Objective

The aim of the assignment is to locate printed text on notices, these text areas should be identified within rectangular regions with different regions for separated text and text of different sizes. Once the areas of text are detected, the areas found by the program should be compared to the ground truths supplied using DICE coefficient and an average found for the whole set.

Method

- Resize Images

The images were downsampled to a smaller image as it was found that the solution worked better across all the images if they were of a similar size.

- Convert to greyscale

The image is first converted to grayscale using the cvtColor function and the CV_BGR2GRAY code. Grayscale is used predominately in image processing and edge detection because of the intensity of the converted image. Grayscale allows for edges to be distinguished and does not suffer from the complexity issues of processing colours. It is also needed when thresholding the image.

- Morphological gradient

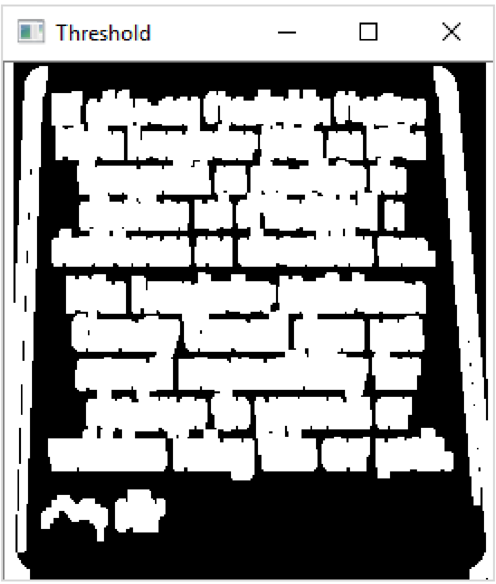
From here, the morphological transformations are used by creating a structuring element or kernel which decides the nature of operation, for morphological gradient, the kernel is created with a width of 3 pixels and height of 1 pixel. Morphological gradient is the difference between the dilation and the erosion of the image, it allows for each pixel in the image to show the contrast intensity of the pixel nearest the image.

- Dilation

Next step was to dilate the image, two dilations were done in a row using different structuring elements. Dilation increases the white region in an image and also can help in joining the broken part of the text which will be useful when joining the regions later. The whiter image is also needed for thresholding as it will search for pixel values above a certain value.

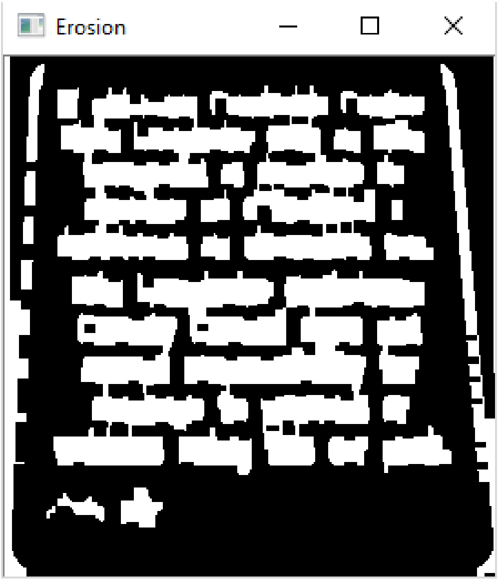
- Threshold

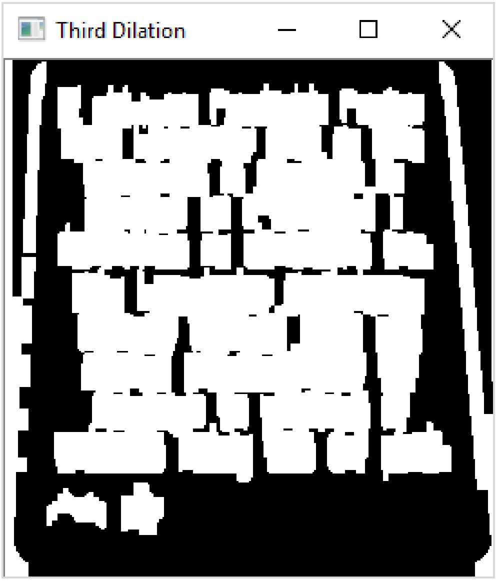
The threshold is done next, thresholding is accomplished by searching for pixels above a certain value and if the pixel fits the criteria, then the pixel is assigned a new value, if the pixel is below then it is black. The threshold value is 150 and any pixels above this are changed white, this allows for the image to be binarised and easier to connect regions.



