OCR Handwriting Project Outline

Matthew Mulhall

May 15, 2019

1 May 14, 2019

1.1 Summary of Design Decisions

The project will follow an abstraction based design: letters, words, lines, and entire documents. Every document can be broken down into these respective groups of abstraction.

(i) An entire document.

(ii) A collection of lines in a document.

(iii) A collection of words that are consecutively placed on each line.

(iv) Single characters that make up the words.

It can be seen that each level abstraction relies on the previous, going all the way down to the individual letters that are on the document. Given the nature of that abstraction Dr. Johnson suggested we start from the ground up, meaning first we will be building the data set for letters, and training a model to recognize other letters of similar (1800's English) style. Our current priority is to build this large data set of characters for our neural network to pull from. After this set is built up we will work on figuring out the optimal design of our model and start to train it. After this section is completed we will have a network that can identify individual characters. From this base level we will then work on the next level of abstraction, that will be able to identify the words in a line. The project will follow a similar style of abstraction based progress until we can use every level to read an entire document.

1.2 Some specifics

We currently have 8 documents that have been allocated for our project. The first 5 will be used to create the data set of images. On top of simple screenshots, we will be using GPUs to transform the images to get the most mileage out of each photo. The last 3 will be later allocated into development and strict testing sets. These will be allocated as the training set is developed.

1.3 Significant Developments

(i) Matt created a python script that renames the pictures in the subdirectories according to a naming scheme, this allows for saving files without having to worry about typing the name into the save box. Doing this means the whole process takes 10x less time. When taking photos one can either: focus on a letter saving several in a certain directory (fastest), or save all photos to a "dump" folder and place them afterwards in their correct directory.

Process for quickest imaging:

(i) Pick some page of the manuscript, and select one letter, for example 'a'.

(ii) Using Lightshot, take a screenshot of the letter, and click the save button. Navigate to the lowercase a directory and save.

(iii) For all subsequent a's, take the photo and use shortcut CTRL-S and it will auto-save to the same directory.

(iv) After you find as many as you can on the page, or several pages, move on to the next letter.

(v) Before pushing to git, run the renameUtilityScript.py file which will rename all of the files to the appropriate schema.

(vi) The script must be properly configured for the location of your local repository.