Software Requirements Specification

for

Autonomous Underwater Vehicle RoboSub Computer Vision

Version 1 approved

Prepared by Heriberto Gonzalez, Horacio Mondragon

 California State University Department of Engineering

December 12, 2020

**Table of Contents**

Table of Contents................................................................................................................. pg # 2

Revision History................................................................................................................... pg # 3

 1. Introduction................................................................................................................ pg # 4

 1.1. Purpose........................................................................................................... pg # 4

 1.2. Intended Audience and Reading Suggestions................................................ pg # 4

 1.3. Product Scope................................................................................................ pg # 4

 1.4. Definitions, Acronyms, and Abbreviations .................................................. pg # 4

 1.5. References...................................................................................................... pg # 5

 2. Overall Description.................................................................................................... pg # 6

 2.1. System Analysis…......................................................................................... pg # 6

 2.2. Product Perspective........................................................................................... pg # 6

 2.3. Product Functions........................................................................................... pg # 7

 2.4. User Classes and Characteristics.................................................................... pg # 7

 2.5. Operating Environment.................................................................................. pg # 7

 2.6. Design and Implementation Constraints........................................................ pg # 7

 2.7. User Documentation...................................................................................... pg # 8

 2.8. Assumptions and Dependencies.................................................................... pg # 8

 2.9. Apportioning of Requirements...................................................................... pg # 8

 3. External Interface Requirements............................................................................... pg # 9

 3.1. User Interfaces............................................................................................... pg # 9

 3.2. Hardware Interfaces....................................................................................... pg # 9

 3.3. Software Interfaces........................................................................................ pg # 9

 3.4. Communications Interfaces........................................................................... pg # 9

 4. Requirements Specification....................................................................................... pg # 10

 4.1. Functional Requirements............................................................................... pg # 10

 4.2. External Interface Requirements................................................................... pg # 10

 4.3. Logical Database Requirements.................................................................... pg # 10

 4.4. Design Constraints......................................................................................... pg # 10

 5. Other Nonfunctional Requirements........................................................................... pg # 11

 5.1. Performance Requirements............................................................................ pg # 11

 5.2. Safety Requirements...................................................................................... pg # 11

 5.3. Security Requirements................................................................................... pg # 11

 5.4. Software Quality Attributes........................................................................... pg # 11

 5.5. Business Rules............................................................................................... pg # 11

 6. Legal and Ethical Considerations.….......................................................................... pg # 12

Appendix A: Glossary........................................................................................................ pg # 13

Appendix B: Analysis Models........................................................................................... pg # 14

Appendix C: To Be Determined List................................................................................. pg # 15

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Date | Reason For Changes | Version |
| Heriberto Gonzalez | 12/12/2020 | First Draft | 1 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**1. Introduction**

This software is intended to be implemented in the Autonomous Underwater Vehicle (AUV) under construction by the California State University Senior Design team under the Engineering Department. The goal of the AUV is to compete in an international competition hosted by Robonation to perform tasks in an underwater environment. These include movement, navigation, object detection and recognition, manipulation of objects, target elimination. This document focuses on the computer vision portion of the AUV.

**1.1 Purpose**

The purpose of this document is to explain the computer vision portion of the AUV being constructed by the Senior Design Engineering team to any future users and developers of the product so that they can design and program their own object detection and recognition system.

**1.2 Intended Audience and Reading Suggestions**

The types of readers that the document is intended for is for developers and users that intend to continue the work done by the 2020-2021 Senior Design Team. The reason for this is due to the inability of the entire team to test the AUV within unforeseen circumstances.

**1.3 Product Scope**

The software of mission planning takes data obtained from the other portions of the AUV and transfers them around different states allowing it to perform the tasks required of it. The software when released will be used by future users of the AUV to add upon and improve its features

**1.4 Definitions, Acronyms, and Abbreviations**

AUV – Autonomous Underwater Vehicle

ROS – Robot Operating System

CNN – Convolutional Neural Networks

YOLOv4 – You Only Look Once [ CNN Algorithm ]

DARKNET – Open source Neural Network Framework

OPENLABELING – Open source Image and Video Labeler

CUDA – Parallel Computing Platform and application programming interface model

FPS – Frames Per Second

**1.5 References**

YOLOv4: Optimal Speed and Accuracy of Object Detection – <https://arxiv.org/pdf/2004.10934.pdf>

Darknet Website (no longer updated) – <https://pjreddie.com/darknet/>

Darknet GIT Repository – <https://github.com/AlexeyAB/darknet>

OpenLabeling – <https://github.com/Cartucho/OpenLabeling>

**2. Overall Description**

**2.1 System Analysis**

Objective:

1. Object Detection – Be able to detect when an object of interest is present in view of our AUV
2. Object Recognition – Classify and Recognize the various objects throughout the course
3. Output Objects Estimated Location – Once an object of interest has been Detected and Recognized we should be able to output roughly where in frame our object is at, relative to the angle of the camera

Task:

1. Use Convolutional Neural Network to develop a Deep Learning Model to detect and classify objects
2. Gather and develop an image dataset of the objects from the competition to be used to train the CNN Model

**2.2 Product Perspective**

The software in this SRS is a component of the AUV under construction by the Senior Design Engineering team. The AUV is also built for a competition that is typically run every year, so it will have similarities to other AUVs built by competing teams. The motivation for building this product is to see if the finished AUV performs better than other AUVs in a competitive setting.