- Erosion/Dilation

An erosion and dilation is performed once the image is thresholded, erosion erodes the boundaries of the foreground object. This reduces the thickness of the white parts, disconnects regions, and removes small white noises that may be detected later. A dilation performed after this due to the erosion shrinking the region. The dilation doesn't allow the small white noises to return but increases the size of the regions and connects them.



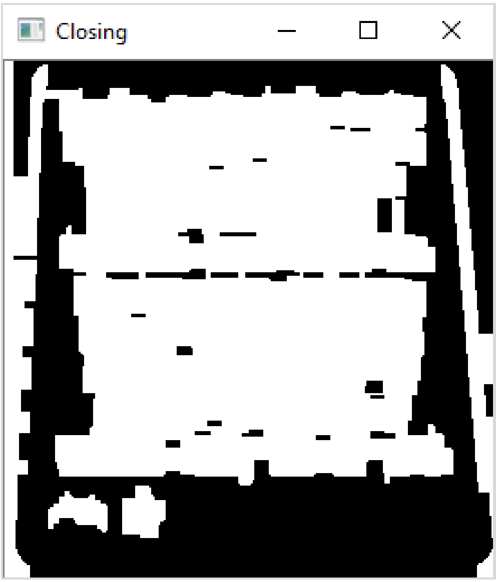


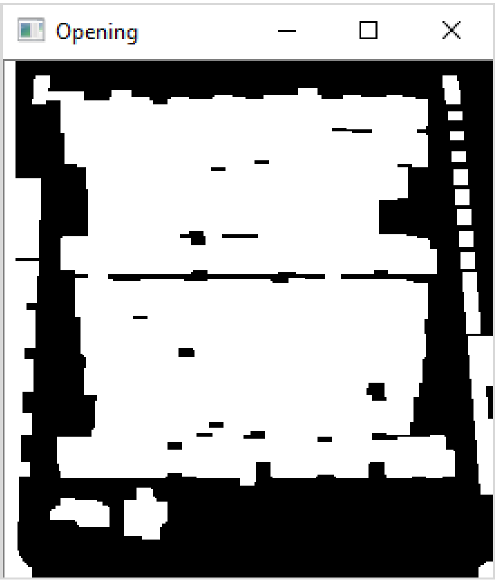
- Closing

A closing is then done, closing is a dilation followed by an erosion. It is good for closing small holes in the region which will connect our regions and allow for the rectangular boxes to find regions that are large enough to be text boxes.

- Opening

An opening is done last, it is the reverse of a closing, an erosion followed by a dilation. It is helpful to use at the end of a solution as it is good for removing noise





- Region Detection

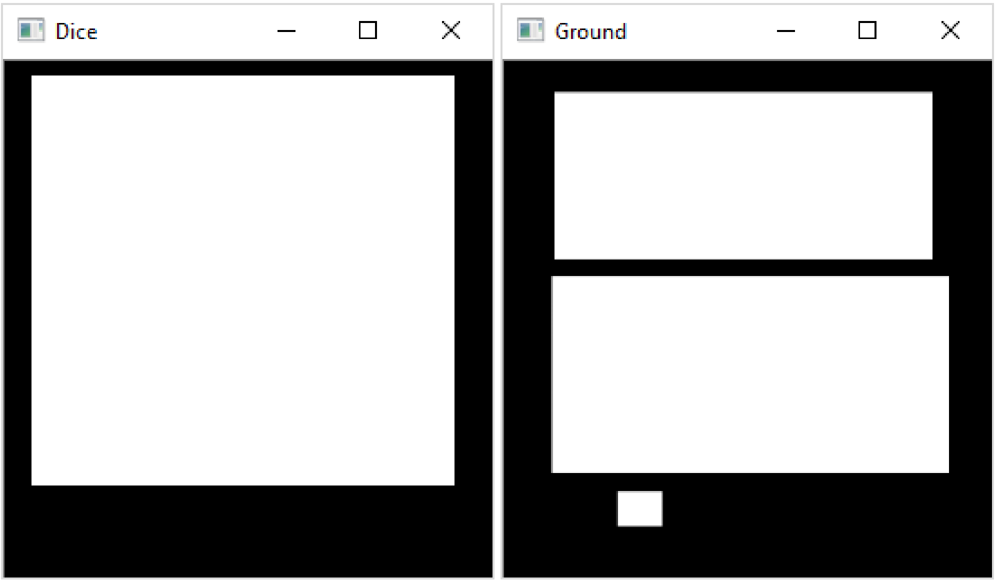
After all these operations are performed on the image, OpenCV's findContours function is used to find regions in the image that are of a certain size. The function finds the points of the region as stores them. The region is bounded by a rectangle at its edges, if the width of the rectangle found is greater than the height of the rectangle then it is added to the rectangle vector, as text is usually wider than it is high. The stored rectangle points are then drawn on the images.

