Image Processing & Analysis

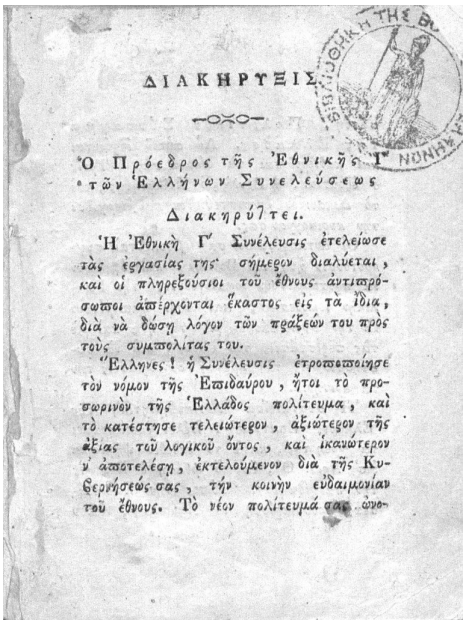
Christos Panourgias

Contents

1 Introduction

This paper aims to present image processing and analysis techniques for cleaning an image containing text and creating boundaries surrounding the words of the text. The goal of this task is to remove noise and other unwanted image features and then segment the text into individual words or lines. To achieve this, a combination of image processing techniques using MATLAB will be used.

The following image will be used which contains text, but also contains a stamp and noise created by the natural wear and tear of the paper on which the text is written. Image processing techniques will be applied to clean up the image and remove these unwanted features. Then thresholding and edge detection techniques will be used to recognize the text in the image.



ΔΙΑΚΗΡΥΞΙΣ

Ὁ Πρόεδρος τῆς Ἐθνικῆς Γ'
τῶν Ἑλλήνων Συνελεύσεως

Διακηρύττει.

Ἡ Ἐθνικὴ Γ' Συνέλευσις ἐτελείωσε τὰς ἐργασίας της· σήμερον διαλύεται, καὶ οἱ πληρεξούσιοι τοῦ ἔθνους ἀντιπρόσωποι ἀπέρχονται ἕκαστος εἰς τὰ ἴδια, διὰ νὰ δώσῃ λόγον τῶν πράξεών του πρὸς τοὺς συμπολίτας του.

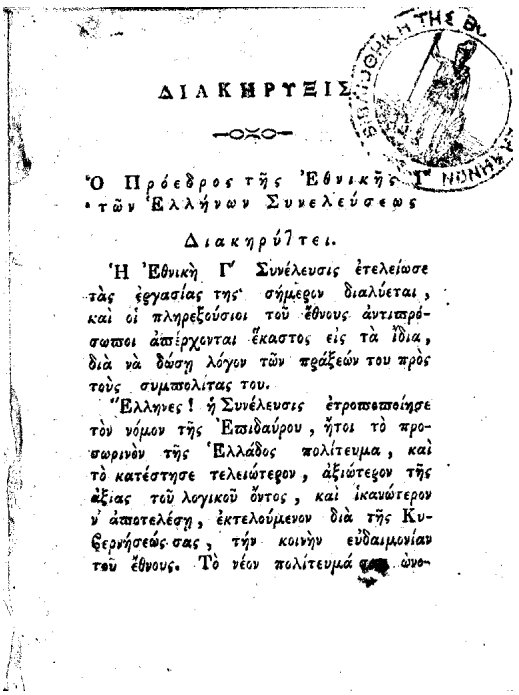
Ἕλληνες! ἡ Συνέλευσις ἐτροποποίησε τὸν νόμον τῆς Ἐπιδαύρου, ἤτοι τὸ προσωρινὸν τῆς Ἑλλάδος πολίτευμα, καὶ τὸ κατέστησε τελειώτερον, ἀξιώτερον τῆς ἀξίας τοῦ λογικοῦ ὄντος, καὶ ἱκανώτερον ν' ἀποτελέσῃ, ἐκτελούμενον διὰ τῆς Κυβερνήσεώς σας, τὴν κοινὴν εὐδαιμονίαν τοῦ ἔθνους. Τὸ νέον πολίτευμά σας ὠνο-

Once the text is detected, region-based segmentation techniques will be applied to segment the text into individual words or lines. This will include creating borders around the text that clearly separate different words or lines.

