CSCE 4561 Capstone II

Spring 2021

Project:

NASA/Robotic Mining Competition Rover

Group Members:

Andrew Burroughs, Calvin Franz, Z. Gunner Lawless,
Jett McCullough, Carson Molder

April 27th, 2021

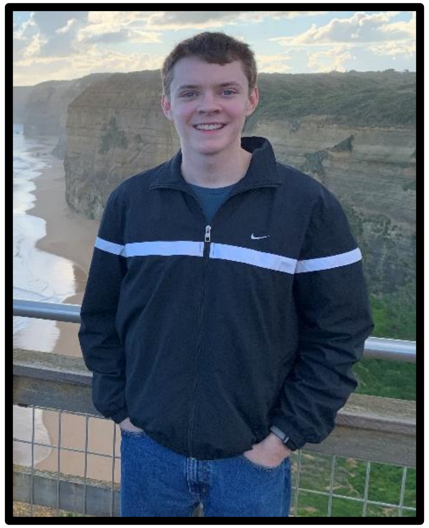
About us

Andrew Burroughs
Comp. Sci.
Senior

Calvin Franz
Comp. Sci.
Senior

Z. Gunner Lawless
Comp. Sci./Eng.
Senior

Jett McCullough
Comp. Sci.
Senior



Carson Molder
Comp. Eng.
Senior

ARKANSAS RAZORBOTZ

UNIVERSITY OF ARKANSAS
College of Engineering
Computer Science
& Computer Engineering

Problem

It costs about $1.2 Million/kg to send materials to the moon (2019 est.)[1]

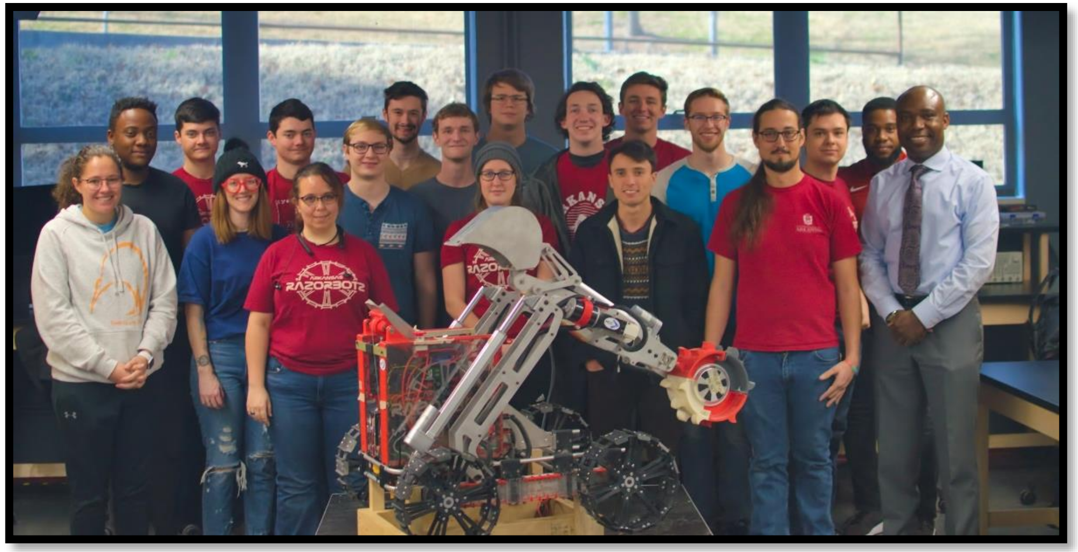
- NASA’s Artemis program:
  - 2024 — Return to Moon
  - 2030s — Visit Mars
- Extended missions are too expensive
- Need to gather and process materials while in space
- We did not have as much experience with robots

Image Source: nasa.gov [1]

Objective

Our goal was to develop the software for a robot that can autonomously mine rocks on the moon.

