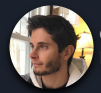
Autonomous Search With AI

Group 28



Current Speaker:
Patrick Bauer

Group Members & Roles

Patrick Bauer

Team Lead

Simulation Environment

Navigation

Pablo Trivino

Object Detection

Robot Sensors

Mark Pedroso

Object Detection

AI Training

Noah Avizemer

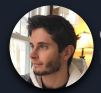
Simulation Environment

Mapping

Nathanael Cassagnol

Navigation

Sponsor:
Joseph Rivera



Current Speaker:
Patrick Bauer

Project Objective

Simulate a fully autonomous robot that searches for a target in an urban environment using AI.

Current Speaker:
Patrick Bauer

Requirements - Simulation

- Simulated urban environment (400ft x 400ft)
  - Several streets with buildings, trees, cars and other obstacles.
- Fully autonomous navigation
  - After the simulation starts, no human assistance.
- GPS denied environment (no global coordinates)
- Must notify user when target is found and simulation is complete.

Current Speaker:
Noah Avizemer

Requirements - Object Detection

- Bounding box shall accurately surround target
- Must accurately identify the target
  - > 0.7 confidence.
- Model predictions shall be correct at least 97% of the time.
- Model shall make predictions at least 2 times per second.

Current Speaker:
Mark Pedroso


bb8

Implementation

- Software
  - ROS Noetic
  - Gazebo 9
- Algorithms
  - SLAM
  - YOLO

Current Speaker:
Patrick Bauer

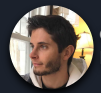
ROS

- Open-source robotics operating system
- Consists of infrastructure, tools, and capabilities for programming robots
- Industry standard for programming robot behavior
- Capability to implement packages and algorithms such as YOLO and SLAM

Current Speaker:
Patrick Bauer

Gazebo

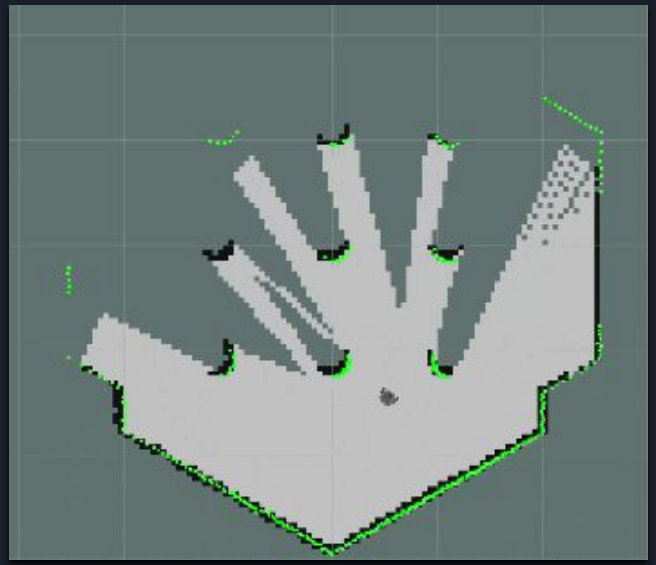
- Open-source robotics simulation software
- Simulate robotic vehicles in various 3D environments
- Supports various sensors and controls for the robot that are necessary for the simulation
- Easy integration with ROS



Current Speaker:
Patrick Bauer

Simultaneous Localization and Mapping (SLAM)

- Allows for an agent to navigate through an environment, avoid obstacles, and generate a map using LIDAR sensors.
- Uses occupancy grid-based mapping



Current Speaker:
Noah Avizemer

You Only Look Once (YOLO)

- Object detection algorithm
- Great precision/fps ratio
- It is a single shot detector
  - Useful for real-time object detection
- Training can be done on standard GPU
- Detects one object per cell
- Can generalize objects well

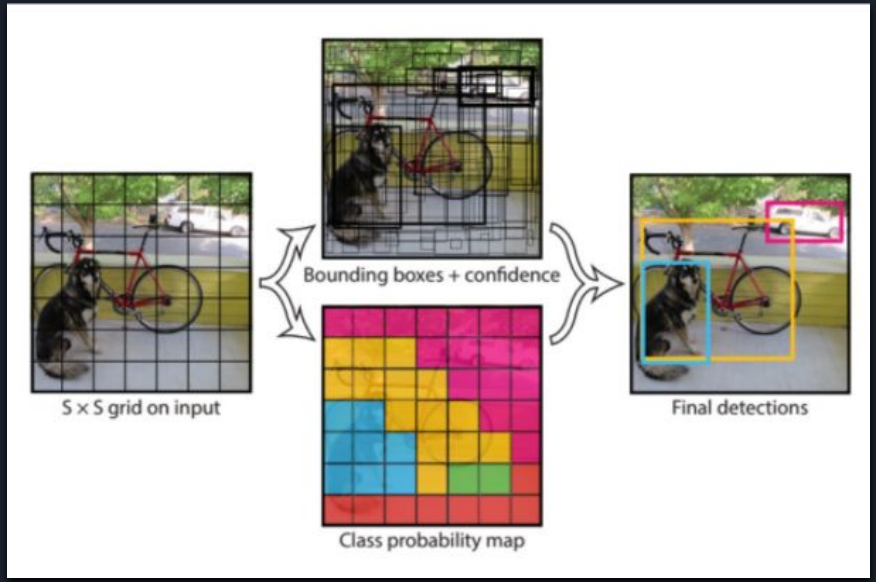


Figure 1. Redmon, J. (2018)

