

Cody Nichoson

ECE 332

October 5, 2021

MP#1 - Connected Component Labeling (CCL)

Description of Algorithm

The purpose of this algorithm is to identify individual shapes in a provided bitmap image. For my implementation, I used a function to contain all calculations to simplify the task of applying the method to multiple images. After initializing several variables based on the size of the image, the function then iterates through each individual pixel in the image. The pixel's upper and left neighbors are examined, and an element in a zeros matrix (`label_arr`) is populated with a value based on the relationship between the neighbors and the pixel in question. An equivalency table is appended every time the two neighbors are different non-zero values. This equivalency table then has all duplicates removed and is sorted.

This first pass results in a `label_arr` that contains a number of identified shapes, but with multiple label values per shape. To correct this, a second pass is made to compare label values with equivalency pairs stored in the equivalency table. In the end, all equivalent labels are replaced with their lowest common value. Additionally, the label array for test image 3 (`gun.bmp`) is filtered to remove any shapes smaller than a specified threshold. The results are then plotted and displayed using Matplotlib.

Results and Analysis

The resulting images show each individual shape in the test images highlighted as a different color and showcasing a different label when moused over. If printed, the label array would show each shape formed from a cluster of similar labels in an array of zeros.