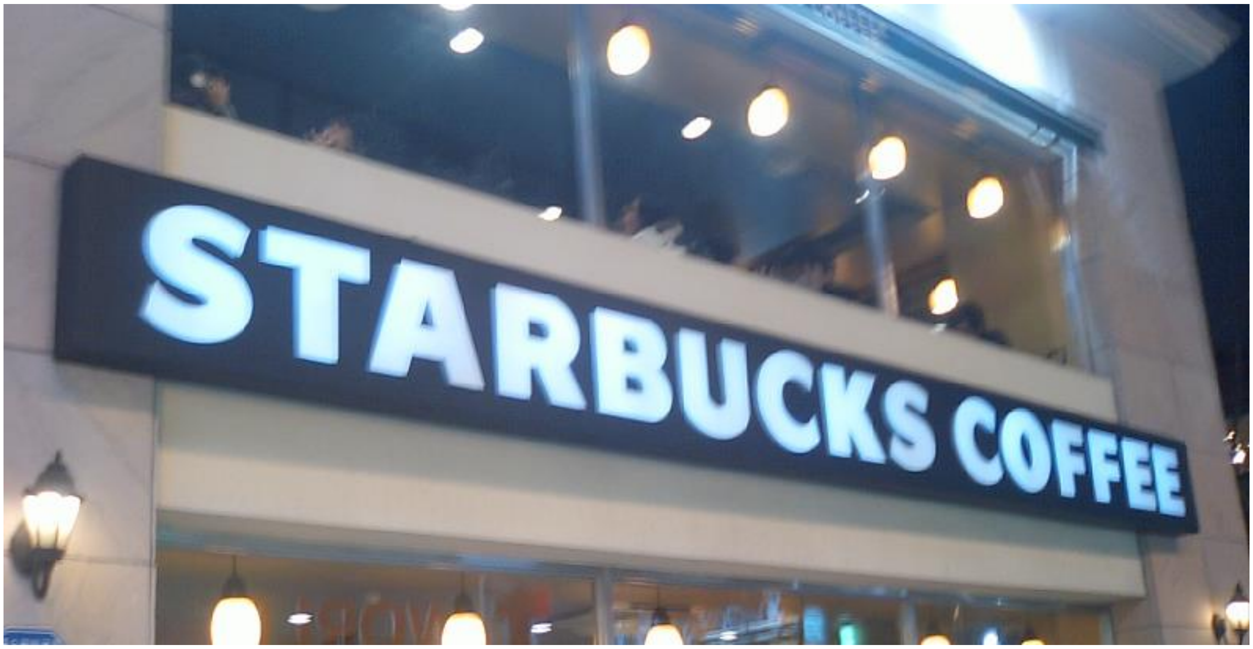
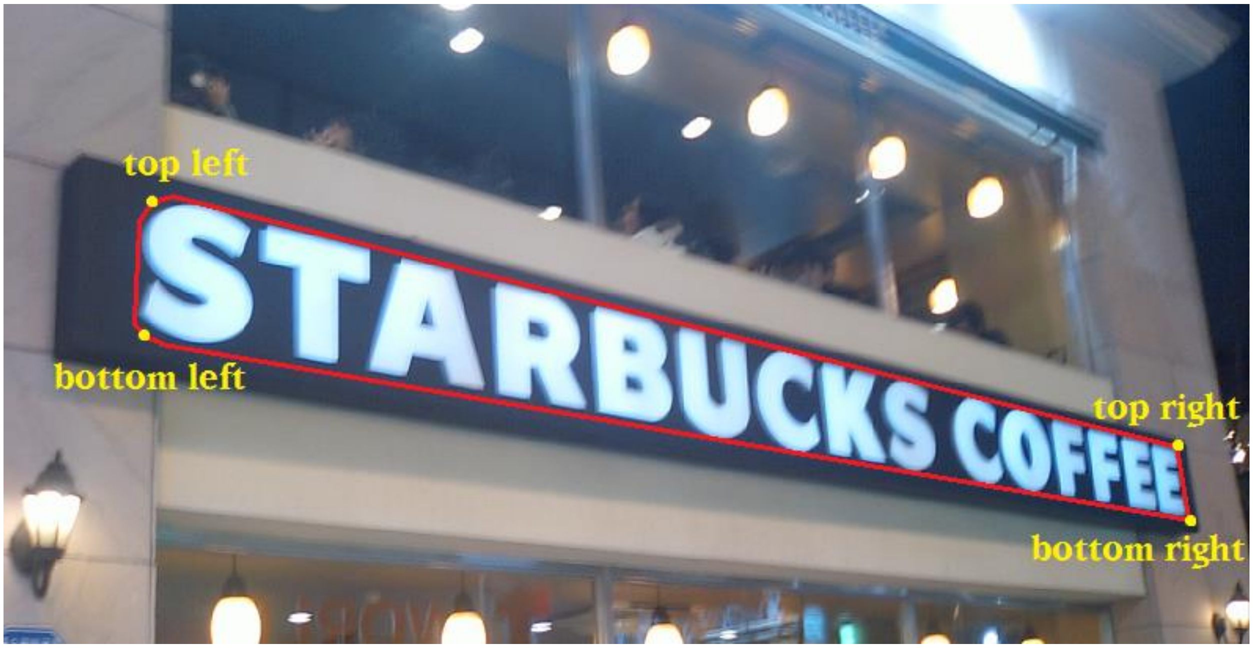
Alternate pipeline, mainly to handle noticeably **slanted** text (proposed pipeline **doesn’t** use EAST):

