Print and Probability

Lynn Dang Xuan Wang Aneesha Ramaswamy Jonathan Zamora

I. Research Context and Problem Statement

The study of historical documents provides modern researchers with the opportunity to deepen their understanding of the past so that their findings can better inform the present and future. While researchers have been able to study these documents by hand one at a time, Optical Character Recognition [OCR] techniques can alleviate the inconvenience of manually transcribing documents while also offering some practical benefits. For example, they can help us understand old linguistic practices and find the exact publishers for anonymous documents. Furthermore, although modern OCR models have made substantial progress in the field of historical document transcription, there is still a great deal of work left to do before OCR models can accurately transcribe documents with irregular patterns.

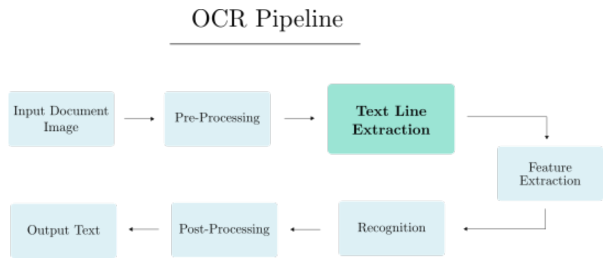

Fig. 1. OCR Pipeline

An essential component of the OCR pipeline (shown in the figure above) is the text line extraction process. Text line extraction segments a page based on an approximation of where text lines are located, and the quality of each segment determines how accurately the text lines can be read later in the OCR pipeline. Unfortunately, issues such as noisy or under-inked characters, ink blotches, and inconsistent document formatting have complicated the text line extraction process for historical documents [2].

Previous approaches to the problem of text line extraction have implemented deep neural networks as a part of their line segmentation process [3] [4] [7] [8]. The most common deep neural network approach used by researchers involves applying a Convolutional Neural Network [CNN] to the task of line segmentation and OCR. The CNN approaches help generate probability maps that determine the likelihood of a pixel being a part of a text line or not. A significant drawback found within each of these CNN approaches, however, is that they make independent predictions for each pixel, so the overall probability map may not be as accurate since there is no extra step to check if the pixels truly make a line. Other techniques have implemented graph-based methods and dynamic programming in order to either approximate separating seam paths between text lines [9] [6] or to cluster pixels together so the program can determine which clusters are lines [5].

One of these methods which we will be building on in our work is *dhSegment*. While these methods have shown promising results for modern documents, they did not achieve comparably great results when applied to historical documents due to their irregular text line patterns since the grand majority of papers that utilized these methods ran their tests on much less rigorous datasets than those currently being worked on within our ”Print and Probability” project team.

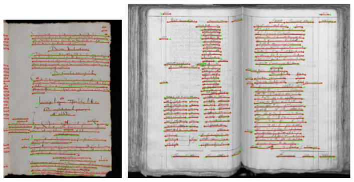

Fig. 2. dhSegment Baseline Extraction

Within our work, we will be utilizing two deep neural network models as the central components of our line extraction model for historical documents in order to improve line extraction accuracy. The first model will build upon the previously mentioned *dhSegment* solution and the second model will be a unique additional component intended to increase the accuracy of the process. Elaborating on our proposed solution, we will be automating the line segmentation process for historical documents by using a robust model combination approach that can take in any probability map of pixels and correlate the pixels such that it can determine whether or not they make up a line. In doing this, we will have achieved our ultimate goal of accurately transcribing text lines in historical documents.

II. Our Approach

Prior text line extraction approaches have followed the general model in which a probability map is generated from a historical document, and a certain threshold probability is specified as a cut off to determine if a pixel is or isn't part of a line. To improve this model, we will instead be utilizing a model combination approach as shown below in Figures 3 and 4.

