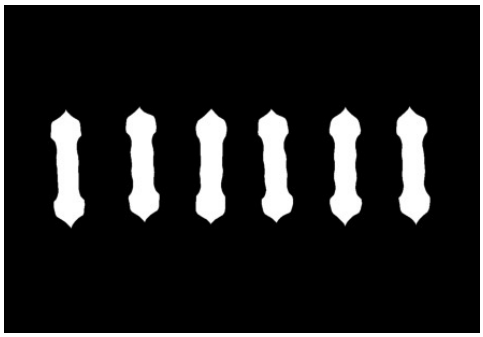
CS413 Image and Video Analysis Coursework Report

Sam Moon – 2001779

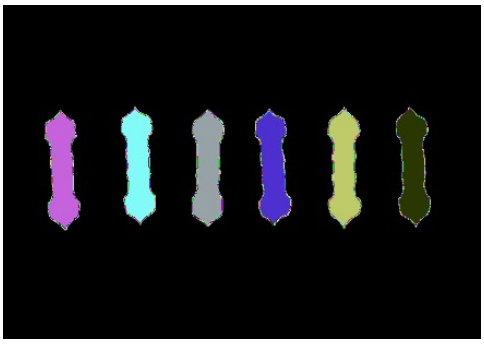
Part 1 – Resistance is Futile

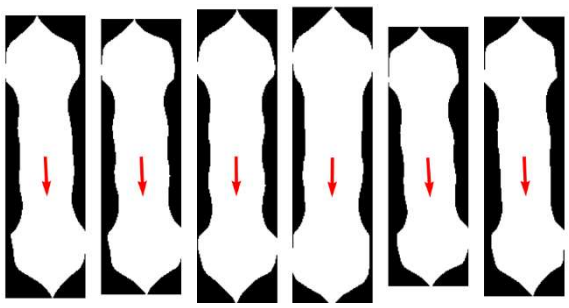
My solution to this part partially solves the problem – is correctly calculates the resistance values for all resistors in the first image, however, does not do so in the other two images. It correctly detects many of the other resistors but does not compute the resistance correctly.

The solution firstly picks out the foreground objects from the image using a Gaussian blur and luminance threshold. The terminals of the resistor are then removed using a Euclidean distance transform threshold to remove parts of the foreground too close to the background, effectively 'pinching off' the thinnest parts of the foreground objects.



The connected components of the resulting binary foreground image are then partitioned out and stored. For each connected component of each image, the bounding box is found and for each component which is within the valid size range, the centroid and principal axis are found. These are used then to rotate every component into a vertical orientation.



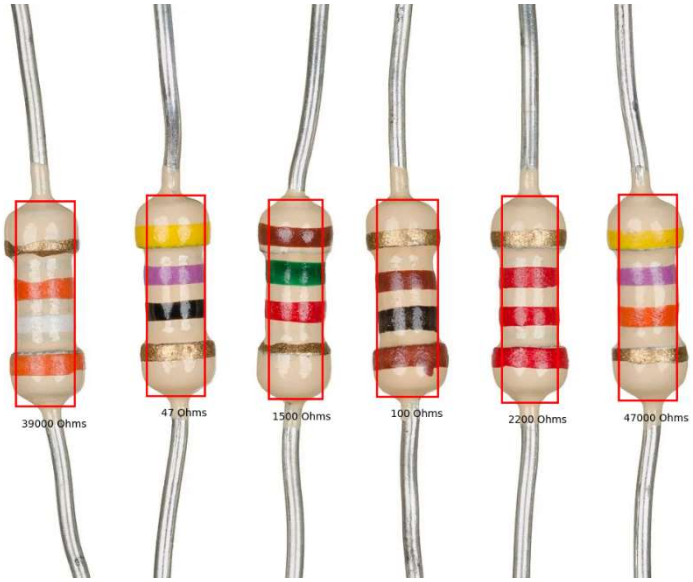


Then, for each rotated resistor, the mean colour along each horizontal row of pixels is taken to give a 1-dimensional colour progression strip. Additionally, the median colour in this colour progression is taken as the background colour of the resistor.

The colour strip is then converted to the Lab colourspace, which using simple Euclidean distance measure more accurately aligns with human perception of colours than using RGB or HSV. A set of known colours is defined to compare the colours in the strip to. Each colour in the strip is assigned to its closest colour from this set (plus the background colour) by Euclidean distance to give a more discrete colour progression.

The coloured bands are then collected together and the background colours stripped out to give the list of 3-6 colours which make up the resistor's resistance indicator.

