Image Processing Assignment 1

Edward Savage

University of Lincoln, School of Computer Science

15561454@students.lincoln.ac.uk

The objective of this report is to describe the approach to analysing the given images with various image processing techniques to achieve an automatic recognition of components.

Task 1 – Pre-processing

Firstly, the image was loaded to the MATLAB workspace using the `imread()` function. The image was then converted to grayscale using the rgb2gray() function. Bilinear interpolation was then applied to resize the image via `imresize()` and parsing the 'bilinear' parameter (See Fig.1).

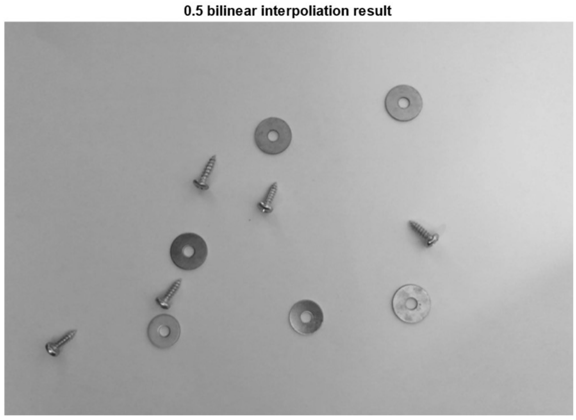

Fig.1.*(Bilinear interpolation result)*

The `imhist()` function was then applied to understand intensity distribution (See Fig.2, left), when compared to the histogram after enhancement (See Fig.2, right), we can see quite a difference in distribution.

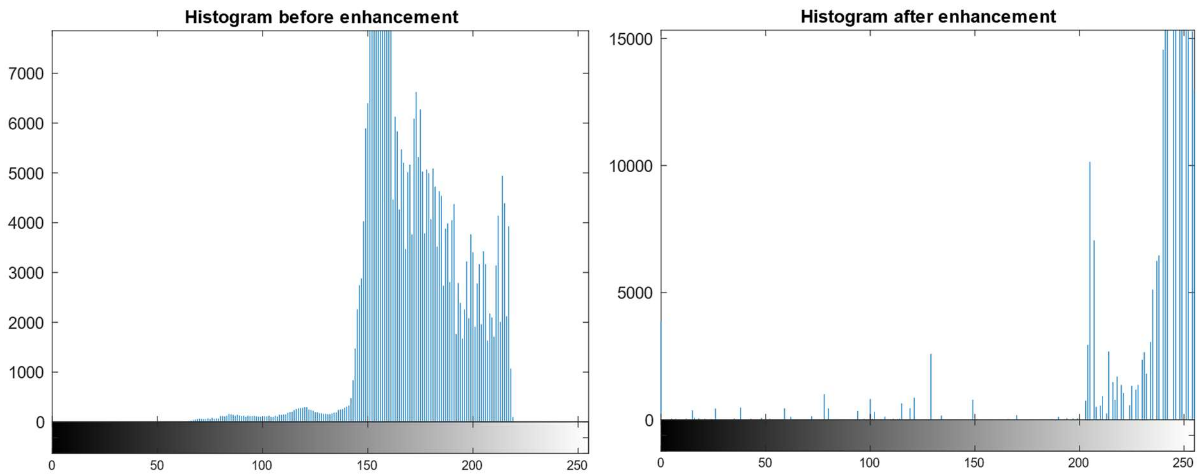

Fig.2. *(Histogram before enhancement (left), Histogram after enhancement (right))*

Most values have gained intensity, the enhancement procedure has increased background values and decreased foreground values, leading to a uniquely expanded histogram.

Image enhancement went through several methods, starting with flattening. Utilising gaussian smoothing with low standard deviation provided an approximation of shading for each component, giving an easier approximation for background and foreground elements. Local image contrasting allowed for edge aware enhancement and further flattening of components.

