Detection of Page Objects from the HathiTrust Digital Library

Arpandeep Khatua, Mentors: Glen Worthey, Boris Capitanu

Department of Electrical and Computer Engineering, Grainger College of Engineering, University of Illinois at Urbana-Champaign

HATHI TRUST

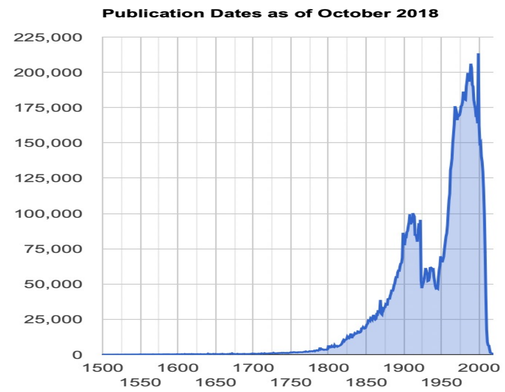
ILLINOIS

NCSA | National Center for Supercomputing Applications

INTRODUCTION

The HathiTrust Digital Library has over 16 million volumes which is about 5 billion pages. These pages are scanned using OCR and stored in the database, however due to the presence of non-textual elements on the pages the documents are largely unlabeled. There 2 purposes of our project:

1. Detect the various non textual elements on each page using computer vision models.
2. Create a mask and tag each page which can be stored as meta data for each page or document thus improving the search metrics.



Plot 1: Number of publications in HathiTrust Library per year.

MODEL

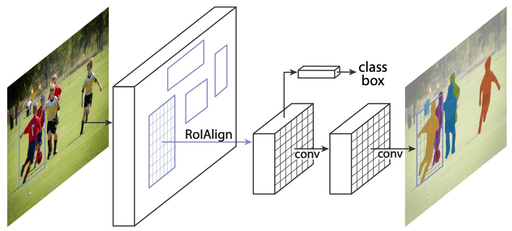


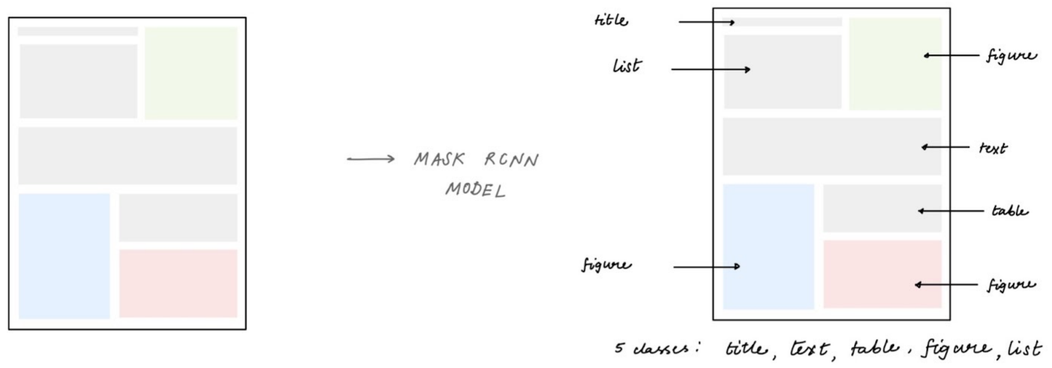
Figure 1. The **Mask R-CNN** framework for instance segmentation.

Detectron2 is a ground-up rewrite of Detectron that started with maskrcnn-benchmark. The platform is now implemented in PyTorch. With a new, more modular design, Detectron2 is flexible and extensible, and able to provide fast training on single or multiple GPU servers. Detectron2 includes high-quality implementations of state-of-the-art object detection algorithms, including DensePose, panoptic feature pyramid networks, and numerous variants of the pioneering Mask R-CNN model family also developed by FAIR.