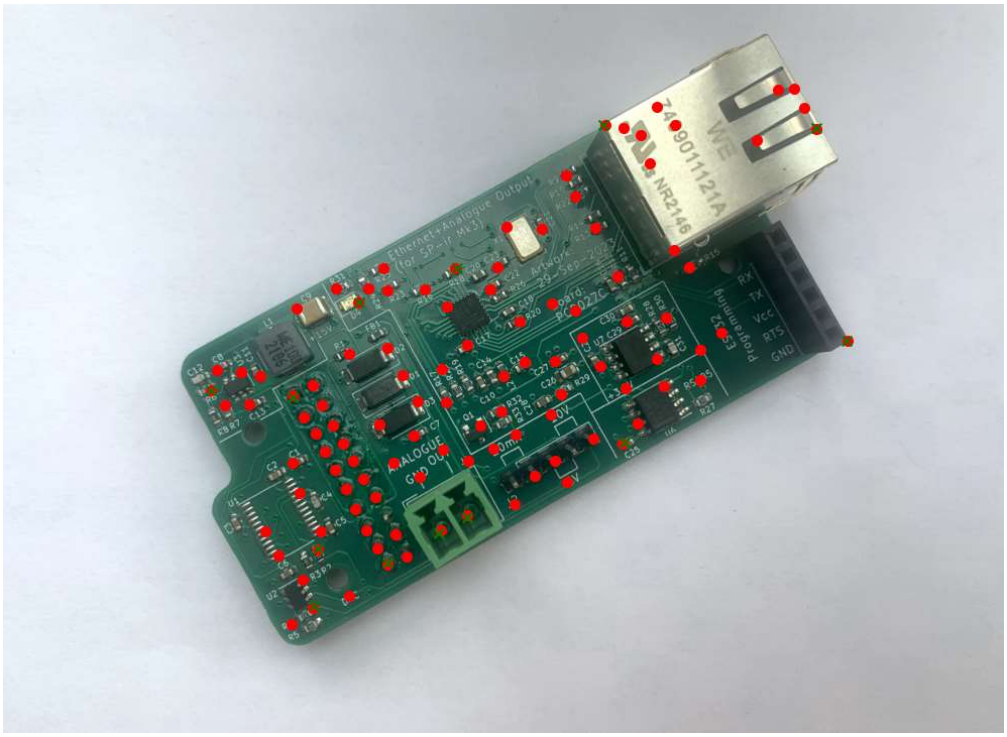
From this list the resistance can be easily calculated following the arithmetic rules. Finally, the bounding boxes of each detected resistor are drawn onto the images and the calculated resistance values for each are written alongside the corresponding resistor.