2 Part A

2.1 Image Cleanup

First the image is converted to grayscale and then, using Otsu's method via graythresh(), it is converted to binary, as shown below



ΔΙΑΚΗΡΥΞΙΣ

Ὁ Πρόεδρος τῆς Ἐθνικῆς Γ' Συνελεύσεως
τῶν Ἑλλήνων

Διακηρύττει.

Ἡ Ἐθνικὴ Γ' Συνέλευσις ἐτελείωσε τὰς ἐργασίας της· σήμερον διαλύεται, καὶ οἱ πληρεξούσιοι τοῦ ἔθνους ἀντιπρόσωποι ἀπέρχονται ἕκαστος εἰς τὰ ἴδια, διὰ νὰ δώσῃ λόγον τῶν πράξεών του πρὸς τοὺς συμπολίτας του.

Ἕλληνες! ἡ Συνέλευσις ἐτροποποίησε τὸν νόμον τῆς Ἐπιδαύρου, ἤτοι τὸ προσωρινὸν τῆς Ἑλλάδος πολίτευμα, καὶ τὸ κατέστησε τελειώτερον, ἀξιώτερον τῆς ἀξίας τοῦ λογικοῦ ὄντος, καὶ ἱκανώτερον νὰ ἀποτελέσῃ, ἐκτελούμενον διὰ τῆς Κυβερνήσεώς σας, τὴν κοινὴν εὐδαιμονίαν τοῦ ἔθνους. Τὸ νέον πολίτευμά σας ὠνο-

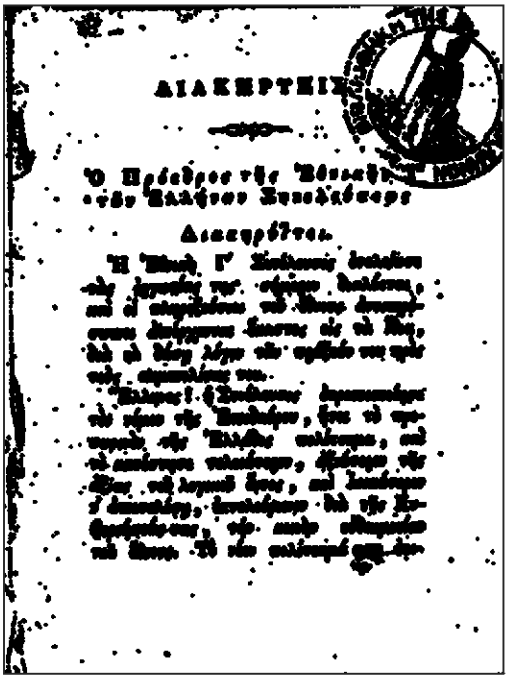
We want the stamp removed as it is not a feature of interest. To achieve this the image will be dilated until the stamp is the largest coherent area of the image, then the largest coherent area will be found and then the pixels of that area will be replaced with graylevel of the pixel of the background, which in our case has the opposite graylevel of the one shown in the image, i.e. 0.

Next we want to remove the noise (several pixel which
appear due to natural wear of the paper) and the outline in the image which is created due to the shadows of the image. First, to remove the border, we'll add padding to the image and apply dilation so that the pixel of the padding join the pixel of the border. So the border will be the largest coherent area of the image. By repeating the procedure mentioned in order to remove the stamp, the removal of the outline is also achieved. By applying erosion to the image we remove the noise. Dilation is then applied so that the letters of the words become more visible.

For the above morphological operations, the disc construction element was used because it increases the size of the objects in all directions, which joins the elements better and results in their complete removal. The process is detailed below.

2.1.1 Seal Deletion

We create a disc shape construct with radius 4 using strel(), then use the construct to apply dilation to the image using imdilate(), this process produces the following result.

