ELECOMP CAPSTONE **SYMPOSIUM**
TUESDAY, DECEMBER 22, 2020

THE UNIVERSITY OF RHODE ISLAND

THINK BIG WE DO™

Applying ML Research for Undersea Object Detection

Design a software package that can quickly and accurately detect objects underwater

Team Members: James Morris (ELE), Anthony Neves (CPE), Andrew Gomes (ELE)

Technical Directors: Thomas Santos, Dana Brown | Consulting Technical Directors: Noah Daniels, Al Gaines ('88), Najib Ishaq ('20), and Jeremy Peacock ('20)

Project Motivation

A difficulty of utilizing Unmanned Underwater Vehicles (UUV) is accounting for physical obstacles in the path of that UUV that may require additional navigational commands. Rite-Solutions is collaborating with Dr. Noah Daniels and his graduate research team to develop a machine learning model that can identify objects on the seabed. This technology could be applied across several commercial industries, as well as the Navy. The technological advantages of the machine learning model include avoiding potentially hazardous objects for the UUV, searching for and identifying objects, and eventually aiding in autonomous UUV operations.

The software package being designed to satisfy the above criteria requires translation of existing Python/Tensorflow code into a language with a higher focus on safety, Rust, to implement the software onto standalone devices properly. To test the software package prototype, a MATLAB program will be created to simulate an active side-scan sonar, where results can be validated under different conditions of the sea.

Anticipated Best Outcome

The primary objective of the project is to design a software package that can quickly and accurately detect objects underwater. In order to accurately test this software package, the simulated sensor data from MATLAB must be created and verified to be working properly as well. As a result of the translation of the existing Python code into Rust, the effectiveness of the new programming language Rust can be evaluated as well for future products with similar purposes made by Rite-Solutions. The best possible outcome would be realized with an effective Rust and MATLAB software package that can simulate side-scan sonar data and detect underwater objects from that data accurately.

Key Accomplishments

Sonar Emission: The MATLAB simulation of a side-scan sonar requires both an emission of energy that can be applied to an object, as well as sensors that receive feedback from energy bouncing off of an object. The emission of this energy from a sonar sensor has been simulated in a very simplistic form, which is the foundation of the High-Level Design Flowchart for our MATLAB program, the simplified version is in **Fig. 1**

MATLAB Examples: Several MATLAB resources are influential to the Sonar emission, detection, and object detection codes that we are creating. While creating a simulation of a side-scan sonar in MATLAB, we are also editing and applying the "Underwater Target Detection with an Active Sonar System" mathworks program to our own coding. By utilizing this example, the group can evaluate which methods of sensor and signal representation best fit our goals as we progress.

Objects as MATLAB Targets: Replacing the targets of Sonar example codes with solid objects instead of designated targets strength objects is a key step in the development of the MATLAB code. The desired output is the feedback from the sonar simulation when energy is reflected off of the object, therefore targeting that object with the sonar emission is critical, target profile is shown in **Fig. 4**.

Learning the Ins and Outs of Rust: To better translate the provided Python/TensorFlow code into Rust we were tasked with learning the language through the official Rust documentation book (henceforth referred to as 'The Rust Book'). This book contains a total of twenty chapters ranging from a "guess the number" program, to building a multithreaded web server. While the information provided in these chapters are important to know looking towards the translation of the provided code, we have been informed that the later chapters contain information that is not vital to this project. At the moment, we have up to chapter 8 completed.