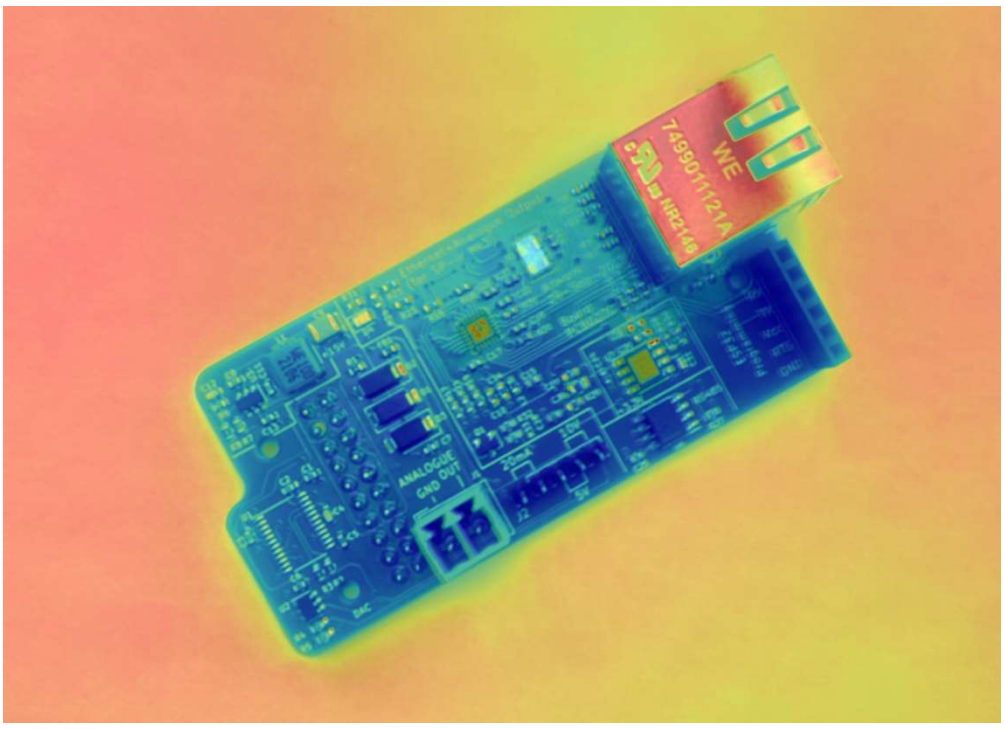


Part 2 – Spot-the-Difference

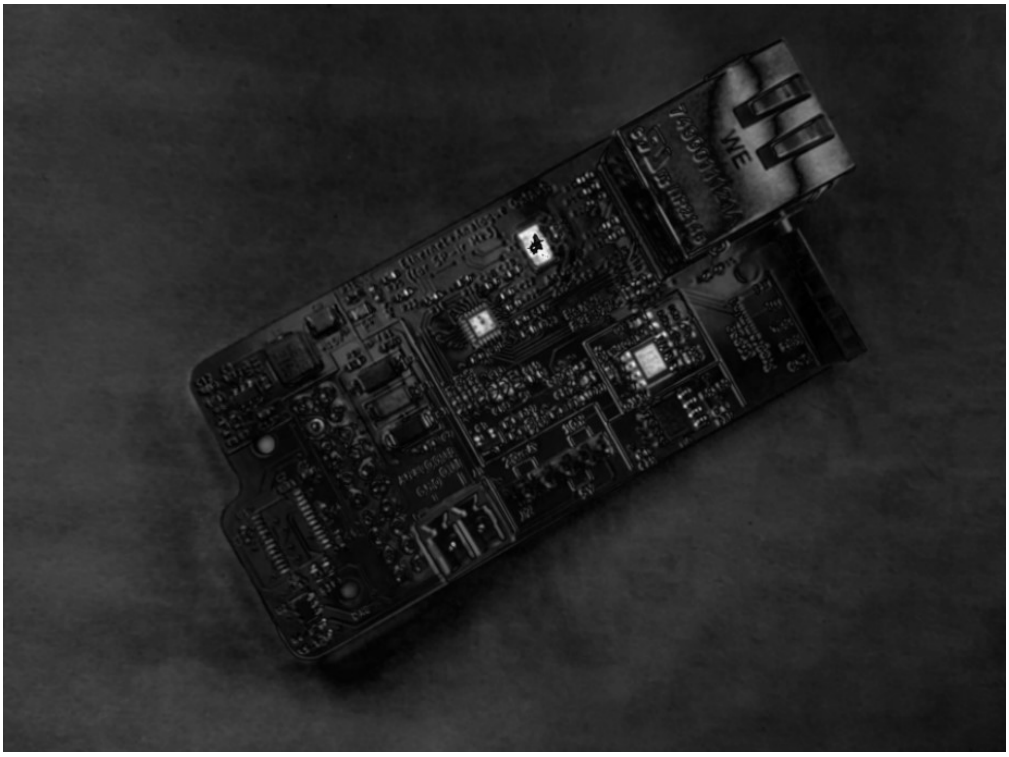
My solution to this problem correctly identifies the position of the three main components missing from BOARD2 but which are present on BOARD3.

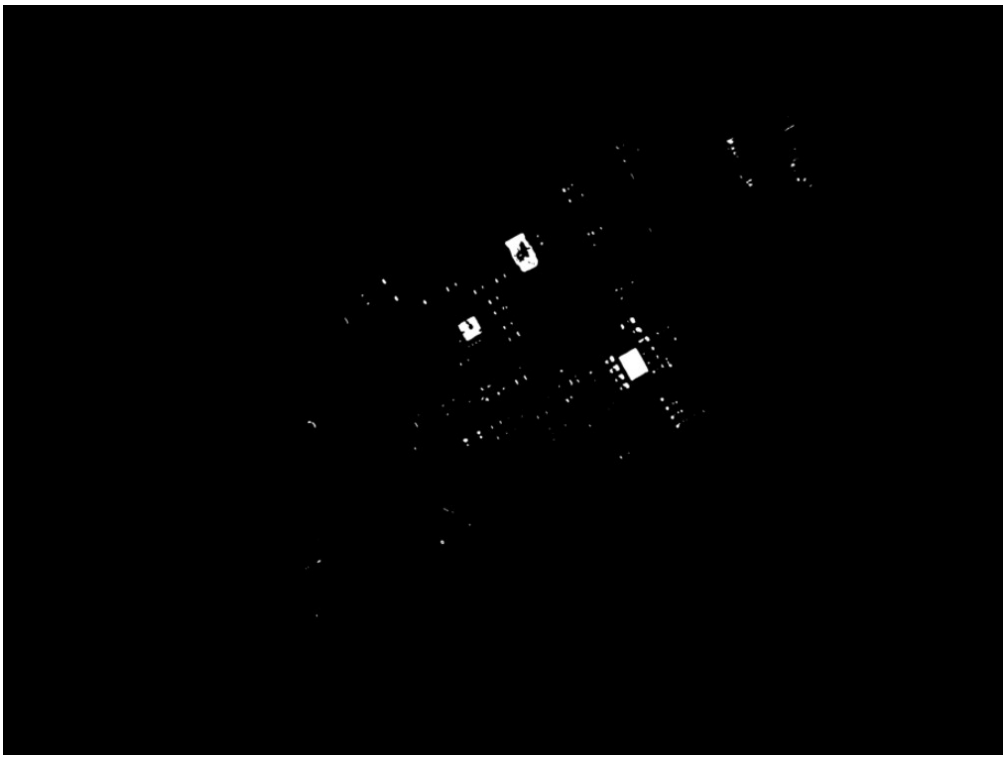
The solution begins with Harris corner detection ran on BOARD2-1 and BOARD3-1. I then manually provided a set of matching points across the two images and the solution computes the homography matrix to transform the points in BOARD3 to the matching points in BOARD2. This matrix is then used to warp BOARD3 onto BOARD2.





