System Requirements Specification

for

TurtleTech

Version 5.0 approved

Prepared by Christopher Angland, Laurel Dodson, Parker Brown, RoseEllen Hoke

TurtleTech

27 April 2023

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
TurtleTech Team	10/07/22	The team compiled a draft for sprint 1	1
TurtleTech Team	11/08/22	Updated draft for progress made after sprint 2	2
TurtleTech Team	12/03/22	Updated and final revision for sprint 3	3
TurtleTech Team	4/27/23	Updated and completed final documentation	4

1. Introduction

1.1 Purpose

The purpose of the TurtleTech project is to produce a payload casing for the Nvidia Jetson device and streamline the process of setting up the Jetson while also producing a functioning neural network system. The payload casing will be swappable between different drones. The neural network system will be able to identify right whales. This document provides a detailed description of the software requirement for the TurtleTech project and will explain the interface requirements and all the system's features. This SRS will cover the whole functionality of the TurtleTech system.

1.2 Document Conventions

The system requirement specification is a crucial document that outlines the functional and non-functional requirements for a software system. This document is broken down into six sections with various amounts of subsections in each section. The sections are identified in the table of contents and then are place at the top of each section and are labeled with a numerical value that corelates with the table of contents. The subsections are assigned a decimal numerical value and are based off the sections value. The subsections are bolded above each subsection. The six sections in this document are the introduction, overall description, external interface requirements, system features, other nonfunctional requirements, other requirements, and the glossary.

1.3 Intended Audience and Reading Suggestions

This project is sponsored by Northrop Grumman and the Brevard County Zoo and is open to the public for research. This project is under the guidance of Dr.Akbas and will benefit the functionality of the TurtleTech neural network system and the TurtleTech hardware team.
The System Requirement Specification document is written to provide guidance and educate the reader on specific requirements the TurtleTech system must meet and what will not be met as well. The document is broken into six sections. The first section introduces the purpose of the TurtleTech team and why this document is being written. The second section provides an overall description of the system and provides the reader with all the necessary background information, in order to understand what is trying to be accomplished. The third section covers the external interface requirements which includes the hardware, software and user interfaces. The fourth section includes the system features, and the fifth section covers the other nonfunctional requirements that were not previously discussed in the document.

1.4 Product Scope

The purpose of this project is to produce a payload casing for the Nvidia Jetson device and streamline the process of setting up the Jetson while also producing a functioning neural network system. The payload casing will be swappable between different drones. The payload casing will be configured in partnership with the TurtleTech Hardware team and will be tested for durability and practicality.
The neural network system will be able to identify right whales. The neural networking system will only identify one specific animal at a time, it will not be able to differentiate multiple things at once.

The ERAU team will produce a 3D printed payload casing and a functioning neural networking system by the end of the 2023 Spring semester.

1.5 References

Censys Technologies. (2022, October 7). *Sentaero BVLOS*. Censys Technologies Corporation. Retrieved November 8, 2022, from https://censystech.com/sentaerobvlos/

Imaging, L. (2019, January 1). *LI-XNX-CB-6CAM-FP*. Leopard Imaging. Retrieved November 8, 2022, from https://www.leopardimaging.com/product/nvidia-jetson-cameras/nvidia-nx-mipi-camera-kits/li-xnx-cb-6cam-fp-poe/

TurtleTech-erau. GitHub. (n.d.). Retrieved October 7, 2022, from https://github.com/TurtleTech-ERAU

Turtle Tech. Northrop Grumman. (2022, June 29). Retrieved October 7, 2022, from https://www.northropgrumman.com/sustainability/turtle-tech/

Build from source on Windows :   tensorflow. TensorFlow. (n.d.). Retrieved November 8, 2022, from https://www.tensorflow.org/install/source_windows

Gartzman, D. (2019, June 15). Installing multiple python versions on Windows using Virtualenv. freeCodeCamp.org. Retrieved November 8, 2022, from https://www.freecodecamp.org/news/installing-multiple-python-versions-on-windows-using-virtualenv/

Getting started with Jetson Nano Developer Kit. NVIDIA Developer. (2022, November 1). Retrieved November 8, 2022, from https://developer.nvidia.com/embedded/learn/get-started-jetson-nano-devkit

How to downgrade python from 3.7 to 3.6. Stack Overflow. (1965, December 1). Retrieved November 8, 2022, from https://stackoverflow.com/questions/52584907/how-to-downgrade-python-from-3-7-to-3-6

Nicknochnack. (n.d.). Nicknochnack/TFODCOURSE. GitHub. Retrieved November 8, 2022, from https://github.com/nicknochnack/TFODCourse

