University of Arkansas – CSCE Department

Capstone I – Final Proposal – Fall 2021

NASA Robotic Mining Competition with Arkansas Razorbotz

Nicholas Beck, Cade Courtney, Levi Davis, Michael Ebbs, AnElizabeth Henry, Tristen Teague

Abstract

For our Capstone project we will be teaming up with Arkansas Razorbotz to compete in the NASA Robotic Mining Competition "Lunabotics Challenge." The objective of our project is to work with the other students on the Arkansas Razorbotz team to design and implement a lunar mining robot to compete in this competition. Overall, the scope of the project that has been given to our capstone team is to evaluate the computer science and computer engineering aspects that go into our robot design. Between now and May 25, 2022, our team will evaluate existing code from the previous years' robot and refactor recycled code so that it is more organized, runs more efficiently, is designed more specifically for this year's competition, and so that it can be recycled for future robotic competitions that the Arkansas Razorbotz compete in. The goal of both our capstone team and the Arkansas Razorbotz team is to produce a robot which will win the "Lunabotics Challenge" hosted by NASAS in May of 2022. Although the significance of this project plays a large role in allowing everyone on our capstone team to demonstrate the knowledge that has been afforded to them through the university, the bigger significance of this project lies in the capability of the robot that we produce. As many reputable schools will be taking place in this competition, we will see how precise engineering design and robotic autonomy cam impact the efficiency of lunar mining robots and their ability to mine rare/expensive earth materials from the moon. This in turn could create a way for us to mine these materials while cutting down on the environmental impact created by mining these materials on earth.

1.0 Problem

NASA has put forth their intent on completing what is known as the Artemis Mission to the moon. As a part of their mission, they plan to land the first woman on the moon by 2024. This has recently been delayed to 2025 due to the Coronavirus outbreak. The intent of the plan is to further develop our exploration of space discoveries while preparing for humanity's next "giant leap", the mission to Mars. In the preparation for that step, however, NASA looks to establish the presence of humanity and infrastructures on the moon to support lunar surface exploration as their priority. It is essential for the necessary resources and infrastructure to be in place on the moon in order to properly support arrival and sustainability. To do this will require an extremely large number of resources. For example, the commercial cost of delivering payloads to the moon is about $1.2 Million per kilogram [3]. As each mission to send technology into space comes at a high cost, it is imperative that each piece of technology NASA sends is extremely efficient and well developed for the purpose it will be serving.

Lunar rovers are a notable example of a piece of technology that NASA will need to spend an extreme amount of time developing before they send it into space. The rover can be used for a large variety of functions on the moon, therefore making it an invaluable piece of equipment for astronauts to use. But as there is a limited number of resources available on the moon (food, water, oxygen for example), astronauts have a limited period of time to spend on doing tasks and gathering research. Thus, the need for the feature of autonomy for lunar rovers comes into play. The lunar rover will be subject to extreme environments that are different than that of earth, but in order to prove useful and to receive every benefit of the cost that NASA spent on getting it to the moon, the rover must also prove to be reliable over an extended period of time. The challenge of maximizing efficiency of every component of a lunar rover is a difficult one to say the least. So, as a part of the Artemis project, the Artemis Student Challenge was created as a competition for college-level teams to develop a robot that could be capable of excavating lunar surfaces and mining resources to be transported back to earth. The challenge in this competition, however, lies in the restrictions that each team must follow when developing their robot. For example, limitations for the robot's weight, size, and method of operation have been set outright by NASA in the competition handbook. In the end, the goal for each team is to have developed what they think is an efficient version of a “lunar rover” that can operate in the harsh and unique environment of the moon while autonomously and reliably completing mundane tasks such as navigating and mining.

In this competition, our capstone team plays an integral role as a part of the Computer Systems sub-team of the Razorbotz team at the University of Arkansas. We will be responsible for delivering the code and computer interfaces involved with the functionality of our robot, including the ability of our robot to work autonomously.

2.0 Objective

The objective of this project is to design a prototype robot that will master the complexity of maneuvering through rough terrain while mining the regolith simulants created by NASA in the competition. The robot will receive two attempts at maneuvering through the terrain to collect the maximum amount of regolith within the 15-minute time limit of the competition while operating remotely or through autonomous operations using ROS2 (Robot Operating System). The robot must adhere to the size and weight limitations put forth by NASA in the 2022 Lunabotics Challenge Guidebook. The limitations of our design are included below in the “Design” subsection entitled “Requirements and Design goals.” Here are a few examples of weight and size limitations for the robot as well as minimum expectations that the robot must meet in the competition:

- The design of the lunar rover must satisfy the smaller undeployed volume of 1.1 m length x 0.6 m width x 0.6 m height.

