An Analysis of the Use of Machine Learning to Detect Marginalia

By Jeff Niznik, Book Traces Research Assistant

At least for a human, identifying a page as having or not having handwriting is easy. Book Traces' mission to find and catalogue interventions in books written before 1923 has been built upon manual searching. Public submissions from curious readers and detailed entries from Book Traces student workers have led to thousands of documented marginalia, inserted artifacts, and other miscellaneous findings. There is huge potential for expansion beyond the University of Virginia, and several other universities have been inspired to start their own projects and adopt techniques similar to what worked well here. But setting up the same large-scale inventory sampling can take time, effort, and funding. There is an immense volume of old books out there, and looking at every single page isn't practical. By introducing an automated way to identify these interventions, the process would be streamlined, and libraries the size of UVA's could be explored in a fraction of the time. It may be less thorough compared to human searching, but this is a small sacrifice for the efficiency and reach of a system that can search page images. This is true especially considering the work already done by the Google Books Project, which digitized an unprecedented number of works around the globe.

While using machine vision to identify handwriting is somewhat of a niche task, there is plenty of existing research in similar areas. For example, optical character recognition (OCR) is being used to decipher and digitize ancient texts. Faced with a similar problem of expensive, time-consuming manual transcribing, researchers in the humanities welcomed advances in deep learning in the early 2000s that improved not only speed, but also error rates (Salian). For these more challenging problems, Recurrent Neural Networks, or RNNs, are normally the best tool. The "recurrent" aspect, keeping with the theme of neurological analogies, describes somewhat of

a working memory that the models have. This ability to take into account recent information is key for natural language processing and time series data (Nigam). A digital humanities project called HisDoc[1] has been researching the analysis of historical documents with complex layouts. One model they use is a Long Short-Term Memory, a type of RNN. They have been able to classify regions of a work, attributing authors or scribes for handwritten characters and drawings.

Our task is different in that we aren't concerned with what the handwriting and marginalia actually say, at least in order to find them. Prioritizing the visual over lingual aspects of our data makes the choice of a Convolutional Neural Network over an RNN clear. CNNs are the go-to model for image classification, and have been dominating competitions in recent years. The basic process for a CNN involves feature extraction by running a small grid, called a filter, across the pixels of the image multiple times over and synthesizing the results, and prediction, by classifying the features into categories. The multi-layered structure of these nets allows them to learn what aspects indicate that image belongs to a certain class or contains a certain object. One drawback of using CNNs, and deep learning models in general, is that they aren't interpretable. Machine learning differs from other areas of statistics in that it is results-oriented. In some cases, this is a frustrating property that hinders our understanding of what the model is using as a basis for prediction. In other cases, including ours, a prediction-focused framework is a great fit. When we are confident that the results are reliable, there is little disadvantage in casting our page scans into somewhat of a black box. There is the potential concern of how the model would handle inputs that are unlike anything it has ever seen. But having spent hours looking through digital libraries, we can say there isn't too much potential for surprise out there. Page scanning errors by Google can look strange and do pose a threat, but they are also rare. Examples could be included

[1] https://diuf.unifr.ch/main/hisdoc/

in the interventionless half of the training set, and the worst-case scenario would be a very occasional missed intervention on a warped or obscured image.

As a proof-of-concept that this type of prediction is possible with our data, we developed a CNN model using the Keras API to TensorFlow to classify images as containing or not containing handwriting. As mentioned earlier, the broad category of interventions goes beyond handwriting to include insertions like newspaper clippings, flowers, letters, and photos, and other designations formalized as part of the Book Traces vocabulary. But the focus of the image classification model we created was solely handwriting. A future project's model could identify the other types of interventions beyond the two categories we chose. The only obstacle is collecting enough examples of each class so that the model can generalize, as this was a challenge even for us with only two categories. Data augmentation and/or simulated data could help here. Although the work of describing which type of intervention occurred and where was already done, downloading and labelling the images was still time-consuming. We selected entries on the Book Traces database that included marginalia, annotations, or inscriptions, and searched to see which were also available on the HathiTrust digital library. This resource allows us to narrow results to only items scanned at UVA, and shows the barcode in the URL so we can be sure we have an exact match. Most of the books chosen had more than one type present, and the sampling favored books that had many examples of handwriting, to speed up the labeling process. But because the numbered pages on Hathi start counting at the cover, and included a variable number of endpapers, they are not an exact match with the recorded locations of interventions already found by Book Traces. Selecting images to download was still fairly quick, as handwriting is visible even on thumbnail-sized pages that can be scrolled through with a few rows of around 7-13 pages on screen at a time. Knowing which books to check in the first place

also made the task of labelling significantly faster, as roughly one in eight pre-1923 books contain interventions at all (Book Traces White Paper). A total of around 80 books were picked, and an average of 25 pages per book were included. Keeping the total number of pages with and without handwriting equal was important. If the model was given an unbalanced set of images, such as whole books with only a few instances of handwriting, it would end up predicting the larger class much more often, even if given exclusively images with handwriting.

