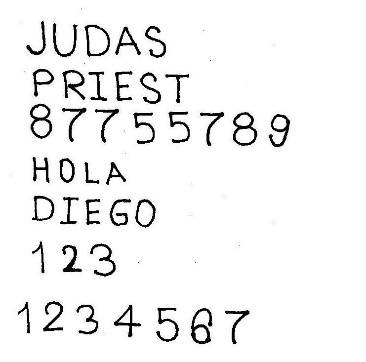
C:\Users\Administrator\Documents\Classes\SO2\CSSE463\Project\Term Project\Real code and tests\test\5\1\letters\1.png

So our first step is to preprocess the input image that has many words and lines into pictures of single letters. We want to keep the information of different words and lines. So we do not want to just get all the components directly. The way is mainly divide each line from a paragraph, then divide each word from a line, and finally divide each letter from a word.

Our dividing algorithm base on the average position of each component. When we divide the lines, we get a list of the average position of each component on x axis. For the letters on the same line, the value should be about the same. If the different between two consecutive values exceeds a certain threshold, we will divide them to be separate lines.

After getting a picture of one line, we would like to divide the line into words. This time we will want their average position on y axis. We believe that the interval between words and letters would be quite different. If the difference is larger than a certain value, we treat them as different words.

Then we get a picture that only contains one word in it. We may just use connect components to figure each word out and crop them by bounding boxes. But not all words are connected. For example, “i”, “j” are letters that are made by two parts. Again, we get the average position on x axis for all the pixels that is not 0. We treat it as an average line for a letter. If a letter is only made by one component, its average position should be about the standard we set. If one component is really far away from it, it is probably the dot of “i” or “j”. We will combine it with the nearest component according to the position on y axis.

When we store our pictures it will look like what I shown in the picture.

For our test cases, we will not only have single letters, but also pictures that are extracted from real documents. To apply to our trained letters, we first need to find out the words in the given picture. We mainly follows a flow that we

But our method now only works for letters that are clearly separated. If the handwriting is really cursive and all the letters or even words are connected, we don’t have a good way to separate them correctly.

Our idea of dividing mainly base