Method\_2:

a):

To perform morphological operations on the image “proj5”, it has to be transformed into a binary image. Here the threshold is set to be 50 according to the histogram of the image which shows that the most of the gray values of the image are distributed at 0 and 75.

b):

i):

To reduce the noise including the dash lines and the cross lines. One of the possible approach is to erode the image by a 3×3 mask. The erosion of the image A and B is denoted as A⊖B and defined as:

*A⊖B = {z | ⊂ A, ∀z ∈ } = those pixels z such that Bz included in A.*

The result of image A eroded by B is a set of all points z such that B, translated by z, is contained in A. In terms of the 3 × 3 neighborhood mask, the erosion can be realized by the logic that only the pixels with each of the 8-neighbor-pixels have a binary value of 1 can be attributed as 1 in the new image.

It can also be realized by the following logic:

*f(x , y)⊖B = min{f(s , t)} for (s , t)∈N(x , y)*

This shows that the erosion can be realized by constructing a new image whose binary value of each pixel is the minimum value (0 or 1) of its 8-neighbors.

The reason that the noise of the image could be eliminated by this way is that the neighbors of the dashed lines and the cross lines are mostly background with the binary value of 0 such that the minimum value of those pixels with noise and its 8-neighbors is 0. It’s apparent that the size of the mask could be enlarged into a 5×5 mask so that more noise could be eliminated, however, the foreground will also be eroded by a larger scale.

ii):

Opening is one of the morphological operations which can be applied to extract the tall letters in the image. By eroding the image with a thin structing element which has the size of the horizontal length of the tall letters, we can isolate these letters. That is, we perform the erosion by a mask whose horizontal size is the same as the characters to be extracted. Then, the image could be reconstructed by dilation of the original image using that image as the mask and the eroded image as the marker and repeat the process until the result image doesn’t change. The whole process could be presented as:

*A1 = A⊖B “marker” image*

*A constraint mask*

*1st iteration:*

*() = ( ⊕ ) ∩ A*

*• • •*

*nth iteration:*

*() = ( ())*

c):

The boundary detection can also be realized by morphological operation. If we take a close look at the process of erosion, we may find that the erosion is nothing but cutting the edge of the image by a scale of the size of the mask, hence we can detect the edge by taking one more operation after the erosion, which is given by the following equation:

*β(A) = A−(A⊖B)*

where *β(A)* is the image that we want and B is a 3×3 mask. In MATLAB, the actual output is 1 minus this result in order to get a image whose background is 0 and foreground is 1.

Result:

The figure 2 shows the image with corruption reduced and figure 3 shows the image with D I L and P extracted.

A screenshot of a cell phone

Description automatically generated

*Figure 1: The Binary Image*

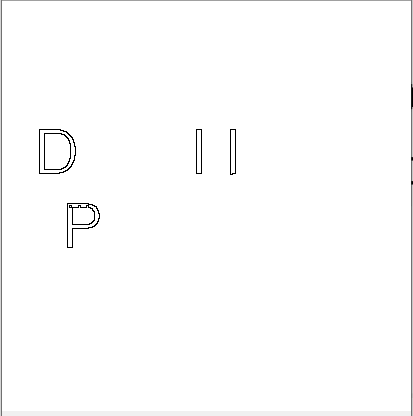
**

*Figure 2: The Image with Corruptions Reduced*

*A close up of a logo

Description automatically generated*

*Figure 3: The Image with Characters Extracted*

**

*Figure 4: Edge Detected Image*