WildTracker
Interface Control Document

Prepared by	Alexander Santander, WT18G4-OD	Date	29/10/2018
Checked by	Delenn Palmer, WT18G4-PM	Date	29/10/2018
Approved by	Alexander Santander, WT18G4-OD	Date	29/10/2018
Authorised for use by	Felipe Gonzalez, Supervisor	Date	29/10/2018

Queensland University of Technology
Gardens Point Campus
Brisbane, Australia, 4001.

Revision Record

Document Issue/Revision Status	Description of Change	Date	Approved
1.0	Final Issue	29/10/2018	Alexander Santander

Table of Contents

Paragraph **Page No.**

List of Figures

Definitions

WT	WildTracker - Software implementation of an autonomous detection and tracking tool for wildlife
QUT	Queensland University of Technology
HLO	High Level Objectives
PMP	Project Management Plan
GUI	Graphical User Interface
TF	TensorFlow
CNN	Convolutional Neural Network
OpenCV	Open Source Computer Vision
OD	Objective Detection
DL	Deep Learning
ML	Machine Learning

1 Introduction

To ensure that a system is constructed properly, all of the designed interface connections between each piece of software needs to be outlined. The Interface Control Document (ICD) is where these interfaces are explained. Defining these interfaces in this manner allows the subsystems to be constructed separately while ensuring that they can be integrated at the final integration phase.

1.1 Scope

The ICD provides an overview of all of the connections and interfaces between all the elements of each subsystem. This document is a point of reference for a user to view or check any of the interfaces within the system. This document does not contain information relating to why a piece of software was chosen (as this can be found in RD/4), but does discuss the connection between each software's interface.

1.2 Background

QUT ASL is a world leading research centre based in Brisbane, Australia. They conduct research into autonomous technologies which support the development of autonomous aircraft or drones for remote sensing with on-board sensor systems for a wide range of commercial applications.

The QUT Airborne Systems Lab (ASL) has commissioned students of EGH455 in collaboration with WWF and Wildlife Australia to design and build an autonomous detection and tracking tool for wildlife. Group 4 has been tasked with designing an Unmanned Aerial System (UAS) application that must have the ability to identify and report the number of, the size of and volume of wildlife present in footage retrieved by drones. In addition, the data acquired and processed must be accessible both in the real-time use of the application and after the video has been exported.

2 Reference Documents

2.1 QUT Systems Engineering Documents

RD/1	WT18G4-SUP-Customer Needs	Autonomous detection and tracking tool for wildlife
RD/2	WT18G4-SR-01	WildTracker Project: System Requirements Document 2018
RD/3	WT18G4-PMP-04	WildTracker Project: Project Management Plan 2018
RD/4	WT18G4-FD-01	WildTracker Project: Final Systems Design 2018

2.2 Numbering Scheme

For ease of identification, a numbering system has been developed.

For the requirement REQ-M-01:

REQ – This is a requirement derived from the client's brief and the associated HLOs

M – Denotes a *mandatory* requirement, whereas D denotes the *desired* requirement.

3 System Overview

The two subsystems in which data is generated and analysed is the Graphical User Interface (GUI) and the Object Detection (OD) system (which can be split into two sections), one that processes the data and the DarkFlow model which analyses the video. The GUI has inputs such as the file name and location, as well as the threshold and elevation values. These values must then be passed through to the backend for processing. From there, each frame is sent to the DarkFlow model for analysis. After said analysis from the DarkFlow Neural Network, the information is sent back to the GUI so that it may be visually updated. The connections by which the data travels from one subsystem to the other is detailed in the sections below.