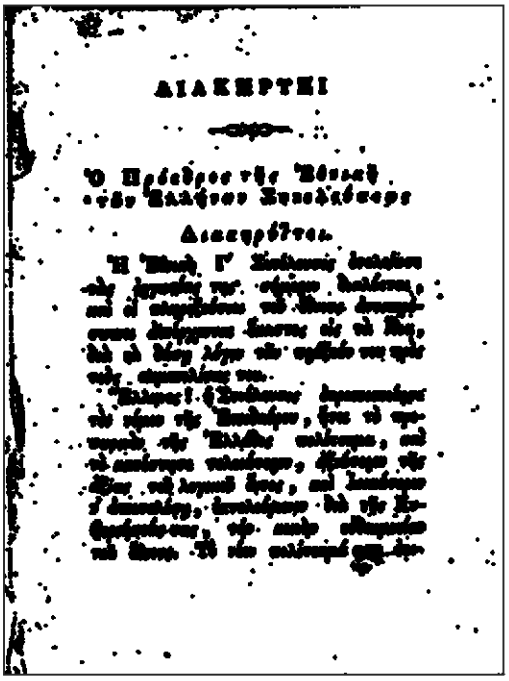


As mentioned above, the seal is now the largest coherent area of the image.

We create the function max_comp(cc_arreas) which accepts a structured array containing information about the area of each contiguous area (this can be extracted if the function
regionprops(conn_comps, 'Area')) and returns the pointer of the contiguous region that has the largest area. We also create the function remove_bigComp(bw) which accepts a binary image,

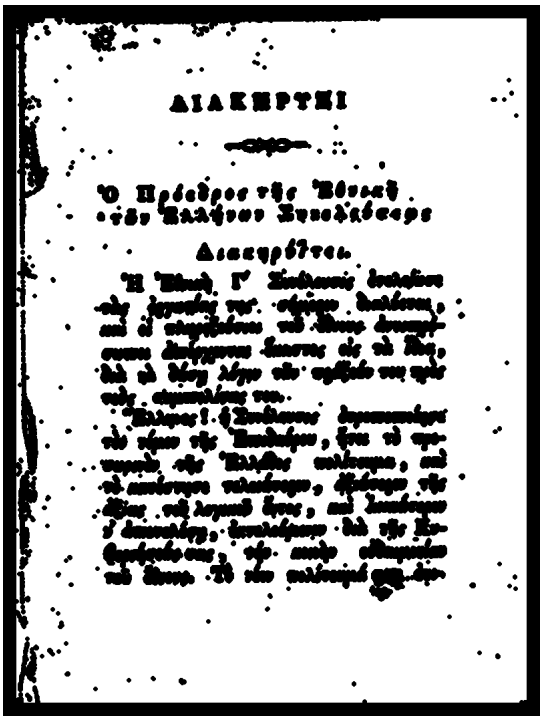
- compute contiguous regions with connectivity 8 using bwconncomps(bw, 8)
- calculates the areas of contiguous regions using regionprops(conn_comps, 'Area')
- finds the contiguous region with the largest area and changes the graylevels of its pixel to 0.
- Returns the original image without its largest contiguous region.

Using the above two functions produces the following result.

