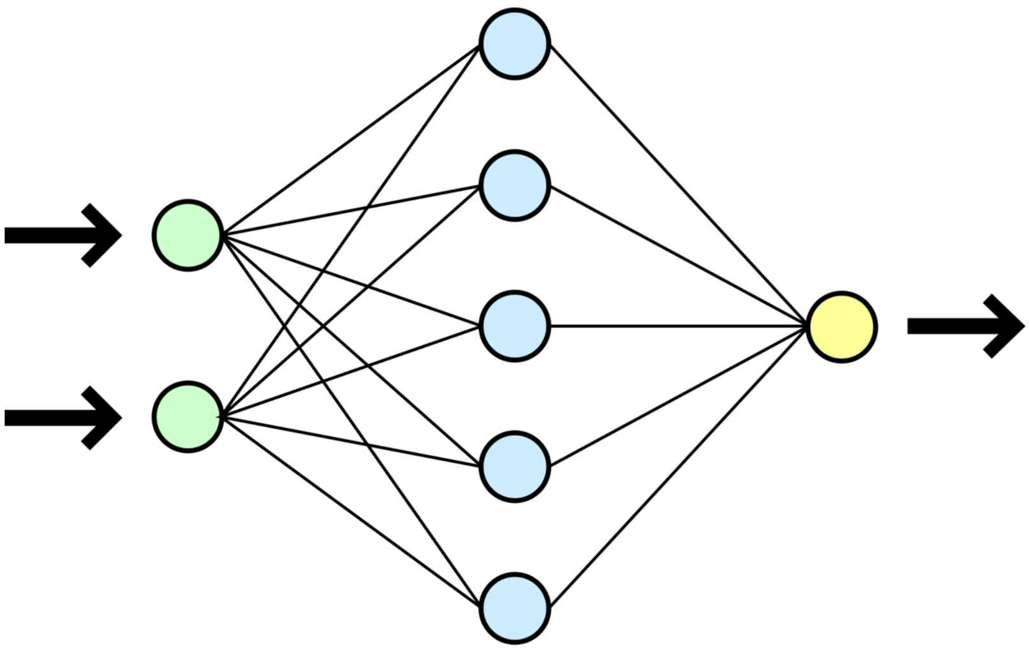

Figure 48 – Neural Network

When dealing with images, a convolutional neural network (CNN) is usually applied. A CNN consist of a regular neural network, with the input being a window of a predetermined size. The window slides across the entire image, and with the help of the hidden layers, gives rise to features in the image.

The hidden layers are attributed with the actual detection of objects and are often called filters or kernels. These hidden layers are what are being found or developed during the machine learning process and are expressed by coefficients. Initial hidden layers for a model usually give the ability to detect edges and other simple features of an image for example.

Machine Learning Basics

For the initial milestone, we set a goal of being able to train a machine learning model to do grape cluster detection. To get to the point of detection, we first needed to be able to understand how the machine learning process operated. Figure 49 shows a simplified machine learning model. From the figure, one can deduce that the machine learning method is comparable to a black box, where it is very hard to determine what the machine is actually learning.

Figure 49 – Machine Learning Model (simplified)

To best understand what is going on when object detection is taking place, it is important to define and break object detection into two terms, shown below in Table 13.

Table 13 – Basic ML terms and definitions

Terminology	Definition
Classification	To determine if a classifier is present in frame.
Localization	To determine where a classifier is located in frame.
Precision	Proportion of positive identifications that were actually correct.
Recall	Proportion of actual positives that were identified correctly.

There are also two other terms, Precision and Recall, which are more mathematical constructs. Shown below in are the mathematical representations of precision and recall.

$$Precision = \frac{True_{Positives}}{True_{Positives} + False_{Positives}}$$

Equation 1 – Precision Definition

$$Recall = \frac{True_{Positives}}{True_{Positives} + False_{Negitives}}$$

Equation 2 – Recall Definition

Precision and Recall are inversely proportional to each other, meaning an increase in Precision will result in a decrease in Recall. This can be shown in graphical format as a Precision-Recall curve, an example of which is shown below in Figure 50.

