* What did you try and why do you think it did not work?

Initially, I tried detecting the cones and boundaries automatically using edge detection and line detection techniques. However, this method wasn't always reliable due to lighting issues, shadows, and unclear cone boundaries, which often led to lines being incorrectly detected or placed in the wrong areas. I also experimented with manually drawing the boundary lines based on the known positions of the cones, but this approach only worked for specific images and couldn't adapt to different cone placements, making it less flexible. Ultimately, I settled on a compromise by using a combination of automatic detection with some manually set rules, such as filtering lines based on their slope. This allowed me to improve the accuracy and robustness of the detection, ensuring that the red boundary lines followed the cones more reliably.

The libraries I used were OpenCV numpy and matplotlib.