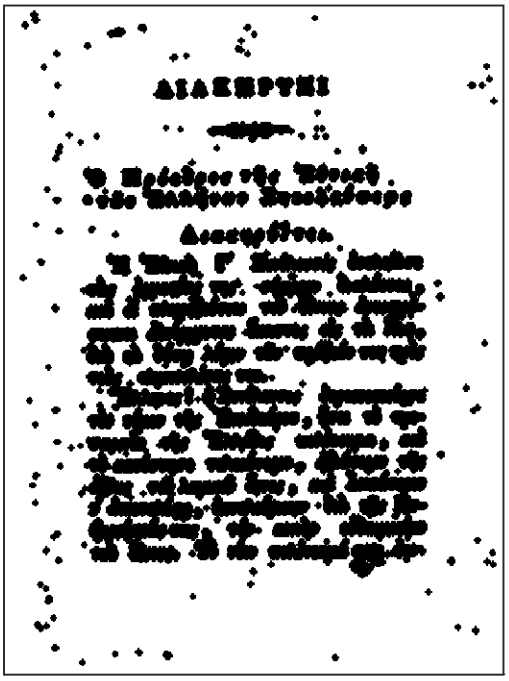


2.1.2 Border Removal & Noise

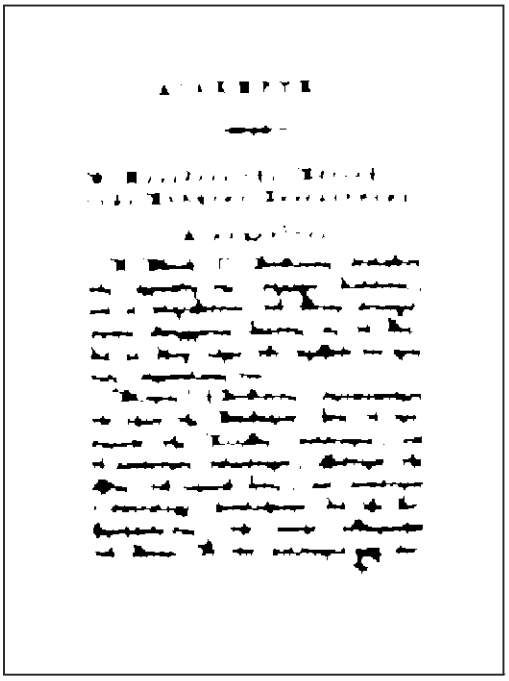
Adds 20 pixel thick padding to all sides of the image via the padarray() function. A disk dilation of radius 2 is applied using imdilate() so that the margin is merged with padding and creates a large cohesive area. This process produces the image below.



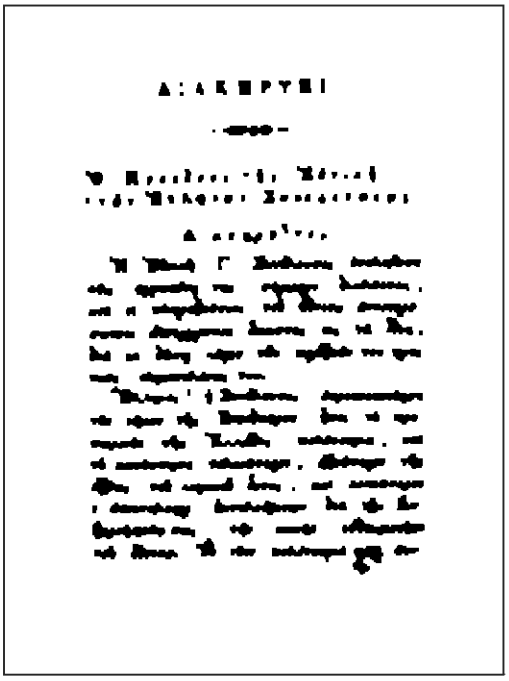
Again using the function that removes the largest coherent region, remove_bigComp(), the margin is removed and the result below is produced.



To remove the noise a disk erosion of radius 8 is applied to the image using imerode(). This process produces the following result

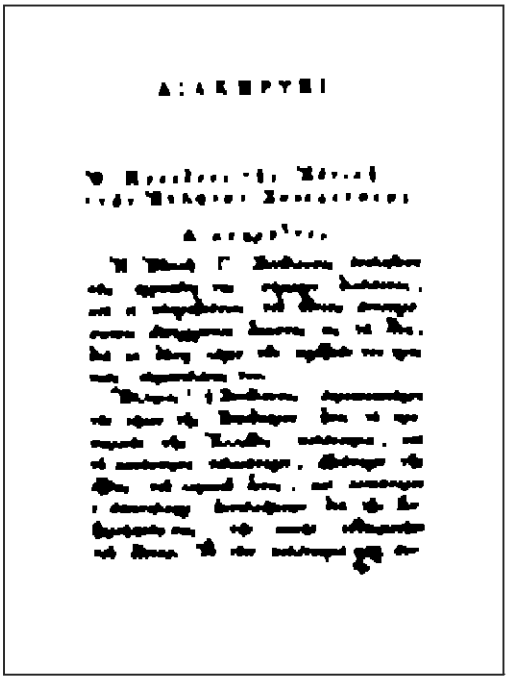


It is noticed that the letters of the titles have low density, so a disc dilation of radius 3 is applied using imdilate() and the following result is produced.



We want the image to contain only words, so the element between the title and subtitle must be removed. First the dimensions of the image will be saved using the size function, then a mask is created the same size as the image and with all its values being units. Then, with the help of the dimensions of the image, the part of the object is located and the zero value in the mask is replaced in it.

Finally, we apply the mask to the image using the function immultiply() and the following result is obtained.