With the two images overlayed, the pixel difference between the two images is calculated via the Euclidean distance in the RGB colour space. This is then thresholded to give a binarized difference image. A Euclidean distance transform threshold is also applied to remove artifacts resulting from the boards not being perfectly aligned in some areas.



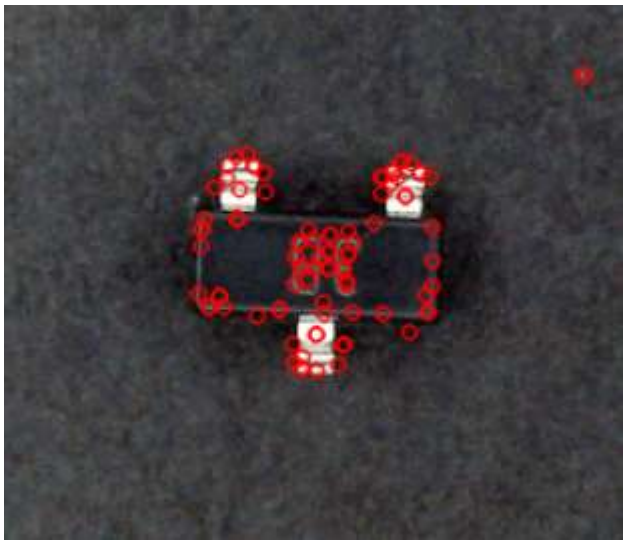


Next, the connected components of this binary image are found, and the bounding box found for each. The connected components of valid size are found and stored. Finally, the bounding boxes of the valid connected components are drawn onto BOARD3, marking the components which were missing in BOARD2.

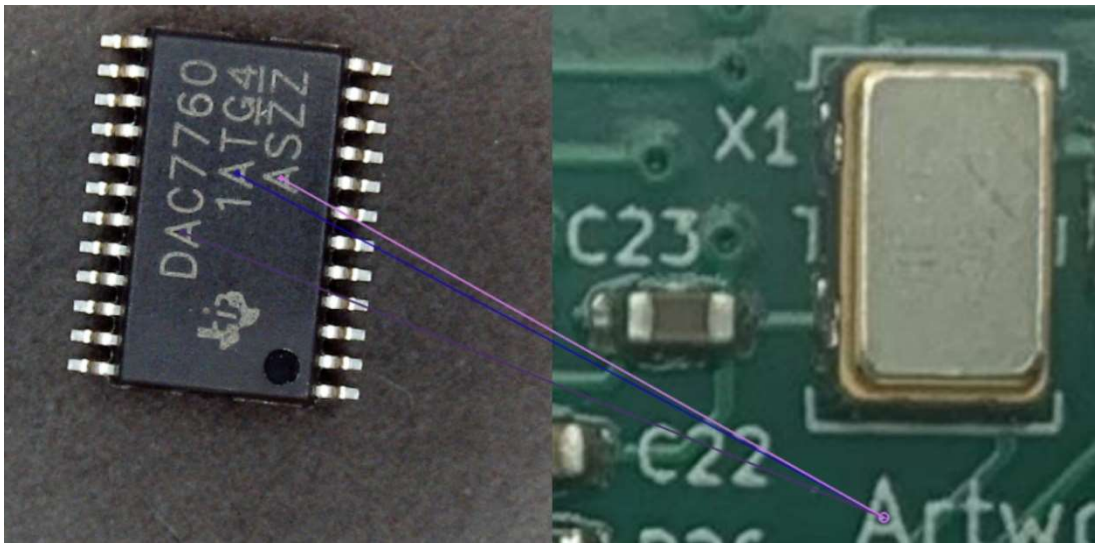
Part 3 – Pick-and-Place

My solution is this part is incomplete.

The solution begins by finding the SIFT key points in each component image to provide a bag-of-words to later match with the components of the board.



Next, the solution attempted to iterated through a series of windows of the board image of different sizes, find the SIFT key points in each window and find match the window to the component image which mostly closely resembles it with regards to the SIFT features. However, this part of the solution does not work as intended.



Word Count: 629