- The lunar rover must weigh less than the maximum mass of 80 kg.
- The rover must have a “KILL SWITCH,” which is a red emergency stop button that will shut down the lunar rover’s electrical systems.
- A minimum amount of 1.0 kg of gravel must be mined and deposited during either of the two competition attempts.
- Subsystems used to transmit commands / data and video to the telerobotic operators are counted toward the mass limit. Equipment not on the robot used to receive data from and send commands to the robot for telerobotic operations is excluded from the mass limit.
- Robots may deploy or expand beyond the envelop after the start of each competition attempt but may not exceed a 1.5 m in height.
- Multiple robot systems are allowed, but the total mass and starting dimensions of the entire system must comply with the volumetric dimensions given in this rule.
- The robot must provide its own onboard power. No facility power will be provided to the robot during the competition runs. There are no power limitations except that the robot must be self-powered and included in the maximum mass limit.
- The robot cannot employ any fundamental physical processes, gases, fluids, or consumables that would not work in an off-world environment.
- Components (i.e., electronic and mechanical) are not required to be space qualified for Lunar or atmospheric, electromagnetic, and thermal environments.

The Razorbotz team is constructed of multiple sub-teams that contribute work to specific parts of the robot. Our sub-team, the Computer Systems sub-team, is responsible for designing the remote controls and autonomy of the robot. During the competition, as stated above, the robot may operate in one of two ways: remotely or autonomously. Additional points may be earned by the team for implementing autonomous operation within the robot. Examples of all autonomous features that could earn additional points in the competition are listed below:

- Excavation Automation
- Dump Automation
- Travel Automation
- Full Autonomy (One Cycle) - Successful completion of one cycle of Excavation, Dump, and Travel
- Full Autonomy - Robot is in hands free control for the entire 15-minute period of the competition and has completed two or more cycles of Excavating, Dumping, and Traveling

Secondary objectives within the project include but are not limited to: Gaining experience in hardware and software design, learning more about the process of autonomy and different architectures of machine learning, working with a diverse and multitalented team of individuals, and further developing our professional skills to take with us on our future career paths.

3.0 Background

3.1 Key Concepts

This year, autonomy and refactoring are the biggest goals of the Computer Systems sub-team. Along with this come many key technologies that will help us along the way.

The robot will be using ROS (Robot Operating System) to create a real time operating system primarily using C++ and Python. Ros 2 uses a simple publisher and subscriber architecture. Publisher nodes are responsible for sending messages that subscriber nodes will pick up and read if it is relevant to them. In the case of the robot, all of the motors have publisher methods that send real time information about the motor to subscriber nodes that need the information. Most of the subscriber and publisher nodes are already implemented in the robot's code, we will focus on polishing the existing code and changing nodes if new components are added in the future.

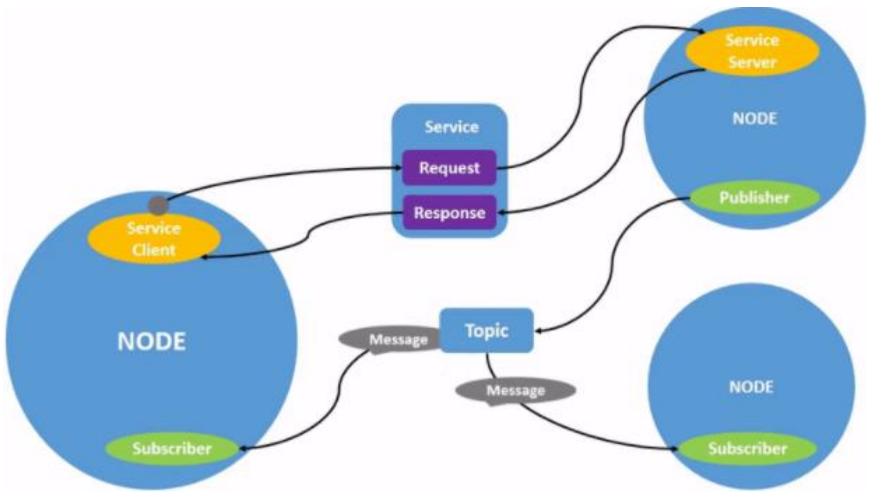

Figure 1 from Understanding ROS 2 nodes," Open Robotics, 2021 [7]