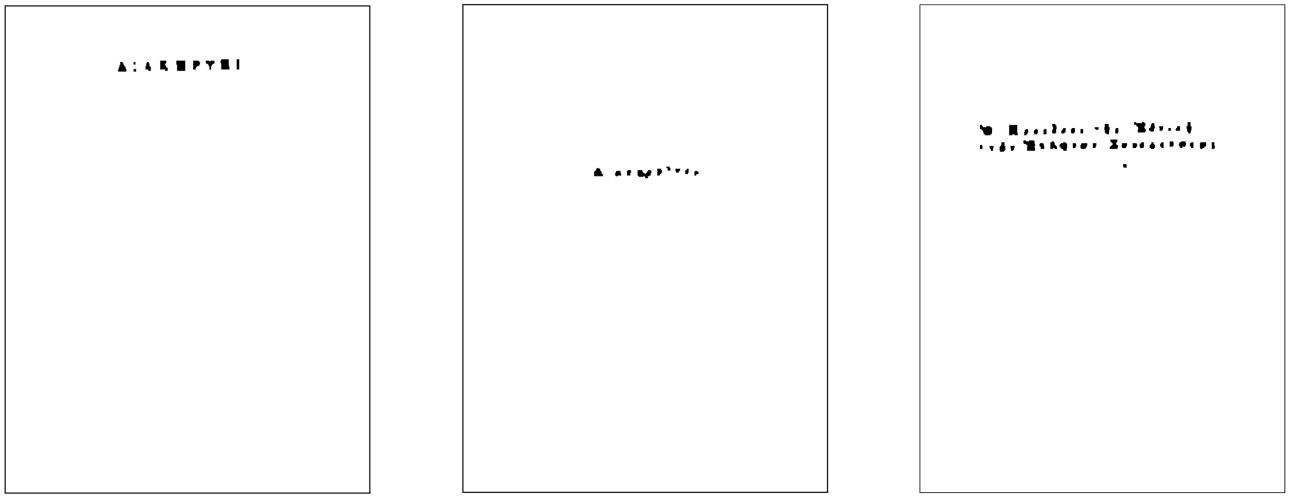
2.2 Word segmentation

Due to the application of the above morphological operations, the text presents compounds and separations of words. Correct word segmentation requires that each word be covered by a unique coherent region. To achieve this horizontal line expansion will be applied, so as to join the letters of a separated word without running the risk of joining words whose position is on different lines. It will also replace graylevel of pixels that join two words with zero graylevel (The morphological operations are applied to the negative image, so pixel with zero graylevel they will turn white).

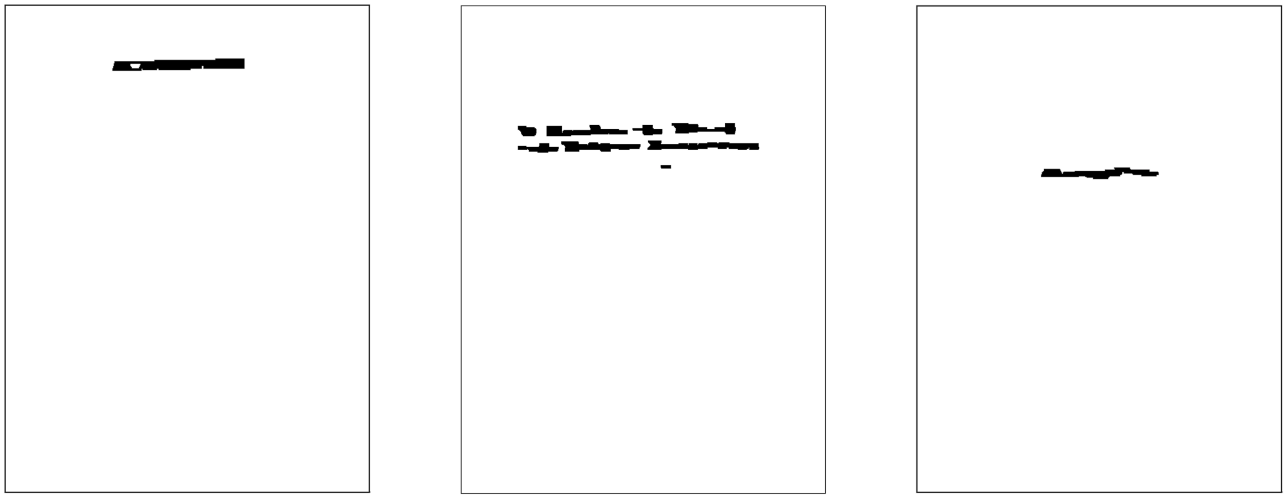
It is noticed that the title and the text below the title have separated words, so horizontal line expansion will be applied so that they join. However if expansion is applied with a component in the same line, unwanted joints will occur. Therefore different masks will be applied which isolate the points where we want to apply dilation and then they will be added to the original image by joining.

