1 Requirements

Using Python and OpenCV write a program that reads a image file containing a scanned page of handwriting text containing multiple lines.

2 Algorithm

In order to solve this problem three steps are necessary:

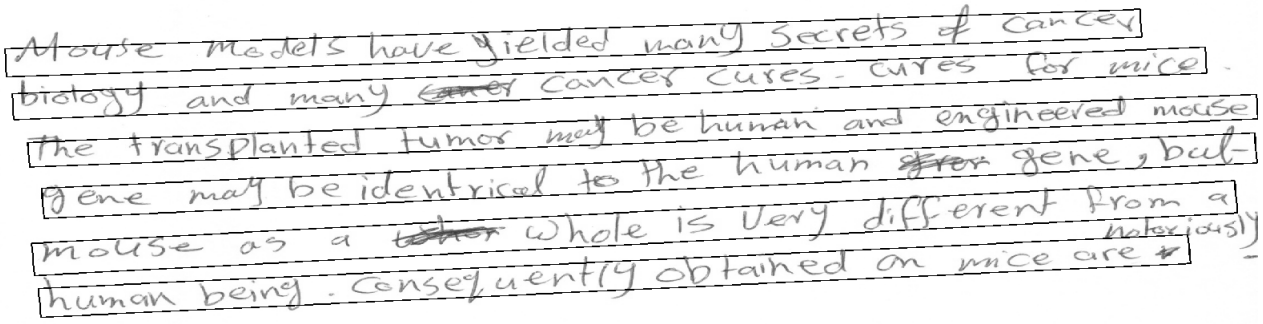
1. Image preprocessing
2. Line extraction
3. Character extraction

2.1 Image preprocessing

Firstly, to be able to process the image it needs to be binarized - this is achieved with the OpenCV's function `threshold` using Otsu's algorithm specified by the flags `THRESH_OTSU | THRESH_BINARY_INV`. Next the image is smoothed using a median blur filter and a morphological closing operation is performed to eliminate some remaining empty holes in the text.

2.2 Line extraction

To be able to extract lines of handwritten text that can be also unaligned asa a line or unaligned characters, one can use the Hough line transform in order to get a homogeneous representation of each line of the text. In order to apply the Hough transform, the OpenCV function `HoughLinesP` is used with the parameters `minLineLenght`, `maxLineGap`, `threshold` and the angles ρ and θ tuned such that the drawn lines cover each line of text.



2.3 Characters extraction

To extract the characters, at first, the extracted lines are cropped from the image, then the image is thinned using OpenCV's `ximgproc.thinning` in order to get a skeleton of the characters, next, a horizontal histogram is computed (the sum of pixel values on each column). Next, the possible delimitation lines are computed as follows: On the histogram:

- If the sum is equal to 0 - a possible delimitation line is present
- If the sum is 1 (or 255) - there is either a character ligature or a part of an open character (like 'u')