Re-imagining Chronicling America: Extracting and Classifying Visual Content from Historic Newspaper Scans

Benjamin Lee
Ph.D. Student, UW CSE;
Innovator-in-Residence,
Library of Congress
bcgl@cs.washington.edu

INTRODUCTION

Chronicling America is a partnership between the Library of Congress and the National Endowment for the Humanities to digitize historic newspapers published between 1789 and 1963 [3]. Over 15 million pages of historic American newspapers have been digitized to date, complete with high-resolution scans and machine-readable OCR[1] [3]. Indeed, *Chronicling America* is an invaluable resource for academic, local, and public historians; educators and students; genealogists; and members of public to explore American history. To help semantify the *Chronicling America* corpus, the Library of Congress Labs launched a crowdsourcing initiative called *Beyond Words* to identify photographs, illustrations, cartoons/comics, and maps in WWI-era newspapers in *Chronicling America*, as well as transcribe captions and record the content creators. Over 6,700 verified *Beyond Words* annotations have been collected to date.

This paper presents my work in constructing a deep learning pipeline for automating the extraction of photographs, illustrations, cartoons/comics, and maps from historic newspaper scans in *Chronicling America*. In particular, I leverage the *Beyond Words* data as training data for finetuning pre-trained Faster-RCNN implementations from Detectron2's Model Zoo [11, 17].

My **contributions** are as follows:

1. I present a new public dataset for extracting visual content from historic newspaper scans, which can be found here.

2. I contribute a finetuned Faster-RCNN model for this task, which achieves 57.4% bounding box average precision (AP).

3. I present an initial attempt at weakly supervising the extraction of captions for the identified content using the machine-readable OCR for the newspaper scans.

4. I present a demo for "A Day in Newspaper History," in which I extract visual content from every newspaper page from a specific day in history and cluster the photographs and illustrations using the embedding space of ResNet-18 with T-SNE for dimensionality reduction.

[1]The full specifications, such as scan resulution and OCR format (METS/ALTO) can be found here.

CODE

All code described in this paper was written in `Python` and can be found within the following public GitHub repository: `https://github.com/bcglee/newspaper-navigator`. For any questions, please email the author at bcgl@cs.washington.edu.

RELATED WORK

Other researchers have built pipelines for extracting visual content from historical documents. PageNet utilizes a Fully Convolutional Network for pixel-wise page boundary extraction for "historical handwritten documents" [14]. dhSegment is a deep learning framework for "historical document processing," including pixel-wise segmentation and extraction tasks [1]. The AIDA collaboration at the University of Nebraska-Lincoln have applied deep learning techniques to newspaper corpora including *Chronicling America* and the Burney Collection of British Newspapers [7, 9, 8]. These efforts have faced limitations with pixel-wise extraction approaches to training, and none have trained on bounding boxes for the extraction of visual content from historical documents.

Published datasets for historical document deep learning tasks include READ-BAD [2], SIAMESE [16], and DIVA-HisDB [13]. However, none of these datasets are designed specifically for the task of extracting visual content from newspaper scans.

METHODS

Fetching the Chronicling America Data

Chronicling America has a well-documented API for downloading high-resolution newspaper scans. To download scans from specific date ranges, I forked code written by the AIDA collaboration at University of Nebraska-Lincoln for downloading *Chronicling America* images, modifying the code

to downsample images using PIL and allow for more granular date specifications in my queries. My forked repository can be found here: https://github.com/ProjectAida/chronam-get-images.

Constructing the Beyond Words Dataset

To briefly describe the *Beyond Words* crowdsourcing workflow, volunteers are able to "discover pictures in the Library of Congress historical newspaper collections" by contributing their time to three different tasks:

1. *Mark*, in which users are asked to "draw a rectangle around each unmarked illustration or photograph excluding those in advertisements. Enclose any caption or text describing the picture and the illustrator or photographer" [4].
2. *Transcribe*, in which users are asked to correct the OCR of the caption for each marked illustration or photograph, transcribe the author's name, and note the category ("Editorial Cartoon," "Comics/Cartoon" "Illustration," "Photograph", or "Map") [5].
3. *Verify*, in which users are asked to select the transcription that most closely matches the caption. Users are also able to filter out bad regions or provide their own transcriptions in the event that neither transcription is of good quality [6].

