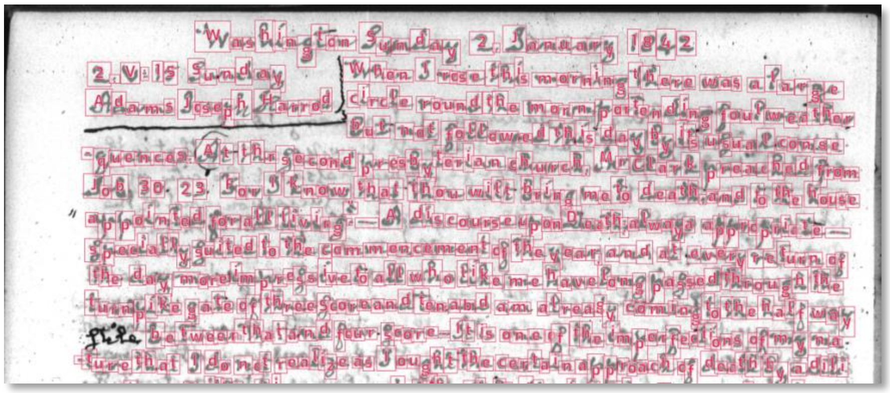
Handwriting Text Recognition (HTR) Project


Washington Sunday 2. January 1842
2. V. 15 Sunday
Adams Joseph Harrod
When I rose this morning there was a large
circle round the moon portending foul weather
But not followed this day by its usual conse-
-quences At the second presbyterian Church, Mr Clark preached from
Job 30. 23. For I know that thou wilt bring me to death, and to the house
"appointed for all living" — A discourse upon Death, always appropriate —
Specially suited to the commencement of the year and at every return of
the day more impressive to all who like me have long passed through the
turnpike gate of three score and ten and am already coming to the half way
stile between that and four score — It is one of the imperfections of my na-
-ture that I do not realize as I ought the certain approach of death, a dili

Catalyst Fund Final Report
Submitted by

Greg Colati, Connecticut Digital Archive
Joseph Johnson, School of Engineering, Computer Science & Engineering
Michael Kemezis, Connecticut Digital Archive
Sara Sikes, Greenhouse Studios

A Collaboration Between

UCONN LIBRARY

CT DIGITAL ARCHIVE
CONNECT • PRESERVE • SHARE

GREEN HOUSE STUDIOS

UCONN SCHOOL OF ENGINEERING
COMPUTER SCIENCE & ENGINEERING

This project was made possible in part by a 2020 award from the Catalyst Fund at LYRASIS.

Goal

The goal of the Handwriting Text Recognition (HTR) project is to develop a large-scale, open-source software for handwriting text recognition for historical documents by training a convoluted neural network (CNN) to recognize handwriting of 19th century scribes. We aim to reach this goal by using a combination of machine learning, artificial intelligence techniques and software platforms, specifically utilizing Graphical Processing Units (GPUs), to transform simple screenshots of characters from handwritten documents by slightly skewing each captured image several times to create a larger training set.

The first stage of our work was completed in the summer of 2019 by an undergraduate computer science student, Matthew Mulhall, who created a training set of over 16,000+ images of 22 different characters (or classes) averaging about 260 images per character from four volumes of the John Quincy Adams Papers (https://github.com/mattlm0831/OCR-Handwriting/blob/master/documentation/project-overview-report.pdf). Through a partnership with the Massachusetts Historical Society, we have access to over 200 pages of handwritten material from John Quincy Adams (JQA) (https://www.masshist.org/jqadiaries/php/) from the early 19th century. Obtained through the team's previous engagement with this extensive collection of key historical documents, these pages have served as the initial data source for our work. We chose this data set because of the regularity of the handwriting in these documents and the fact that they are already transcribed, enabling us to compare our automated work with the manually created transcripts.

Even though this amount of representative data is considered very low for building a CNN, our initial results were very promising. We were able to achieve a 96%+ success rate when testing on four character classes and an 86%+ accuracy when testing on the full 22 characters produced by the work completed in 2019. We determined the next step would be to further push the model and scale up to 70+ characters to increase the data set by 3-5x and perform more testing.

To create a larger data set, we had to do more work to review the remaining three volumes of handwritten documents from the JQA Diaries to identify and capture screenshots of additional handwritten characters. The goal was to capture more images of the 22 character classes from the original work done in 2019 and identify new characters to further build up the training set for our model.