The objects then go through opening-by-reconstruction, followed by closing-by-reconstruction. Through each step the top and bottom 1% of values are saturated with `imadjust()` to reduce intensity at non-relevant components and noise. Lastly, the values are eroded with a small square structure to close off unconnected components (See Fig.3).

Enhanced image result

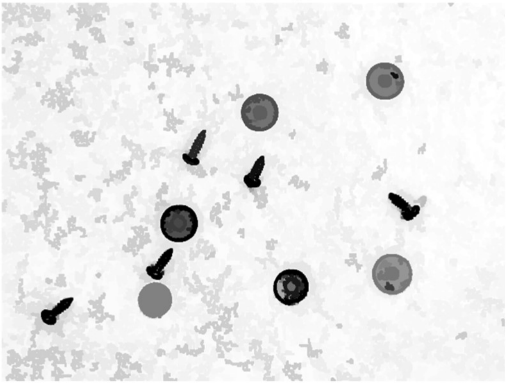

Fig.3. *(Resulting enhanced image)*

The reconstruction methods were preferred over the toolbox `imopen()` and `imclose()` morphological operations as the mask for reconstruction is applied using a flood-fill from seed points, which are defined by the erosion and dilation methods. The toolbox functions use a structuring element rather than connected components to perform the reconstruction.

Task 2 – Edge detection

Both Sobel and Canny methods are sensitive to noise, but Canny does not supress maxima, all edge candidates that are not dominant in their respective neighbourhoods are not considered edges and was hence used with the `edge()` function.

Canny method edge detection

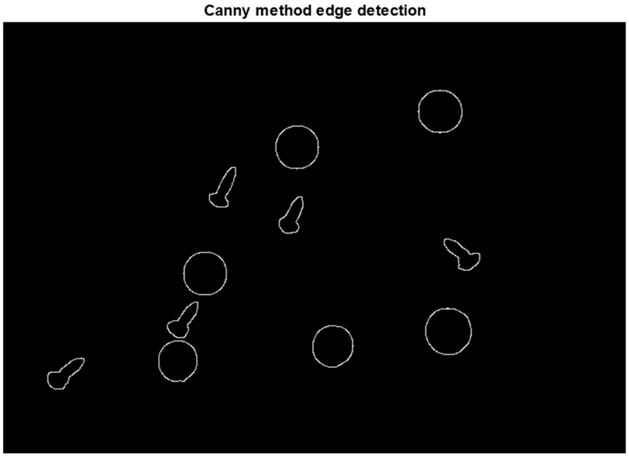

Fig.4. *(Resulting Canny edge detection)*

Task 3 and 4 – Segmentation and Object Recognition

For this task we label the connected components. Firstly, we fill any remaining holes in objects by using the `imfill()` function with parameter 'holes' and then apply `bwlabel()`. Each object returned was a unique entity (See Fig.5, left), where we wish for each to fall within defined classes. To classify, we create a table with the axis dimensions using the `regionprops()` function. We then initialize a container image with the same dimensions as the parsed binary image.

Utilising the table return, we gather the number of objects detected, and begin to iterate through them. We use the axis dimensions to compare sizes, categorising based on how much distance there is between an axis. Once we classified the objects to relevant containers, we replace the container values with a respective classifier identifier 1,2 or 3 at the positions of the object and convert the label to RGB (See Fig.5, right).