Current Speaker:
Mark Pedroso

Object Detection Consideration

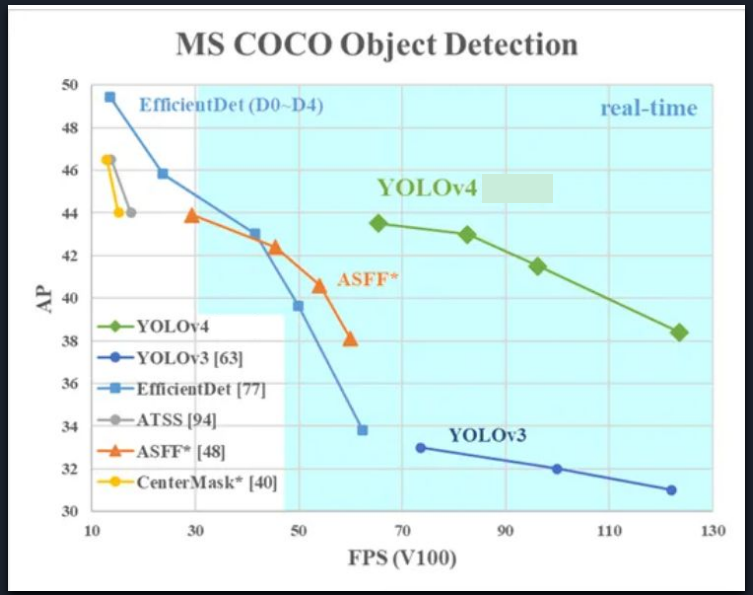


Figure 2. Bochkovskiy, A. (2020)

- YOLOv4 seemed best suited for this project considering it had the best ratio of time to precision
- 10% and 12% increase in AP respectively from YOLOv3

Current Speaker:
Mark Pedroso

Final Decision for Object Detection

- Yolov3 is more established and stable
  - Yolov3 - 2018
  - Yolov4 - 2020
- Yolov3 has more available documentation for implementation

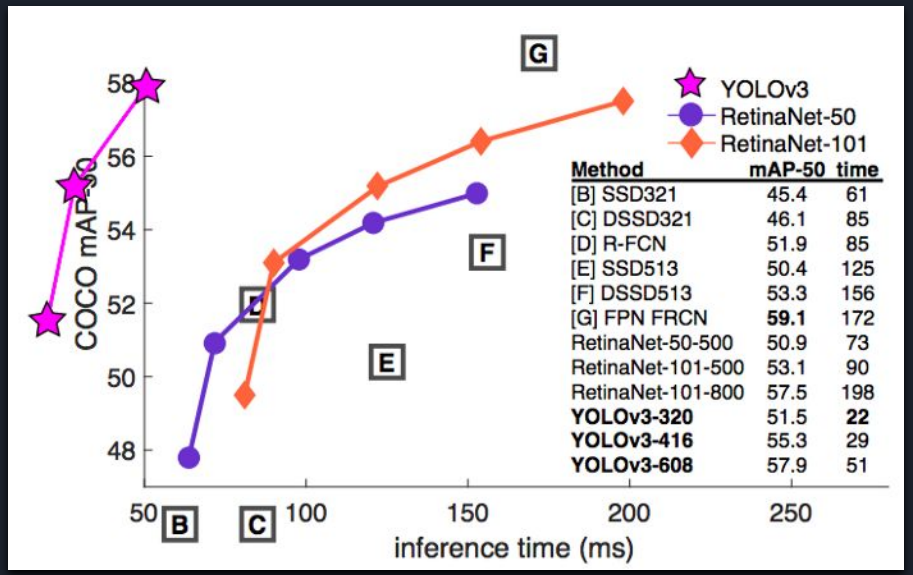


Figure 3. Redmon, J. (2018)

Current Speaker:
Mark Pedroso

Custom Dataset

YOLOv3 requires a dataset of images to learn

- 1,344 BB8 images gathered for training
  - located within the simulation.
  - Varying backgrounds
  - Different angles & distances
- Labeled using labelImg
  - Created ground truth boxes, and labeled them with the appropriate class
- Dataset split into training and testing sets
  - 80% training, 20% testing