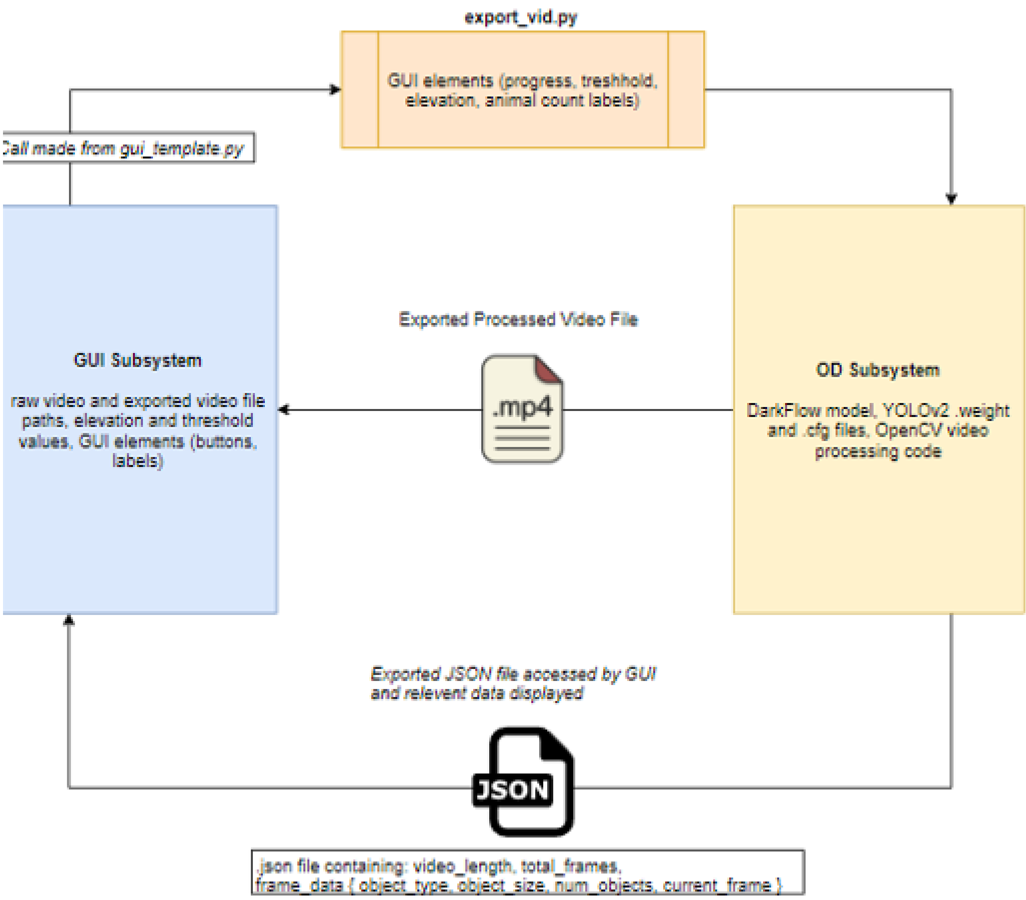

Figure 1 – GUI & OD Subsystem Connections

4 Subsystem Interface Specifications

4.1 Subsystem Interface 1

The first and most notable interface connection would be that of the GUI to the backend processing of the video. Once the video is loaded in to the front end and the user hits the "play" button, all required variables are called through the function definition within the file that analyses the video. This includes: the progress of the video, the filename, the location of the file, the threshold value and the elevation value. From here the filename and destination are used to initially open the video, the progress value is used to ensure the status bar was visually updated, the threshold and elevation values are used to calculate the tolerance of the analysation and the sizing of the animals respectively. The diagram below illustrates the flow of information.

