

# Visual Detection with Context for Document Layout Analysis

Carlos X. Soto

Brookhaven National Laboratory  
Upton, New York, USA  
csoto@bnl.gov

Shinjae Yoo

Brookhaven National Laboratory  
Upton, New York, USA  
sjyoo@bnl.gov

## Abstract

We present 1) a work in progress method to visually segment key regions of scientific articles using an object detection technique augmented with contextual features, and 2) a novel dataset of region-labeled articles. A continuing challenge in scientific literature mining is the difficulty of consistently extracting high-quality text from formatted PDFs. To address this, we adapt the object-detection technique Faster R-CNN for document layout detection, incorporating contextual information that leverages the inherently localized nature of article contents to improve the region detection performance. Due to the limited availability of high-quality region-labels for scientific articles, we also contribute a novel dataset of region annotations, the first version of which covers 9 region classes and 822 article pages. Initial experimental results demonstrate a 23.9% absolute improvement in mean average precision over the baseline model by incorporating contextual features, and a processing speed 14x faster than a text-based technique. Ongoing work on further improvements is also discussed.

## 1 Introduction

Mining scientific literature at scale for information that can be processed automatically is a valuable and much sought after technique for domain researchers and data scientists. Whereas mass processing of articles was once largely limited to keyword searches and citation crawling, modern natural language processing techniques can now deeply search for specific and broad concepts, explore relationships, and automatically extract useful information from text.

However, getting the article text in the first place is not always simple. Though some publications offer full-text articles in open-access, machine-readable formats (e.g. plain text, HTML or XML),

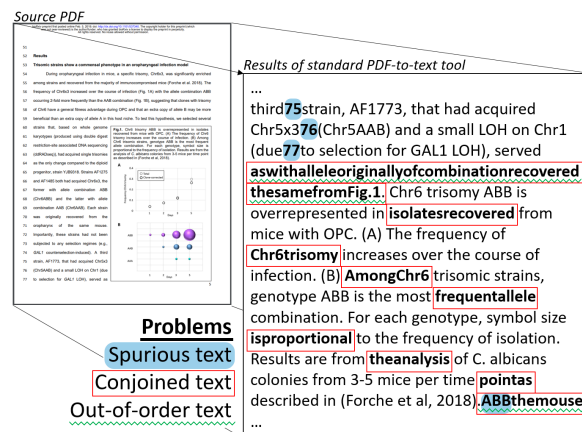


Figure 1: Text extraction from PDF articles is generally a noisy process because of the significant inconsistencies in PDF publishing tools and the edit-and-republish nature of print-ready document preparation by journal publishers and document hosting services.

this is far from the norm. Instead, much of the scientific record remains in PDF files that are much more challenging to automatically process. This challenge remains whether the PDFs originate from scanned pages (thus requiring OCR) or were published from digital source material. Although there are tools for automatically extracting text from PDFs, formatting is highly inconsistent. Standard tools often mix headers, footers, table and figure captions, page numbers, and other extraneous text into the main text being extracted (Bast and Korzen, 2017). And text order is not always well preserved (Figure 1). This is not quite as problematic if text is extracted for simple tasks like keyword searches. However, more complex tasks like Named-Entity Recognition often rely on contiguous, cleanly segmented text for successful processing. And while manually cleaning and reformatting text may be an option for small sets of documents, this quickly becomes impractical when dealing with larger corpora.

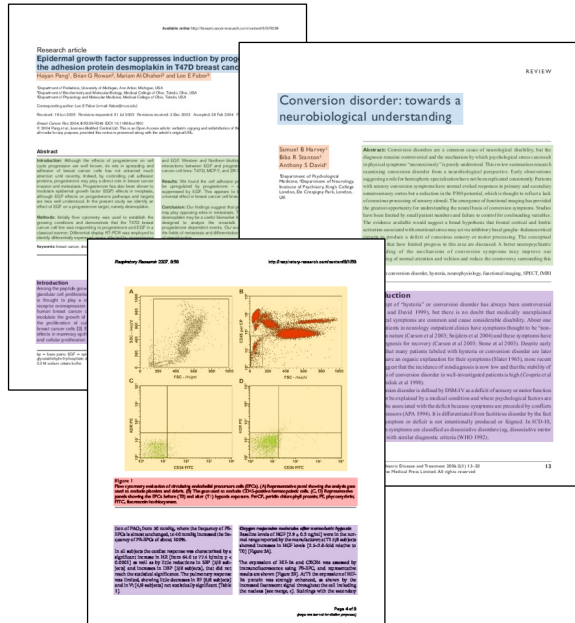


Figure 2: Examples of ground-truth region labels, showing varying journal styles. Note that even when text is illegibly small, the salient regions are visually apparent (e.g. title, abstract, figure caption, etc.).

To facilitate large-scale automatic knowledge extraction from scientific articles, this paper presents an adaptation of the Faster R-CNN object detection model for the task of document layout detection, with the addition of contextual features. The method **visually** detects major regions in article pages and classifies them into a set of labels, including main-body text, tables, figures, and others (Figure 2). To train and evaluate the detection models, a novel dataset of 100 region-annotated scientific articles was created, totaling 822 labeled pages. This work is an ongoing effort: the dataset is being expanded, additional contextual features are being developed, and further evaluation is being conducted.