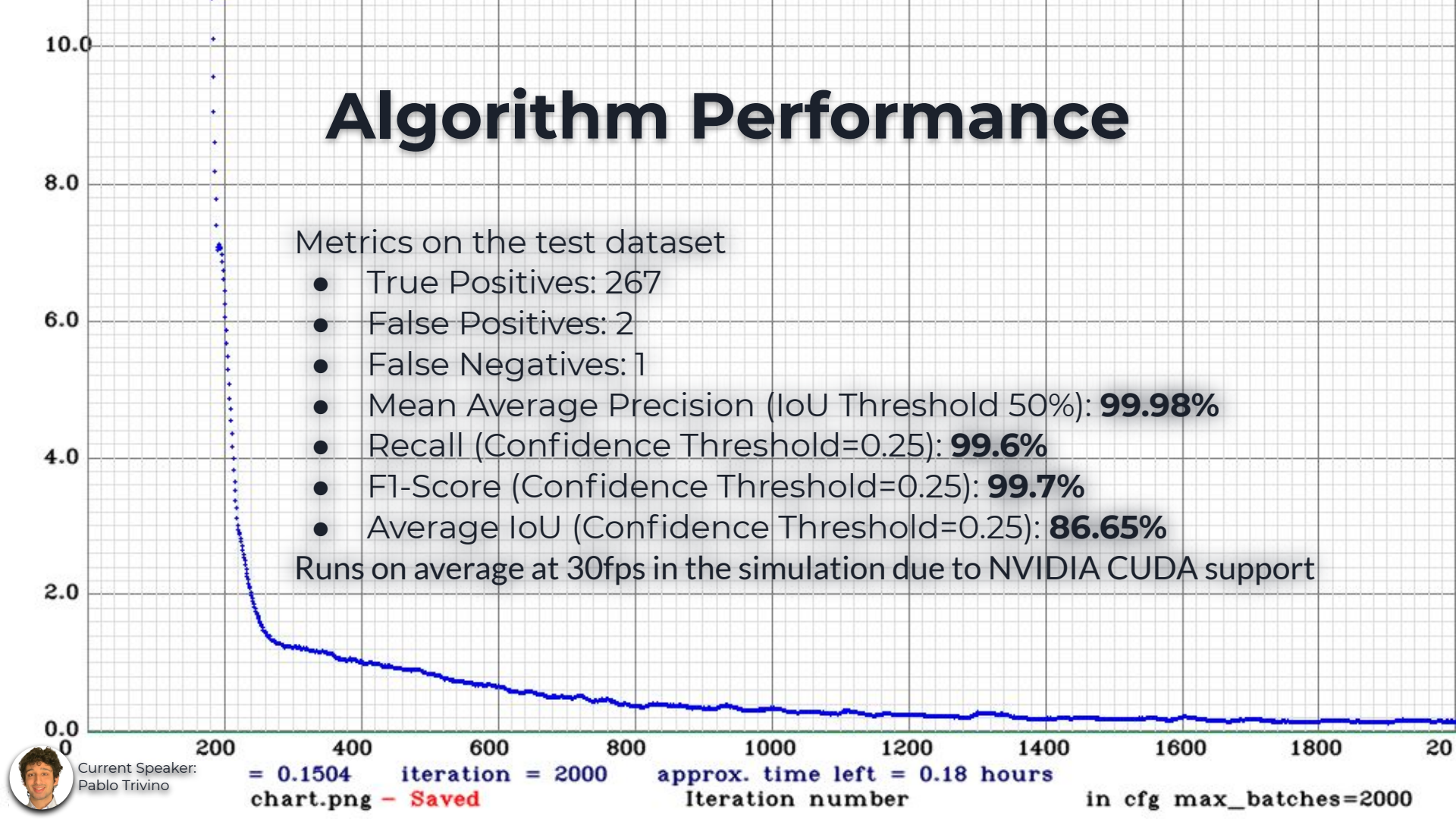
Current Speaker:
Pablo Trivino

Training Methodology

- Model trained in Google Colab using the custom dataset
  - Training time: ~4 hours
- Modified hyperparameters
  - Batches: 64
  - Subdivisions: 16
  - Width/height: 416 x 416
  - Max Batches (2000 * class number): 2000
  - Steps (80%, 90% of max batches): 1600, 1800
- Data augmentation used to improve training
  - Built in image resizing in YOLO config file
  - Dataset contains images from many angles, positions, and backgrounds
- Model tested using the testing set after training
- Final weights file generated by the training was downloaded and used for detections in our local machine

Current Speaker:
Pablo Trivino

Algorithm Performance



Metrics on the test dataset

- True Positives: 267
- False Positives: 2
- False Negatives: 1
- Mean Average Precision (IoU Threshold 50%): **99.98%**
- Recall (Confidence Threshold=0.25): **99.6%**
- F1-Score (Confidence Threshold=0.25): **99.7%**
- Average IoU (Confidence Threshold=0.25): **86.65%**

Runs on average at 30fps in the simulation due to NVIDIA CUDA support

Current Speaker:
Pablo Trivino

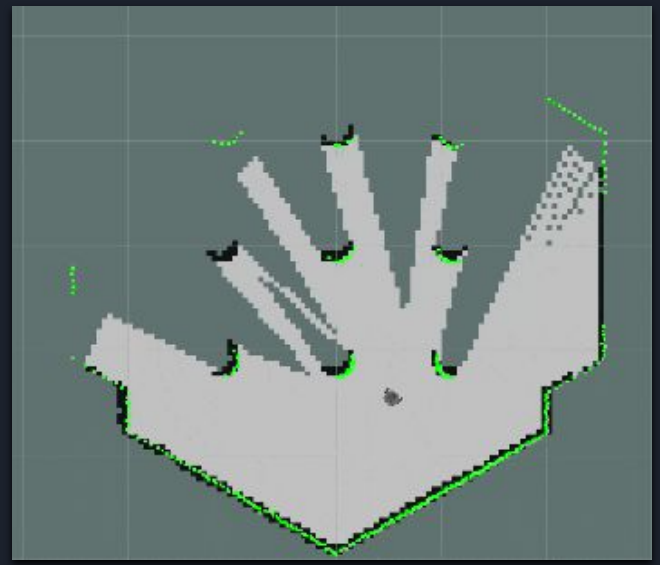
Simulation - Environment

- 400ft x 700ft urban environment
- Lots of different obstacles
- Simulated pavements and grass have different friction attributes to simulate real world conditions.

Current Speaker:
Noah Avizemer

Navigation - Gmapping & SLAM

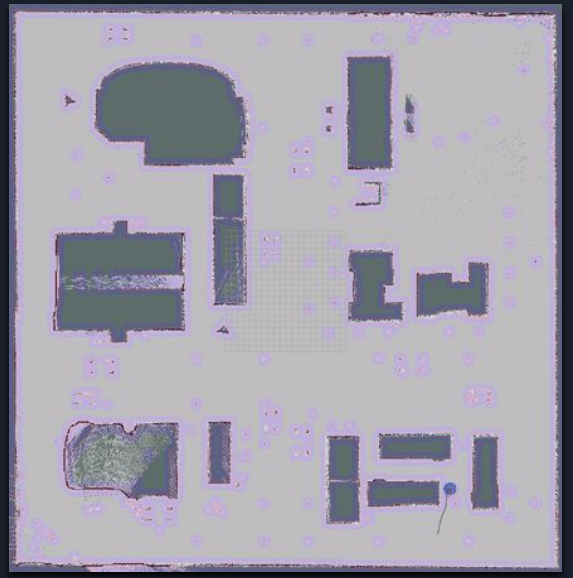
- Map of environment created using Gmapping SLAM
  - Costmap is generated from lidar
- Turtlebot autonomously navigates to areas with unmarked pixels to explore the environment.
- Exploration reveals new areas to turtlebot allowing for the map to grow.



