Quality Checks and Improvements

Occlusion Check

Complexity: 6.5, Potential for improvement: 8

The occlusion check basically checks if a given pair of annotations overlap each other in two dimensional space. We measure the estimated occlusion by a simple formula of taking the average between the occluded Y space, and occluded X space, if and only if they overlap in both dimensions. The algorithm is a brute force iteration through the entire list of annotations, checking if a given annotation occludes with another, which is inefficient. Furthermore, we could improve by dynamically generating severity levels according to the ratio of measured occlusion relative to the occlusion value given

Stray Click Check

Complexity: 2, Potential for improvement: 3

The stray click check merely checks if an annotation is the consequence of a stray click on the task. It checks the size of the annotation, rendering an annotation unlikely if it is too small, or contains a float sized width/height
We could improve this by dynamically generating severity by measuring the ratio between the size of the annotation and the minimal threshold to trigger it.

Color Check

Complexity: 8.5, Potential for improvement: 9

The color check performs a check on objects that are annotated with the incorrect background color. This check is prone to errors since it measures the color difference in euclidean space, and shades can be interpreted wrongly. We can improve by averaging the colors in the background, and searching for a closer match, or tuning the RGB colors we might expect to see in the tasks.