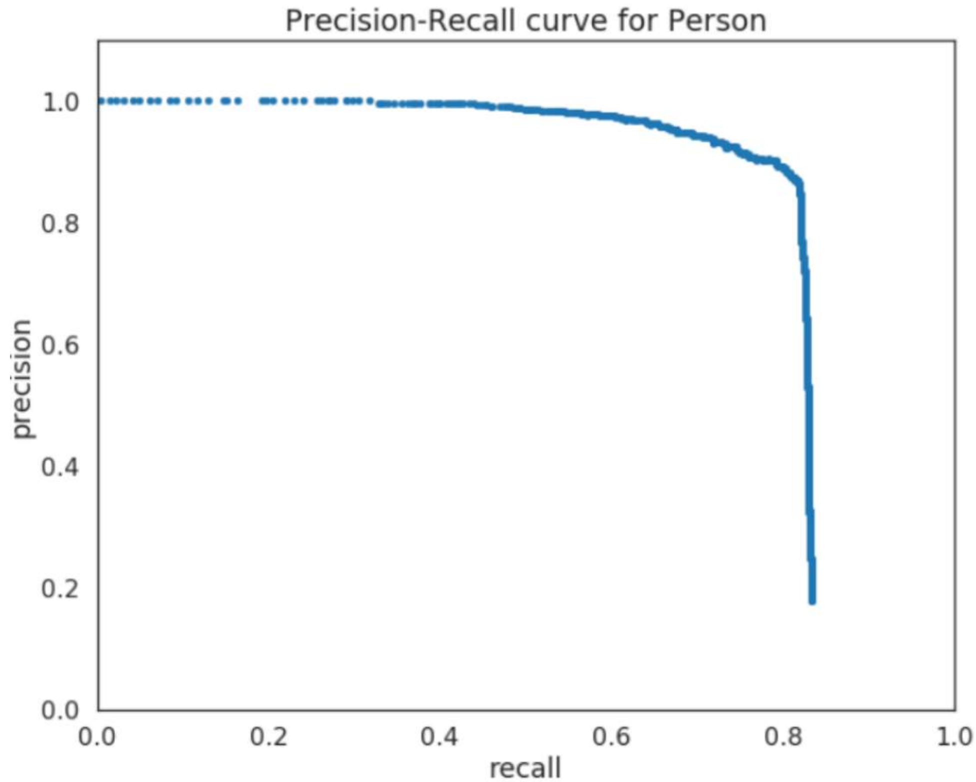

Figure 50 – Precision-Recall curve for classifier Person

Using the Precision-Recall curve, one can calculate the Average Precision (AP), which is the moving average of the curve. According to a source, the "Precision-Recall curves summarize the trade-off between the true positive rate and the positive predictive value for a predictive model using different probability thresholds." This is a useful metric that deals with quantifying classification.

$AP = \frac{1}{n+1}\sum_{i=0}^{n} P(R_i)$ where $P(R_i)$ is the Precision at Recall R_i.

Equation 3 – Average Precision

Another metric which is meant to measure the localization is the Intersection over Union (IoU). This metric finds the variation between the ground truth and predicted localization. The intersection represents the areas which are common between the two sets of bounding boxes, while the union represents the total area covered by the two sets of bounding boxes. This measurement quantifies localization.