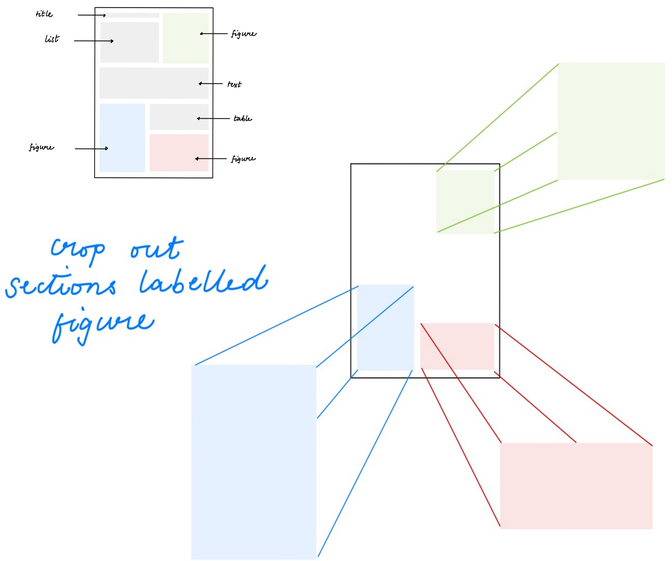
Detectron2

PROCEDURE

The Detectron2 model is trained for panoptic segmentation using the Mask R CNN model baseline. We then used the PubMed dataset to re-train this model to detect 5 different classes of objects on a page as listed below. This was done using an open-sourced implementation to get the trained model.



1. We take the image of a page and use it as an input to the re-trained modified detectron2 model. This model gives us a list of (x, y) coordinates for the masks and labels each class on the page.
2. Extract the boxes labeled as figures and pass it through a binary classifier to classify whether it is a music score or not.
3. If not music score, we use a large pretrained NASNET model to further classify each figure into the ImageNet classes.
4. However, to reduce the fine classification of 1000 classes, we use a K-Means algorithm to cluster the ImageNet classes into 12 coarse classes.



{insect, animal, plant, person, clothes, vehicle, household objects, sport, technical, construction, food, instrument}

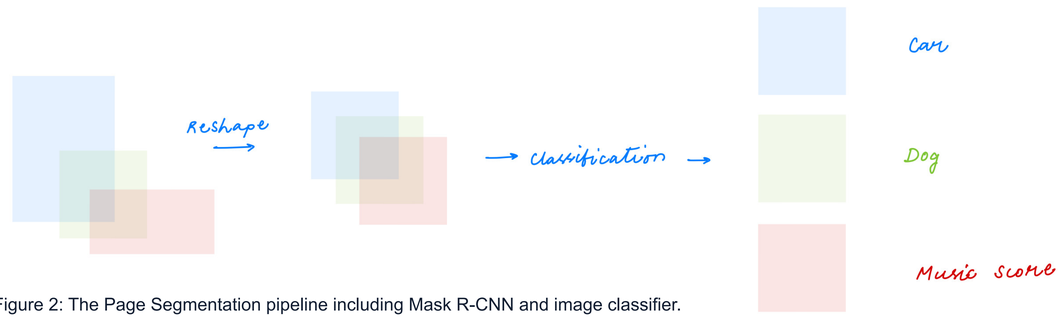


Figure 2: The Page Segmentation pipeline including Mask R-CNN and image classifier.

RESULTS

Figure 3: Results using random test samples. (HathiTrust data classified)

CONCLUSIONS

We currently are running this pipeline to process HathiTrust page data using the Carbonate system at Indiana University. After preprocessing the input and classifying and detecting the region of 18 different classes, we plan to add these parameters to the search algorithm to provide users with more finer search parameters. We are working on improving the run time which is as 2 pages per second using 1 V100 NVIDIA GPU.

ACKNOWLEDGEMENTS

I would like to thank HathiTrust for allowing me to work on this project and use their data. I would like to thank IU for providing compute resources to run my model and a big thanks to Boris and Glen for guiding and helping me at every point of this project.

ILLINOIS