Each annotation is considered verified once it has made it through these steps. At least 6 different individuals interact with each annotation (2 people at each step, with at least 51% agreement) during this process.

The verified *Beyond Words* annotations that I utilized as training data can be found here. To convert the JSON file available for download into a deep learning training set, I wrote a script to pull down the *Beyond Words* newspaper images and format the annotations according to the COCO dataset format, which is a standard data format for image recognition tasks.

The *Beyond Words* dataset in COCO format can be found here and is available under a CC0 license ("No Rights Reserved") in the public domain. The dataset contains 3,437 newspaper scans with 6,732 verified annotations. The breakdown of annotations is as follows:

- Photographs: 4,193
- Illustrations: 1,028
- Maps: 79
- Comics/Cartoons: 1,139
- Editorial Cartoons: 293

Training the Image Extraction Model

As described in Related Work (Section 3), researchers have encountered limitations when training deep learning models such as dhSegment using pixel masks for this task, as well as similar ones. For the task articulated in this paper, I decided instead to train on bounding boxes, which agrees with the nature of the *Beyond Words* workflow, in which volunteers were asked to draw bounding boxes over pictures/illustrations and their captions, rather than segment out the photographs/illustrations at the pixel level (or segment the contents of the photographs/illustrations). Daniel Gordon's advice to consider bounding box training and to take a look at FAIR's newly-released Detectron2 research platform for object detection, which has a number of pre-trained Faster-RCNN implementations in its Model Zoo, proved to be ideal for training [17].

RESULTS & EXPERIMENTS

Model Selection

All of my deep learning training was done in a Google Colab notebook, which can be found in my repository. All performance metrics reported are using a single NVIDIA Tesla K80 GPU provided by Colab.

I performed model selection over two different pre-trained Faster-RCNN models (more details on the pre-trained models can be found on the Model Zoo page:

1. `faster_rcnn_R_50_FPN_3x`, the *fastest* of the pre-trained Faster-RCNN implementations according to inference time
2. `faster_rcnn_X_101_32x8d_FPN_3x`, the *most accurate* of the pre-trained Faster-RCNN implementations according to box average precision

In particular, I wanted to assess the tradeoff between accuracy and inference speed; the Resnet 50 implementation requires ≈ 0.1 seconds per image, whereas the X101 implementation requires ≈ 0.25 seconds per image, so speed is a necessary consideration when building a pipeline over 15+ million images.

In Figure 1, I present the validation accuracy (box average precision) for both pre-trained models as a function of iteration, using an 80%-20% training-validation split. In both cases, I set `BASE_LR` $= 0.00025$ and `BATCH_SIZE` $= 64$, with data augmentation of `RESIZE_SHORTEST_EDGE` and `RANDOM_FLIP`.[2] `faster_rcnn_X_101_32x8d_FPN_3x` achieved the higher validation AP of 57.4%, but my guess is that this result is not statistically significant; I would need to perform cross-validation to test this properly. It is also worth noting that the training/validation data is noisy, in the sense that even validated annotations are susceptible to human error; for example, in some cases, advertisements are improperly marked. Training `faster_rcnn_R_50_FPN_3x` for 390 epochs (30,000 epochs) required just over an hour on the single Colab GPU. Training `faster_rcnn_X_101_32x8d_FPN_3x` required approximately seven and a half hours.

In Table 1, I present a summary of the two models. The primary takeaway is that using `faster_rcnn_R_50_FPN_3x` in a pipeline that runs at scale on all 15+ million *Chronicling America* images is preferable because average precision is only marginally worse, but the inferene speed is 2.5 times faster than for `faster_rcnn_X_101_32x8d_FPN_3x`. Interestingly, both models had the lowest average precision with the

[2] In particular, I set `ResizeShortestEdge(short_edge_length=(640, 672, 704, 736, 768, 800), max_size=1333, sample_style='choice')` & `RandomFlip()`. It is worth noting that these are the only two data augmentation methods currently supported by `Detectron2`.

