Text Detection and Recognition from Natural Scenery
Interim Progress Report

Ishtiak Zaman (izaman@indiana.edu)

Introduction

The project has two major parts. First part is to detect and localize text in an image. This part will be able to crop out only the text area from the whole image that can be used in the second part. I am using Stroke Width Transform [1] method to detect and localize text.

In the second part, we will have a cropped text area. I will use CNN to train the english letters and numerals from Chars74k [2] dataset using Caffe [3]. Once the training is done, I will use fixed length sliding window to figure out each characters in the text area using the trained network.

Background and Related Work

There has been many works related to text detection and text recognition using various kind of methods. Most commercially available OCR software can successfully read the texts from printed documents, but fails to read texts in natural scenery due to different orientation, color, font, noises etc.

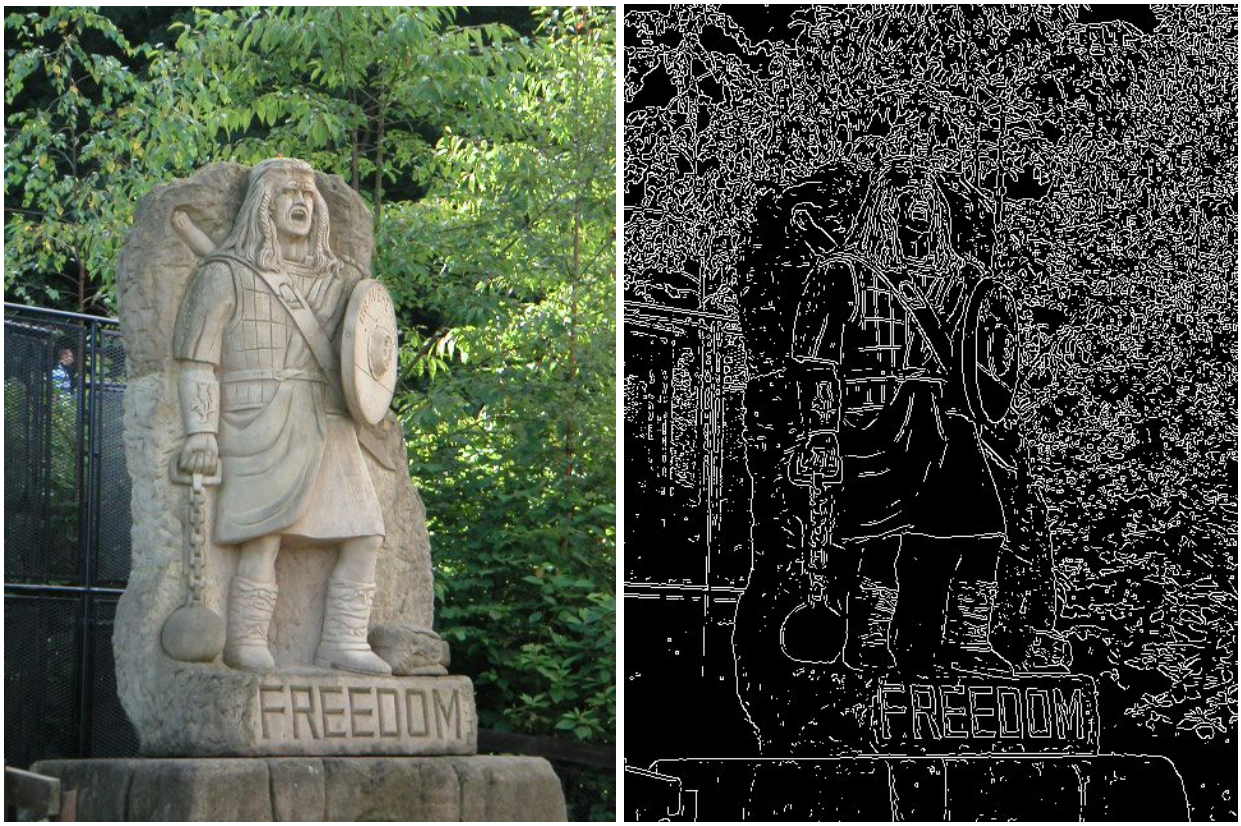
The PhotoOCR [4] uses three different kind of methods to detect texts, a mixture of Viola-Jones style boosted cascade of Haar wavelets, connected components from a binarized image and uses graph cuts to find a text / non-text labelling by minimizing an MRF with learned energy terms, and anisotropic Gaussian filters. Finally a trained network recognizes the text.

There has been excellent work on detecting text regardless of text size, font, color, orientation, language using Stroke Width Transform [1]. Text characters have similar stroke width compared to other elements in a natural scene, this property is used to detect text successfully. I am using similar technology in this project to detect text using Stroke Width Transform.

Progress So Far

Progress for part 1: I have made significant progress on part 1.

The first step is to get a thin noiseless edge map. I've used Canny edge detector [5] to detect the edges. This step is completely done. The following figure shows an image and detected edges.



The next step is getting stroke width of every pixel of the images. When an edge pixel is found, the method starts searching through a straight line in the direction of the gradient direction of that pixel to find another edge pixel with similar gradient direction. If such a pixel is found, all the pixels between the two pixels are assigned a stroke width that is the euclidean distance between the two pixels. This method is applied for all pixels to get a map of stroke width of all pixels. This step is fully done. The following figure shows the stroke width transferred image.

In the image above, the color of every pixel specifies the stroke width assigned with that pixel. We can see that the group of texts have similar stroke width. The next step is to group together neighbor pixels with similar stroke width, which would keep the text pixels and skip out other elements in the image with weakly grouped by stroke width. I am currently working with this part, which would be finished by April 13.

The current codes are present in the github repo that generates the images above.

Compile: make

Run: ./final <input-image-name>

Progress for part 2: I have made a few progress on part 2. Part 2 starts from the result

of Part 1. By the time I am working on Part 1, I've made the following preparation for Part 2:

- Learnt using Caffe on Bigred2 supercomputer.
- Collected dataset from MNIST and Chars74K.

Revised Research Plan

My initial plan was to use CNN to detect and localize text. I've revised my research plan to finish the detection and localization part by native computer vision methods. The part 2 research plan is still the same.

April 13: Finish Part 1, text detection and localization.

April 14-21: Work on CNN for part 2. Tune parameters to maximize accuracy.

April 21-22: Design poster.

April 25: Finish the whole project.

April 27: Present the final poster.

April 25-30: Polish the whole project by tuning parameters, testing with different type of images. Try to maximize accuracy and minimize false positive results.

May 1: Submit final project and report.

References

1. B. Epshtein, E. Ofek, and Y.Wexler. Detecting text in natural scenes with stroke width transform. In CVPR, 2010.
2. http://www.ee.surrey.ac.uk/CVSSP/demos/chars74k/
3. http://caffe.berkeleyvision.org/
4. PhotoOCR: Reading Text in Uncontrolled Conditions. Alessandro Bissacco, Mark Cummins, Yuval Netzer, Hartmut Neven. In ICCV 2013.
5. J. Canny, "A Computational Approach To Edge Detection", IEEE Trans. Pattern Analysis and Machine Intelligence, 8:679-714, 1986.