Current Speaker:
Patrick Bauer

Navigation - Grid Search

- At runtime waypoints are generated for each point on the grid.
- Modular, can be fit to any environment dimensions.
- A* Algorithm used to navigate to each waypoint.
- If a waypoint cannot be reached, the turtlebot moves to the next one.
- Spaced optimally ensure turtlebot thoroughly searches the entire environment.



Current Speaker:
Nathanael Cassagnol

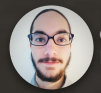
Navigation - Obstacle Avoidance

- Map-based, A*, pathfinding used to plan paths around obstacles.
- Lidar used to avoid nearby obstacles.
- Avoids getting stuck, damaging itself, or running into anything else.

Current Speaker:
Nathanael Cassagnol

Navigation - Target Approach

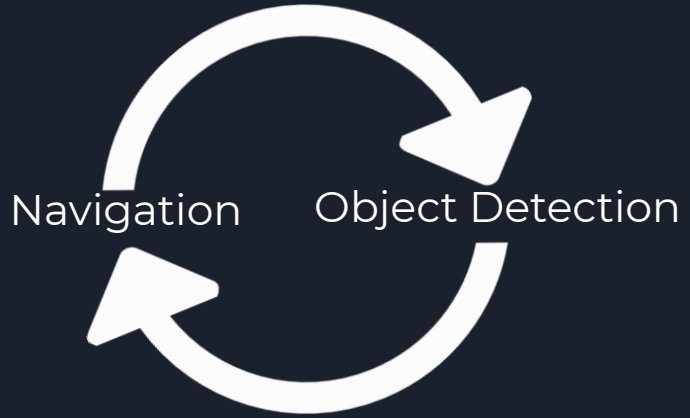
- Camera feed is run through our trained AI and the results are fed to navigation.
- If target object detected (>0.6 probability), grid search is paused.
- Turtlebot approaches target based off the target's bounding box.
- If target confirmed mission ends successfully.
- If false positive occurs, grid search continues.



Current Speaker:
Nathanael Cassagnol

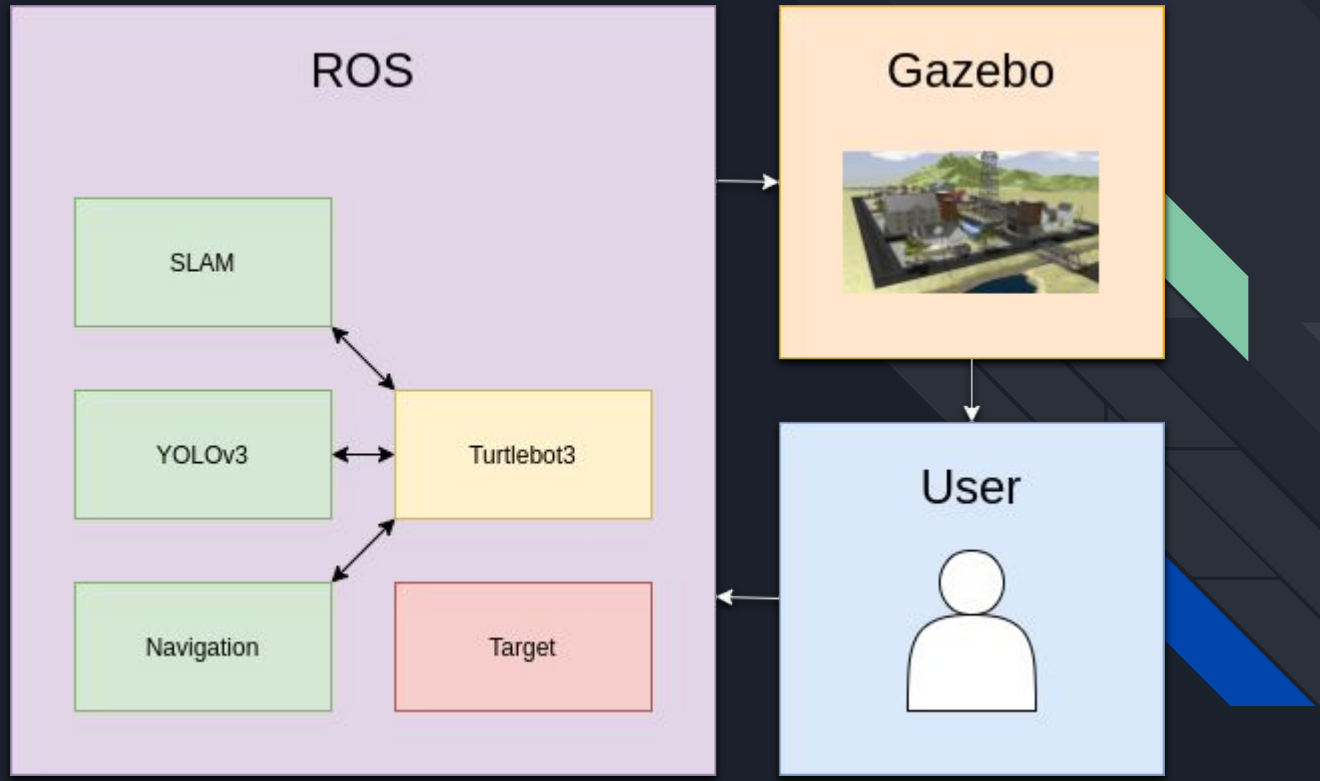
Simultaneous Navigation & Detection

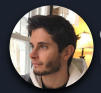
- Need to navigate the environment and search for the object simultaneously.
- 2 python threads
  - One controls the navigation of the turtlebot.
  - One checks for objects detected.
  - Shared resources for notifying each other about status