"illustrations" category, presumably due to the lower number of examples relative to the complexity of differentiating illustrations from photographs in historic newspapers (which is often difficult because the scans are produced from microfilm, which saturates photographs).

In Figures 2, 3, and 4, I show some predictions on the validation set. In the captions I elaborate on performance; overall, the results are quite promising.

Weakly Supervising OCR Extraction

In addition to providing high-resolution scans of newspaper pages, *Chronicling America* specifications require that each newspaper page be submitted with machine-readable OCR of the text in METS/ALTO format. This format includes localized bounding boxes for each string of characters recognized by OCR (separated by whitespace). Thus, the coordinates of a predicted bounding box for visual content can be leveraged to extract corresponding textual content by finding all text that appears within the bounding box (the *Beyond Words* workflow already utilizes this to provide OCR for volunteers to correct in the second step, *Transcribe*). Because the *Beyond Words* volunteers were told to include captions within bounding boxes during the *Mark* step, the image extraction model has learned to include captions within the bounding boxes for visual content. Even though this process is noisy, captions can oftentimes be extracted via this simple method of extracting textual content. In my repository, I have included a python script for extracting this textual content from the OCR as additional metadata for the predictions. Though quite simple, this method provides a baseline for extracting captions and other textual content.

Visualizing "A Day in Newspaper History"

With this pipeline for extracting visual content from historic newspaper scans, we can begin to ask a wide ranges of questions about the corpus. For example: what does all of the visual content across newspapers look like on a specific day of history?

To answer this question, I used img2vec to generate ResNet18 embeddings for extracted visual content [12]. These 512-dimensional embeddings are taken from the penultimate layer. To visualize these embeddings, I used T-SNE, a dimensionality reduction algorithm commonly used for visualizing high-dimensional spaces [15]. In particular, `Scikit-Learn`'s implementation is straightforward to use.

In Figure 5, I present a visualization of the extracted visual content for June 7th, 1944 (the day after D-Day). It is exciting to see that images are clustered not only by style but also by semantic content; for example, in Figure 6, I show a cluster that contains headshots.

CONCLUSION

In this paper, I have presented a pipeline for extracting, categorizing, and captioning visual content in historic newspaper scans. I have introduced a new dataset for this task using annotations from *Beyond Words*, the Library of Congress Labs's crowdsourcing initiative for annotating and captioning visual content in WW1-era newspaper scans in *Chronicling America*; the dataset is available publicly at the GitHub repository for this project. I have described my experiments with finetuning different pre-trained Faster-RCNN implementations in Detectron2's Model Zoo on the dataset that I have released. I have also outlined my code for extracting captions for identified visual content by leveraging the machine-readable OCR for each newspaper scan. Lastly, I have presented my visualization of "A Day in Newspaper History" that enables users to explore all of the visual content from newspapers in *Chronicling America* published on the same day in history.

FUTURE WORK

One promising avenue for exploration is the unverified *Beyond Words* annotations. Because the pipeline requires at least six individuals to touch each annotation before it is considered verified, there is a backlog of about 50,000-100,000 annotations that have not been fully verified. Because the drawing of bounding boxes around visual content is the first step in the pipeline, it is reasonable to expect that the overwhelming majority of unverified annotations have high quality bounding boxes. Thus, an immediate extension of the work presented in this paper is to train on these unverified annotations and assess how performance changes. In particular, I expect performance to improve. Also, just making these unverified annotations available as a dataset for public use is a worthwhile endeavor.