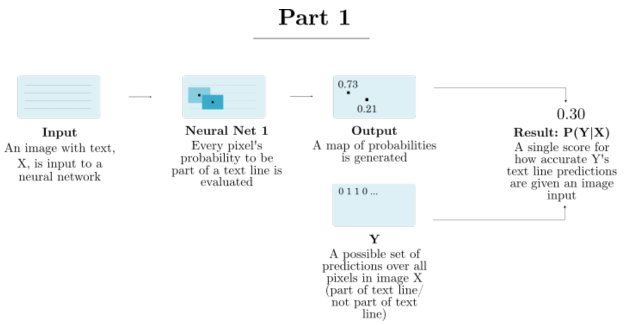
Part 1
0.73
0.21
0.30
Input
An image with text, X, is input to a neural network
Neural Net 1
Every pixel's probability to be part of a text line is evaluated
Output
A map of probabilities is generated
Result: P(Y|X)
A single score for how accurate Y's text line predictions are given an image input
0 1 1 0 ...
Y
A possible set of predictions over all pixels in image X (part of text line/ not part of text line)

Fig. 3. dhSegment/previous CNN solution model

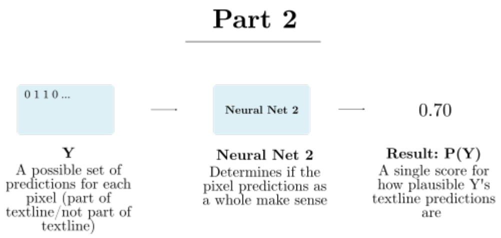
Part 2
0 1 1 0 ...
Neural Net 2
0.70
Y
A possible set of predictions for each pixel (part of textline/not part of textline)
Neural Net 2
Determines if the pixel predictions as a whole make sense
Result: P(Y)
A single score for how plausible Y's textline predictions are

Fig. 4. Our added component

A. Model 1: $P(Y|X)$

The first component of the model, Figure 3, is similar to previous text line extraction approaches. It will be a neural network trained to output a score for $P(Y|X)$ to determine the probability of how plausible Y is, given an image X. In this case, Y would be a possible predicted set of labels for pixels belonging to a line in the given image X. For example, if a pixel in Y is part of a text line it could be labeled with a '1' and if it was not, a '0'. The very first time Y is generated, it will be random. $P(Y|X)$ makes its predictions based on a generated map of probabilities from an image X and the model's built-in algorithm, outputting a single score for how accurate Y's predictions are given an image input.

This Neural Net 1 will likely be accomplished based on the use of *dhSegment* or any other previously used CNN architectures[1]. While the first part of our model provides a basic solution to text line extraction for historical documents, it only makes predictions for each pixel using the neighboring pixels without explicitly verifying whether the predicted baselines would actually make sense as lines. As a result, some of the predicted baselines produced by this model may not make much sense, but they would still be counted as lines. To improve upon this existing technique, we have proposed a second model that will be used in combination with the first model.

B. Model 2: $P(Y)$

For the second part of this process, we will train Model 2, displayed in Figure 4, to correlate neighboring pixels and learn what lines should generally look like, considering important characteristics such as whether the lines are straight, horizontal or parallel. This new model will take in the same Y as Model 1 and a possible set of predictions for each pixel, and it will output $P(Y)$, a single score for the predicted lines of pixels over the entire image to see if they make sense as a whole. This score, however, will need to be incorporated with the output of the first model to further improve the accuracy of the predicted baselines.

C. Search Function

To do so, we will perform a linear combination of each model's output and determine the most appropriate Y that outputs the most accurate baselines. We will compute this using the argmax to provide the maximized probability of $[P(Y) * P(Y|X)]$, ensuring that the pixel belongs to a text line. Finding this argmax is challenging because it is an NP-Hard problem, so heuristic complex methods would require a large amount of time to reliably calculate it.

In our approach, we will instead implement an approximate method to find the argmax through the use of a greedy algorithm. From this argmax greedy algorithm's output, the labels of Y will be continuously improved and fed back into Model 1 and 2 to find better baseline predictions. Based on this output, we can then make a more concrete decision about where the baselines are on our historical documents, further allowing us to segment and extract text lines from our historical documents.

III. Evaluation Plan

To evaluate our model, we have prepared a large dataset of ground truth files from years of manually annotated text lines along with some of our own annotations. This dataset will be comprised of documents printed from the early modern era (1500 - 1800's) with challenging printing inconsistencies. While most of this dataset will be used for the task of training our model, a portion of it will be reserved to test the accuracy of the model. For the models tested, we will compute the intersection over union of the pixels, comparing the line regions of the predicted output of the model and the ground truth from the annotated pages. This will essentially determine the overlap in pixel labels from the predicted model and the ground truth labels. If these labels have a strong overlap, that means that the predictions are more accurate. Another accuracy metric that will be measured is taking the average difference of the distances from the predicted baseline to the true baseline according to the y-coordinates. The shorter the distances of the predicted baselines from the true baseline, the more accurate the predictions are.