- NASA Artemis Student Challenge
  - Traditionally at Kennedy Space Center
  - Competition between universities for the best mining robot
- Razorbotz
  - Team of UARK students competing in RMC
  - Led by our project champion, Professor Uche Wejinya



Last Year's Team and Rover

UNIVERSITY OF ARKANSAS

ARKANSAS RAZORBOTZ

UNIVERSITY OF ARKANSAS
College of Engineering
Computer Science & Computer Engineering

UNIVERSITY OF ARKANSAS

Background

- Refactor existing code
- Automation with computer vision
  - Identify objective locations and key targets
  - Path generation/discovery
  - Tools and libraries
    - ZED SDK
    - Git
    - Darknet
- Robot Operating System 2 (ROS2) to program robot
  - Multi-language support (Python, C, etc.)
  - Logical units ("nodes") coded for autonomy, navigation, etc.

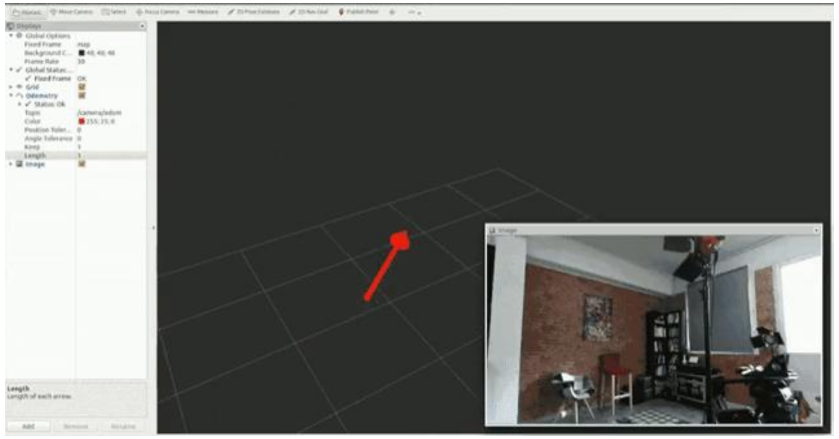


Image Sources: ZED Docs [2], pjreddie.com [3]

ARKANSAS RAZORBOTZ

UNIVERSITY OF ARKANSAS
College of Engineering
Computer Science & Computer Engineering

Facilities and Equipment

Mechanical Engineering Robotics Lab

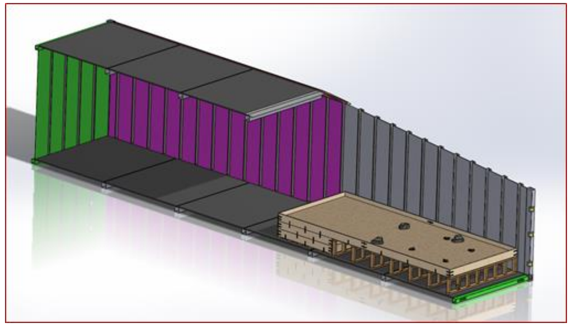
Campus Test Pit

- Contains moon-like rocks and dust

Autonomy / AI development

- Jetson Nano
  - Small embedded system that can run AI workloads for the robot
- Nvidia GeForce RTX 2080 Ti

Campus Test Pit:



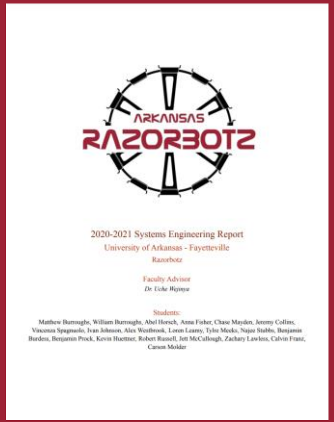
Jetson Nano:

Image Source: nvidia.com [6]