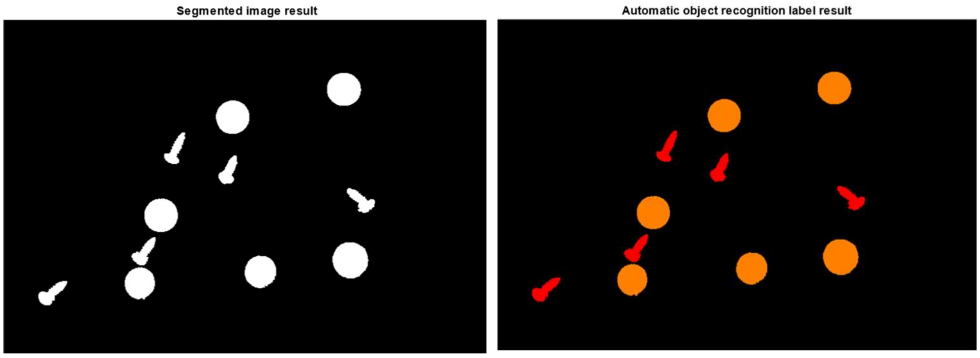

Fig.5. *(Segmented Image (left), objects recognised (right))*

Task 5 – Robust method

The resizing process has remained, but now uses bicubic interpolation to provide smoother tonal gradations. Flattening with `imflatfield()` utilises gaussian smoothing to its sigma value, which was set to 10. This reduces shading distortion and allows for better contrast processing (see Fig.6, left).

Local contrasting at a high sensitivity rate was then applied, as to not pick up noise, but strengthen object boundaries. A Laplacian filter is also applied to remove remaining noise and perform edge-aware smoothing (See Fig.6, right).

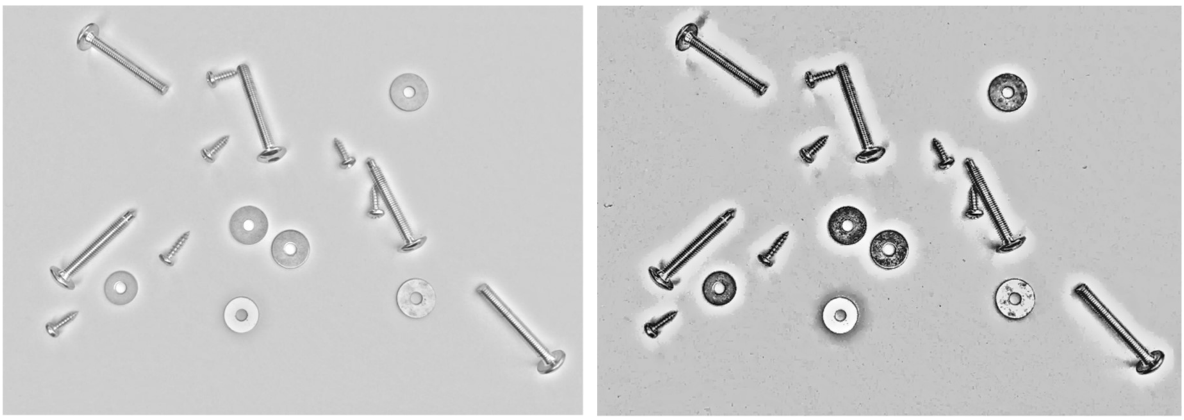

Fig.6. *(Bicubic reduction and flatted (left), Contrast adjusted, and Laplacian filtered (right))*

A square structure is used to perform opening-by-reconstruction, just small enough to fit within circular objects (see Fig.7, left), we then repeat the process but with a disk structure. These preserves both square and disk like structures, which are the main properties of the objects we wish to identify (see Fig.7, right).

