Project Overview and Structure (Han):

This research project was done based on the work and code from [Core ref.]. Basically, there are a few critical steps involved, and they are pB edges, superpixels, parametric maxflow grouping, and recursive object closure detection. The following is their brief descriptions, and their details will be in the next sections.

The first step is to compute the pB boundaries for the image. The pB boundary detector was first proposed by Martin et. al. [ref] to combine brightness, color, and texture as local features to detect boundaries for natural images. The result of the pB boundary detector is used to help use determine if the contour as enough edge support. The more edge support we have, the more confident we are about the selected closure.

The second step is to compute superpixels, which are a higher level than the fundamental pixels in an image. Since we intend to use a group of adjacent superpixels to represent an object and use its boundary to approximate its closure, generating superpixels and labelling them for an image are necessary for this project. There are a few different algorithms to compute superpixels, and we chose SLIC, NCuts, and turbo pixels as the main candidates. We also intend to compare the results by using these three different superpixel algorithms.

The third step is to group multiple adjacent superpixels. According to [Core Ref], the grouping of multiple superpixels together to get their closure as an approximation of the boundary of an object can be formulated as an optimization problem. This problem can then be reformulated as an energy flow problem and can be solved by parametric maxflow.

The last step is to wrap the above three steps as a function and use it to detect the objects on the same level and the next level. For the same level object detection, we intend to search all valid objects with the criteria of reaching a certain size. For the next level objects, we intend to search all objects within any object in the parent level. By recursively repeating these two steps, we can find a tree-structure, which can represent the relationships among all possible objects in an image.