Read Me:

Execution Environment:

Python 3.9 (https://www.python.org/downloads/)
Stable Open-CV Release (python3 -m pip install opencv-python)
Stable Numpy Release (python3 -m pip install numpy)

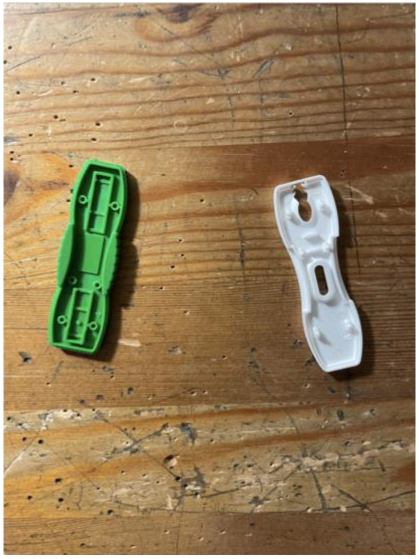
Running the Demo Code:

```
python3 detect_orientation_cartridges.py <path to test image>
i.e.
python3 detect_orientation_cartridges.py test_images/img.png
```

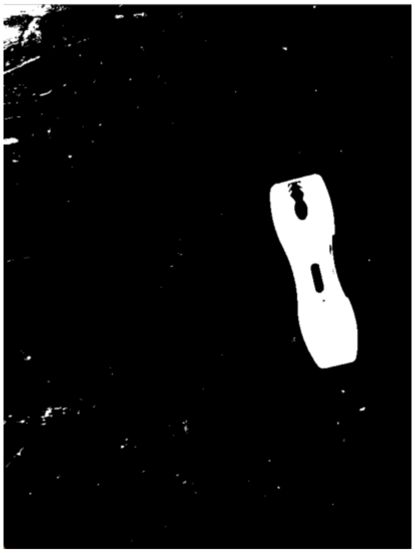
Algorithm:

The description of the algorithm is given in the comments in the code. However, here is a quick summary of how it works.

1. Start with the Raw Image

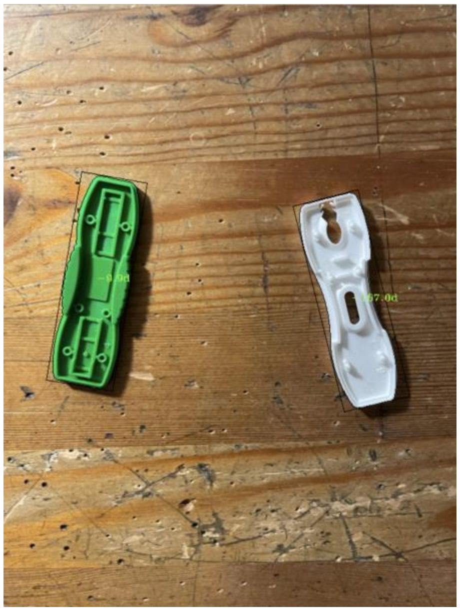


2. Convert the RGB image to the HSV color scale and apply a threshold to only select the white/green cartridges



We have segmented the image by color and are only left with the white sample.

3. Detect Contours and fit a Bounding Box



The OpenCV Find Contours function is used to identify the cartridge contours. Then the open cv minimum area rectangle is used to fit a rotated rectangle to the contour. Lastly, the contour moments are calculated, and we use simple relationships to determine the rotation of the contour (see https://en.wikipedia.org/wiki/Image_moment#Examples).

As you can see in the above image, the orientation angles of the contour are labeled in green. The green sample is rotated 9.9 degrees CCW, while the white sample is rotated 167 degrees CCW.