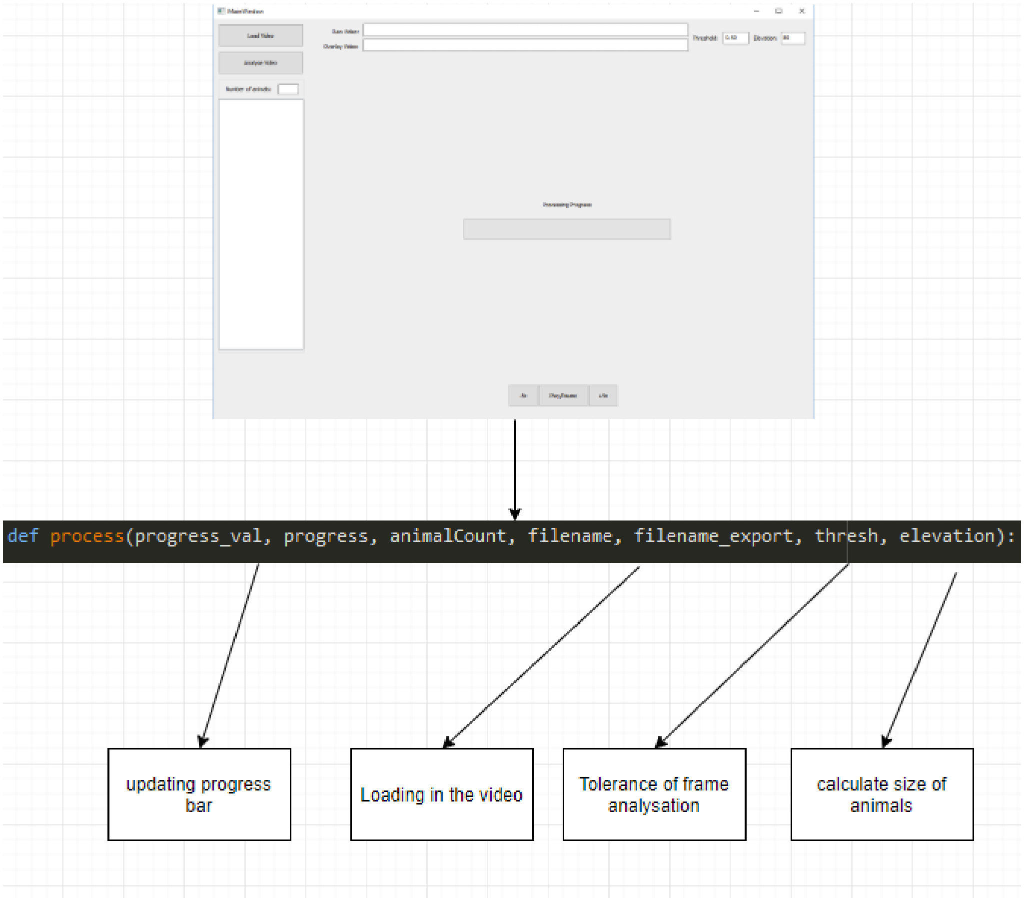

Figure 2- GUI & Backend Connection

4.2 Subsystem Interface 2

This second subsystem interface involves sending each frame to the DarkFlow model for analysis. The actual transferral of data is rather straightforward. Once the model has been initialized with the set weights, a simple function call is made to pass through the current frame. This functional call is a simple 'for loop', which iterates through each frame being sent through.

```
# Perform detection on frame
results = tfnet.return_predict(frame)
```

Figure 3 - Analysis of Frames

4.3 Subsystem Interface 3

Once the information has been processed through the backend, it must then be returned to the GUI, so the video can be visually updated. The most efficient way of completing this was to create a JSON object that passes through all the data required. This object contains arrays of information including: the current frame, the number of animals within the image, the type of animals and whether the animal is large or small. These variables were gathered by directly reading them from the DarkFlow model, feeding data through 'if/else' statements and certain functions that were used for calculations.

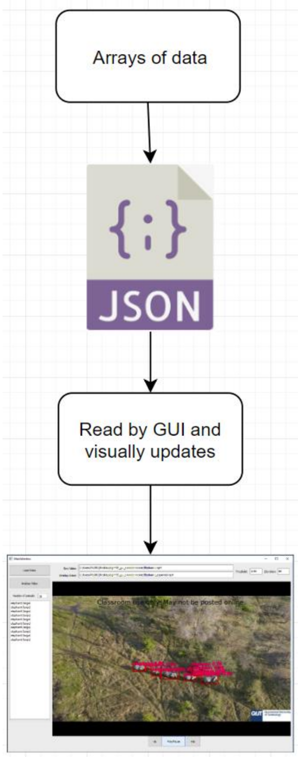

Figure 4 - JSON Updates GUI

5 Conclusion

The means by which these three interface connections were created was one of the most challenging tasks of this project. The most important facet of machine learning is the manipulation of data so that the model may read and analyse in the most efficient way. To facilitate this, all avenues by which the data flows must seamlessly interconnect. Through creative thinking and reasoning this smooth connection was achieved.