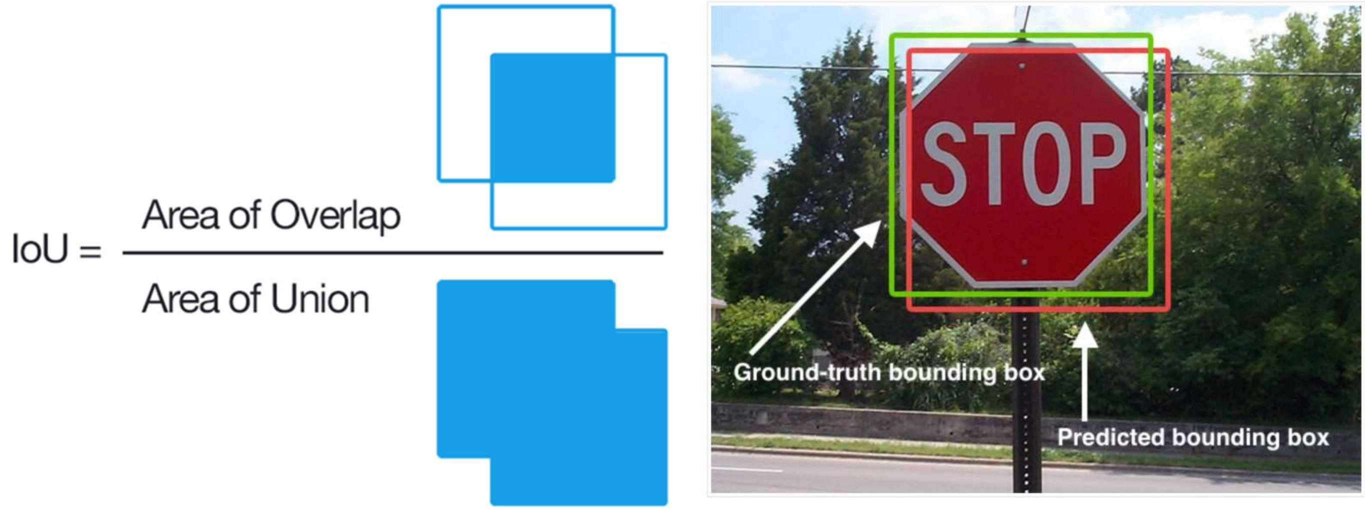

Figure 51 – IoU examples

However, the IoU is directly related to the Precision-Recall curve. For different IoU thresholds, the Precision-Recall curve shifts, resulting in an in-direct relationship between classification and localization. This gives rise to the mean average precision (mAP), which is the mean of the average precision values at different IoU thresholds. The mAP metric is particularly important because it is indicative of a model performance. Shown below is a Precision-Recall curve with different IoUs, in Figure 52.