**2.3 Product Functions**

The functions of the software involve:

* Classifying the objects of interest within the competition
	+ This will be done using the YOLOv4 deep learning algorithm
* Output the name of the object and an estimated location of the object
	+ This will be done using the Darknet framework + YOLOv4, along with mathematical calculations of bounding boxes

**2.4 User Classes and Characteristics**

The user classes that are anticipated to use this product are software developers that seek to utilize Darknet and YOLOv4 to develop a CNN to detect and classify objects.

**2.5 Operating Environment**

Environment:

* Operating System: Ubuntu 18.04
* Software:
	+ Darknet
	+ CUDA
	+ OpenLabeling
* Hardware: Jetson TX2
	+ GPU 256-core NVIDIA Pascal™ GPU architecture with 256 NVIDIA CUDA cores
	+ CPU Dual-Core NVIDIA Denver 2 64-Bit CPU
	+ Quad-Core ARM® Cortex®-A57 MPCore
	+ Memory 8GB 128-bit LPDDR4 Memory

**2.6 Design and Implementation Constraints**

Constraints of the AUV are:

* Communication between the people designing software to the AUV and the people designing the hardware of the AUV.
* Ability to test the AUV within the restrictions of the 2020-2021 year

**2.7 User Documentation**

* Setup Guide for Darknet 🡪 PDF format
* Setup Guide for YOLOv4 🡪 PDF format
* Tutorial Video on Training Custom YOLOv4 Model 🡪 YouTube Video

**2.8 Assumptions and Dependencies**

Factors that may affect the requirements in the document are:

* The use of other Ubuntu versions
* A programming language other than Python

**2.9 Apportioning of Requirements**

Delayed requirements involve the inability of the current Senior Design team to test the AUV due to recent lockdowns and quarantines.

**3. External Interface Requirements**

**3.1 User Interfaces**

This software has no user interface at the moment however one could have an interface that displays a live feed of the AUV field of vision.

**3.2 Hardware Interfaces**

* Cameras: Blue Robotics Low-Light HD USB Camera
	+ Field of View (Horizontal): 80°
	+ Field of View (Vertical): 64°
	+ Focal Length: 2.97 mm
* Hardware: Jetson TX2
	+ GPU: 256-core NVIDIA Pascal™ GPU architecture with 256 NVIDIA CUDA cores
	+ CPU: Dual-Core NVIDIA Denver 2 64-Bit CPU
	+ Quad-Core ARM® Cortex®-A57 MPCore
	+ Memory: 8GB 128-bit LPDDR4 Memory

**3.3 Software Interfaces**

Darknet GIT Repository – <https://github.com/AlexeyAB/darknet>

OpenLabeling – <https://github.com/Cartucho/OpenLabeling>

**3.4 Communications Interfaces**

Not Applicable…

**4. Requirements Specification**

1. The system shall be able to take in live feed form the cameras on board the AUV and run it through the CNN Network to output accurate classifications of the objects within the course
2. The system should be able to calculate a rough distance or angle at which the AUV is detecting the object of interest this could be done with pixel calculations

**4.1 Functional Requirements**

1. The software shall be trained on a dataset large enough for the CNN to learn and distinguish the difference between the various objects in the course
	1. 500-1000 images per object
	2. Each Image in the Dataset must be labeled using OpenLabeling
2. The software shall be trained using Darknet and using the YOLOv4 algorithm
	1. Edits must be made to the algorithms variables to be applicable to the need of the dataset that it is given
	2. Darknet requires a txt file listing the paths to all the images in the dataset along with the config file that would be edited for dataset
	3. Training can vary in time depending on the GPU used to train the CNN

**4.2 External Interface Requirements**

The Computer Vision software will take input from two Blue Robotic Low Light cameras where each cameras video feed will be passed into the Darknet software where it is then compiled and ran onto the YOLOv4 CNN and the output will be the name of the object of interest that is in present view of the AUV and a estimated angle at which the camera is viewing the object of interest.

**4.3 Logical Database Requirements**

Not Applicable…

**4.4 Design Constraints**

The software is reliant on a powerful enough GPU to run the CNN network and to perform within specifications ( 10-15 FPS ) the algorithm takes advantage of the GPU image processing to perform its CNN network onto each frame of the live video. Also, as of now the GPU must be a NVIDIA graphics card to be able to run Darknet using CUDA.

**5. Other Nonfunctional Requirements**

**5.1 Performance Requirements**

The performance requirements for the computer vision software are to perform at a decent FPS around 10 to 15 would be plenty for the AUV to make quick and accurate classifications and detections.

**5.2 Safety Requirements**

Not Applicable…

**5.3 Security Requirements**

Not Applicable…

**5.4 Software Quality Attributes**

Not Applicable…

**5.5 Business Rules**

Not Applicable…

**6. Legal and Ethical Considerations**

Not Applicable…

**Appendix A: Glossary**

Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.

If this section is very short you may include it in section 1.4. If your list is very long you may include it here and put a reference to this Appendix in section 1.4.

**Appendix B: Analysis Models**

Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams.

**Appendix C: To Be Determined List**

Collect a numbered list of the TBD (to be determined) references that remain in the SRS so they can be tracked to closure.