CMPEN/EE455: Digital Image Processing I

Computer Project # 5:

Morphological Image Processing

Pavan Gurudath

Date: 11/01/2017

OBJECTIVES

The project is aimed at understanding the changes in an image due to morphol ogical processes such as dilation and erosion. Furthermore, techniques to remove noise, obtain certain portions of an image from a larger scene using erosion and dilation are explored. The main objective of this project involves the following:

1. **Thresholding an image to obtain a binary image and removal of noise:**
    - To threshold the image at a suitable value that would generate a binary image consisting of 0 and 255 pixel intensity only.
    - To remove the pepper noise grid and the lines from the input image.

2. **Extract the tall characters from the image**
    - To obtain the marker points of all the tall characters i.e. D, P, I, l.
    - To repeatedly dilate and hence reconstruct the tall character.

3. **Edge detection:**
    - To extract the edges of the tall characters.

METHODS

This project concentrates on the morphological processing i.e. dilation and erosion. In order to perform the objectives of this project, an input image named *'proj5.gif'* has been utilized.

This project can be executed by running the *'main.m'* file. The *main.m* file calls the following function: '*dilation.m'*, *'erosion.m', 'reconstruct.m'* and *'edge_detection.m'*. They execute the functions of performing dilation, erosion, reconstruct the tall characters from the marker points and to obtain the edges of the tall characters respectively.

The following subsection details the method in which the project statements are solved.

1. **To threshold the image and remove noise:**
   - The noisy image is first converted to binary image.
   - After observing the histogram of the noisy image, a threshold value of 50 was chosen since the histogram mainly consisted of values 75(gray) and 0(black). The thresholding removes the several different gray value noise.
   - The pepper noise and the lines are removed by applying a 3x3 structuring element and thereby eroding the image.
   - After erosion, certain pixels are removed and certain uneven surfaces appear. A smooth image is obtained by dilating the image.
   - The process of eroding an image A using a structural element B can be defined by:

$$\boldsymbol{A \ominus B = \{ z \mid B_z \subset A, \forall z \in Z^2 \}}$$

where B_z is the translation of B i.e. $\boldsymbol{B_z = \{b + z \mid b \in B}$

   - The process of dilating an image A using a structural element B can be defined by:

$$\boldsymbol{A \oplus B = \{ z \mid B_z \cap A = \emptyset, \forall z \in Z^2 \}}$$

2. **To extract the tall characters from the image**
   - In order to extract the tall characters, we use the image to find an appropriate mask that would remove the small characters. After using the data cursor in the imshow function and measuring the pixel length of the tall characters, we find that the appropriate mask would be of length 45 in the vertical direction.
   - We use a 45x1 structuring element and erode the clean image in order to obtain the marker points of the tall characters.

   Using these marker points, we obtain the entire tall character as follows:

   - This algorithm is executed by the *'reconstruct.m'* function which takes the marker image and the clean image as its input.
   - Two temporary variables are created and a loop is run and checked using a flag variable.

- The marker image is dilated repeatedly through every iteration. In each iteration, the intersection of the dilated image and the original image is taken so as to restrict the dilation to it character shape only.
- The intersected image of the dilated and clean image is checked in every iteration if it is equal to the previous image, and thereby decided when to stop the algorithm. If the intersected image is not the same as the one in the previous iteration, then the loop continues and the temporary images are assigned appropriately.
- The algorithm discussed above, where conditional dilation is being performed could be mathematically represented as follows:

$$1^{st}\ \text{Iteration: } D_A^{[1]}(A_1) = [A_1 \oplus B_8] \cap A$$

$$\vdots$$

$$n^{th}\ \text{Iteration: } D_A^{[n]}(A_n) = [D_A^{[n-1]} \oplus B_8] \cap A$$

In this way, the tall characters are recreated.