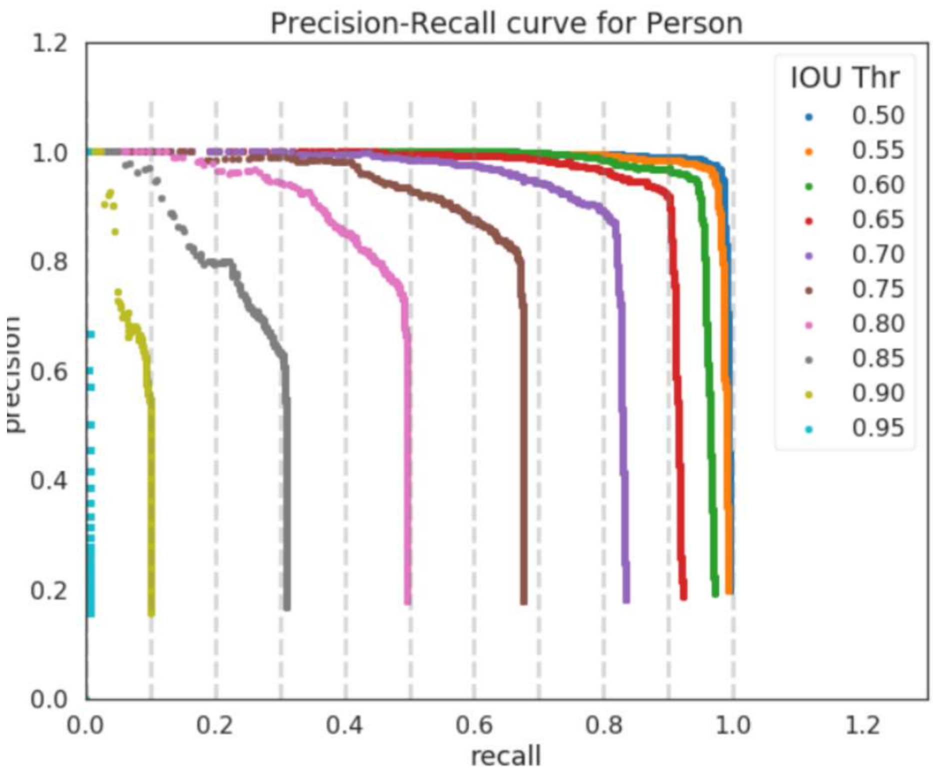

Figure 52 – Precision-Recall-IoU curve

In terms of the training process, images are fed into the machine, and the machine then creates a model based on features it has interpreted identify the classifier, via the use of a convolutional neural network. So, for each image fed into the machine, a matching annotation file also needs to be present, which shows the machine where in the image the classifier is present. However, due to the various methods and programs used for training machine learning models, there are different annotation formats.

Machine Learning Datasets

Since we downloaded a precompiled machine learning dataset from Google Open Images, the dataset came fully annotated in the Google Open Images format. However, we needed the images to be in the correct format for Darkflow/Darknet YOLO. The annotation formats for both datasets are shown below in Figure 53.

```
Google Open Images Format

<Classifier> <Xmin> <Ymin> <Xmax> <Ymax>

E.g. Grape 100.43 32.23 123.5 200.3

YOLO Format

<Classifier> <Xcenter> <Ycenter> <Width> <Height>

E.g. 0 0.5 0.43 0.25 0.28
```

Figure 53 – Image Annotation Formats

To convert between the two datasets and verify that the datasets were in fact annotated correctly, two python scripts were created which converted the annotation format and allowed a user to view what exactly was annotated for each image. The scripts can be found in the relevant appendix section of this report.

Machine Learning Metrics

Since machine learning is very obscure, it would be great to have a way to evaluate the accuracy and efficiency of our trained model and understand how the model accuracy changes depending on the dataset, and the amount of time the machine is training for. The metrics consider both classification and localization, as well as losses that occur during the machine learning process.

7.2 Appendix B: SWOT Analysis

To briefly summarize important details from the internal and external analysis of the VineBot project, a compilation of strengths, weaknesses, opportunities and threats are gathered below in Table 14.This SWOT analysis includes information discussed previously in the proposal.

Table 14 – SWOT Analysis

Strengths	Weaknesses
• Accumulated Team Experience • Previous Progress • Testing Flexibility • Minimized Losses with Precision Grape Picking	• Limited Business Experience • Divided Priorities (School)
Opportunities	**Threats**
• Growing Agricultural Automation Demand • Growing Agriculture/Wine Markets • Lack of Precision Picking Solutions	• Speed of Non-Precision Solutions • Drought

7.2.1 Strengths

Our strengths include accumulated team experience, the previous VineBot work completed for the SoE, and the inherent flexibility and adaptability of the product.

Besides his experience in the telecommunications field, and skill as a computer vision programmer, Navraj is a forward-thinking competent documenter who ensures that notes are taken throughout the design process. These notes are invaluable when it comes to troubleshooting and evaluating alternatives. Navraj has a knack for creative research and consistently finds resources that enable progress. Brayden brings to the table four years of professional mechanical design and automation experience. His work with CNC machines has provided us expert insight into motion control which is a large component of past, present, and future work. Brayden's experiences in the industrial workplace have given him a practical and pragmatic viewpoint into design. Terry has experience in the industrial robotics and automation industry. He also introduces creativity to any project taken on and is a dependable group member. Projects overseen by Terry are seen through to completion. He is also from BC's main wine growing region and has experience harvesting wine grapes.