2.2.1 Word Letter Association

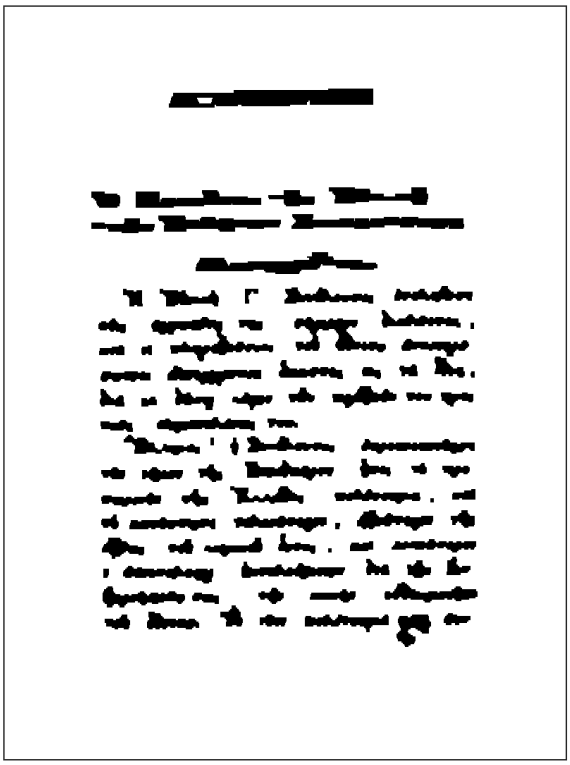
Using the image dimensions and the zeros() function, three masks are created which have zeros in all their positions. Units are then placed in the positions of the words to which we want to apply expansion. Multiplying the image with these masks (separately for each mask) produces the following result.



In the above images, a horizontal line diastole is applied with a horizontal line component of length 25, 15 and 28 respectively. This produces the following results.

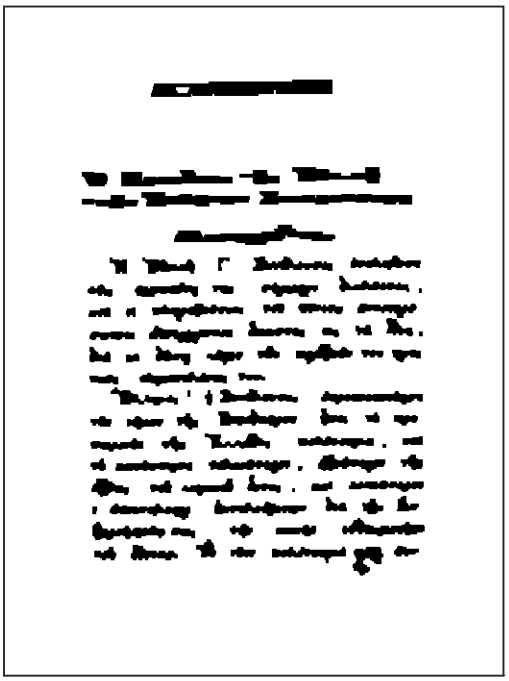


Using the join operation, we join the original image with these three images.