- Dice Coefficient

$$Dice\ Score = \frac{A \cap B}{A + B}$$

- A is the total number of pixels that are white in my image.
- B is the number of pixels that are white in the ground image.
- A∩B is the total number of pixels that are white in both A and B. So it the intersection of the regions of ones in A and B.

The ground truth images are made using the given points of the rectangles and placed onto the images. The images are converted to black and the bounding rectangle boxes filled in with white. This is also done using the bounding boxes found by my solution. The white pixels in these two images are counted and used to calculate the dice coefficient as per the equation above.



Results

These are the resulting images, on the left is the result and the second is the ground truth for comparison.

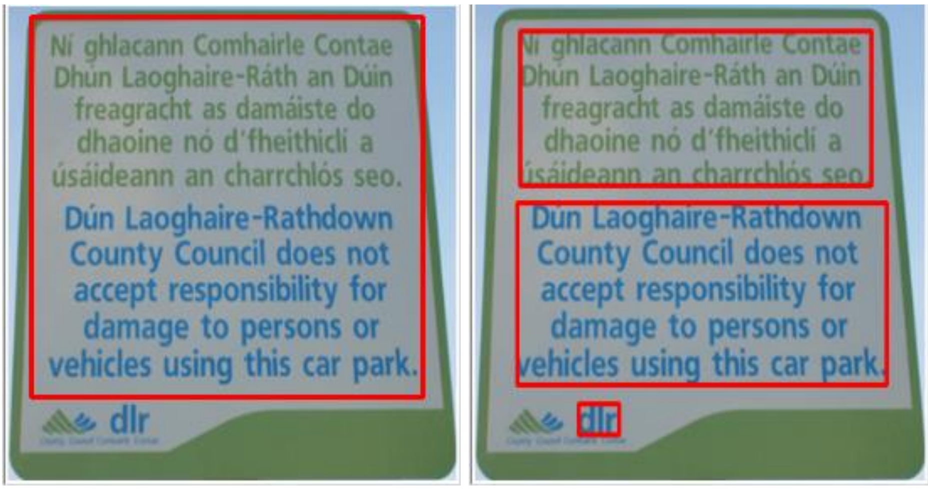


Image 1

Image 2

Image 3

Image 4

Image 5

Image 6

Image 7

Image 8

Conclusion

The results aren't terrible, but there are some glaring issues particularly in image 2 and and image 8. Image 2 is by far the worst and the solution in general is weak in notices with large spaced letters and little signs within them, as the regions would be quite large and the program would detect these regions and bound them. This can also be noticed in image 7 as the the whole sign for the speed limit is captured as opposed to just the text. As can be seen in image 2 and image 8 the solution has trouble connecting the larger letters to the region, width dilation was an option but this ruined other images.

A few of the images are split into too many sections compared to the ground truth, this was a result of being unable to connect the regions without messing up other images. For images like image 1 where the text is one big box as opposed to two smaller ones this was a result of making lower limit of the region size higher as there was a lot of noise in images like image 6 and image 8. The biggest issue is that the solution is certainly optimised for these images.

The DICE coefficient is used to test the accuracy of our boxes to the ground truth, it is a measure of the similarity of the regions found in the program.

```
Dice coefficient for Notice 1: 88.4398
Dice coefficient for Notice 2: 28.6099
Dice coefficient for Notice 3: 87.4572
Dice coefficient for Notice 4: 82.7159
Dice coefficient for Notice 5: 91.2963
Dice coefficient for Notice 6: 92.4751
Dice coefficient for Notice 7: 83.3979
Dice coefficient for Notice 8: 77.7266
Average Dice coefficient is: 79.0148
```

Overall the solution is fairly decent but image 2 is a glaring failure. The ground truth metric is a decent way to determine the accuracy of the solution but there is the issue regarding the arbitrary determination of the ground truth as one person's ground truth is not the same as another, this means there could be more than one answer to a solution.

Another issue is that with the ground truths being supplied the overall performance of the solution can be optimised to the images. Better dice scores were obtained by seemingly arbitrary changes that would better fit the sample images.

A couple test images gave decent results, but suffer from similar defects as image 2, as seen in the first picture.