ARKANSAS RAZORBOTZ

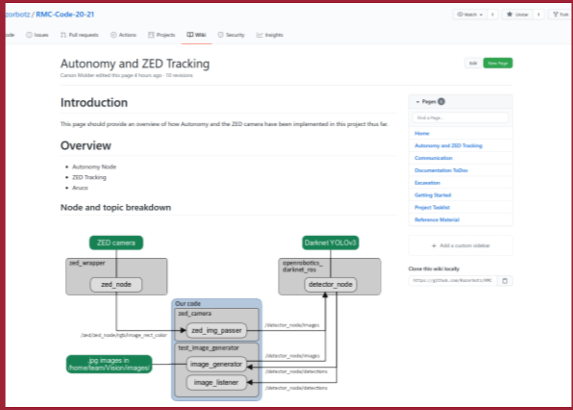
UNIVERSITY OF ARKANSAS
College of Engineering
Computer Science
& Computer Engineering

Deliverables



Design document

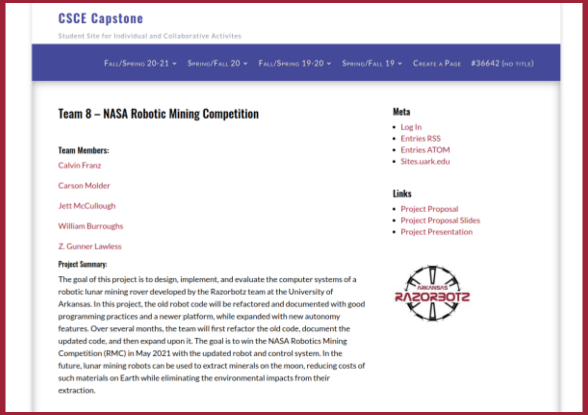
ROS2 nodes



Documentation

Dockerfiles

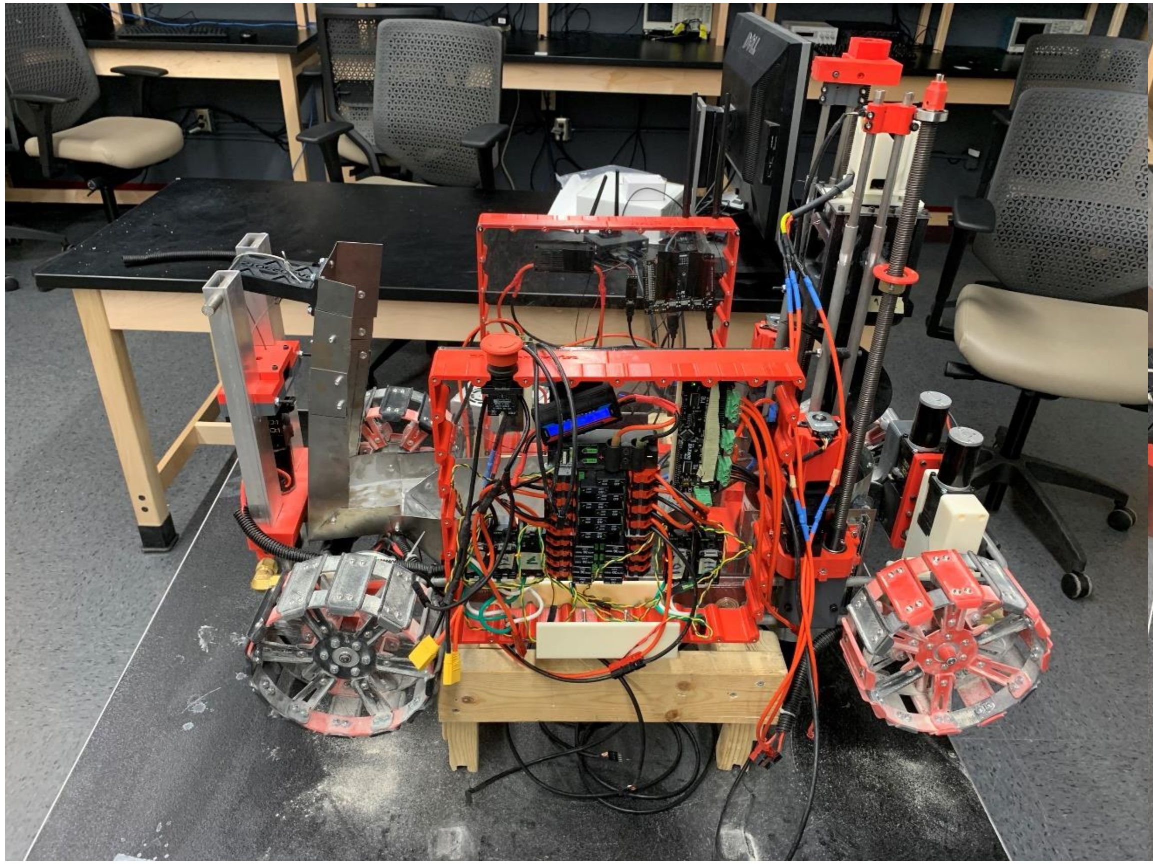
Rock image datasets

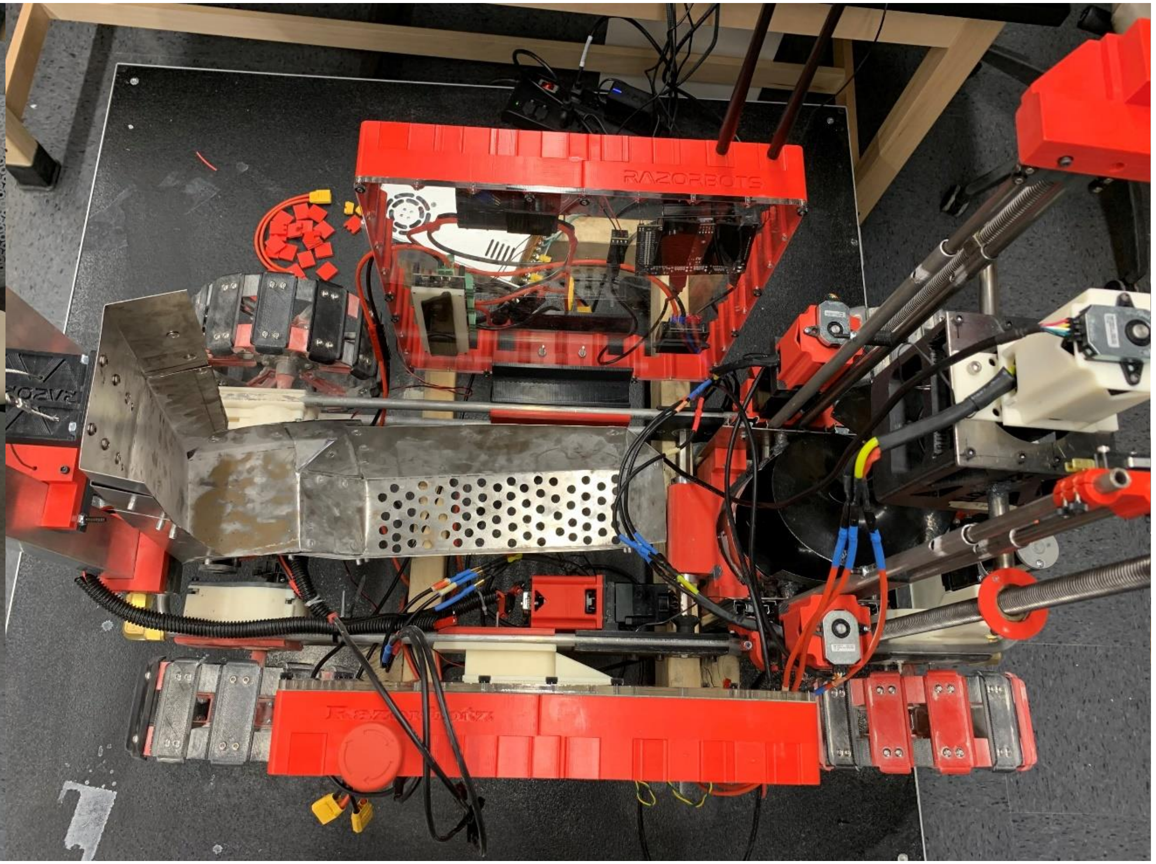


Final report and website

UNIVERSITY OF ARKANSAS

Current State of the Robot





ARKANSAS RAZORBOTZ

UNIVERSITY OF ARKANSAS

College of Engineering
Computer Science
& Computer Engineering

Manual Control Demo

Vision Overview

- ZED Camera
  - Depth Sensing Capabilities
    - Each pixel has an associated depth (Depth Map)
  - All detected objects given depth
- YOLO: “You Only Look Once” [8]
  - Neural network that performs real-time object detection
  - Trained on COCO dataset (DEMO) [9]
    - Over 80 categories
    - book, person, dog, bicycle, airplane, etc.
  - Runs on Darknet
    - Coded in C, more compatible with ROS2 than PyTorch
- Lunar Rock dataset