We set out to hire two computer science students to continue working to expand the dataset and adjust the CNN as needed. To consider this project a success, we would accomplish the objective of expanding our dataset while also keeping the accuracy of our model at the same level or better than our previous accuracy rate. While we had issues with hiring student workers due to Covid restrictions concerning student work on campus and student hiring in the 2020-21 academic year, we have made considerable progress in strengthening the model, while also creating new tools and streamlining workflows to enhance the efficacy of the overall project.

Process

Expanding Dataset

The primary goals of the UConn LYRASIS student team, under the direction of Professor Joseph Johnson, was to increase data collection and begin the initial phases of text recognition on additional journal page characters from the John Quincy Adams (JQA) Diaries. Using a character collection tool built by Joseph Berchard, developed as part of this stage of the project (see more below), the team took a combined 25,000+ individual screenshots of letters from Volumes 40 to 43 of the JQA Diaries. These captured characters were automatically categorized into groups by letter type, using the newly developed capture tool, allowing the team to use these characters to begin developing prediction models much quicker than in the past. The prediction models use specially developed algorithms (https://github.com/JQA-Machine-Reading-Group), which make use of deep learning, in the form of a multi-layered artificial neural network (ANN). ANN is an attempt to mimic the function and structure of the biological human brain, which has proven in latest research to be a highly effective approach for computer vision (which is the focus of this project). However, the approach of using ANNs is only the beginning for making an accurate prediction model. One main concern is to determine how many layers to build into a model and how many nodes to include in each layer. Many optical character recognition models have over 100 layers, such as the one designed by Yan LeCun for processing US mail (L. Bottou et al.).

The student team used Keras libraries (https://keras.io/) and Jupyter notebooks (https://jupyter.org/) to develop convoluted neural networks (CNNs) on seven base letters (a, e, n, s, t, i, and l) for this phase of the project. After various iterations and layer implementations, a prediction accuracy on the test dataset has ranged between 85% and 95%. As we continue to increase the size of the dataset, this accuracy is expected to increase. Furthermore, other methods such as K-Fold Validation[1] are being implemented to artificially introduce variation in the data for increased accuracy.

Enhancing Model

The team developed and tested three separate predictive models during the project. The team analyzed each of the model designs under the umbrella of ANNs and investigated which ones performed best. This will help inform our research for the future of the project, as well as other outside researchers about the optimal settings for analyzing historical documents that are hand-written with a feather pen.

Model 1

The first model consists of three convolutional layers with depths, or the number of layers of nodes in a neural network, of 32, 64, and 64 respectively. Each

[1] K-Fold Validation is a means of splitting the dataset into several slices – such as 5 groups each constituting 20% of the entire dataset - which can then be used to train the model and then test the accuracy of that trained model. The process gives researchers a distribution of testing accuracies from which they can make a more reliable estimate of actual performance once the model is unleashed on completely new data.

convolutional layer is followed by a Rectified Linear Unit (relu) activation layer[2] and maxpooling filters[3]. In the final convolution, the data is flattened, parsed through a 128 deep Dense layer, and outputted through a 7 deep Dense layer for the 7 classification classes (letters a, e, n, s, t, i, and l).

After training on these 7 letters, this model implemented Stratified K-Fold Validation (10=folds) on the validation dataset for an average validation accuracy of ~85%. The letter dataset has a minimal number of screenshots of all the other letters due to their lower frequency of use. However, running this model on all 26 letters (lowercase) with the same stratification gave an average validation accuracy of 77.17%.

Model 2

The second model consists of four convolutional layers with depths of 32, 32, 64, and 64 respectively. Each convolutional layer is followed by a relu activation layer and maxpooling filters. In the final convolution, the data is flattened with a dropout of 0.5, then parsed through a 64 deep Dense layer and output through a 5 deep Dense layer for the 5 classification classes (a, e, n, s, t). With an initial test on the validation data, it has an accuracy of ~86% across the five letters chosen.

Model 3

The final model consists of three convolutional layers with depths of 256, 512, and 1028 respectively. Each convolutional layer is followed by a relu activation layer and maxpooling filters. In the final convolution, the data is flattened with a dropout of 0.5, then parsed through a 64 deep Dense layer and output through a 5 deep Dense layer for the 5 classification classes (a, e, n, s, t). With an initial test on the validation data, it has an accuracy of ~95% across the five letters chosen.