3. To extract the edges of the characters

- Edge detection is the process of detection of boundaries of objects within images.
- Once the tall characters are extracted, it is fairly simple to detect the edges of the characters. This task is carried out by the *'edge_detection.m'* function.
- In order to accomplish this, the filtered tall characters are eroded by a 3x3 mask.
- The difference between the tall character image and the eroded image, preserves the edges of the characters.
- However, as per the notes on L-26, it is said that the edge detection is done by finding the difference between the dilated image and the eroded image. However, this would introduce one layer of pixels around the character which is unnecessary unless the edge had to be thickened. We have implemented both, but it is fair to say that the first method is more appropriate.

Note: If the character 't' is also to be considered as a tall character, then instead of a 45x1 mask, a 44x1 mask has to be applied in order to extract it. This can be implemented by uncommenting lines 46-54 and commenting lines 56-58.
Also, it is observed that the final output 'P' has certain discontinuities. This is because we have used erosion and dilation in order to remove the grid pepper and line noise. If we use a median function, as implemented in lines 25-32, instead of erosion then, these abnormalities in 'P' can be avoided.

RESULTS

This section details all the images that are obtained after completing each of the objectives. Firstly, the image that is considered as the input image for this project i.e. 'proj5.gif' is as shown in Fig 1.

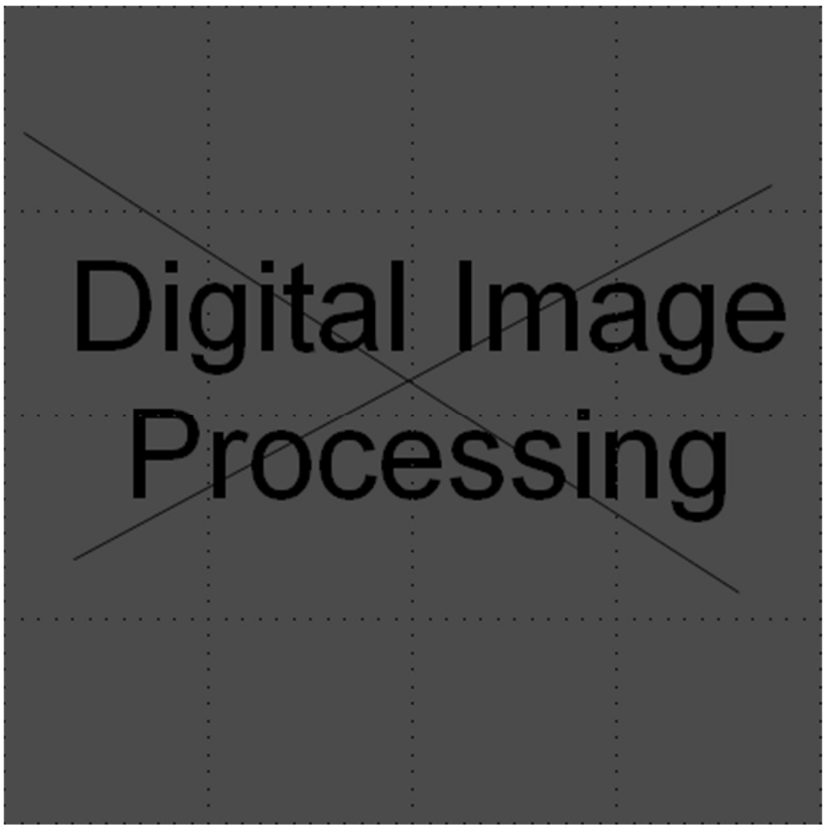

Figure 1: Input Image

The abovementioned objective's results are documented as follows.

1. To threshold the image and remove noise:

The histogram is plotted for the input image in order to find an appropriate threshold value. Using the image shown in Fig 2 we chose the threshold value to be 50.