The SoE project has provided us with a functioning prototype vehicle, and most of the financial resources required for the capstone project. We have already ordered many of the parts required for our project and as a result have more time to work towards the capstone. Most of the research we have done for the previous project is applicable to this one as well.

Early design decisions have given us flexibility in testing motion and vision. We have built sixteen feet of scale vineyard model to test our work with. The realism of our model allows us to conduct indoor and outdoor testing at our leisure for the duration of the project. Many of our design decisions are not specific to picking grapes. The robotic picking abilities may apply to other crops as well such as tree fruits or cannabis.

7.2.2 Weaknesses

Our weaknesses result from lack of business experience and having to share our time with schoolwork.

Besides working in businesses and taking management, marketing, and finance classes, we bring no real entrepreneurial experience to our project. It can be easy to assume that because a project is feasible or interesting, that there is demand for it. Until we have feedback from our stakeholders in the wine industry, we cannot determine value or demand. We will need to provide a minimum viable product to garner this feedback. Also, due to our lack of internal marketing and advertising specialists, it would be difficult to promote this idea and gain attention in our targeted industry, without external help.

Besides managing the capstone project, our time must be shared with our schoolwork. The best we can do to mitigate this, is incorporating as much capstone research and work as we can into other class projects.

7.2.3 Opportunities

Some of our opportunities include growing markets, growing demand, and lack of precision solutions.

Possible market opportunities include Canada's growing agricultural industry, and wine industry. As demand for local labour continues to grow, solutions will be sought out for making the task of farming as efficient as possible. VineBot can be easily adapted to work in different agricultural fields, which allows further expansion after a working proof of concept. Projected labour shortages also provide a possible opportunity for VineBot to be adopted. Since agriculture will continue to grow, and the agricultural labour force may not be able to meet demands, automating farming tasks may help lighten the load, so to speak.

In terms of staying competitive, technological advances in agriculture may not be extremely valuable now, however, it is a possibility soon. Being a pioneer in an agricultural shift towards autonomous technology driven farming could give British Columbia, and even Canada, a competitive edge over regions known for agricultural production, like California, Europe and Japan.

There are currently no precision grape picking solutions on the market. Current work is done by imprecise and violent mass-harvesting machines that are too expensive for small to medium sized enterprises.

7.2.4 Threats

Our main threats include the speed at which current solutions can harvest grapes, and the possibility of drought in the future.

Despite the damage to the grapes involved in current mechanized picking solutions, they are extremely fast and have begun to receive some approval from even the harshest vineyard purists when necessitated.

Some analysts worry that the increase in farmable land brought by climate change in Canada, could be offset by water shortages induced by drought.

7.3 Appendix C: Milestone Meetings

Below are meeting summaries and deliverables for the three milestone meetings we planned. Note, the decided not to have the originally planned five milestone meetings due to a lack of time.

7.3.1 Meeting 1

Summary

This report describes the projected initial milestones and our completion status for those milestones with respect to an autonomous wine crop harvesting robot, VineBot. It is addressed towards the staff of the BCIT Electrical Engineering faculty, the BCIT School of Energy and our project mentor, Craig Hennessey. Wine grape harvesting can be a rigorous, time-consuming, and labour-intensive task which has already begun to be automated. There is also a social push towards organic and sustainably produced products and produce, which could be an agricultural automation opportunity. Our senior engineering capstone group aims to improve on current systems by developing a robotic vehicle that can monitor and harvest grape crops autonomously and selectively. Ideally, this robotic vehicle should be able to navigate a vineyard without the need for human intervention, work during all times of the day, and successfully detect and harvest wine grape crops. This is going to be accomplished by using cutting-edge vision technologies, robotics, and various engineering methods we've picked up at BCIT. This report will outline what we hoped to accomplish in terms of some initial milestone goals, and how we carried out the proposition. We will not however go too-far in depth into technical aspects of the project. In the following sections of this report, a summary of our project status will be introduced, including a meeting summary for the milestone meeting that took place on Tuesday, January 29th, 2019. A list of planned and actual deliverables will then be discussed for the present milestone, leading into some key findings that were encountered during the milestone, and a section outline the necessary changes that should be made to the project plan. Finally, a conclusion and expectations for the next milestone will be discussed.

