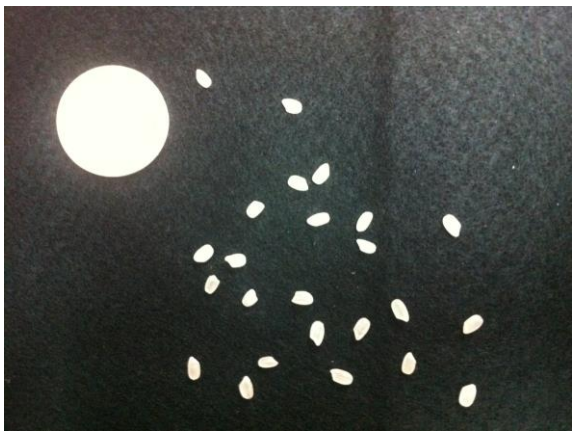
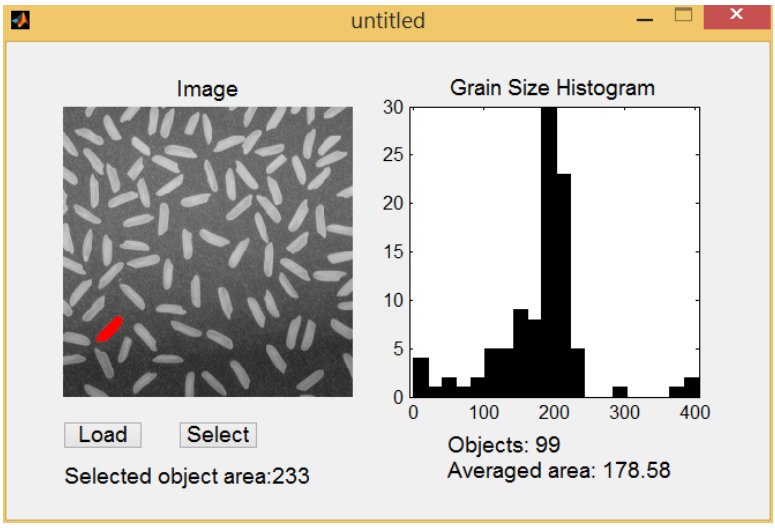
Lab 9

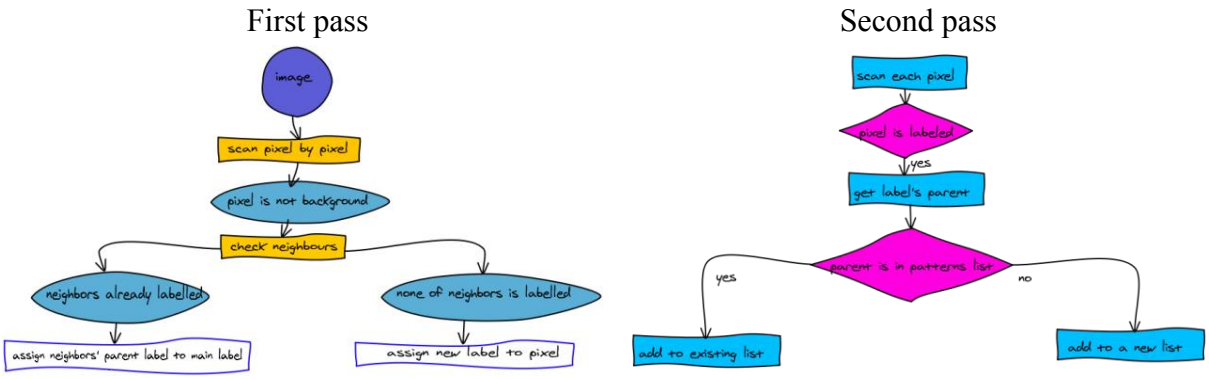
1. Write a (not necessary GUI) program to calculate the number of rice grains for the given image "09Rice.jpg" (see below). Identify the size, centroid, perimeter, major axis length, minor axis length, and orientation for each grain. A ten-dollar coin was put with the grains when the photo was taken; hence, you can calculate the actual sizes of the grains rather than providing their sizes in pixels. Try NOT to use built-in function `regionprops`.



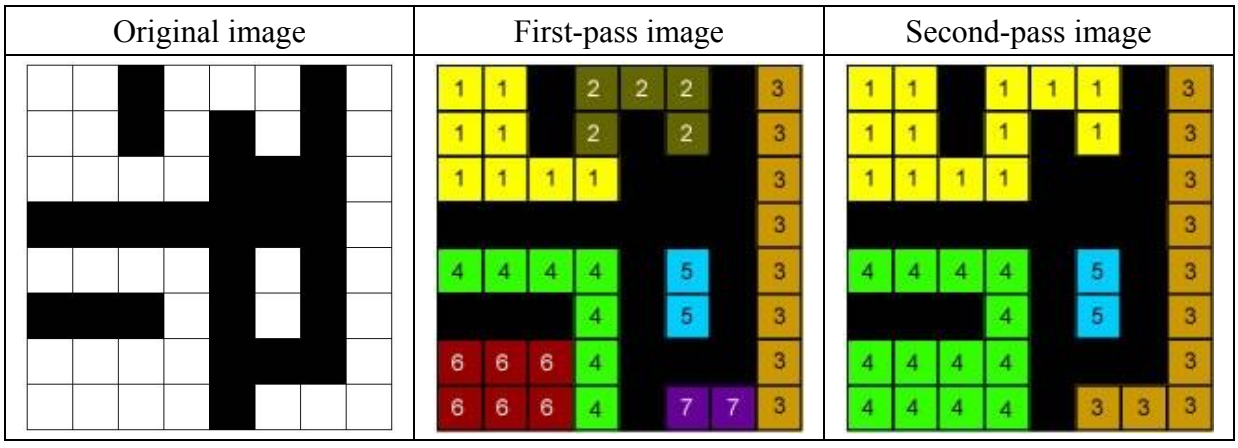
2. Design a GUI program to analyze the "09rice.png". The program should allow the users to load an image and then display the histogram of the connected components in the image. The program should contain a "select" button with which the users can interactively select objects to be analyzed in the loaded image. Try NOT to use the built-in functions for calculating the grain statistics. You may need the built-in function `ginput` to allow the users for selecting objects of interest.



3. Connected-component labeling is an algorithm used to detect connected regions in binary digital images. The algorithm subsets of connected pixels in an image a unique number. The algorithm can be implemented using a two-pass approach. The first pass assigns temporary labels to the pixels. The second pass replaces each temporary label by the smallest label of its equivalence class.



Below find an example for the original (binary) image, first-pass image, and second-pass image. In the first pass, the algorithm searches for connected pixels row by row. If there exists a neighbor pixel in a previous row, the labels of the current pixels are assigned to the label of the neighbor pixel. Otherwise, the labels of the current pixels are assigned to a new number. In the second pass, a table contains the equivalence relationships between the labels are generated. The labels are then updated according to the table.



Set ID	Equivalence label
1	1, 2
3	3, 7
4	4, 6
5	5

Write a function that implements the two-pass connected-component labeling algorithm.