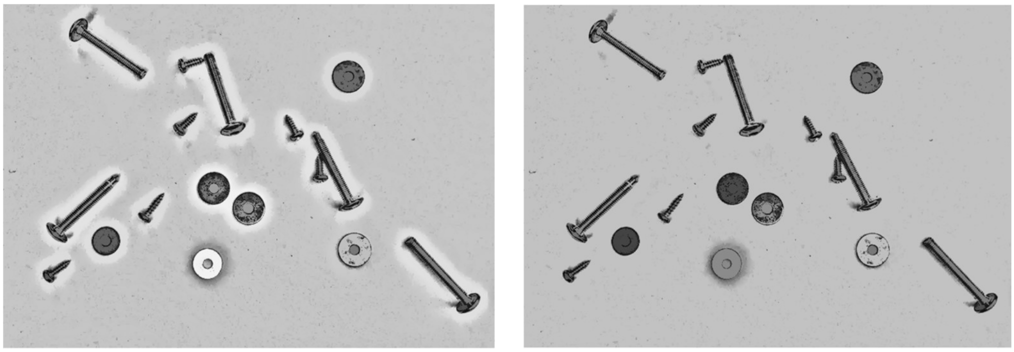

Fig.7. *(Square opening-by-reconstruction (left), Disk opening-by-reconstruction(right))*

Running through multiple opening operations created a fusion between shadow and object (see Fig.8, left). This was handled with square structure erosion, filling, and then erosion again with a disk structure. The objects are then parsed through a low-threshold edge-aware adjustment to facilitate binarization There was remaining noise with a structure less than 5 pixels, which was removed via `bwareaopen()` before being inversed (See Fig.8).

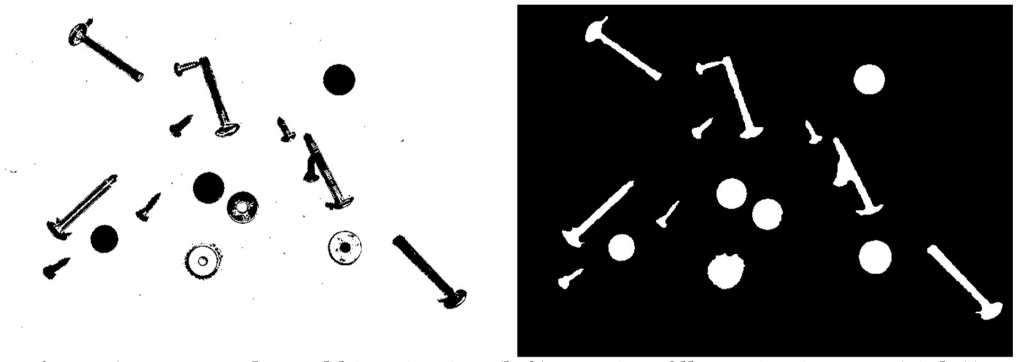

Fig.8. *(contrast-adjusted binarization (left), erosion-fill-erosion inverse (right))*

Task 6– Performance Evaluation

The enhanced binary image was then parsed through the automatic object recognition method defined earlier with an adjustment to include long screws as 3. The ground truth images were also loaded. The labelled groups were iterated upon, giving two label matrixes for each image. Dice score, recall and precision were then evaluated with the relevant parsed two labelled images to produce a table for all 10 images. (See Fig.9).

T

10x3 table

	1 Dice score	2 Recall	3 Precision
1 IMG_01	[0.883190883190883,0.694834074580910]	0.588935035635543	0.844558823529412
2 IMG_02	[0.886293333333333,0.692858454194970]	0.593101041242511	0.818758532080623
3 IMG_03	[0.884168088075671,0.679815209665956]	0.588076642130696	0.817273153209433
4 IMG_04	[0.865161059046449,0.497560975609756,0.696004577095452]	0.310379996131460	0.498334274421231
5 IMG_05	[0.846436519848869,0.687025659559089,0.742725127801809]	0.313442121729238	0.501436087018342
6 IMG_06	[0.885646644276827,0.653542086498217,0.704061482627955]	0.305418270019879	0.516197638935980
7 IMG_07	[0.880622837370242,0.706365503080082,0.716283634579210]	0.367614443419971	0.568609312365094
8 IMG_08	[0.888878776847470,0.700827406009581]	0.584837718455925	0.816092724105588
9 IMG_09	[0.878200262231730,0.724807395993837]	0.541337971302565	0.766355962011959
10 IMG_10	[0.873479760392086,0.689917058059358]	0.513529434866277	0.788328208838247

Fig.9. *(Dice score, Recall and precision table)*