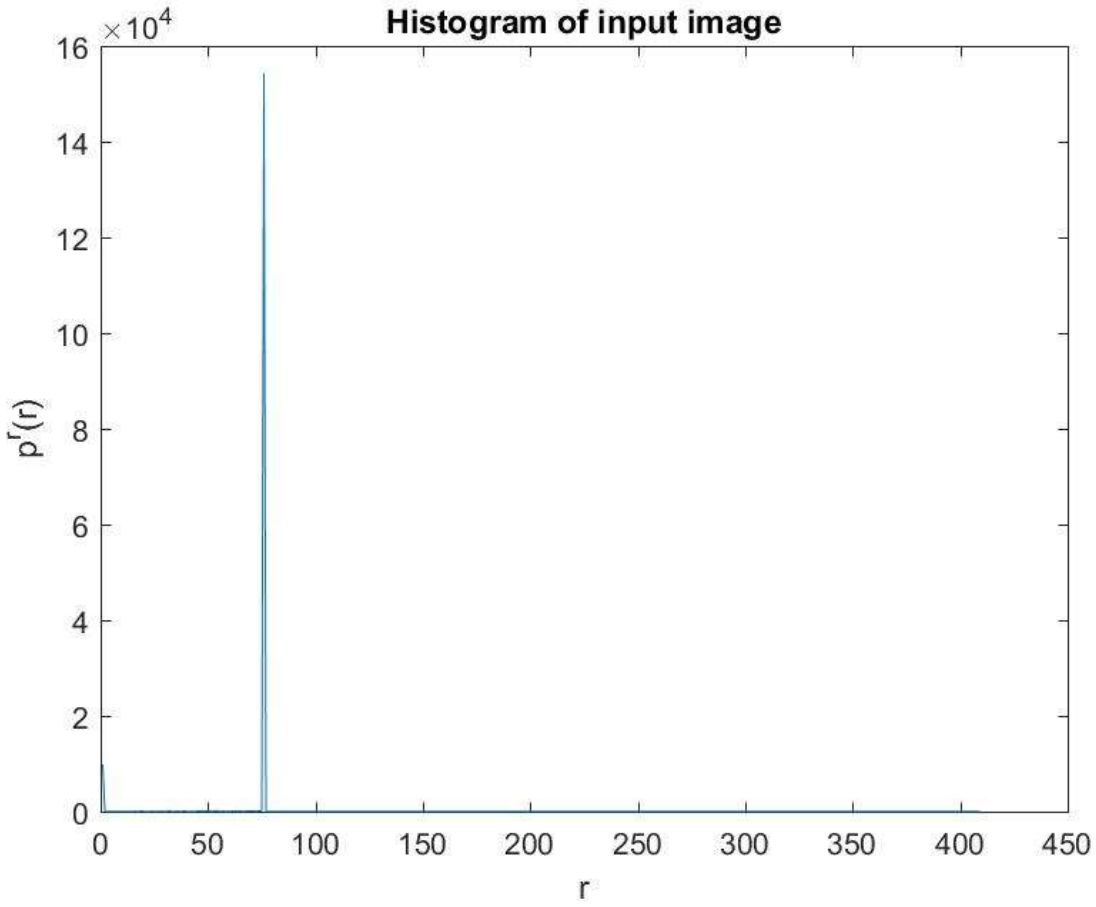

Figure 2: Histogram of input image

While it may not be apparent in the above figure, but the input image does contain pixel intensity values between the 0 and 75. After applying the threshold of 50, the image is made binary with the background as black (0) and the foreground as white (255) as shown in figure 3. (In the question, it is said that black

(0) constitutes the foreground and white (255) constitutes the background. We have assumed that the question means in terms of the input image).