We will be adapting preexisting code to increase the autonomy of our navigation systems so that the lunar bot can navigate around obstacles and identify various objects. The obstacles and objectives will be detected automatically with a ZED camera mounted to the robot. We will use a large set of images to train the robot to detect the obstacles automatically. Paths will be automatically generated after obstacles are detected by passing sensor input to the computer vision model. The computer vision model will be created with PyTorch[4] along with the ZED SDK and some extra python.

3.2 Related Work

Luckily, we have several years' worth of attempts from other students to look back on and improve upon [5]. Previous years, however, did not have to focus on autonomy. [6] Their primary goal (until 2020) was excavation and navigation. Now we have to refactor the old code in ROS2, document everything, and expand on it to implement autonomous navigation and excavation. All the requirements for autonomy and excavation are listed in the NASA Lunabotics handbook [3] which provides a baseline of information for us to design around and a complete breakdown of the competition scores.

4.0 Design

4.1 Requirements and Design Goals

Requirements

- The design of the lunar rover must satisfy the smaller undeployed volume of 1.1 m length x 0.6 m width x 0.6 m height.
- The lunar rover must weigh less than the maximum mass of 80 kg.
- The lunar rover cannot employ any fundamental physical processes, gases, fluids, or consumables that would not work in an off-world environment.
- Must be able to communicate with the lunar rover remotely.
- The computer system's telecommunication is required to have a total average bandwidth of no more than 5.0 megabits/second.
- Must be able to control the lunar rover's drive train and excavation motors.
- The lunar rover is required to only excavate in the designated excavation zone.
- The rover must have a "KILL SWITCH," which is a red emergency stop button that will shut down the lunar rover's electrical systems.
- The lunar rover must avoid obstacles throughout the course.
    - At least three randomly placed boulder obstacles of approximately 30 cm to 50 cm.
    - At least two craters of varying depth and width, no wider or deeper than 50 cm.

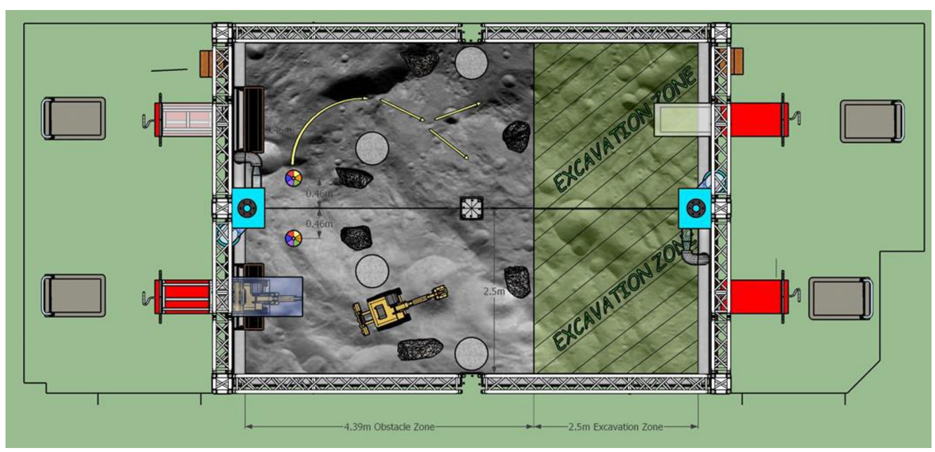

Fig. 2 Mining Arena From "NASA Lunabotics 2022, Registration, Rules and Rubrics," NASA, 2021

Design Goals

- Update current lunar rover functionality to the newest ROS2 distribution.
- The lunar rover will be fully autonomous.
    - Autonomous arena navigation
        - The rover must be able to localize objects to detect obstructions in the rover's path.
        - The rover must be able to detect the excavation zone.