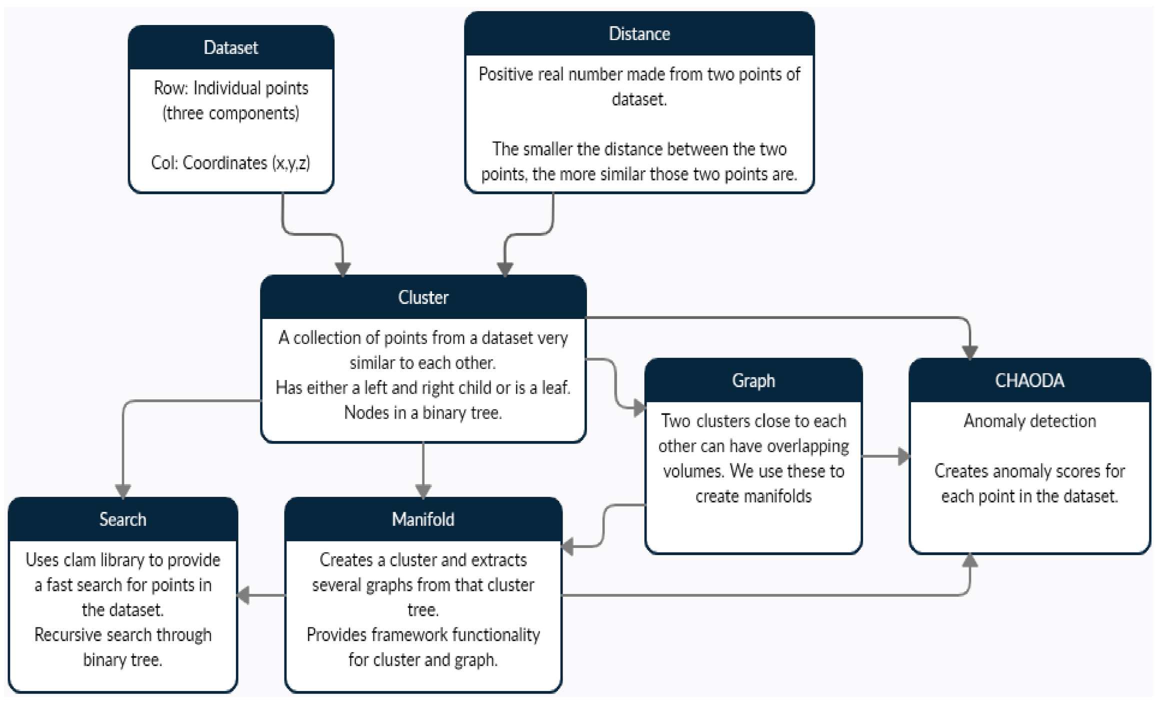

Fig. 3: Diagram of the function and connections between CHAODA, Search, Cluster, Graph and Manifold, which are all being translated into Rust from Python.

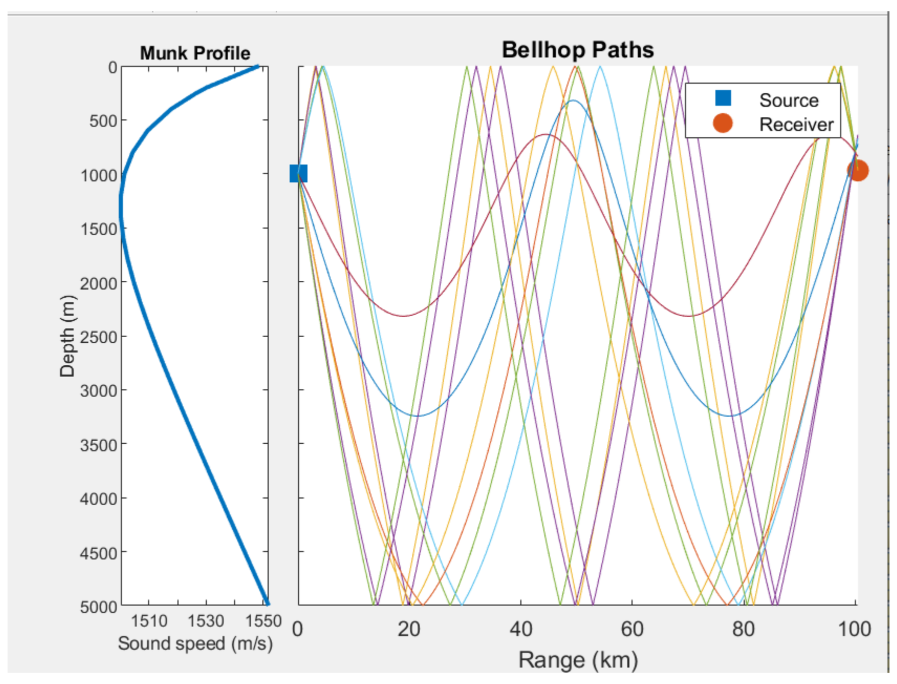

Fig. 2: The speed of sound according to the water depth used in the MATLAB simulation, along with the Bellhop Paths between the Source and Receiver.