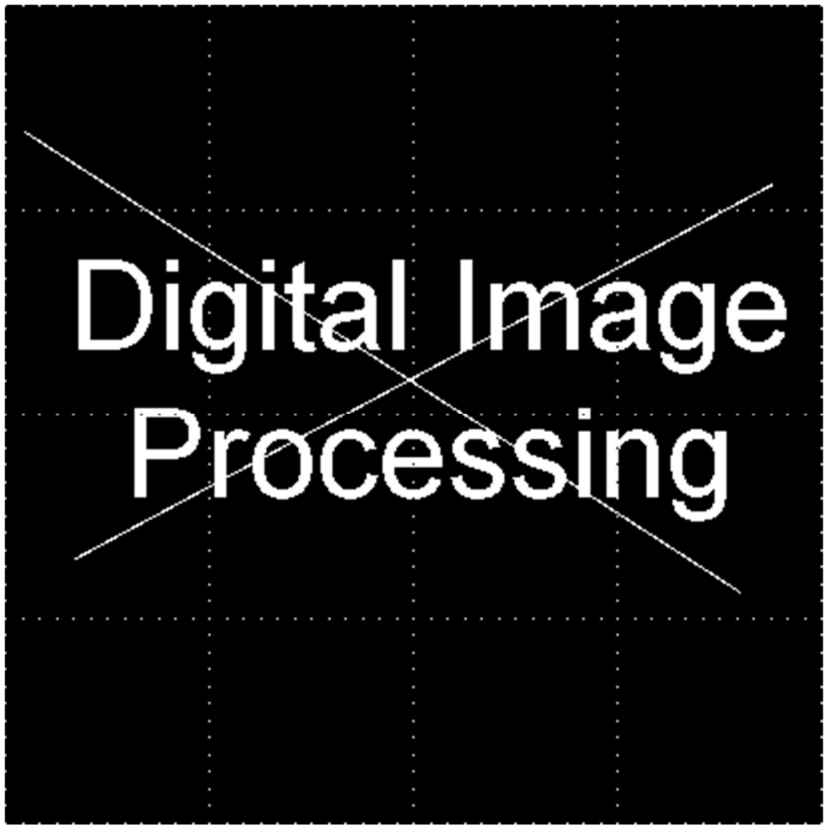

Figure 3: Thresholded image

This image undergoes erosion and dilation in order to remove the pepper noise as well as the line streaks. The erosion is done using a 3x3 structuring element. The eroded and dilated image are as displayed in figure 4 and figure 5 respectively.

