System Requirements Specification

for

TurtleTech

Version 1.0 approved

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

TurtleTech

3 December 2022

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

1. Introduction

1.1 Purpose

The purpose of the TurtleTech project is to produce a functioning neural network system on a Nvidia Jetson device that identifies aerial turtle images against non-turtle images in real-time. This document provides a detailed description of the software requirement for the TurtleTech project and will explain the interface requirements and all of the system's features. This SRS will cover the whole functionality of the TurtleTech system.

1.2 Document Conventions

This will have to be completed as we write the document and make special indications

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 TurtleTechs neural network system.
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 the TurtleTech project is to produce a functioning neural network system on a Nvidia Jetson device that identifies aerial turtle images against non-turtle images in real-time. 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 functioning neural networking system by the end of the Fall 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

Sea turtles are an extremely important part of the ocean's ecosystem, managing other species' populations and even providing habitats for other marine life forms. Unfortunately, many sea turtle species are endangered due to primarily man-produced issues. The protection of sea turtles is crucial

for maintaining the health of the world's oceans, and this product helps one specific effort in that conservation. (talk about turtles being an indicator)

The current method for tracking sea turtle migration is through satellite telemetry, which requires a physical transmitter to be placed on the turtle's shell. This process requires a lot of equipment and manpower, to go out and get trackers onto sea turtles and monitor their position. (add more detail)

Turtletech is an initiative that is researching how to track sea turtles in a more effective way than currently available, by using UAS (unmanned aerial systems) to track loggerhead sea turtles off the coast of Florida. Unmanned aircraft don't require nearly as much man power or equipment to operate, and can survey large amounts of area in a short amount of time. (add more detail)

Our product is software that will be used on Turtletechs UAV (unmanned aerial vehicle) which will autonomously track turtles seen by the payload camera in real time. (validate & expand statement)

2.2 Product Functions

The product shall:

- Apply a visual neural network algorithm to images taken in real time
- Visually identify any turtles and/or turtle tracks seen in the images
- Save images with visually identified turtles and/or turtle tracks with time stamp
- Have proper documentation for functionality and development
- Be adaptable for tracking other species (whales)

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 thermal and electrical. Aside from basic operational requirements, 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, process 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 seeing issues with potential over-heating and brownouts, so the product may have to incorporate some sort of temperature recording if possible.

2.5 Design and Implementation Constraints

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 "turtle". Due to this design constraint, the system is more prone to false positives, maybe identifying random flora floating at the surface to be a turtle. The storage on the Jetson processor does limit 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 (power, weight, & thermal).

2.6 User Documentation

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, in regards to 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. Along with user manuals, troubleshooting guides/methods will also be included.

2.7 Assumptions and Dependencies

The most critical assumption and dependency for this product is that all of the hardware, being the camera and processor, will work properly while in flight. Without these components, the product cannot function. Part of this assumption is that the components will have a consistent power source for the product so it can function properly.

Product function assumptions are currently that we are focused solely on turtle detection, with whales being a potential future add on. For this version of the product, we are assuming that there will be no wireless transmission of data recorded in flight, and images are saved locally.

The dependencies for the software portion of the project relies heavily on Tensorflow. Along with Tensorflow, we use some other libraries, but one of the main additional libraries we use is object_detection, cv2, numpy, and matplotlib.

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 via folder 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.

The development environment for the algorithm will be VSCode, able to compile and debug Python programs, as the neural network will be created using Python language before being ported to the Jetson AGX Xavier. Porting the programs to run on the Jetson Xavier will require using a Linus terminal to command-line copy the algorithm over to the SD card of the Jetson (this will also require a converter, likely micro-SD to USB, to connect a computer and the storage SD Card. These software development tools have individual specifications and designs, which will be explored further as programming progresses, and form the minutiae of the interface the user will be using to design the system.

3.2 Hardware Interfaces

The hardware for this system is a Jetson AGX Xavier interconnected to a Censys Sentaero. On board along with the Jetson are additional payload devices, including cameras in the wings, nose, and fuselage of the plane. The commands for flight of the plane are delivered by control inputs transmitted via an antenna nearby the flight area, and supported devices fall under the 24V voltage that can be supplied by the drone battery. This interface will not be a difficulty for our Jetson, as it is rated for 12V. However, the Jetson is currently exhibiting a recurring failure where it stops recording and processing data around 15 minutes into each flight. The current solution being implemented for this problem is providing a separate power source (Li-Po battery) that only powers the processor.

