Date: March 19th, 2019

From: Adam Kwok

To: Doctor Kaputa and anyone else interested in my work

Subject: PyQT4 utilizing OpenCV Tennis Ball Detection Analysis

Introduction

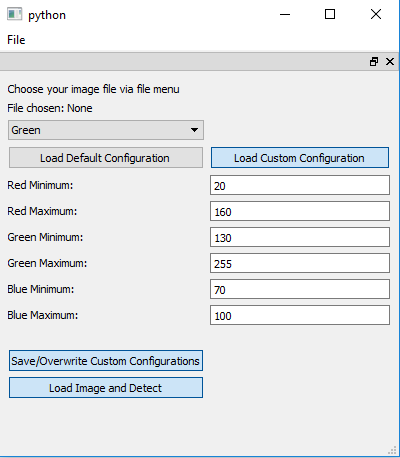
For the CPET-563 drone project, we utilized OpenCV for the detection and tracking of tennis balls. The important factors in this design are that it was developed to include capabilities of loading and saving configurations from a file, selecting tennis ball color (green or blue), loading different images from a file menu, and displaying the x and y coordinates of the ball. These requirements were to ensure the following:

* Repeatability of tracking across a wide variety of images
* Consistent tracking across the necessary colors
* Experience populating python variables from configuration files
* Establishing a clear point for the drone to track
* Determine the limits (especially distance and cover) of OpenCV tennis ball tracking

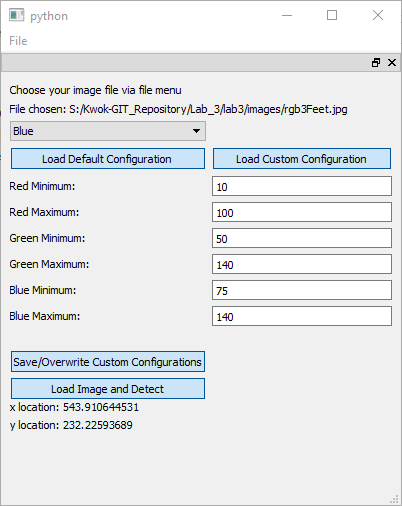
The below analysis investigates the limitations and error cases established and recognized for the future implementations. This module was designed for detecting tennis balls in image files, but with the intended use of being implemented on a Snickerdoodle’s FPGA and utilized with a video feed.

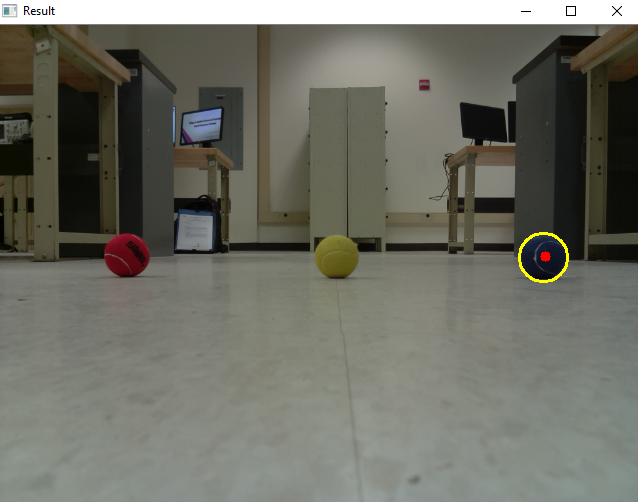
Analysis

With the goal of detecting tennis balls, the OpenCV application was focused on homing in on color and shape. This meant that the minimums and maximums of red, green, and blue were established for the necessary shades of blue and green targeted these values comprised the “default” values that the user could always populate the data fields with the default of the color selected in the combo box with “Load Default Configuration”. Selecting “Load Custom Configuration” would populate the data fields in a similar manner, but with values derived from configuration files respective with which color was selected in the combo box.



These configuration files could be overwritten by modifying the values in the PyQT line edits and selecting the “Save/Overwrite Custom Configurations” button. When the “Load Image and Detect” button is hit, the image with the tennis ball detected is displayed and the x and y coordinates are displayed beneath the load and detect button.





Conclusion

* Because the code provides clear x and y coordinates for the centroid, the drone is provided with a single point to track which aids in each of clarity.
* Since error was found tracking when there was a similar color in the same image, developers should take note of background colors before tests.
* Providing the user with a default configuration setting as well as a custom configuration file to modify allows the user to avoid overwriting optimal settings while still allowing them to customize their functionality right from the gui.
* Obscuring the tennis ball results in a skewed centroid and complete loss of tracking once more than approximately 75% of its face is covered.