

My project is a drone that can scan a store shelf and determine if the representation of a specific brand is adequate for what the vendor paid for. In most retail stores, a brand will pay for space on a shelf in order to be more eye catching to the customer and eventually get more sales, which is why bigger brands are generally seen more. However because of either employee ineptitude or general non-compliance up to 15% of a brand's space on a shelf that it paid for can be lost, which can lose the brand upwards of 5% loss in overall sales.

Several apps have been released to fight this such as Clobotics, however they all require an employee to go around manually and take snapshots. I wanted to build a system where a shop owner could simply place the drone at a starting point, click a button and have their inventory done automatically within minutes.

Since all inventory in a shop is usually on the shelves and I wanted to not have the drone rely on GPS I fitted the clover's pi camera in front for object detection and navigation. Rather than using map-based navigation since one can't expect a shop owner to place markers perfectly on the ground, opencv2 has a built in aruco marker detection method that identifies and draws a bounding box around all markers in frame. So once the drone takes off from the starting position, it identifies where the marker is in frame and uses the clover's `navigate()` method to center the marker and its area to place the clover approximately 1 metre from it.

What's expected is that the shop owner places markers at every different brand's section in ascending order down an aisle. The drone has a specific marker ID it's looking for and once it's centered itself on the nearest marker, it determines its ID and moves either left or right depending if the identified marker is lower or higher than the target. This way the clover moves down an aisle stopping at each marker until it reaches the target.

In order to identify the number of items of each brand, I've trained a tensorflow model and stored its metadata in a caffemodel file for the python script to pull from. Every time the drone stops at a marker, a method is run that applies a convolutional neural net to the frame and determines how much of a specific item (right now the model is trained to recognize soda cans but can easily be taught other items) it at said marker (remember that each marker is to be placed at each brand's individual section). In the video below, you'll see a POV video of my drone with two markers placed on the wall in my garage. What happens is that the drone takes off, centers itself on the first marker and as its ID is 140 and the target ID is 138, it's driven left to the second marker which has a soda can hung next to it. It then identifies the soda can, conveying the number of cans of that brand to the shop owner before centering itself on the marker and landing due to it reaching the target marker.

My main goal with this project was accessibility. I wanted to make a drone that could navigate purely based on its camera with computer vision tools that aren't exclusive to MAVLink or ROS drones. By fitting the camera to the front not only would a user be able to get more functionality out of it than simply map-based navigation but would be able to make more useful applications such as shop keeping or security through facial recognition. With alterations to the left, right,

front, back, up and down commands this script can be applied to any hackable drone and requires little to no prior experience to use.

Check out the video below to see it in action:

<https://www.youtube.com/watch?v=jsVY0DM9Sew>