Rosebrock, A. (2020, April 18). How to configure your nvidia jetson nano for computer vision and Deep Learning. PyImageSearch. Retrieved November 8, 2022, from https://pyimagesearch.com/2020/03/25/how-to-configure-your-nvidia-jetson-nano-for-computer-vision-and-deep-learning/

Tensorflow. (2021, May 7). Models/tf2_detection_zoo.MD at master · Tensorflow/models. GitHub. Retrieved November 8, 2022, from https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/tf2_detection_zoo.md

Tensorflow. (n.d.). Models/model_builder_tf2_test.py at master · Tensorflow/models. GitHub. Retrieved November 8, 2022, from https://github.com/tensorflow/models/blob/master/research/object_detection/builders/model_builder_tf2_test.py

Tensorflow. (n.d.). Models/research at master · Tensorflow/models. GitHub. Retrieved November 8, 2022, from https://github.com/tensorflow/models/tree/master/research

2. Overall Description

2.1 Product Perspective

Right whales are a very rare species, with estimates of the number of individuals in the world ranging from three hundred to five hundred. The small size of the remaining population and the extensiveness of the species' range means that sightings are rare, and accurate and up-to-date information about location and condition of these whales if difficult to come by. To assist this species, more information about the whales' migratory behaviors would be beneficial, and this product helps one specific effort towards their conservation.

The current method of identifying whales requires extensive knowledge of mild differences between individuals, including barnacle patterns and characteristics such as scarring on the bonnet (head) or flukes. This process requires significant background training and introduces delay, and so is not an efficient method of whale monitoring.

TurtleTech is an initiative that is researching how to track marine animals in a more effective way than currently available, by using UAS (unmanned aerial systems) to track (currently) turtles off the coast of Florida. Unmanned aircraft don't require nearly as much manpower or equipment to operate and can survey large amounts of area in a short amount of time to quickly process gathered images for whales.

However, our current system relies upon one UAV, and has difficulties mounting the payload used for capturing the whale images. We intend to create a system that is adaptable both to new species and new aircraft, allowing us to have a more accessible system that can be used in more situations to provide additional benefit to marine researchers and the TurtleTech team alike.

Our product is a hardware case that can store a circuit board payload. The framework case will be easily removable from the UAV and fit the hardware restrictions for use in different situations, and the software on the circuit board will be applicable to whales, in order to identify whales from the images captured by the UAV.

2.2 Product Functions

The product shall:

- Hold the Jetson board and other components of the payload
- Adhere to design requirements imposed by the hardware team
- Be removable from the UAV chassis
- Be manufactured to previous standards of creation used on the UAS
- Apply a neural network algorithm to images taken
- Identify and output and whales located in the images

- Have proper documentation for use of the Jetson and neural network

2.3 User Classes and Characteristics

There are a handful of different user classes that will either directly use or implement this product. These classes are pilots, software engineers, hardware engineers, and researchers.

For the pilots, they will likely not be involved in the use of the product. As of now, they are simply required to fly the physical aircraft carrying the product, while other user classes handle the interaction of the product directly. The technical expertise of this user class is limited, since their skills are all centered around the operation of the aircraft itself. However, in an ideal employment environment, much further into the development of this project, pilots may be required to directly interact with the product. In this scenario, pilots will most likely collect the data from the product and compile it into a more understandable format for other user classes.

Software engineers are going to have the most overall use of the product during the development of this project. This involves not just the team developing the neural network, but future engineers who are going to update and improve this product and adapt it for other species down the line. Although they have more collective experience with programming the Jetson processor, it is important for the product to still have the documentation and proper coding practices to make it as easy to understand. This also applies to when our involvement with the product ends, and others continue the development and improvement.

The hardware engineers will only be involved with the physical payload requirements, with the current concerns being physical sizing of components and 3D modeling. There are operational requirements, such as limitations of camera mounting points, and efficacy aspects that can be incorporated into the design, such as battery hinges and a potential on/off switch. The payload case design must fit previous manufacturing requirements, be correctly designed to work within the UAV under its current setup, and not interfere with other components. Aside from operations concerning the payload case, the hardware team influences the primary product input: the payload camera. The setup they put in place for the type of lens and focal length of the camera installed influences all aspects of pictures that the product will receive. So, although this user class doesn't necessarily use the product, they are a large part of how effective the product will be in use.

The last user class are the researchers, better known as the customers of this project. This payload is designed to get them the data they need to monitor the turtle population. They will not be directly interacting with the code of the product, but most likely they will be using the interface of the software to compile the data collected. Even with their expertise in how the data applies to the overall mission, it is up to us to design the product so that it is as easy as possible for them to use.