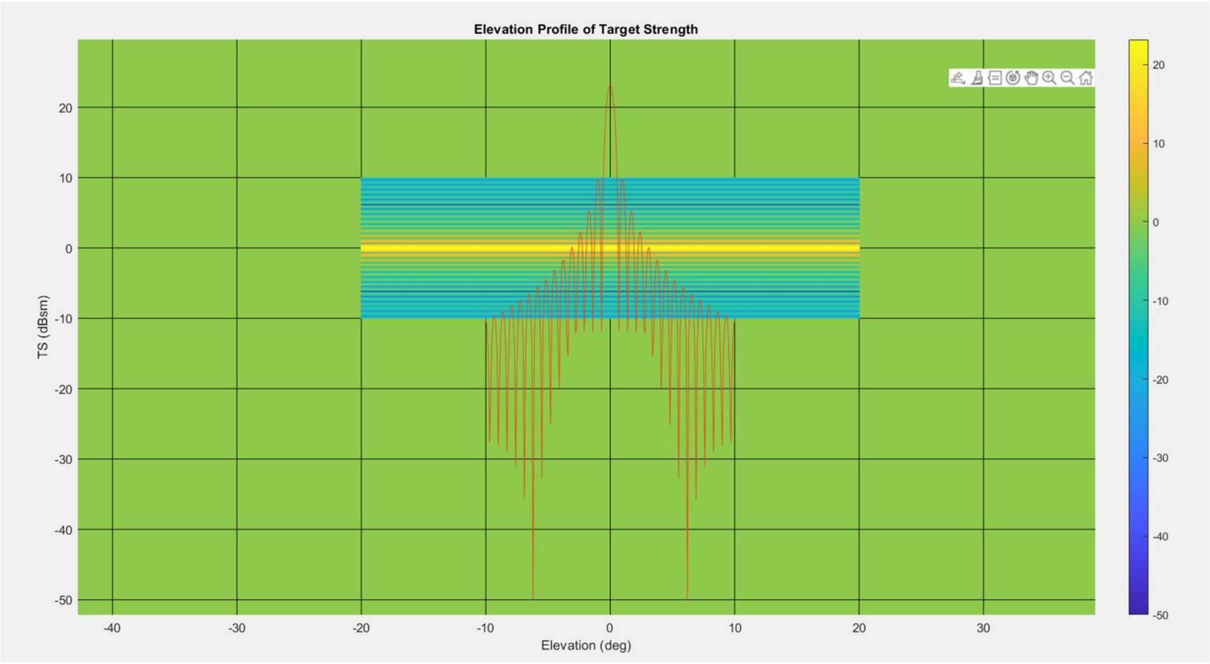

Fig. 4: The profile of the cylinder that replaced the targets for the Underwater Target Detection with an Active Sonar System, rotated to show length and width although it is 3-Dimensional

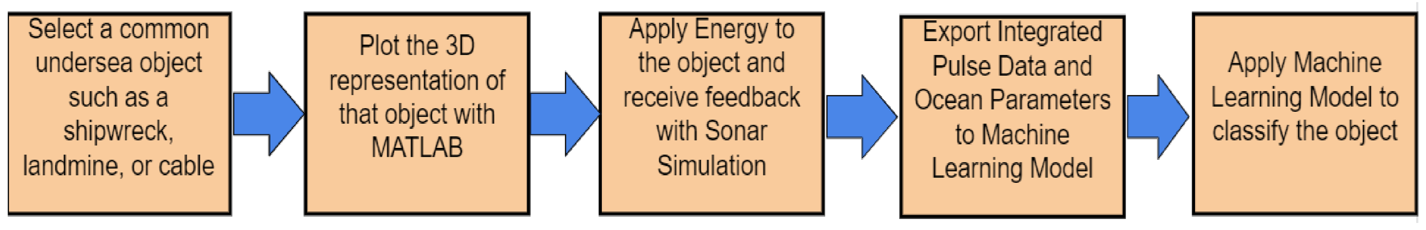

Fig. 1: The High-Level Design Process of the MATLAB code to simulate a simple, active side-scan sonar

Remaining Technical Challenges

Sonar Simulation: To date, a Sonar simulation MATLAB Program has been made, although it is currently simple compared to the end goal. The current program outputs six plots, including the target profile of the object (**Fig. 4**), underwater paths between source and receiver, and the munk profile of the water along with a Bellhop Simulation of the paths between a distant source and receiver (**Fig. 2**).

Testing Machine Learning Model: The existing Python Machine Learning model for Object Detection needs to be tested with Sonar data. The data taken from the Sonar detection code can train the Machine Learning model to recognize different objects from data that simulates a sonar reading.

Environmental Factors: The current emission and detection coding progress in MATLAB exists on an isobathymetry plane with no velocity or noise. In order to make the software package more applicable to commercial needs, factors such as noise, water salinity, depth, temperature, or a bumpy seabed need to be applied. The machine learning model for object detection needs to be trained to recognize objects that are the same, but under different environmental factors.

User Manual: The specifics of the User Manual are yet to be determined, but in principle the manual will cover basic implementation, how to ensure that the program is working correctly, file types that the program creates and how to read or write to the files, as well as basic troubleshooting.

Translation: Provided Python/TensorFlow code must be translated into Rust to acquire the best anticipated outcome. At the moment, work is being done to translate the 'manifest' script and test it with predetermined test cases. The existing Python code with its classes of CHAODA, Search, Graph, Cluster, and Manifold with their relationships can be seen in **Fig 3**.

Learning Rust: To enhance the quality of the provided code's translation from Python/Tensorflow to Rust, we must complete reading through and documenting the remaining chapters in The Rust Book.

Implications for Company & Economic Impact

The work being done by the Rite-Solution's team towards implementing Machine Learning into the UUVs is to aid in operations conducted by the Navy for object identification and precautionary measures of hazard avoidance. The proposed autonomous ability of the UUV allows for further operations in heavily contested and hazardous environments. Speaking from the standpoint of company and economic impact, the implementation of the software package using side-scan sonar data allows for quick and accurate detection and identification of 3D objects located deep underwater that can be incorporated into naval and commercial systems, which include, but are not limited to, UUVs, submarines, aircrafts, and other unmanned systems.

ELECOMP Capstone Program Director: Professor Harish R D Sunak

email: sunak@ele.uri.edu telephone: 401-874-5859 ELECOMP Website: https://uri.edu/elecomp-capstone