[2] A relu layer addresses what is known as the “vanishing gradient problem” – a situation in training neural networks as you begin to introduce more layers. Without intervention, the network will either not be able to learn anything or give poor or wrong solutions about the data due to being unable to properly process all the layers in the network.

[3] A pooling layer addresses a locational translation concern in neural network training - when training the model with large numbers of images of the same object, we don't want the model being thrown off by the fact that a specific character in an image is in a different position than the same character in a different image. To make sure the model learns the pattern across different scales and angles of the classes of characters, character images captured from manuscript pages are taken and turned or skewed slightly, producing different relative positions. The pooling layer deliberately blurs the signal provided by pixels in an image as they are processed by the algorithm by combining the outputs of nodes into fewer nodes so that a more stable signal is generated across images of the same objects but in different positions.

Summary of Models

Model #	Layers	Accuracy	K-Fold Validation
1 (7 Letters)	3 (32,64,64)	85%	Yes
1 (26 Letters)	3 (32,64,64)	77.17%	Yes
2 (5 Letters)	4 (32, 32, 64, 64)	86%	No
3 (5 Letters)	3 (256, 512, 1024)	95%	No

Character Capture Tool

While working to expand the dataset, the team realized the original method of capturing character images from manuscript pages caused massive bottleneck in their work. The system set up initially used a script that leveraged the native screen capture tools on the Windows machine used for this project, that would allow a user to capture one character at a time and place it into the appropriate folder for that character class. Marking up a single page with hundreds of characters with this tool was not the optimal workflow.

Team member Joseph Berchard developed a tool to solve the problem that allows users to upload, markup and save characters from entire handwritten pages identified by the user into the correct directories of each letter (https://lct.jbec.us/). Using this tool, users can view a page of a handwritten document and can click and drag rectangles to denote characters on the page. Users can then denote what character they have identified in the rectangle on the page. The system also has a predictive text function that attempts to predict the character they have chosen on the page. The tool also allows users to select entire words and lines of text for capture. Once an entire page is marked up, the user can save their work and each selected character on a page is saved into the appropriate directory. A full instructional video can be found on YouTube (https://www.youtube.com/watch?v=UyPV7N6FHfE&ab_channel=JoeyBechard).

This tool also captures coordinate page metadata for each character that is collected from a page. Using these coordinates, the team has begun to approach another issue in OCR and handwriting text recognition, identifying lines of text in documents. Currently our research is based on identifying and transcribing characters contained on manuscript pages. The next set after refining this model will be to start identifying lines of text on page images. The image capture tool begins to bridge the gap between identifying characters and identifying lines of text in a document.

Accomplishment

We accomplished three main goals in this phase of the project.

First, we designed and developed a critical tool for expediting the data collection process, the Character Capture tool. The tool helps the team quickly crop images of characters, annotate them, and organize them by character for training and testing using our models. The augmentation of the speed with which we collect data is easily a factor of 10. This increases our ability to create a more complete data set much more quickly and efficiently, allowing us to make more informed inferences about the performance of our models faster. We now have over 30K images because of this newly developed tool, and our goal is to capture over 100K - 200K character images.

Second, the increased size of the dataset, facilitated by the tool described above, allows us to better understand how adding more data impacts the accuracy of our initial models. By increasing our dataset size to 30,000+ character images, we have increased accuracies of our classification models from 80-85% on 5 characters to 90-95% on 10 characters. These findings helped us quantify our needs for the appropriate dataset size to reach a level of 95-99% accuracy on 72 characters (a mix of lowercase and uppercase letters, numbers, and punctuation), which will fall between 200K-500K character images.

Finally, we have started our exploration of the way in which hyper-parameter settings, or settings that help refine nature of a deep learning neural network, affects the accuracy of our models. This includes the way in which regularization, randomized drop out (both are intended to prevent overfitting), the number of layers, the number of nodes per layer, the activation function, and batch normalization affect our results. We are still in the early phases of this, but we have found three models due to this project, which shows promise given a larger dataset.

What We Learned

One of the main lessons we learned is that we can only do so much with one dataset to push this project forward. If our goal is to develop an open-source tool for handwriting text recognition and transcription more generally, we will need to work with other datasets from other institutions. Using the tools, workflows, and models developed as part of the work funded by this grant, we now see a path forward to approach other handwritten collections that we did not have when this project started.

