The Algorithm of Recursively Extracting All Objects (Han):

The benefit of using closures as a means to detect objects is that it not only gets a boundary for an object, but also tells a scope for other objects. Therefore, within a boundary of an object, we can potentially find the objects in the next level. Outside the boundary of the object, we can find other objects at the same level. Therefore, because of these two mechanisms, we can potentially find all possible objects in an image. Furthermore, when the process finishes, a tree-structure should also be built to indicate the relationships among the objects. The following is the pseudo code for our algorithm of searching for all objects in an image.

function **extract\_objects**(image, mask, tree\_node):

if (mask’s size is more than a certain threshold size):

return;

end;

masks\_holder = find\_objects\_same\_level(image, mask)

while (masks\_holder is not empty)

its\_mask = get\_first(masks\_holder);

add its\_mask as a child\_node of the tree\_node;

**extract\_objects**(image, its\_mask, child\_node):

save its\_mask to a file;

end;

end

function **find\_objects\_same\_level**(image, mask):

initialize m\_holder for masks;

a\_mask = find\_an\_object(image, mask);

while (a\_mask > certain size threshold)

add a\_mask to m\_holder;

mask = mask & not(a\_mask);

a\_mask = find\_an\_object(image, mask);

end;

return m\_holder;

end;