    - Autonomous excavation and unloading
        - The excavation process must start once the rover reaches the excavation zone.
- Coding and documentation standards
    - The rover will be programmed in the latest stable release of ROS2, which is the foxy distribution.
    - Object-oriented programming paradigm.
    - Publisher and subscriber programming model will be used for rover communication.
    - We will create new documentation and update old documentation using the tool Doxygen.
    - Documentation structure
        - A brief file description.
        - A detailed file description that includes any topics the node publishes or subscribes.
        - Function descriptions that include parameters and return values and links to the relevant files.

4.2 High Level Architecture

The goal of our team's project is primarily to autonomize the existing capabilities of the Razorbotz mining robot. These capabilities include not only course traversal, but also the excavation and dumping capabilities of the system. As the NASA Robotic competition scoring places quite a heavy weighting upon the autonomy capabilities of the robot, implementing a system that is fully or partially autonomous is very important overall.

Secondarily, the refactoring of old code from previous years is another objective. Improving the modularity and readability of the code of any systems is extremely valuable to not only the current implementations of the system, but also for the evolution of the system over a given period of time. As this project has spanned many years —and hopefully many more— making sure that personnel newly joining the team have concise and easily readable code is a must.

The most important part is the autonomization of the system with object detection and recognition. In order to traverse the environment, the robot must be able to detect and recognize any obstacles in its path so that it can take appropriate action—whether avoidance, interaction, or some other process. Similarly, the system must be able to detect and recognize the appropriate locations to excavate or dump the excavated materials.

To make the transition from entirely human controlled action to semi or fully autonomous capabilities, the system will implement a ZED stereo camera and YOLO (you only look once) object recognition to distinguish objects in the competition arena so that it can take the appropriate action. PyTorch will also be used to train the YOLO object recognition system to detect objects and locations in the arena.

The ZED stereo camera allows for determining three dimensional locations of objects from digital images via two forward facing cameras. The two forward facing cameras attempt to mimic the way humans perceive depth via stereo disparity—the difference in image location of an object seen in one eye, compared to its location as seen by the other. Using this approach, the stereo camera can triangulate a pixel's location in a digital image by comparing the location of the same pixel in one camera with its location in the other camera, and compute that pixel's—or collection of pixels— location in a 3D space (fig. 1). However, detecting an object and recognizing an object are two different endeavors. This is where the YOLO object recognition process comes into play.

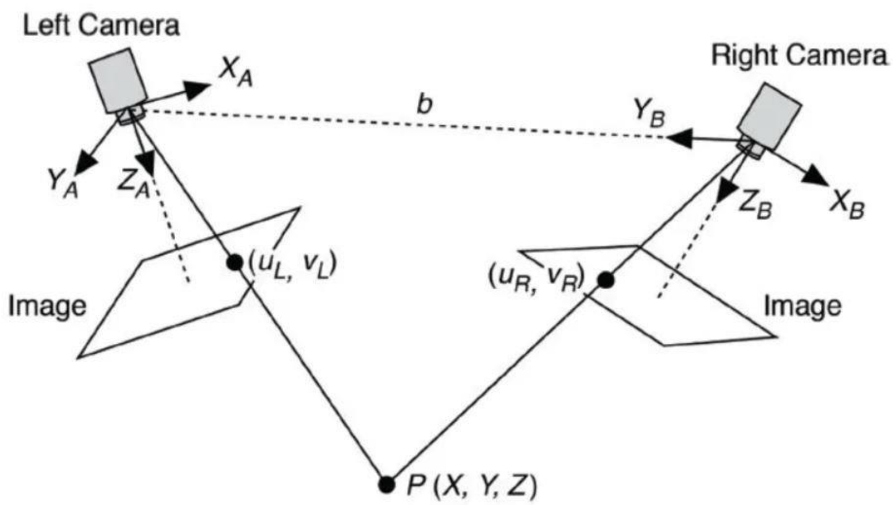

Fig. 3 Stereo Vision Model. From Nair, D. (2012, October 1). *A Guide to Stereovision and 3D Imaging*. Tech Briefs.

YOLO object recognition is an extremely fast vision detection algorithm that is readily becoming the standard for object detection in computer vision. It uses object localization to place a bounding box around an object—or objects— inside a digital image, and then outputs a matrix of values that represent that object's location in 3D space relative to the camera. The variables inside the matrix represent whether there is an object detected, and if so, the height, width, and center of the bounding box, and the object's classification (fig. 2). This algorithm can also distinguish between multiple objects in the same image using intersection over union (IOU). The IOU process assigns each bounding box with an object inside a probability score. This probability is then compared to the probabilities of the other bounding boxes around the same object and then determines which bounding box most accurately represents the object.