Current Speaker:
Patrick Bauer

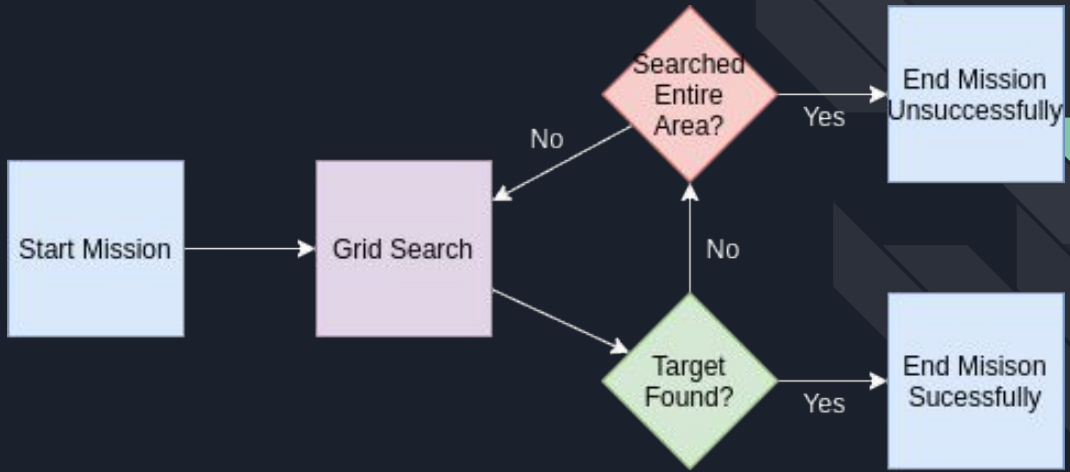
Entity Relationship Diagram: Project





Current Speaker:
Patrick Bauer

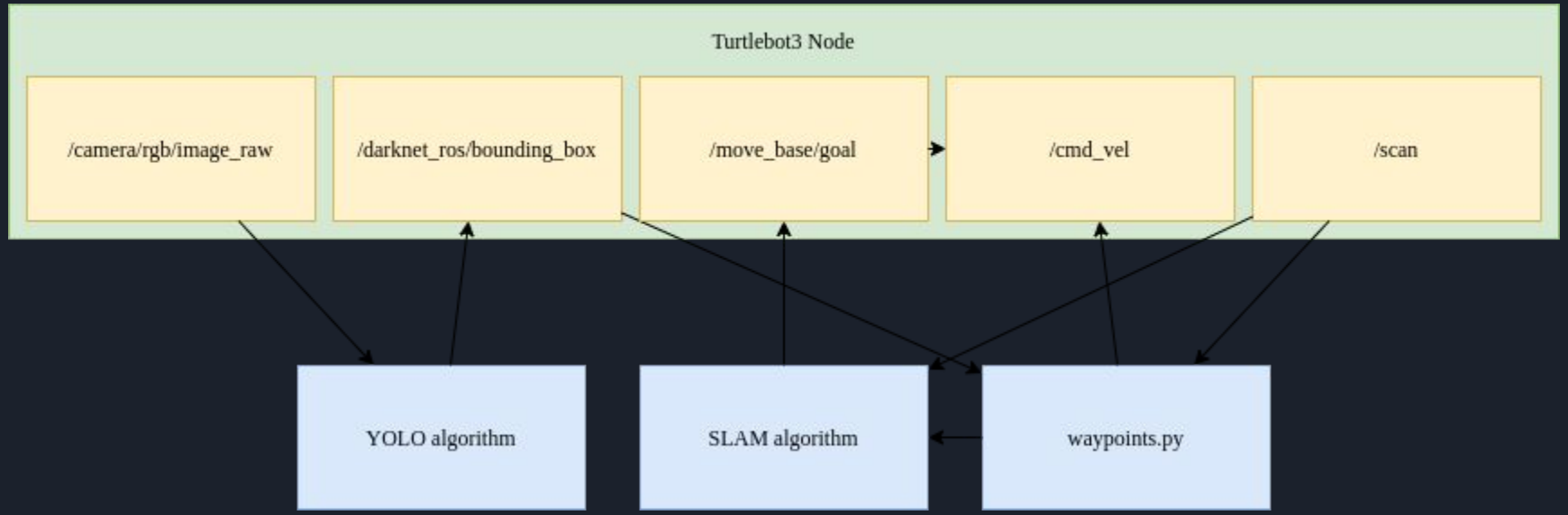
Mission Process Flowchart



Area of Work	Patrick Bauer	Nathanael Cassagnol	Mark Pedroso	Pablo Trivino	Noah Avizemer