This error has not been reproducible in lab conditions, but in order to solve this issue with the hardware interface, we are currently proposing tests to check functionality and attempt to troubleshoot the shutdowns. In addition, the hardware of the Jetson itself limits the size of the program (neural network) that can be run on the system in-flight. We are restricted to the storage of the Jetson processor to collect and process images in our algorithm. Storage on the Jetson processor also has to account for the neural network size alongside the storage of identified images. he Jetson

hardware imposes some restrictions on the capabilities of the program, and these limitations will likely have greater effects on the development of the program as it progresses, which will be covered in further documentation outside of the hardware interface.

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 VSCode version 1.71 and/or Jupyter Notebook. The neural network is composed of inputs, which will be the user-side information provided, a layer of nodes within which comparisons, criteria selection, and data analysis will occur, and an output layer which will save specific inputs, which have been processed to ensure selected targets are within them, and will also be edited to provide important information to the user (such as boxing-in the turtles within the images). The interface will rely 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. The neural network is not defined in its entirety currently, but will be 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 is turtles, but 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 System Feature 1

4.1 Recognize turtles in real time from Nvidia Jetson

4.1.1 Description and Priority

The overall goal of the project is to recognize and identify sea turtles in real-time, from the Nvidia Jetson, deployed on a drone. This is the highest priority task, as once we can accomplish it, we can add additional features.

4.1.2 Functional Requirements

Be able to sustain flight for more than ~15 minutes. The current issue deals with the Nvidia Jetson shutting down around 15-20 minutes of flight. For the above feature to work correctly, we wish to fly the drone hours at a time and to capture images of turtles in real time.

4.2 Record coordinates of turtle sighting

4.2.1 Description and Priority
Once a turtle is spotted or another label is spotted with the camera, a set of coordinates will be recorded and embedded in an image. These coordinates are related to the location where the drone saw the turtle or turtle track. This will be used for research purposes, to further identify patterns in sea turtles.

4.2.2 Functional Requirements
The Nvidia Jetson does not have an onboard route to record coordinates. However, currently we are looking at a solution that can be used on some Linux distros, it is called the "whereami" tool. To install this, we need to install npm, which is a javascript package manager.
see: https://ostechnix.com/get-geolocation-commandline-linux/ for more information.

4.3 Recognize turtle track on the beach in real time from Nvidia Jetson

4.3.1 Description and Priority
In tandem, when training our turtle dataset, we have some images of turtle tracks on the beach. We will add the turtle track label to those images.

4.3.2 Functional Requirements
When training our turtle dataset, we will add an additional label and identify pictures with turtle tracks, to add another element to the neural network.

4.4 Recognize whales in real time from Nvidia Jetson

4.4.1 Description and Priority
As well as training both turtle images and turtle tracks, we plan to eventually add the ability to recognize whales. This is not as high a priority and we will likely leave this for phase two (Spring semester).

4.4.2 Functional Requirements
We have obtained a database of whales from Kaggle, as well as a database from British Antarctic Survey. We plan on training the network with a label "whale", with these images.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

- The Jetson must contain a neural network that processes image recognition in real time. The Image recognition must be processed at efficient speeds.
- The hardware must contain enough storage to fit the data being stored.

- When the neural network identifies a turtle or their tracks, it must mark the recognized object with a box and name.
- Each identified object will have a confidence level attached.
- Each Image will have a watermark will have a timestamp.
- The software must be able to fit the strict Nvidia Jetson Xavier storage requirements ~8GB.

5.2 Safety Requirements

- The software on the Jetson must be stable and not risk damaging the hardware.
- The connected hardware must not exceed a voltage of 12v.

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.
- When communicating between the drone and ground, it is important that it is not possible to intercept the data being transferred.

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, such as whales in the sea or turtle footsteps on the beach.
- The box surrounding the identified image should be clearly defined
- The timestamp should have the correct time attached

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.

Appendix B: Analysis Models

Currently there are no items for the analysis model

Appendix C: To Be Determined List

Currently there are no items to be added to the to the to be determined list.`