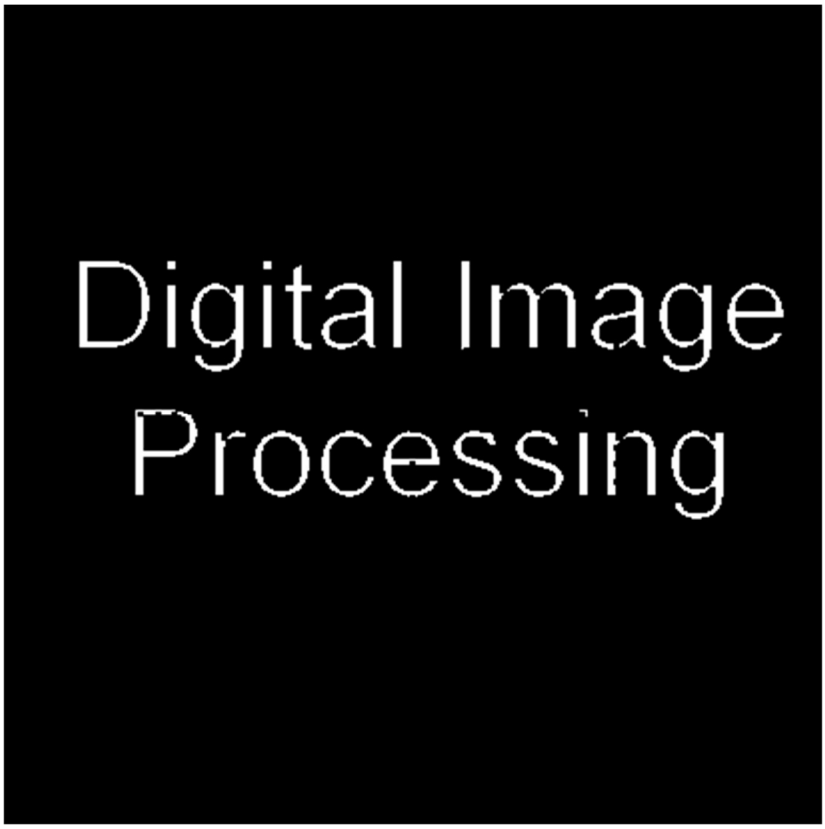

Figure 4: Erosion of pepper noise image

Figure 5: Input image without noise

2. **To extract the tall characters from the image:**
   In order to extract the tall characters, we need to obtain the marker points of these characters. As explained above, the marker points are obtained by eroding figure 5 using a 45x1 structuring element. The obtained result is as shown.

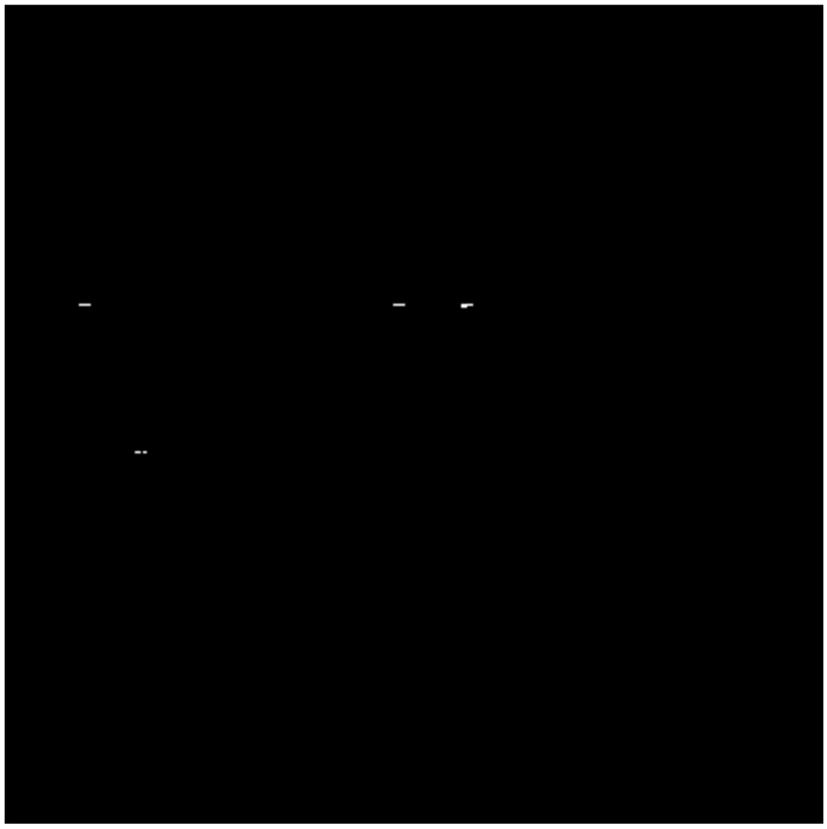

Figure 6: Marker points of the tall characters

Using figure 6 and figure 5, the tall characters are reconstructed by the algorithm presented above. This is executed in the 'reconstructed.m' file. The image after reconstruction is as shown in figure 7.