Current Speaker:
Patrick Bauer

ROS Communications

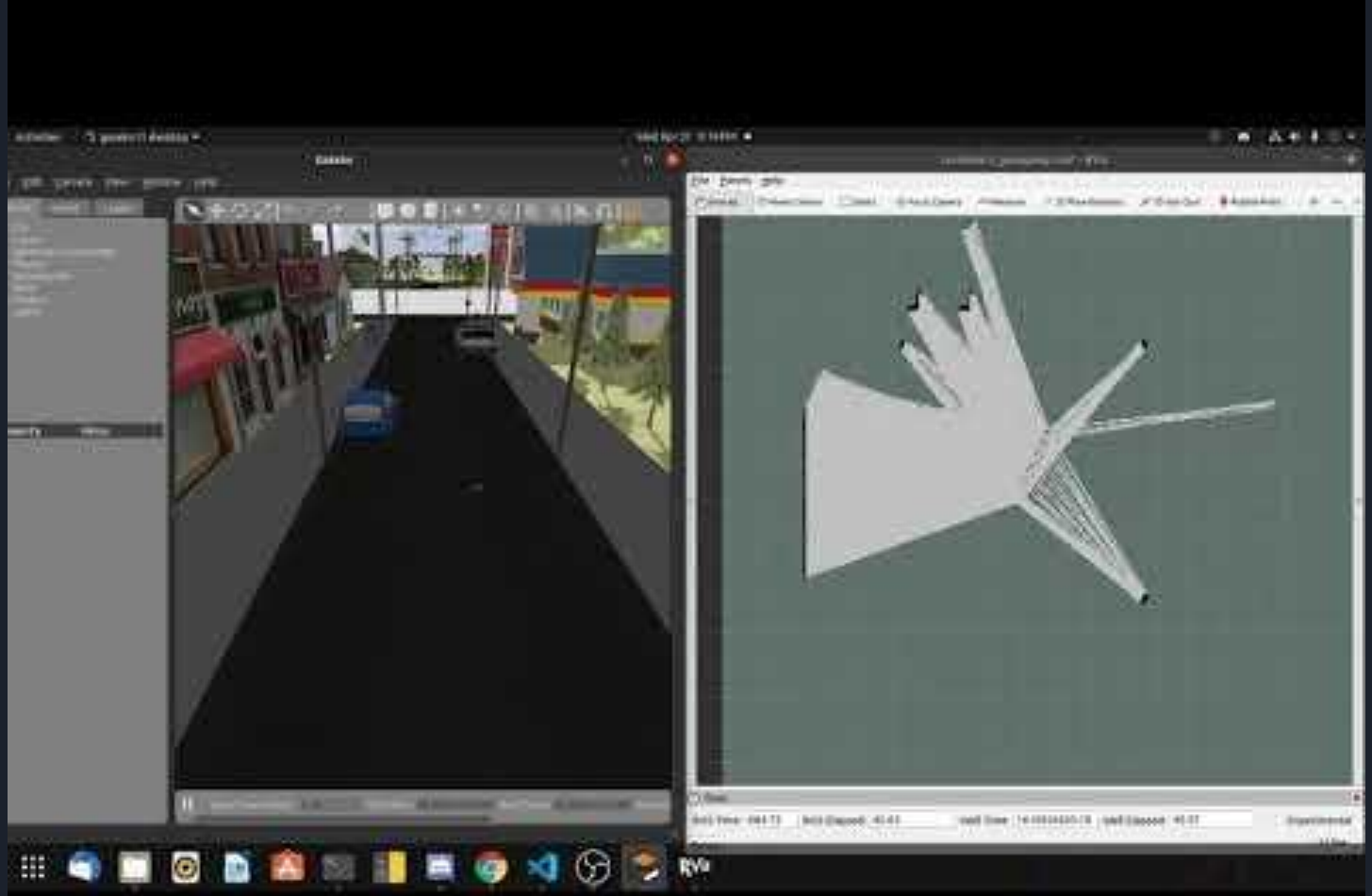


Current Speaker:
Patrick Bauer

Autonomous Mapping Demo

- In order to navigate around the world, it must first be mapped out
  - Uses Gmapping SLAM
- Done autonomously with a turtlebot
  - Navigates to closest unexplored areas.
- Increased sensor range to map outdoor environment.