2.4 Operating Environment

The hardware platform is a Jetson Nano, which runs Ubuntu Linux x64 18.04. The software operates in a real-time environment, having to continually take in images of whales, save them, and store the collected data. The software must communicate and co-exist with the camera system and the storage of data. The quality of photos from the camera in this environment is of fairly low quality, meaning that the system needs to be capable of discerning actual data in this environment. Currently, the payload of the aircraft is experiencing significant wear and failure, so adjustments must be made to create a new and better payload case to hold all the components without the deficiencies of the current system.

2.5 Design and Implementation Constraints

Hardware concerns are again a major limitation to the efficacy of the system, and constraints are placed on the payload case structure by the hardware team, which needs particular portions of the casing framework to hold the camera at specific angles, fit around the battery with the UAV chassis, and other specifications to keep the payload casing compatible with the other parts of the UAV.

The biggest limitation to the product is the quality of the camera used for gathering images, and the computing power of the processor. The actual images that will be taken by the UAV payload are of a lower quality. This means that the neural network is going to be designed around detecting very obscure and generalized algorithms of what defines a "whale". Due to this design constraint, the system is more prone to false positives, maybe identifying large wave patterns at the surface to be a whale. The storage on the Jetson processor severely limits the size of software that can be uploaded onto it, so the file size for the neural network must be considered.

These hardware limitations are the main constraint on the effectiveness of the project. However, it is also far too great of a challenge to improve the hardware, due to cost and vehicle platform concerns, such as power, weight, & thermal issues, so we will specialize our case design as much as possible while maintaining compatibility and refine our software to attempt to make it run on the Jetson.

2.6 User Documentation

The user documentation incorporates proper use and operation of the Jetson system. It also involves explanation of usage of the Autodesk Fusion tool and some clarification of proper methods of manufacturing (e.g. printing). Lastly the user documentation focuses on how to build and run the neural network system.

User documentation is broken down into two components: software design, and use. The software design documentation covers the process of how the code was developed structure wise, what libraries were used and what references the code was built upon. This documentation is more geared for future developers to use, in order to understand the current product so it can be improved and adapted for other projects. Use documentation is for the customer and pilots, regarding how to use the interface of the product and extract the data. This will mostly consist of user manuals, in regards to the actual use of the product compared to its design.

2.7 Assumptions and Dependencies

It is vital for this project that all the surrounding hardware which the system relies upon, most of which is beyond our scope of usage and maintenance, works as intended while in flight. This is a large project and without components such as the chassis not overheating, the cameras capturing images, or the drone flying at all, are required for the product to function. Thus, a critical assumption for this product is that the hardware is operating correctly on flights.

Our functionality assumptions include that we are currently focusing on whale detection, with no planned work continuing on refining the turtle or turtle track programs. For this version of the product, we are assuming that there will be no wireless transmission of data recorded in flight, and images will be saved locally.

In addition, we must assume that the dependencies for the software side of the project will continue to be available and work for our purposes. These libraries and dependencies include Theano, numpy, matplotlib, and OpenCV, among others.

3. External Interface Requirements

3.1 User Interfaces

The software is intended to run autonomously, so the interface with the user will mostly be the interpretation and presentation of results. Users should be able to enter dataset information into the program, for training the AI to recognize specific targets within those images (such as turtles) and store that categorization information to recognize similar targets within the input data that is received in-flight. Users do not directly supply the images received in-flight as those are sourced from the onboard camera during observation while aloft. Rather, images for training data will be entered (currently) individually into the neural network, which will develop a set of criteria to check further images for the expected targets. Entering training data, and receiving the sorted images, with expected targets boxed in on selected saved images, will be the main user interaction with the system, and thus form the user interface for utilizing the system.

For the payload mounting system, the majority of the user interface will be interacting with the pegs of the mounting system, and the mount itself. The fastener for the pegs will not be accessible to the users once the system is fully assembled. For the interaction with the pegs, it is required for them to secure the payload mount to the full payload bay while being accessible when the payload bay is fully assembled. The pegs also need to be sturdy enough to be a lasting solution for fixating properly-sized payloads to the payload bay.

3.2 Hardware Interfaces

As mentioned previously, the hardware interface for the project will be the mounting system developed by the team. This system is designed to have the users interact only with the peg and mount of the system. The mount can be designed to accommodate multiple payloads to fit into the payload bay, with the current focus being the Jetson Xavier and camera rig.