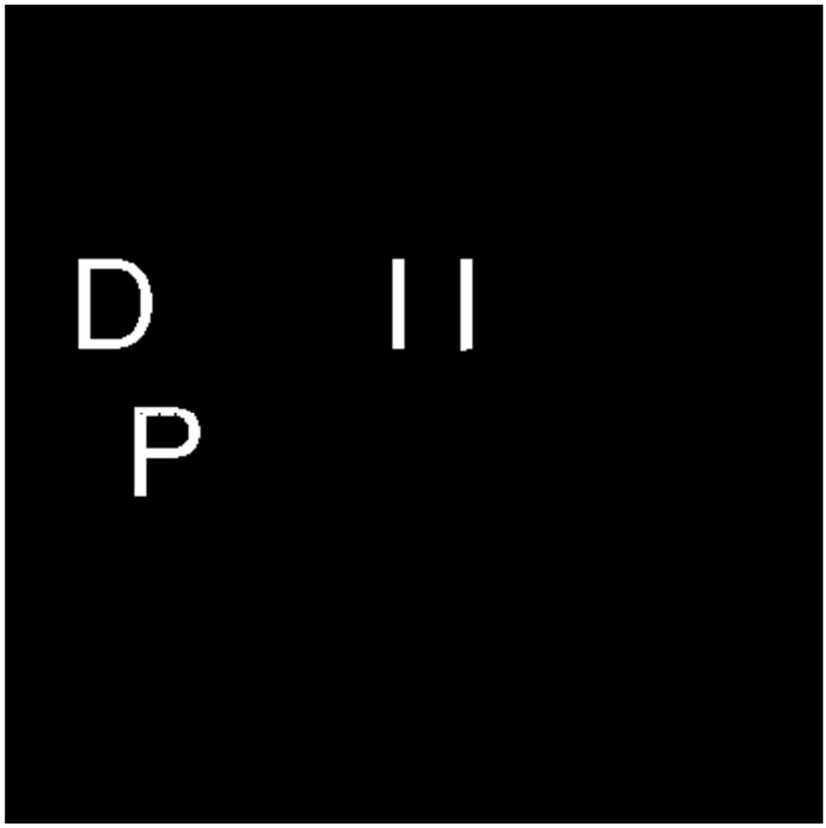

Figure 7: Reconstructed image

3. **To detect the edges of the characters:**
   Using the reconstructed image, the edges of the characters are extracted in two ways as mentioned in the Methods section.

The first method is what we believe is to be appropriate in order to extract the exact edges of the characters, which is the difference of the eroded reconstructed image from the reconstructed image. Figure 8 represents this image.

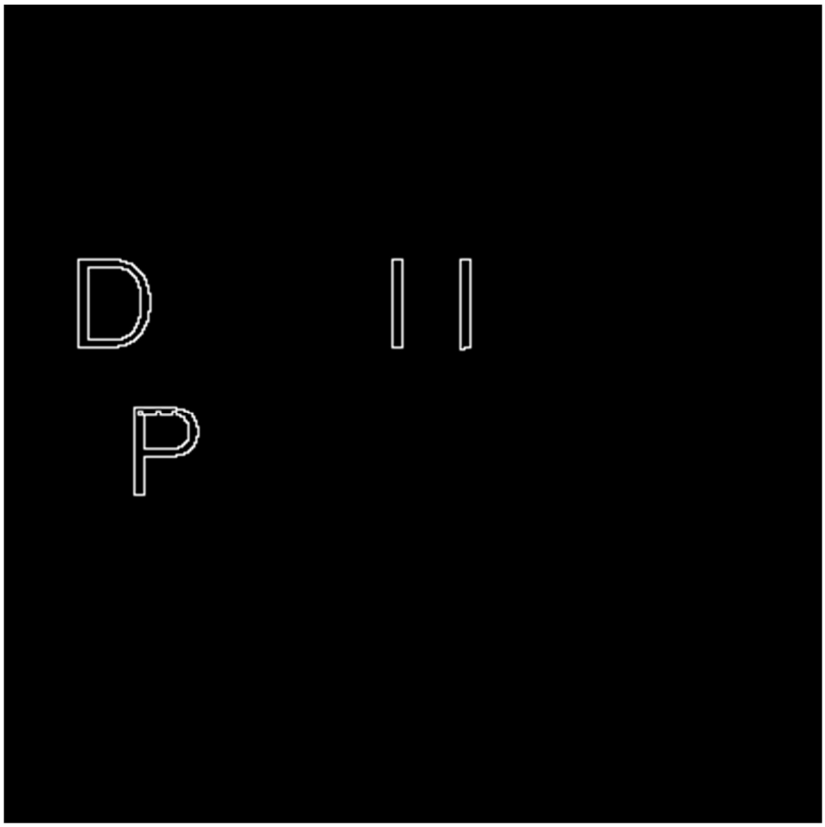

Figure 8: Actual edges of the tall characters (thin)