We also found that we need to better organize, document, and disseminate our overall project. Different parts of code that make up the entire project live in different code repositories on GitHub and on local machines. While most of the code is freely accessible, we need to consolidate and document everything in one central code repository. We will work with Greenhouse Studios (see more below) to set up a central Git repository to consolidate all the code produced for this project so far. Once UConn employees are allowed back on campus in the fall 2021, we will work to extract any useful code or datasets from the local machine primarily used for this project and make them available in the Git repository.

We also found out that coordinating and implementing a project such as this one 100% virtually is possible, but not ideal, and was especially challenging during the early part of the Covid-19 pandemic. Hiring students was also surprisingly difficult, due to most students taking courses remotely both semesters and deciding to work less, or not at all, because of the disruption of their expected college schedules. We also must contend with hiring procedures that are based on in-person practices, such as ID verification for certain paperwork that needs completed in-person on campus. With most students not attending class on campus, we had to work with the hiring team at UConn Library to find other arrangements.

What's Next

The LYRASIS funding helps us progress to the second phase the Handwriting Text Recognition project we started two years ago. Obtaining outside funding for the project allowed us to build on the progress we made with internal UConn funding in the summer of 2019. Now that we were able to expand and refine the working model, as well as develop new tools to allow for more efficient mark up of handwritten documents, we can now confidently reach out to new external partners to collaborate on the next phase of the project, as well as apply for funding from other sources to push the project forward. As we have learned in wrapping up this phase of the project, a dedicated project manager is something we need, and we think we have found a solution to this need internally at UConn Library.

Upon discussing the project with staff at Greenhouse Studios (GS), a unit in UConn Library, and presenting the progress made during this grant funded period, GS has agreed to pick up the project and add it to their portfolio of other research projects starting this fall. GS has a proven track record of success with other digital humanities and technology projects, including Sourcery (https://sourceryapp.org/), a digital archival document delivery application, and Charles VR (https://greenhousestudios.uconn.edu/charlesvr/), a virtual reality reconstruction of the coronation of Holy Roman Emperor Charles V. Formally taking this project to GS is beneficial to the HTR Project and GS in several ways. First, GS has the infrastructure and appropriate support in place to successfully manage digital projects, including dedicated project managers. Second, this move would take the project into the area of Computational Humanities and other technical areas of Digital Humanities that we have so far not investigated. And third, the project would open a new area for GS to collaborate with the harder sciences, something they currently do not do.

Additionally, we currently see several avenues for continued technical work on this project: further expanding the dataset, benchmarking the model against human transcriptions, and further developing the image capture tool. First, we will look to expand the training set even further by incorporating authors from other time periods adjacent to JQA Diaries. Using the connections developed by the Connecticut Digital Archive (CTDA) and GS, we will reach out to institutions around New England with handwritten manuscript collections and partners to further add to our data set using the processes outlined above. Additional data will help us explore hyper-parameter settings on the model to make it more accurate.

Second, we will look to partner with institutions that have existing transcriptions of handwritten documents and work with them to run our model on the handwritten images. We hope to get a benchmark of how accurate our model is in transcribing these documents by comparing the output of our models to the “official” transcriptions.

We see enormous potential in further developing the character capture tool developed as part of this project. Starting in fall 2021, we will work to move this tool to a place that will be accessible to future team members to further develop, administer and maintain. While the Convoluted Neural Network (CNN) and the deep learning process involved in creating the network are still in development, the character capture tool is something that we feel can be moved to systems maintained by Greenhouse Studios and UConn Library to be used to demonstrate the progress, promise, and viability of the project to interested parties. As a large national consortium, LYRASIS would be able to play a key role in identifying members who would be interested in partnering with us in the next two phases of our project. While our connections in New England are strong, we would like to work with a variety of institutions from around the country on this project.

Project Website

As part of this project, we have created a landing page for this project in the Connecticut Digital Archive Resource Center. The project page can be accessed at this link:

https://confluence.uconn.edu/display/CTDA/Handwriting+Text+Recognition

Project Press Release

The Communications and Engagement unit at UConn Library worked with the Office of University Communications to produce an article about the project and the grant that appeared in the Communications web publication, UConn Today:

https://today.uconn.edu/2020/07/uconn-library-school-engineering-expand-handwritten-text-recognition/

Citations

L. Bottou et al., "Comparison of classifier methods: a case study in handwritten digit recognition," Proceedings of the 12th IAPR International Conference on Pattern Recognition, Vol. 3 - Conference C: Signal Processing (Cat. No.94CH3440-5), 1994, pp. 77-82 vol.2, doi: 10.1109/ICPR.1994.576879.