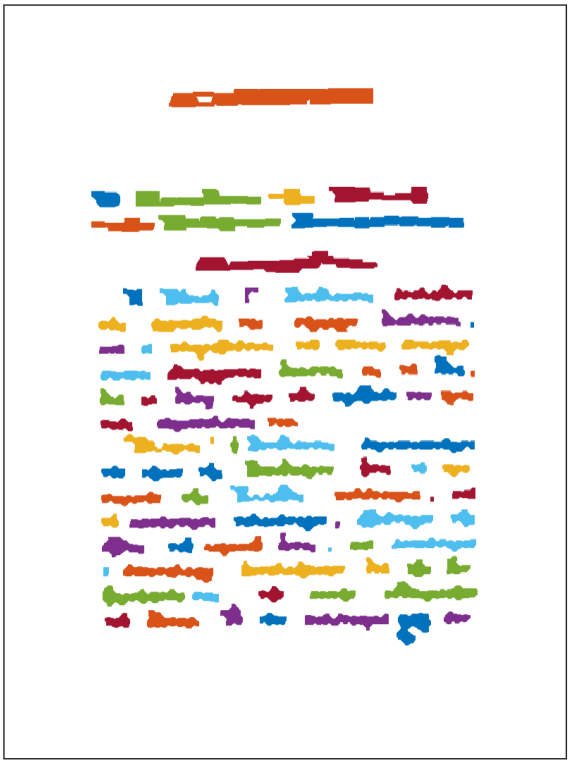
2.2.2 Separating Compound Words

To separate the joined words that occur between the second and third lines of text we will locate the pixel between those lines and replace them with 0 (ie white in the normal image). This is how the image below is produced.



2.2.3 Finding Cohesive Regions

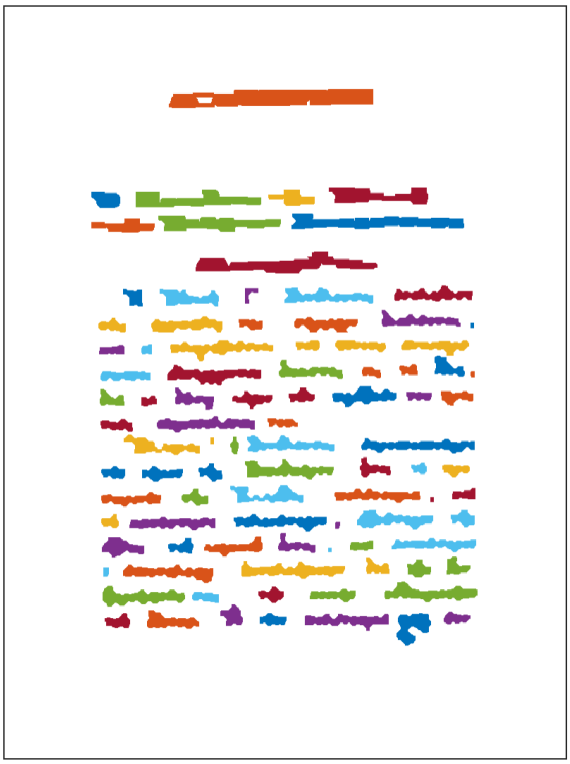
Using the function bwlabel() with connectivity 4, we find all the coherent regions of the above image, and then using the function label2rgb() we display these regions in a different color.



The number of regions found with connectivity 4 is 93. Due to the good quality

of word separation, the number of regions found with connectivity 8 is again 93 as shown below.

However if there was a difference, it would be that connectivity 8 includes the diagonals along with the vertical and horizontal pixel (based on the current pixel) in the figure while connectivity 4 only includes the verticals and horizontal pixel. Each type of connectivity may be more suitable than the other in different situations, this depends on what areas it is desired to create.



2.2.4 Bounding Boxes

Having used max(labels(:)), the number of coherent regions created has been calculated. Using a loop we find the maximum length and width for each coherent region and thus create rectangles that enclose those regions and, by extension, the words of the text. We create the rectangles using the line() function, add them to the original image, and the following result is produced.

2.2.5 Variation of Bounding Boxes

We want to apply a variation to the rectangles so that they contain the punctuation marks of the words. To achieve this the following steps will be applied for each rectangle.

- We'll raise y1 (the coordinate that defines the top side) by 8 pixel up (y1 - 8) so that the tones and splashes are included.
- Will shift x1 (the coordinate defining the left side) by 8 pixels to the left (x1 - 8) so that the tones in the first uppercase letters are included.
- Will shift x2 (the coordinate defining the right side) by 3 pixels to the right (x2 + 3) so that the final letters of the word are contained in the rectangle.

These changes will be applied to where the rectangles are stored. That is, we will have

```
R=[];
for i=1:label4_num
    [r,c]=find(labels==i);
    XY=[c r];
    x1=min(XY(:,1)) - 8;
    y1=min(XY(:,2)) - 8;
    x2=max(XY(:,1)) + 3;
    y2=max(XY(:,2));
    R=[R;x1 y1 x2 y2];
end
```

This will produce the following result.