The second method is according to the definition given in L-26 which is the difference between the dilated reconstructed image and the eroded reconstructed image. The resultant image as shown

in figure 9, would be a thickened image of the edge since it would consist of an extra layer of pixels around the edges of the characters.

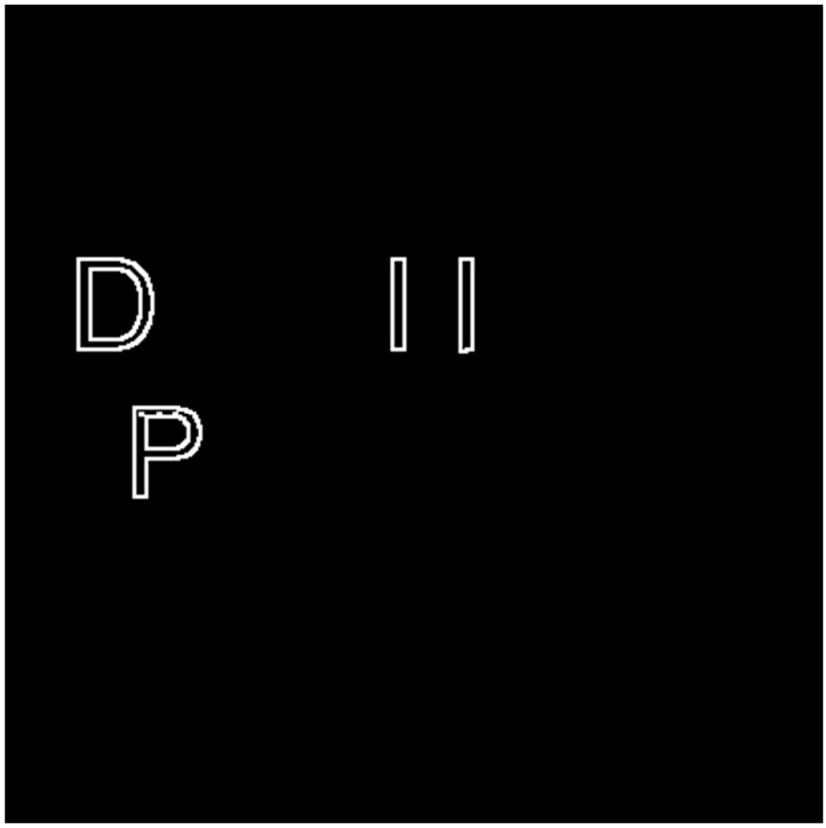

Figure 9: Edges of the tall characters (thick)

CONCLUSION

After completing the necessary tasks to meet the objectives of this project, we can come to a set of following conclusions:

- While performing morphological processing, it can be concluded that several types of noise such as salt and pepper noise, streaks, etc. can be removed with the help of erosion with an appropriate sized mask.
- It is important to study the image before morphological process can be implemented on an image. It can be concluded that there is no perfect size of a mask, and that it depends and varies on the image and the components present in the image. Therefore, in the real world applications, images are usually tinkered with different mask sizes before fixating on a definite mask.
- The algorithm proposed cannot be implemented in the case where the tall characters do not have a straight line, i.e. uppercase S, A, O, etc. Therefore it is necessary to choose an appropriate template.
- It can also be concluded that the distance between two characters must be beyond a particular pixel length such that the dilation does not cause the characters to join which is usually not intended. These cases strengthen the importance of choosing an appropriate template as per the application and the image.