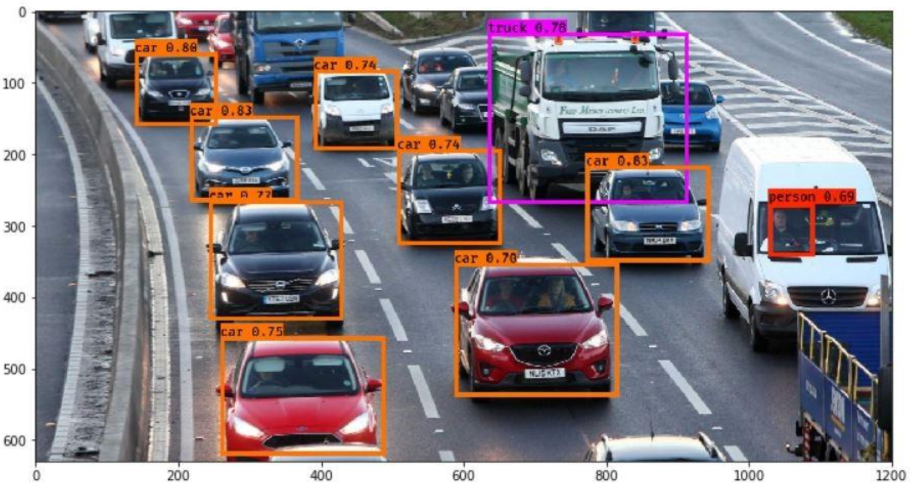

Fig. 4 YOLO Object Recognition Bounding Boxes with Probability Scores. From *YOLO framework: Object detection using yolo.* Analytics Vidhya. (2021, August 26).

The dynamic deep learning framework PyTorch will be used to train YOLO to determine the classification of an object e.g., distinguishing a bicycle from a car, or in the case of this application, recognizing a rock on the planet's surface. PyTorch allows you to import image files of each classification of objects you want the system to recognize, and then uses neural networks to establish weighting from the dataset given. Through this process, YOLO will be able to recognize a rock on the planet's surface, so that it can take appropriate action to navigate around it.

Once the system can detect and recognize objects in the arena field, software algorithms will be implemented via ROS2—a set of libraries /tools specific to robotics applications—to determine and execute the appropriate action for the robot to take. If the system detects an obstacle in its path, then it will be able to autonomously navigate around the object and proceed to its objective. Both the excavation and the delivery of the excavated material to the appropriate unloading site will also be autonomized. The system will also be able to detect any hardware or equipment failures and take the appropriate course of action dependent upon the type of failure.

We are working in collaboration with several other sub-teams with different specialties. Excavation and dumping will be designed and implemented by the Excavation team comprised mostly of Mechanical engineering students. Path planning and automation will largely rely on the specific tools (motor controllers, motors, detection tools etc.) that the Chassis team implements next semester when parts are ordered and eventually arrive.

4.3 Risks

Risk	Risk Reduction
Autonomy system failure	Added the "KILL SWITCH" to stop all electrical components. Train the robots computer vision thoroughly to move through a variety of settings.

Communication between differing fields	Compartmentalized different sections of the robot. Each group has a team leader who handles communication and coordination.
Groups at different stages in the design process	This year two rovers are available to the Razorbot team. This will increase productivity.
ROS2 documentation not as thorough as ROS1	Documentation will be a major part of the computer systems team. The team will be creating foundational documentation for future Razorbotz competitions.

4.4 Tasks

- Refactor previously implemented code
    - Learn ROS2 Foxy distribution
        - Older robot used ROS1
    - Document all previously made code
        - Interactions between the motor and the motor controller
        - Interactions between the joystick and the motor controller
        - Communication between the computer and the robot
        - Interactions between the CAN bus and the computer
    - Adjust existing code to meet new requirements
        - Excavation requirements changed, meaning a new excavation device is needed
        - Change current excavation code to operate new the excavation tool
    - Improving existing code
        - Replace all usage of ROS1 to ROS2 commands
        - Replacing all code relating to the old Talon motors to the new Rev motors
        - Change code for the power distribution panel to allow reading of new devices such as the Rev motors
        - Change existing Zed camera SDK to upgrade to Zed 2 SDK
- Improve UI
    - Existing UI attempts to display the interaction between the robot and a computer tied to it
    - UI is currently broken due to new ROS platform and new parts such as the Rev motors
    - Allow those without deep knowledge of computer science to operate the robot either manually or with full autonomy
- Implementing manual control
    - With new distribution, must be able to manually control robot again
    - Allow movement of robot by using a joystick to control the wheel motors
    - Allow mining within the robot by using a joystick to start the mining process
    - Allow manual dumping of rocks by using a joystick
- Implement full autonomy
    - Traveling autonomy
        - Implementing object recognition with Zed camera