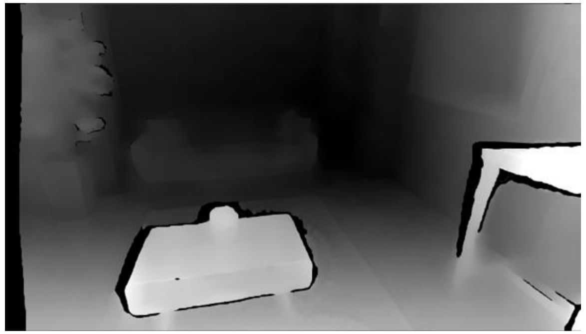


Image Source: stereolabs.com

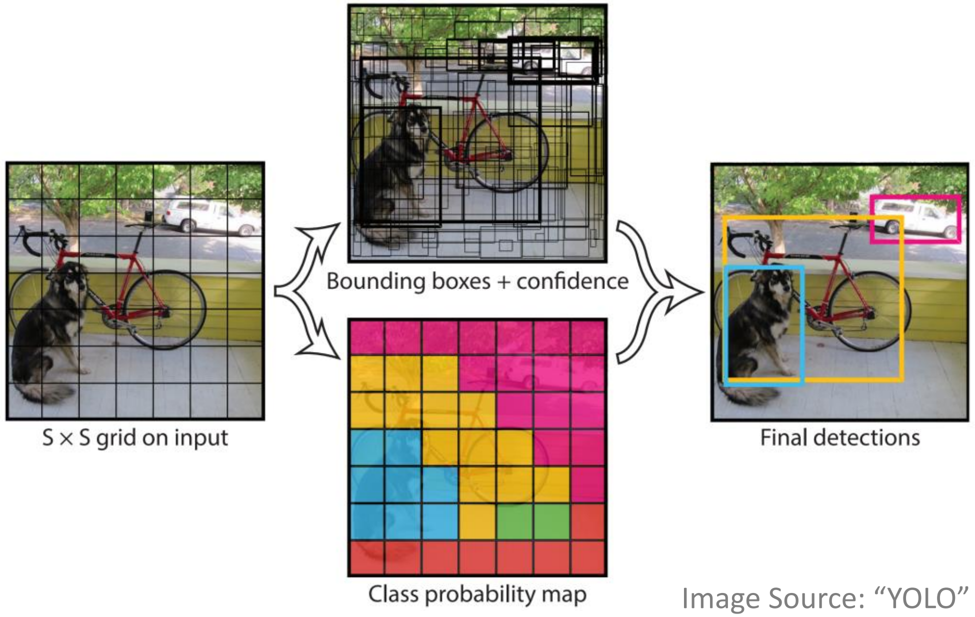


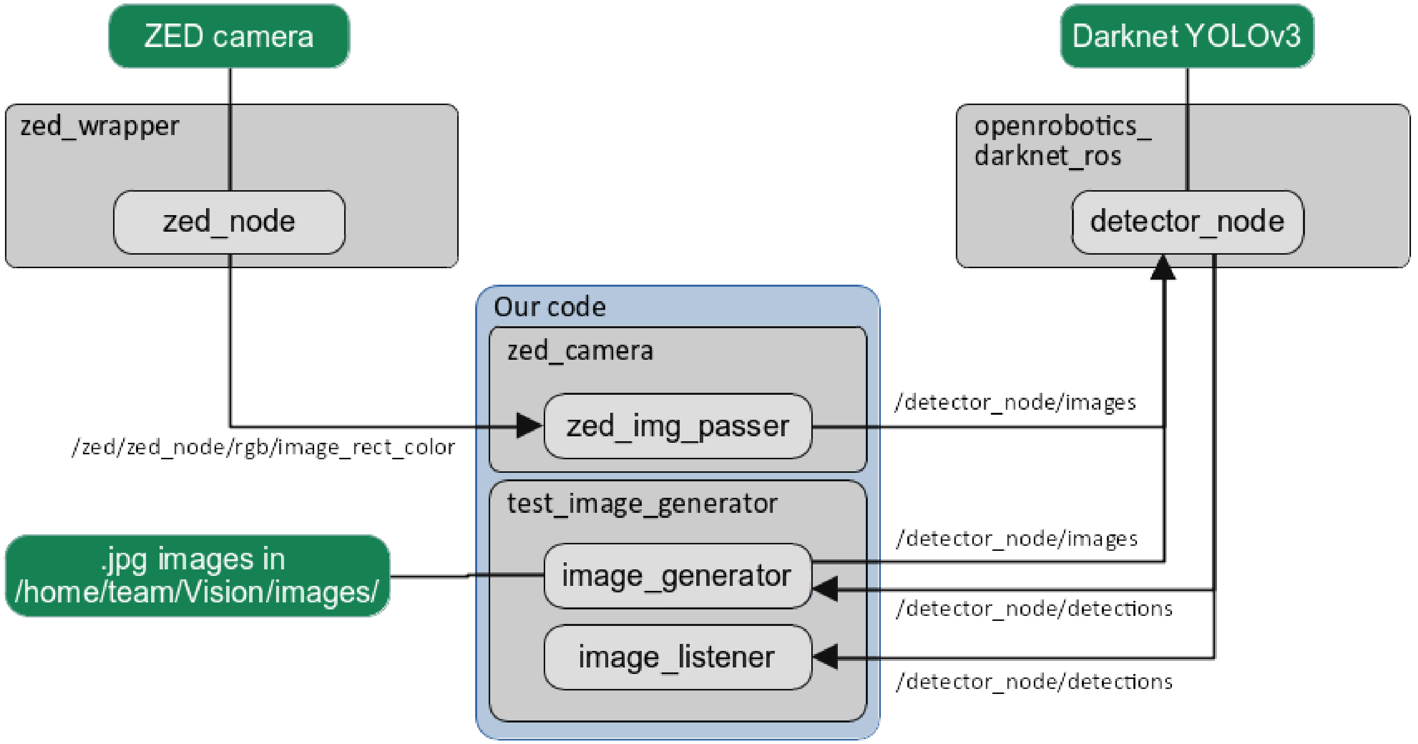
Image Source: “YOLO” [8]

Rock Image Datasets

- Two rock image datasets:
  - Bulk (3700+ images)
  - Small (260 images)
- Images generated from videos taken in testing pit
- Script using OpenCV to extract images
- Manual construction of image labels using labelImg tool [10]
  - Labels in YOLO VOC format
- Datasets are ready-to-use to train Darknet for rock detection

Vision Nodes

- Vision modules
  - `zed_wrapper`:
    - Communicate with ZED camera
  - `openrobotics_darknet_ros`:
    - Communicate with YOLO
  - `*zed_camera`:
    - Passes ZED camera feed to YOLO for object detection
  - `*test_image_generator`:
    - Logs YOLO detections to console
    - Can send a series of test images to check YOLO