In regard to the extracted textual content from the OCR, one could imagine training an NLP pipeline to correct bad OCR using the caption annotations. In particular, in the second step of the *Beyond Words* pipeline, volunteers were asked to correct the OCR that appears over each marked bounding box, resulting in over 6,700 corrected textual annotations. It is straightforward to construct training pairs of input and output to train a supervised model to correct OCR. Weak supervision approaches such as Snorkel could also be explored for this OCR correction [10].

In regard to my demo of visualizing a day in history, there are a number of promising areas to explore. First, I would like to experiment with different embedding spaces, as different models trained on different tasks will result in markedly different embeddings. Second, I would like to improve the visualization by making it interactive with its own UI; I would also like to make small changes such as fixing the aspect ratios of images in the visualization.

ACKNOWLEDGMENTS

BCGL would like to thank Dan Weld and Daniel Gordon at the University of Washington; Jaime Mears, Eileen Jakeway, Meghan Ferriter, Laurie Allen, Deb Thomas, Nathan Yarasavage, Chris Adams, Tong Wang, and the entire NDNP staff at the Library of Congress; and Elliott Wrenn at the United States Holocaust Memorial Museum for their invaluable advice with this project.

REFERENCES

[1] Sofia Ares Oliveira, Benoit Seguin, and Frederic Kaplan. 2018. dhSegment: A generic deep-learning approach for document segmentation. In *Frontiers in Handwriting Recognition (ICFHR), 2018 16th International Conference on*. IEEE, 7–12.

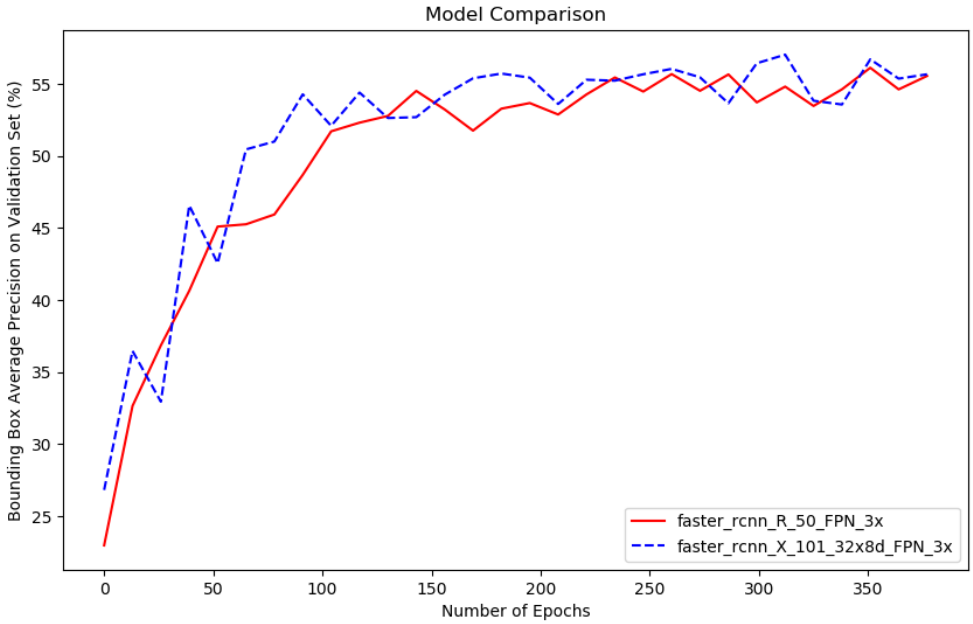

Figure 1. A plot comparing the bounding box average precision for the two models as function of epoch.

Model Comparison		
Model Name	Bounding Box Average Precision (Validation)	Inference Time (s / img)
`faster_rcnn_R_50_FPN_3x`	56.1%	0.1 s / img
`faster_rcnn_X_101_32x8d_FPN_3x`	57.4%	0.25 s / img

Table 1. A table showing the tradeoffs between performance and inference speed for the two pre-trained Model Zoo models that I finetuned and tested.

Figure 2. An example of predicted visual content on an example page from the validation set. We observe that the finetuned model is able to disambiguate constituent parts and classify them. There are a couple of segmentation and classification errors, but given how complex this page is, this is quite promising.