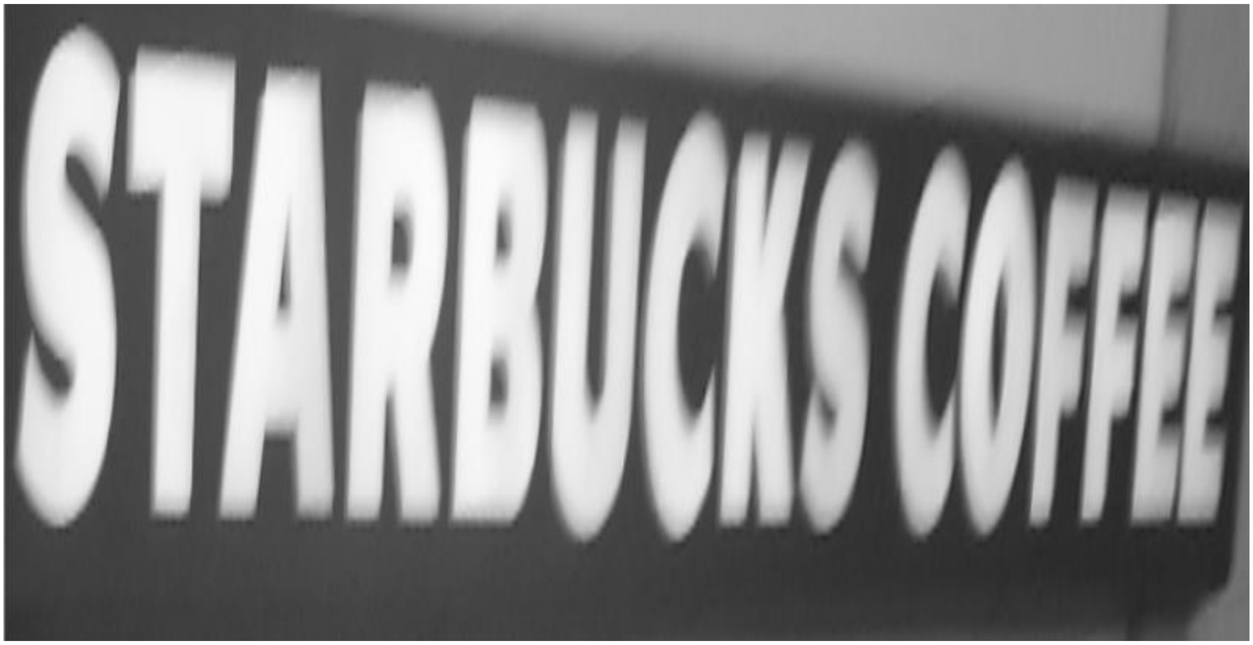
1. Use Canny edge detection on image, prioritize removing edges over keeping them
2. Assume the remaining edges left are part of the text in the image in some way
3. Turn the edges into contours
4. If other contours big enough relative to largest one, assume these contours are text detections
5. For each contour, use the contour’s shape to perform skew correction (if necessary)
6. For each skewed ROI, read the text directly w/ Tesseract



Input image.



Ideal **convex** contour shape (drawn in red) that would be picked up using these 4 corner points (yellow) found based on Euclidean distance from the top left, top right, bottom left and bottom right corners of the entire image.



Final result a perspective transformation using the 4 corners as input. Width and height for this new image should be assumed from estimated width and height of the contour that the 4 corners were associated with. For further reference: https://www.pyimagesearch.com/2014/08/25/4-point-opencv-getperspective-transform-example/