        - Allows the robot to both recognize and maneuver around obstacles
      - Make robot travel between mining sites and dump sites without human intervention
  - Excavation autonomy
    - Implementing an algorithm that works with the new excavation tool
    - Excavation tool must be able to:
      - Mine into the crater
      - Pick up rocks (similar to moon rocks) from mining
      - Store the rocks in a temporary container that will be emptied at dump sites
    - Allow excavation tool to mine, pickup, and store without human intervention
  - Dumping Autonomy
    - Allow dumping of temporary container containing rocks into the dump site
    - Work with Zed camera to locate the dump site with object recognition
    - Make robot do the dumping process without human input
  - Failure Management
    - Detecting failures that could happen within the system such as:
      - Robot being stuck on an obstacle
      - Excavation tool failing to mine/pick up rocks
      - Taking incorrect path to either mining or dumping site
      - Possible overheating of components such as a motor
    - Implementing fail-safes to get around such failures
- Compete in the NASA RMC Competition

4.5 Schedule

Tasks	Start Dates	End Dates
1. Download ROS2 and Review the ROS2 tutorials	11/04/2021	11/25/2021
2. Create/Update Documentation to ensure it is structured with well detailed comments and information as needed.	11/25/2021	12/18/2021
3. Make a more User-Friendly UI Design.	11/25/2021	12/18/2021
4. Update the existing code so that there are simpler functions and or simpler callbacks.	11/25/2021	12/28/2021
5. Work on the camera functionality and data image sets	01/18/2022	02/20/2022
6. Implement travel autonomy to recognize rocks and hazards	01/18/2022	03/01/2022
7. Implement the excavation autonomy	01/18/2022	03/01/2022
8. Implement the dump autonomy to recognize the dump site.	01/18/2022	03/01/2022
9. Test the camera functions.	02/20/2022	04/10/2022
10. Test the travel, excavation, and dump autonomy.	03/01/2022	04/10/2022
11. Implement failure management.	04/01/2022	04/24/2022
12. Perform final tests on the robot.	04/24/2022	04/30/2022
13. Competition in the NASA RMC Competition	05/25/2022	

4.6 Deliverables

- Design Document: For the design documentation, we will submit a separate report explaining how the team decided to go about with the design of the software for the robot. We will make sure to involve well descripted sections that will explain our thought process and some charts and diagrams to help the readers understand. This will be helpful for next year's coders who are working to improve the robot with our code.
- Older Code Documentation: For this project, the past coders left little to no documentation and the code was messy and extremely hard to understand quickly. So therefore, our team will go through and update or add information to make the code more readable for other users in the future. We will also be making sure to update the code so that we can implement good coding habits we have learned over the years. When making these changes we will also be adding detailed information for future coders.
- Project Website: With the final report the team will also have a project website that will stay up to date with the task at which we are waiting to complete or have completed. It will hold all the information of the project once the team has completed it. The website will have a link to the repository and the final report.
- ROS2: The ROS2 will be used for the autonomy portion of the robot. We will use this for excavation, navigation and any other movement or action that the robot will need to perform at competition. These functions will be implemented by software that will have many different files involved to implement the autonomy of the robot.
- Robot Testing Data: The robot will use data images that will help them to identify when a rock is in its path of travel. This will make the robot be able to recognize the rock and move the robot out of the way without having a human physically tell the robot to move.
- Final Report: Once the team has now completed the project, we will write a final report that will explain any steps that were taken in the code. It will also explain any setbacks or any challenges the team faces during the process.

5.0 Key Personnel

Nicholas Beck – Beck is a Senior Computer Science major in the Computer Science and Computer Engineering Department at the University of Arkansas. This is the first year he will be on the Arkansas Razorbotz team, but he has experience in researching machine learning with Professor Justin Zhan. Beck will be responsible for developing the autonomy of the robot's computer system.