Subsets of pages for the books chosen were downloaded as pdf documents, read into R, resized, shifted to greyscale, and saved as separate PNG files. The shrinking and removal of color was done to save computing time, as training would later be done on an old, underpowered personal laptop. But results probably were not affected much by these decisions. The images, sized at 306 by 396, are relatively large for machine learning, and most were already in greyscale anyway. The intricacies of small print text and handwriting call for decently high-resolution images, and small filter sizes for the neural net to be able to find them. Google's scanning process involved editing the images to increase contrast, which made a typical page of text basically black and white. It seems that only in the presence of significant color, like covers or pages with blue library stamps, would the pages include color at all. Also, in some of the old, aging books with staining and speckling, the pages would appear as their actual tan color, presumably to make the words more visible than if the splotches had been made dark with higher contrast. When processing power is not a concern, for example in a parallel computing set-up, these full-size color images could be left alone.

The images were then assigned to different folders based on if they contained handwriting, and what they would be used for. We had a thousand images for each of our two classes, separated into a training set of 600 images each, a validation set of 200 images each, and

a testing set of 200 images each. Care was taken to make sure that none of the images in the testing set came from books seen in the training or validation sets; the test images were completely new to the model. One thousand images per class is very low for image classification, and most examples of using CNNs are running on tens or hundreds of thousands of images. To reconcile this problem, we relied on two techniques: data augmentation, and pretrained models.

Data augmentation involves creating many examples of new images by altering each training image in a number of ways. For example, we specified limits for how much the image generator can rotate, and horizontally or vertically shift them by. Unfortunately, we are uniquely restrained in how much we can alter our images. The people predicting on images of dogs and cats, for example, can zoom, rotate, flip, and shift their images a good amount and still have a picture that clearly contains a dog or cat, even if part of the animal is cut off. But many of our images, especially marginalia, can only be accurately judged if the edges of the image are in view. For flipping, even though the handwriting is preserved, it didn't feel right to train on backwards or upside-down images that are unnatural and the model would never come across in the real world. Scanned page layouts are also consistent in how they appear, so we only rotated by up to 2 degrees and shifted by up to 3% in any direction.

Using a pretrained model, a type of transfer learning, is to build off of existing work for a model usually trained on one of the famous huge databases, like ImageNet, which contains millions of images[2]. The new model picks up where the old left off and applies feature maps it learned to continue learning on its own specific task (Marcelino). The task can be fairly different from the model that was used as a starting point, because the information it contains is useful for

[2] http://image-net.org/about-overview

classifying and differentiating in general. We downloaded and used the VGG16 architecture as the base of our CNN.

The results, a prediction accuracy of about 83% on the testing set, aren't quite ready to revolutionize finding interventions, but they show clear evidence that CNNs will work for our task. Its's useful to consider what would make this problem difficult even if we did have substantially more images to train on. As stated earlier, contrast increasing by Google forces pixels to go towards either black or white. When applied to faint or faded handwriting, the result is spotty or dashed lines that we can see were from writing, but a machine might struggle to differentiate from dots and lines that just random marks or noise. Raw images might not have this problem, but Google's library of scanned images is too convenient to not take advantage of. Nicely-inked hand-drawn art centered on a blank space will be difficult to distinguish from artwork printed on the page. Library check out cards actually do have handwriting on them, but it is not the kind we are interested in. Some books have captions in the margin, which could be mistaken for marginalia. Occurrences like these are what complicates the initial problem of detecting handwriting that we perceive as simple. While we paid attention to evenness among the classes, we did not keep track of getting enough individual examples of unique cases.

Although the output of my and any similar model is binary – 0 for no handwriting, 1 for handwriting – we can also access the predicted probabilities between 0 and 1 generated before they are rounded. With these, we can adjust the cutoff point to make the predictions either better in terms of sensitivity or specificity. For example, if we want to miss fewer interventions in exchange for false positives, we can lower the cutoff to 0.45, making it easier for an observation to be classified as a 1, or containing handwriting. On the other hand, if we are more concerned about wasting time checking books that don't have any interventions, we could raise the cutoff to

0.55 and have fewer false positives, but more false negatives. We can also design a function to combine all of the individual page probabilities for a given book into a single probability that the book has handwriting. We could then compare these cumulative probabilities for many books in a collection or library and search the ones most likely to contain interventions. Because the output is never a perfect zero, we would have to ensure that this function is not biased towards giving long books higher probabilities. By adjusting for this for size-biased sampling effect, we can compare probabilities that books of different lengths have handwriting.

Using supervised machine learning, we don't explicitly code what marginalia and interventions look like or where to find them. Instead we provide large numbers of examples of images that are labeled, and the model trains itself. Because of this, the predictive power of the model is only as strong as the data it is provided with. A query of HathiTrust shows that several libraries have hundreds of thousands of books available in full view, indicating we can see and download each of these works in their entirety. Building a comprehensive training set of thousands of images will not only increase the accuracy, but also prepare it for the variety of possibilities posed by every new page. There is no perfect substitution for careful manual searching. And going through volumes by hand can be beneficial by letting us learn about our own libraries. Book Traces was able to collect other data while searching, find where missing volumes were a problem, and fix books in need of binding or boxes. Manual search in conjunction with image classification techniques can greatly narrow the search. It would make locating the bulk of interventions easier, and could generate interest in what else is waiting to be discovered inside books that aren't digitally available.

Works Cited

Book Traces White Paper

Marcelino: https://towardsdatascience.com/transfer-learning-from-pre-trained-models-f2393f124751

Salian: https://blogs.nvidia.com/blog/2019/01/24/deep-learning-deciphers-historical-documents/

Nigam: https://towardsdatascience.com/understanding-neural-networks-from-neuron-to-rnn-cnn-and-deep-learning-cd88e90e0a90