Figure 3. Another example of predicted visual content. Note how the model learns to include title and captions within the bounding boxes. Note also how advertisements are filtered out automatically.

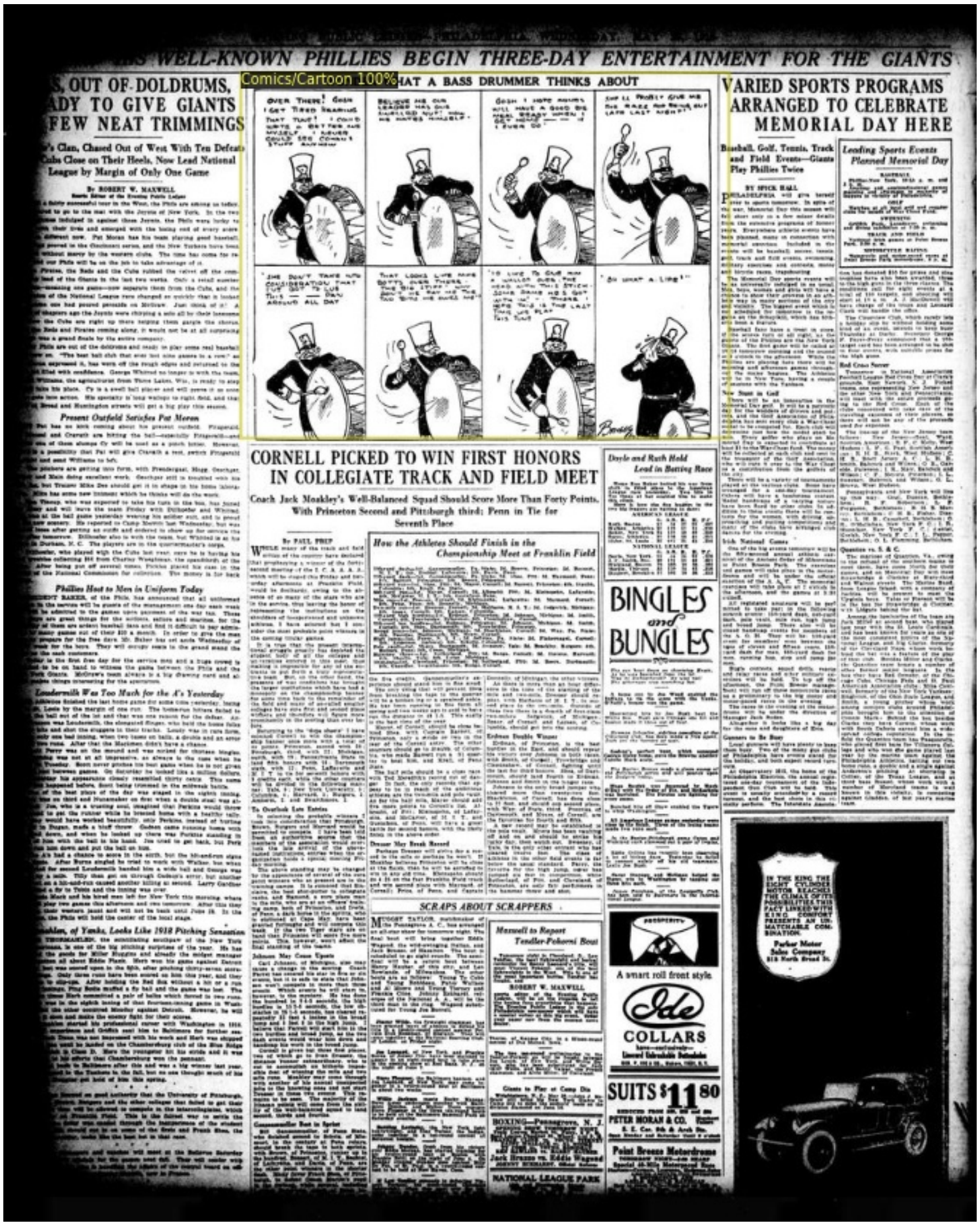

Figure 4. Another example of predicted visual content, this time showing a comic strip that is properly segmented out, despite the eight cells within the strip.