Current Speaker:
Noah Avizemer

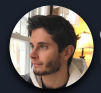


Current Speaker:
Noah Avizemer

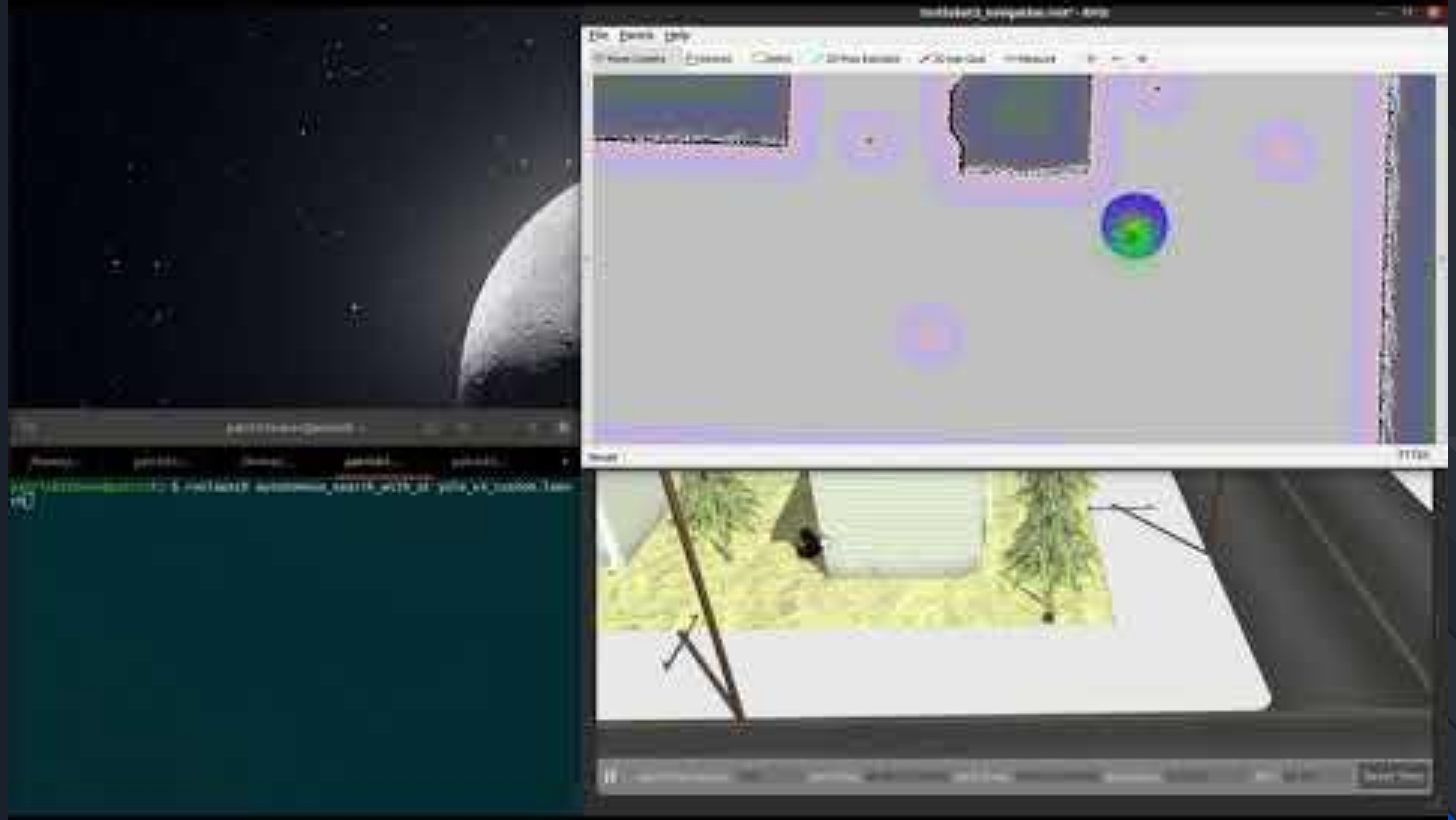
COCO Dataset Demo

Current Speaker:
Pablo Trivino

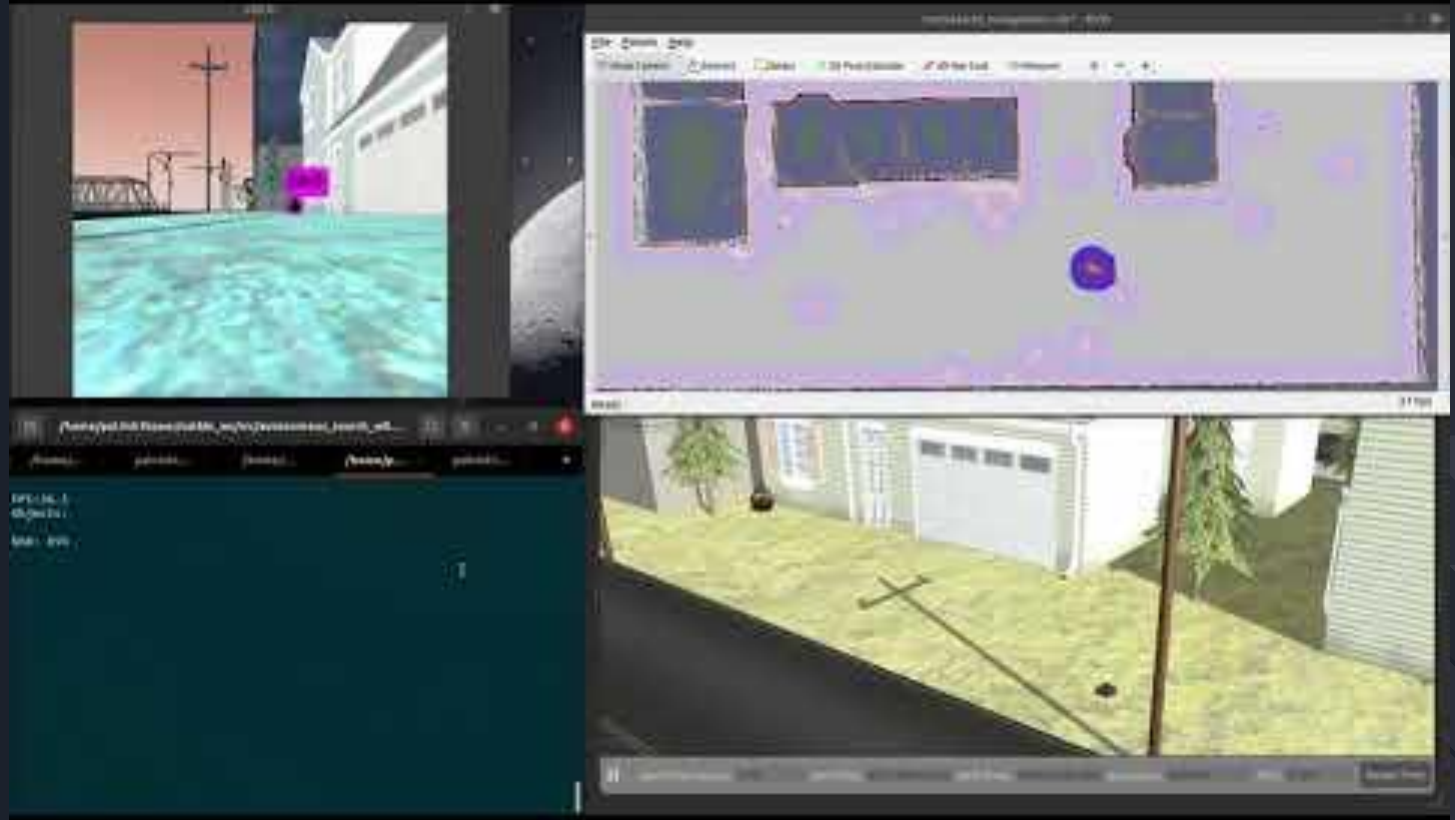
Project Demo



Current Speaker:
Patrick Bauer



Current Speaker:
Patrick Bauer



Current Speaker:
Patrick Bauer

Administration - Budget

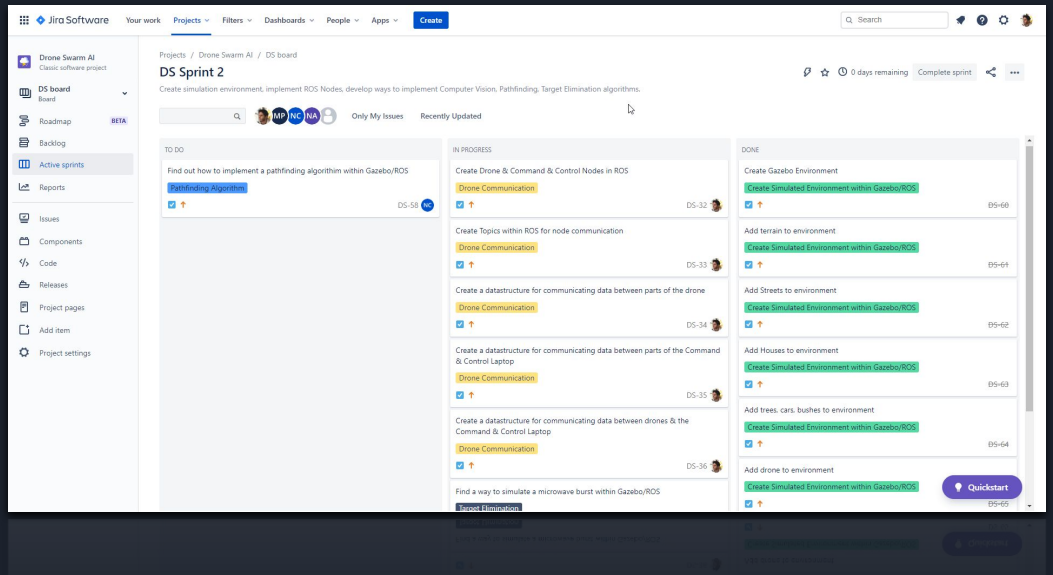
- Took advantage of open-source software and tools
  - Free
- Purchased access to The Construct's ROS development studio
  - $50 a month for premium version (x3)
  - $5 a month for increased storage (x2)

TOTAL: $160

Current Speaker:
Noah Avizemer

Administration - Teamwork

- Bi-weekly meetings with our sponsor through Skype
- Team meetings & communication through Discord
- Project management through JIRA



Current Speaker:
Noah Avizemer

Administration - Challenges

Simulation Setup:

- Dealing with different versions of ROS
- Packages not working
- Deprecated packages
- A lack of documentation on packages and methodologies
- Trouble getting started due to a lack of experience and direction

Object Detection:

- Changed target model multiple times, requiring us to re-gather training data
- Turtlebot cameras do not have 360 degree vision
- Implementing YOLOv3 on a local system
- Low FPS for object detection without GPU

Pathfinding:

- Changed requirements requiring a new approach
- The version of ROS used prevented multi-robot implementation

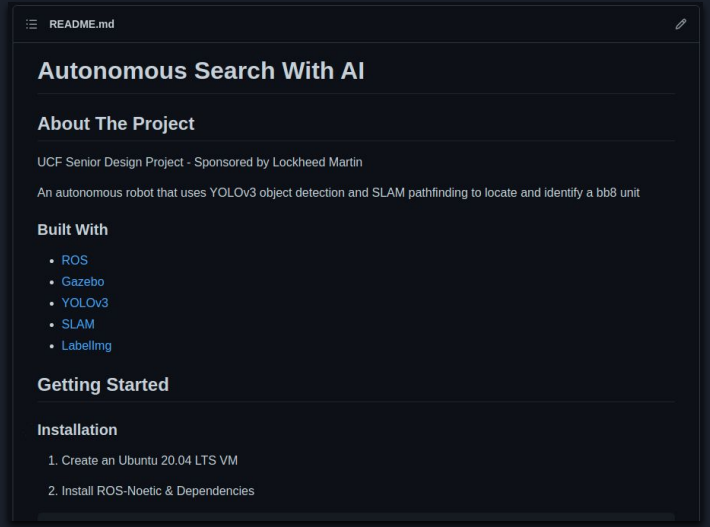
Current Speakers:
Noah Avizemer
Pablo Trivino
Nathanael Cassagnol

Administration - Deliverables

https://github.com/patrick1bauer/autonomous_search_with_ai

Repository contains:

- Autonomous Search With AI package
- Readme with instructions to set up the required software and workspace.
- Final design document including our research and how the project works.


README.md

Autonomous Search With AI

About The Project

UCF Senior Design Project - Sponsored by Lockheed Martin

An autonomous robot that uses YOLOv3 object detection and SLAM pathfinding to locate and identify a bb8 unit

Built With

- ROS
- Gazebo
- YOLOv3
- SLAM
- LabelImg

Getting Started

Installation

1. Create an Ubuntu 20.04 LTS VM
2. Install ROS-Noetic & Dependencies

Current Speaker:
Patrick Bauer

Questions?