Michael Ebbs –Ebbs is a Senior Computer Engineering major in the Computer Science and Computer Engineering Department at the University of Arkansas. This will be his first year on the Razorbotz team. He has completed courses in Embedded Systems/System Synthesis programmed in C/C++/Java/Python and others. He will be responsible for refactoring preexisting systems and developing new autonomous systems.

Cade Courtney- Courtney is a senior Computer Science major in the Computer Science and Computer Engineering Department at the University of Arkansas. Throughout his academic career, he has programmed in multiple languages, C++/Java/Python. For the last three Summers, he interned at Globitech incorporated, where he learned how to work in a team environment and produced code at the production level. He will be working on the rover's autonomy.

AnElizabeth Henry – Henry is a senior Computer Engineering major in the Computer Science and Computer Engineering Department at the University of Arkansas. This will be her first year being a part of the Arkansas Razorbotz team. She has worked for ClickClaims in New Orleans Louisiana for a summer as a software developer intern. She will be responsible for the development of the robot's autonomy.

Levi Davis – L. Davis is a senior Computer Engineering major in the Computer Science and Computer Engineering Department at the University of Arkansas. He has worn many different hats in his lifetime, with responsibilities ranging from CAD engineer for government aircraft parts to installing antennas on broadcast towers. After deciding that he should challenge himself more, he decided to pursue a computer engineering degree. In collaboration with the CSCE Razorbotz team, he is responsible for the developing of the autonomous functionality of the mining robot.

Tristen Teague –Teague is a senior Computer Engineering major in the Computer Science and Computer Engineering Department at the University of Arkansas. This will be his first year on the Razorbotz team. He is a part of the AESIR Labatory where he performs research on Post-Quantum Cryptography on embedded systems. He has experience of working with embedded systems and firmware engineering. Teague will be responsible for working with travel autonomy.

Dr. Uche Wejinya – Dr. Wejinya is an Associate Professor in the Department of Mechanical Engineering at the University of Arkansas. He performs research in robotics and mechatronics and is the faculty advisor for Arkansas Razorbotz.

6.0 Facilities and Equipment

6.1 Facilities

- Mechanical Engineering Robotics Labatory: Lab within the Mechanical Engineering department where the robot is both assembled and tested
- Testing Location: Location where the robot is tested with conditions similar to the layout of the competition

6.2 Equipment

- Nvidia Jetson Nano(s)
- Vex Talon SRX Motor Controller
- Vex Talon SRX Motor(s)
- Victor SPX Motor Controller
- Vex Victor SRX Motor(s)
- Vex Power Distribution Panel
- SPARK Motor Controller
- NEO Brushless Motor(s)

7.0 References

[1] *YOLO framework: Object detection using yolo*. Analytics Vidhya. (2021, August 26). Retrieved November 4, 2021, url:

https://www.analyticsvidhya.com/blog/2018/12/practical-guide-object-detection-yolo-framewor-python/.

[2] Nair, D. (2012, October 1). *A Guide to Stereovision and 3D Imaging*. Tech Briefs. Retrieved November 4, 2021, url:

https://www.techbriefs.com/tb/topic/photonics-imaging/imaging.

[3] "NASA Lunabotics 2022, Registration, Rules and Rubrics," NASA, 2021, url:

https://www.nasa.gov/sites/default/files/atoms/files/lunabotics_guidebook_2022_-_ver_1_1.pdf

[4] "PyTorch: An Imperative Style, High-Performance Deep Learning Library," Adam Paszke, Sam Gross, Franciso Massa, et.al., 2019, url:

https://papers.nips.cc/paper/2019/file/bdbca288fee7f92f2bfa9f7012727740-Paper.pdf

[5] "Razorbotz CS Intro," William Burroughs, 2020, url:

https://www.youtube.com/watch?v=BSq90vnPwJ4&feature=youtu.be

[6] "RMC 2019 Registration, Rules and Rubrics," NASA, 2018, url: https://www.nasa.gov/sites/default/files/atoms/files/rmc2019_registrationrulesrubrics_09062018

[7] "Understanding ROS 2 nodes," Open Robotics, 2021, url:

https://docs.ros.org/en/foxy/Tutorials/Understanding-ROS2-Nodes.html