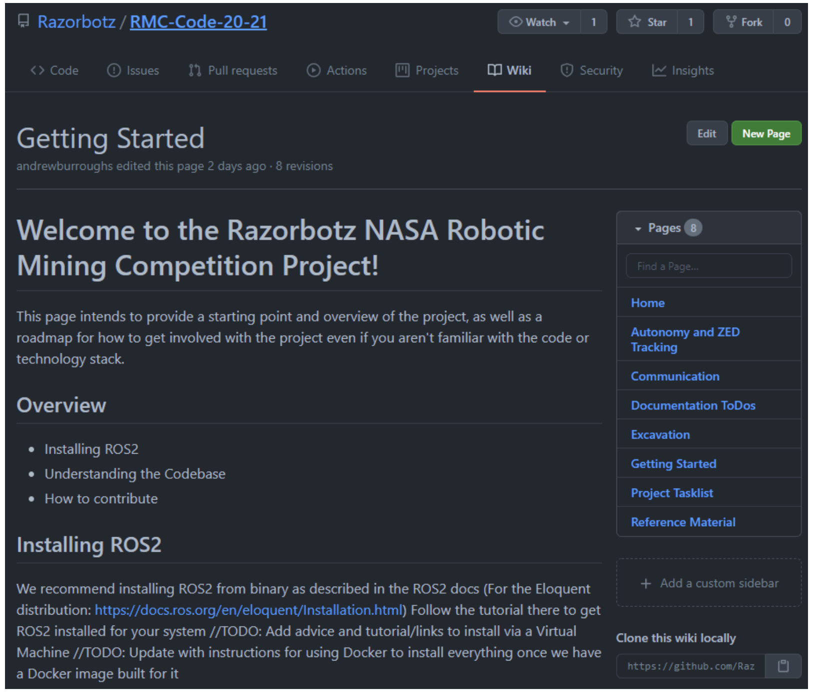
* = new code we contributed

Vision Demo

- Live Demo

Documentation

- Github Wiki
  - New nodes are commented
- "Getting Started" and other guides
    - YOLO vision nodes
    - Communication nodes
    - Excavation nodes
  - Details on how to run nodes
  - Useful references for team members new to programming
- Workflow documents



Dockerfiles

- Steep learning curve to learn ROS2
- ROS2 is difficult to install
  - Lengthy process
  - Dependency issues
- Using Dockerfiles allows:
  - Increased portability
  - Quick install & run time
  - Easier access for non-computer students to learn
- Basic documentation is provided to get other up-to-speed with Docker Images

Future Work

- Full navigation autonomy
  - Use vision data to navigate to key targets
  - Train YOLO on lunar rock dataset
  - Automatically control motors, steering, etc.
- Full excavation autonomy
  - Mining, dumping
- Extending documentation
  - Comment old nodes as they are updated
  - Extend GitHub wiki to include every node
  - Finish guides so they can help familiarize new members with the project
- Complete robot chassis
- Compete in 2022!

References

[1] “NASA Robotic Mining Comptetiton (RMC) Lunabotics 2021, Registration, Rules and Rubrics,” NASA, 2020. url: https://www.nasa.gov/sites/default/files/atoms/files/000_rmc_lunabotics_rules_rubrics_2021.pdf

[2] “Getting Started with ROS and ZED,” Stereo Labs, 2020. url: https://www.stereolabs.com/docs/ros/

[3] Darknet Homepage, Darknet, 2021. url: https://pjreddie.com/darknet/

[4] "ROS2 Github Repository", Github, 2020. url: https://github.com/ros2

[6] “Jetson Nano Developer Kit,” Nvidia, n.d. url: https://developer.nvidia.com/embedded/jetson-nano-developer-kit

[7] "Sterolabs Homepage", Stereolabs, 2020. url: https://stereolabs.com

[8] “You Only Look Once: Unified, Real-Time Object Detection,” Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. *IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 779-788.

[9] “Microsoft COCO: Common Objects in Context,” Lin, T., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollar, P., and Zitnick, C.L. *European Conference on Computer Vision*. 2014, pp. 740-755.

[10] "labelImg Github Repository", Github, 2021. url: https://github.com/tzutalin/labelImg