3.3 Software Interfaces

The bulk of the software will take the format of the algorithm for classifying images, which will be created on an interface of student computers, then edited to run on the Jetson Nano. This is to be accomplished by using a neural network; it will be implemented via Python development in Linux Terminal and/or VSCode version 1.71. The neural network is composed of inputs, which will be the user-side information provided as .csv files of annotated images, a layer of nodes within which comparisons, criteria selection, and data analysis will occur, commands for the inference server and model calls, as well as python files like bbox.py (an output-layer program which will save and edit specific inputs, to provide important information to the user, like listing the coordinate boxes of the whales within the images). The interface will rely on a Python script and neural network developed in Python, the input images and reference datasets being entered into the system, and the operating system of the Jetson board in order to operate as intended.

3.4 Communications Interfaces

The communication of the system primarily occurs within the Jetson itself, with API calls to external inference server to source model weights. The neural network is currently composed of inputs, nodes, and outputs, layered to create an adaptable system that can potentially be trained on different data sets to recognize specified items in photos – as of now the objective has shifted to whales from turtles, but may shift again in the future, especially depending upon the available dataset images. As for input images, data sharing will need to occur within the Jetson between the neural network and its inputs, which can be expanded out to include data transfer between the Jetson onboard the drone and the camera, which will need to transmit input images to the software for processing. Overall, the communications interface core segments will be the communication within the neural network, and that of the camera to the Jetson to correctly transfer good inputs to be processed – as regardless of how the program is designed, the quality of the output information will be affected by the quality of the inputs entered into the system.

4. System Features

4.1 Whale Recognition Algorithm

4.1 Modify Code to be Able to Detect Whales in Real Time

4.1.1 Description and Priority

The first overall goal of the project is to recognize and identify whales in real-time, from the Nvidia Jetson, deployed on a drone. This is one of our overall goals and is a task of the highest priority. The first portion of this requirement is to develop and train a model that can detect whales on a computer.

4.1.2 Functional Requirements

The Nvidia Jetson must be able to be powered on and execute the code given to it. The program must be able to detect whales in real-time using the Nvidia Jetson. The neural network shall analyze multiple images. The Neural Network shall draw a box around the identified object in the image. The neural network shall give a level of confidence for the identified object.

4.2 Payload Mounting System

4.2.1 Description and Priority

Two Components to the payload will be modeled using Autodesk Fusion 360. One is a rotating peg that locks the payload in place. The second is a pivoting Battery Case Holder that makes the battery accessible for charging. Once the models are built, they are presented to the hardware team on their feasibility to the payload. Once models are approved a 3D printed prototype will be built.

4.2.2 Functional Requirements

The payload mount must keep the components (cameras and jetson processor) in a secure enough position to not affect the quality of captured photos during flight. To achieve this requirement, the tolerances between the connecting parts of the mounting system must be made to achieve firm, but moveable connections. This includes the surface between the mounting plate and mounting peg, along with the mounting peg and the floor piece that the peg is snapped into.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

- The Nvidia Jetson must contain a neural network that processes image recognition in real time. The Image recognition must be processed at efficient speeds.
- The Nvidia Jetson must operate as intended, powering on and running the code loaded onto the machine
- The neural network must process images of whales and accurately recognize them.
- The rotating peg must lock the payload in place
- The pivoting battery case holder must have a smooth rotation and pivot with a 90-degree angle.

5.2 Safety Requirements

- Screws must not be used to avoid damage to the casing of the payload
- The rotating peg must not be able to overtighten damaging the casing of the payload
- The code must be stable and not damage what is processing the data

5.3 Security Requirements

- A password to access the jetson will be used so the code is not shared to unauthorized individuals.
- The Jetson must be stored in a secure environment.
- A password is needed to access the stored Autodesk Fusion 360 files

5.4 Software Quality Attributes

- The code must be organized and easy to read.
- The code must be able to be testable given different images of whales
- The box surrounding the identified image should be clearly defined

5.5 Business Rules

- The software team will perform modifications on the code.
- The hardware team will focus on maintaining and solving hardware constraints.
- Our scrum master will handle the organization of team goals and define responsibilities of each team member.
- Each member will work on their share of documentation.

6. Other Requirements

Currently there are no other requirements, but we will add other requirements in the future

Appendix A: Glossary

Pretrained model: A pretrained model is a saved network that has been previously trained on a large dataset.

Model-zoo: A list of pre-trained model that are trained on the COCO 2017 dataset. These models are useful to use as you can use the pre-existing labels, or alter them for your use-case.

Neural Network: A model that simulates the function of a human brain. A neural network consists of a series of layers, hidden layers which include nodes that have weights, and output layer.