3 Part B

This chapter presents the evaluation of text word segmentation using the $recall, precision$, and $f_1 - score$ metrics. The values of these metrics will be retrieved using the information extracted from the calculation of Intersection Over Union where $IOU = \frac{A \cap B}{A \cup B}$ with $A =$ the number of related items retrieved and $B =$ the number of related items in the collection. The evaluation will be checked for Gamma Correction values in the set $\{0.6, 0.8, 1.0, 1.2, 1.4\}$ and for the IOU threshold in the set $\{0.3, 0.5, 0.7\}$

3.0.1 Evaluation

First the tables R and GT are read which contain the coordinates of the rectangles found by the method used above and the coordinates of the real rectangles that surround the words respectively. These tables have rows of the form $[x_{min}, y_{min}, x_{max}, y_{max}]$.

Then we choose a threshold T for IOU from the set $\{0.3, 0.5, 0.7\}$ and calculate the boolean matrix $IOU \geq T$. Each element in the IOU array corresponds to the IOU value for a particular pair of rectangles (one method and one real). If a IOU value is greater than or equal to the threshold ($IOU \geq T$) then it indicates that the corresponding pair of rectangles is True Positive (TP) , because the two rectangles have a high degree of overlap. Summing the elements in the table $IOU \geq T$, we count the number of TP. Thus we have TP = sum(IOUFinal(:)). False Positive (FP) are the rectangles incorrectly identified as TP. In other words, FP are the rectangles that have been identified as having a high degree of overlap with the real rectangles, but in fact do not have a high degree of overlap. The number of FP can be calculated by subtracting the number of (TP) from the total number of actual rectangles. Thus we have FP = bb_num - TP where bb_num is the number of ground truth (Ground Truth) rectangles.

False Negatives(FN) are the true rectangles that have not been correctly recognized by the method. That is, they are the actual rectangles that do not have a high degree of overlap with the rectangles determined by the method. The number of FN can be calculated by subtracting the number of TP and FP from the total number of rectangles found by the method. Thus we have FN = bb_num - TP - FP. This is because if all the rectangles are TP, there will be no FN, and if all the rectangles are FP then there will be no FN. Having now calculated the TP, FP, FN values, we can calculate the evaluation metrics recall, precision and f1-score according to the following formulas.

$Recall = \frac{TP}{TP+FN}$

$$Precision = \frac{TP}{TP+FP}$$

$$f1 - score = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$

3.0.2 Table f1-score

Having determined how to calculate the evaluation metrics to be used, we repeat the process of finding rectangles for each combination (gamma value, threshold) $\in \{0.6, 0.8, 1.0, 1.2, 1.4\} \times \{0.3, 0.5, 0.7\}$ and for each iteration we note the f1-score. This results in the table below.

Πίνακας αποτίμησης **F-Measure**			
	IOU threshold		
gamma value	**0.3**	**0.5**	**0.7**
0.6	0.99	0.89	0.40
0.8	0.99	0.88	0.39
1.0	0.99	0.88	0.37
1.2	0.57	0.46	0.15
1.4	0.54	0.42	0.15

It is observed that the best result is obtained for the combinations (gamma value, threshold) $\in \{(0.6, 0.3), (0.8, 0.3), (1.0, 0.3)\}$ with f1- score = 0.99, which makes sense because the threshold has the lowest value of the three.

Also increasing the gamma value tends to decrease the performance of the method. As the gamma value increases, the image becomes brighter. This can be beneficial in situations where the image is very dark and details are difficult to see. However, this particular image, after gamma value = 0.1 becomes quite bright, so increasing gamma correction further causes the image to be overexposed, resulting in the loss of useful details. This has the effect of reducing the number of TP detected by our method which implies a lower f1-score.

It is noted that, compared to the other values for the threshold, the IOU threshold = 0.7 value presents a greater dispersion of f1-score values, for various gamma value values. The threshold value is used to determine the TP by comparing the IOU values between the rectangles determined by the system and the actual rectangles. A higher threshold value means that the system will consider fewer rectangles as TP, since the overlap between the method rectangles and the actual rectangles must be greater to be considered TP.

When the threshold value is 0.7, the system will be more restrictive in determining TP. This means that the number of TP will be less and the number of FN will be greater. This will result in a decrease in f1-score. Since the method is less sensitive to TP detection, its performance will be more affected by image variations, such as variations in image gamma correction.