Using these accuracy metrics, we will test our model against previous models. One of these models will be the *dhSegment* model by itself to evaluate how well our model can improve upon the results from *dhSegment*. Based on the results, if our model predicts baselines with a higher accuracy than *dhSegment* and other models, our proposed solution will have been a success. However, if our model combination accuracy metrics are the same, it will mean that our model combination approach will have been inconsequential to the line extraction process.

IV. Timeline

Winter 2021:

- Week 1 - 2: Use evaluation plan described above to test and analyze previous approaches based on the Print and Probability datasets. Come up with a concrete plan of how to implement our own model based on previous approaches.
- Week 3 - 4: Create Part 1 of model: Decide which previously used model worked the best and optimize it based on observations from Week 1 and 2.
- Week 5 - 7: Create Part 2 of model: Train a neural network to determine the probability of an image having correctly determined where the lines on a document are (without being dependent on an image).
- Week 8 - 10: Create Part 3 of model: Find and test an approximate argmax function that can approximate the optimal results from Part 1 and Part 2 of the the model.

Spring 2021:

- Week 1 - 3: Review Part 2 and Part 3 of the model and modify if there are any issues. Evaluate the overall model compared to the previously considered models and observe what works better and worse.
- Week 4: Take this week consider different types of approximate argmax algorithms and keep evaluate the overall model. Start working on poster and presentation.
- Week 5 - 6: Based on previous quarter observations, continue to optimize the model for high

text line extraction accuracy and implement bonding box strategies if possible. Continue working on poster and presentation.
- Week 7 - 10: Complete writing up work and presentation.

V. Changes Made Post Peer Review

- Formatted the page for 2 columns
- Added some elaborations to the context and motivations (such as more explanation of the technical terms and adding a broader motivation of why historical documents need to be transcribed)
- Put more emphasis on the fact we are looking for baselines instead of textboxes
- Made it more clear that we are building off of dhSegment as the first model (in problem statement and approach)
- Broke down the model figure into two separate figures to refer more to in the approach
- Added more detail in the approach about what X,Y, and the probabilities of each meant and more motivation to why they are important
- Created another figure to describe the OCR pipeline
- Elaborated on the evaluation plan to explain how we are measuring the accuracy from the distances between baseline predictions, and what exactly the intersection over union of pixels meant
- Added figures to further explain previous approaches in the context
- Changed the Timeline to allow more flexibility and leave space to expand the work, as well as more time to work on the presentation

References

[1] S. Ares Oliveira, B. Seguin, and F. Kaplan. dhsegment: A generic deep-learning approach for document segmentation. In *2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, pages 7–12, 2018.

[2] Taylor Berg-Kirkpatrick, Greg Durrett, and Dan Klein. Unsupervised transcription of historical documents. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 207–217, Sofia, Bulgaria, aug 2013. Association for Computational Linguistics.

[3] T. M. Breuel. Robust, simple page segmentation using hybrid convolutional mdlstm networks. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, volume 01, pages 733–740, 2017.

[4] Kai Chen, Mathias Seuret, Jean Hennebert, and Rolf Ingold. Convolutional neural networks for page segmentation of historical document images. In *Proceedings of the 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, pages 965–970, Kyoto, Japan, 2017.

[5] Tobias Gruning, Gundram Leifert, Tobias Strauß, and Roger Labahn. A robust and binarization-free approach for text line detection in historical documents. In *Proceedings of the 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, pages 236–241, Kyoto, Japan, 2017.

[6] M. Kassis and J. El-Sana. Learning free line detection in manuscripts using distance transform graph. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 222–227, 2019.

[7] O. Mechi, M. Mehri, R. Ingold, and N. Essoukri Ben Amara. Text line segmentation in historical document images using an adaptive u-net architecture. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 369–374, 2019.

[8] Joan Pastor-Pellicer, Muhammad Zeshan Afzal, Marcus Liwicki, and Maria Jose Castro-Bleda. Complete text line extraction with convolutional neural networks and watershed transform. In *IAPR International Workshop on Document Analysis Systems, DAS*, pages 30–35, Santorini, Greece, April 2016. IEEE.

[9] Raid Saabni. Robust and efficient text-line extraction by local minimal sub-seams. In *Proceedings of the 2nd International Symposium on Computer Science and Intelligent Control*, pages 1–6, New York, sept 2018. Association for Computing Machinery.