List of Planned and Actual Deliverables for the Milestone

At this point in time we would have liked to have moved past basic OpenCV processing and have a trained machine learning model doing the cluster detection. The vehicle should have also been able to sample its surroundings with the LiDAR and know its position relative to the rows.

Due to our past work during the last term and our hard work this term, we were able to accomplish most of the planned deliverables. Shown below in Table 16 are some of the deliverables were planned on accomplishing in comparison to what was delivered during the milestone meeting.

Table 15 – Summary of planned and actual accomplishments

Planned Deliverable	**Actually Delivered**
Vision: Machine Learning	Vision: Machine Learning
Vision: Logitech Webcam	Vision: Logitech Webcam: Webcam OpenCV with YOLO
Pose Estimation and Mapping: LiDAR Point Processing	Pose Estimation and Mapping: LiDAR Point Processing
Motion Control: uArm Swift Pro	2D Feedback Control Using Reference Signal

7.3.2 Meeting 2

Summary

On Thursday, April 25th, 2019, our second milestone meeting was conducted. The meeting, held at 11:30am was attended by both Jeff Bloemink and Craig Hennessey, as well as the VineBot capstone team members. During the first part of the meeting, work involving the machine vision and robotic picking mechanism was brought forward. There was some trouble with the robotic picking demonstration, due to our ongoing work on the picking algorithm, however we were able to demonstrate the cluster detection and transformation between the 2D and 3D cameras. Craig Hennessey, after viewing the picking demonstration and noticing a snagging problem, suggested that the robotic picking algorithm should first center the grape clusters in the frame, and then move in to pick the cluster. Brayden was not able to demonstrate his work on the LiDAR SLAM aspect of the project visually, however, was able to briefly describe what he was working on.

List of Planned and Actual Deliverables for the Milestone

Shown below in Table 16 are some of the deliverables were planned on accomplishing in comparison to what was delivered during the second milestone meeting.

Table 16 – Summary of planned and actual accomplishments

Planned Deliverable	Actually Delivered
Vision system based on 2D colour camera and depth camera	Vision system based on 2D colour camera and depth camera
Integration of NVIDIA Jetson image processing board	Integration of NVIDIA Jetson image processing board
Robot arm 2D and depth tracking	Robot arm 2D and depth tracking
LIDAR point processing, SLAM, and path planning	LiDAR point processing
Full system integration	Integration of cameras, robot arm, image processing board, main control board.
Mechanical design of camera mount	6th generation of camera mount complete

7.3.3 Meeting 3

Summary

On Thursday, May 7th, 2019, our third and last milestone meeting was conducted. The meeting, held at 11:30am was attended by Craig Hennessey and VineBot capstone team members, with our capstone coordinator Jeff Bloemink absent. During the first part of the meeting, work involving the stem cutter was introduced, including a preliminary version of the stem cutter that wasn't very good, and then followed by the current version of the stem cutter which was a significant improvement. A short demonstration of the robotic picking and stem cutting mechanism was brought forward. We also briefly discussed how we created some bio-degradable stems. Craig Hennessey, after viewing the cutting demonstration commented that although he was impressed with the cutting mechanism, he would most likely stay away from using sharp blades as a part of a course project. Brayden was able to demonstrate his work on the LiDAR SLAM aspect of the project visually and was able to talk about the PID control system he implemented for the drive wheels.