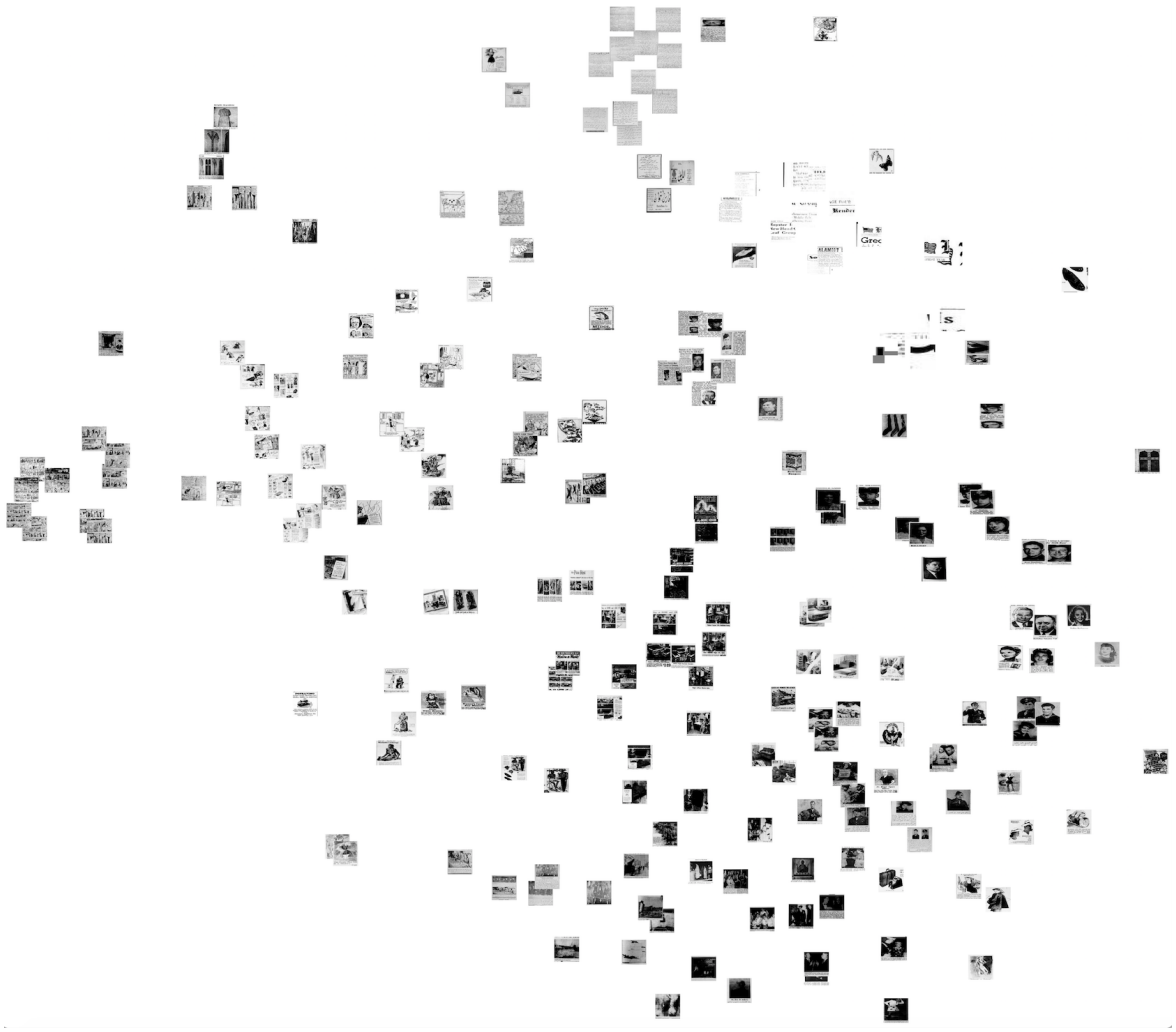

Figure 5. A visualization of extracted content from newspapers on June 7th, 1944, created by running my pipeline for visual content extraction. This was created by generating embeddings for the extracted content using the penultimate layer of ResNet-18 and running T-SNE for dimensionality reduction.

Figure 6. A zoomed-in section of the visualization in Figure 5. Note that semantic features are clustered together: here, for example, we see headshots.

[2] T. Grüning, R. Labahn, M. Diem, F. Kleber, and S. Fiel. 2018. READ-BAD: A New Dataset and Evaluation Scheme for Baseline Detection in Archival Documents. In *2018 13th IAPR International Workshop on Document Analysis Systems (DAS)*. 351–356. DOI: http://dx.doi.org/10.1109/DAS.2018.38

[3] National Endowment for the Humanities. 2019. Chronicling America | Library of Congress. (2019). https://chroniclingamerica.loc.gov/about/api/

[4] LC Labs. 2017a. Beyond Words ("Mark"). (2017). http://beyondwords.labs.loc.gov/#/mark

[5] LC Labs. 2017b. Beyond Words ("Transcribe"). (2017). http://beyondwords.labs.loc.gov/#/transcribe

[6] LC Labs. 2017c. Beyond Words ("Verify"). (2017). http://beyondwords.labs.loc.gov/#/verify

[7] Elizabeth Lorang. 2018. Patterns, Collaboration, Practice: Algorithms as Editing for Historic Periodicals. (2018), 14.

[8] Elizabeth Lorang and Leen-Kiat Soh. 2019a. Application of the Image Analysis for Archival Discovery Team's First- Generation Methods and Software to the Burney Collection of British Newspapers. (2019), 21.

[9] Elizabeth Lorang and Leen-Kiat Soh. 2019b. Using Chronicling America's Images to Explore Digitized Historic Newspapers & Imagine Alternative Futures. (2019), 19.

[10] Alexander Ratner, Stephen H. Bach, Henry Ehrenberg, Jason Fries, Sen Wu, and Christopher Ré. 2019. Snorkel: rapid training data creation with weak supervision. *The VLDB Journal* (July 2019). DOI: http://dx.doi.org/10.1007/s00778-019-00552-1

[11] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In *Advances in Neural Information Processing Systems 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett (Eds.). Curran Associates, Inc., 91–99. http://papers.nips.cc/paper/5638-faster-r-cnn-towards-real-time-object-detection-with-regio pdf

[12] Christian Safka. 2019. christiansafka/img2vec. (Nov. 2019). https://github.com/christiansafka/img2vec original-date: 2017-09-21T08:19:17Z.

[13] Foteini Simistira, Mathias Seuret, Nicole Eichenberger, Angelika Garz, Marcus Liwicki, and Rolf Ingold. 2016. DIVA-HisDB: A Precisely Annotated Large Dataset of Challenging Medieval Manuscripts. In *2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR)*. IEEE, Shenzhen, China, 471–476. DOI:http://dx.doi.org/10.1109/ICFHR.2016.0093

[14] Chris Tensmeyer, Brian Davis, Curtis Wigington, Iain Lee, and Bill Barrett. 2017. PageNet: Page Boundary Extraction in Historical Handwritten Documents. In *Proceedings of the 4th International Workshop on Historical Document Imaging and Processing (HIP2017)*. ACM, New York, NY, USA, 59–64. DOI: http://dx.doi.org/10.1145/3151509.3151522

[15] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing Data using t-SNE. *Journal of Machine Learning Research* 9 (2008), 2579–2605. http://www.jmlr.org/papers/v9/vandermaaten08a.html

[16] M. Wevers and J. Lonij. 2017. SIAMESET. http://lab.kb.nl/dataset/siameset

[17] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. 2019. Detectron2. https://github.com/facebookresearch/detectron2. (2019).