## 2 Related Work

In addition to the numerous tools that exist for extracting text from PDFs – [Bast and Korzen \(2017\)](#) provide a quantitative evaluation of 14 common ones – there are a variety of approaches for analyzing documents to determine their contents’ layout and/or extract information of particular types.

Systems like CERMINÉ ([Tkaczyk et al., 2015](#)) and OCR++ ([Singh et al., 2016](#)) extract the raw or processed markup from “born-digital” PDFs (e.g. using tools like pdf2xml) and apply a variety of text processing methods to deduce docu-

ment structure and apply semantic labels to text blocks. These methods may be rule-based (e.g. regular expressions and heuristics) or machine-learning based (e.g. SVM classification). Such approaches may achieve high-quality extraction and labeling. However they rely on extracted PDF source markup (not always available, e.g. for scanned PDFs), work only on text blocks (ignoring tables and figures, which may also be valuable), and are typically quite slow (2 to 10 seconds or more per article). Analogous approaches for web pages – such as those by [Cai et al. \(2003\)](#) and [Song et al. \(2004\)](#) – also examine the document source (in this case the page DOM) to build representations for further processing.

Alternatively, visual-based techniques for document layout analysis have tended to focus on text segmentation ([Tran et al., 2015](#)), ([Chen et al., 2015a](#)), ([Chen et al., 2015b](#)), ([Garz et al., 2016](#)), especially for historical documents; [Eskenazi et al. \(2017\)](#) survey dozens of such methods. More closely aligned with the aim of this work are those by [Lang et al. \(2018\)](#), who use HOG features and random forests to recognize text, images, charts and tables in newspapers, [Oliveira and Viana \(2017\)](#), who use 1D CNNs to recognize text, images, and tables in scientific articles, and finally [Singh et al. \(2018\)](#), who use a LSTD model to recognize various customizable text and image classes in multiple domains, with mixed results.

Differently from these previous methods, the approach in this paper aims to leverage contextual information in a modern visual object detection approach. Of particular importance to this effort is using a high-quality dataset with labels for relevant key regions of scientific articles. Though some of the cited works in this section published their own custom datasets, these proved to be too granular, too coarse or too noisy for use in this work (see Section 4).

## 3 Visual Document Layout Detection

A modern visual approach is taken to document region detection, with the aim to produce labeled bounding boxes for regions of interest (ROIs) in each document page. Although the text contents in many regions may be useful in classification, our vision-only approach has the benefit of working in any language, and can yield impressive performance even without text features (see Section 5). To be clear: our approach does not access a

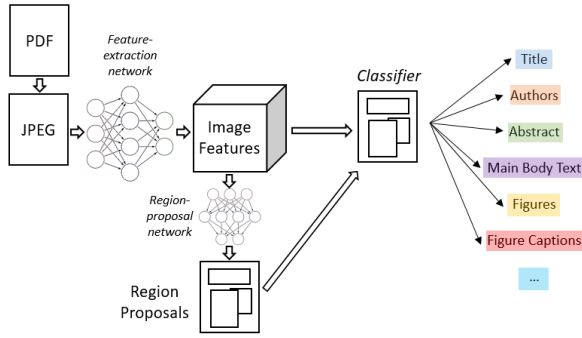


Figure 3: Model for visual region detection using Faster R-CNN. After rendering each page of a PDF to a JPEG and extracting its image features using a convolutional ResNet-101 network, regions of interest are proposed and classified by two additional network which are trained on ground-truth labels.

document’s source at all (neither embedded text, nor underlying PDF markup), instead relying only on JPEG renderings of the document’s pages.

The baseline technique used for this work is the highly successful object detector Faster R-CNN (Ren et al., 2015) – a so-called two stage detection model. This model takes an image as input and first predicts candidate regions of interest (i.e. image regions that are likely to contain objects), then classifies each candidate region into one of a pre-defined set of object classes. The output is a set of bounding boxes identifying recognized objects in the image (e.g. people, cars, dogs, etc.). Algorithmically, Faster R-CNN relies on several component neural networks, beginning with a deep convolutional neural network – ResNet-101 (He et al., 2016), specifically – to perform feature extraction on the input image. The resulting feature map is then fed into a region-proposal network (which generates candidate object bounding boxes) and a classification network (which assigns region labels to those candidates). A regressor network also fine-tunes the ROI bounding box parameters for better fit. Figure 3 shows how this model can be applied directly to the task of detecting salient regions in images of document pages.

An important aspect of standard object detection techniques like Faster R-CNN, is that they only use image features within a region of interest to label that region. This approach is advantageous to general purpose object detection, because objects may be located anywhere in a photo or video., and has been highly successful even when objects are cropped or occluded by other objects (Lin et al., 2014). However, in domains such

as document layout detection, where the ‘objects’ to be detected are generally well-structured elements of the page (and indeed of a collected set of pages), it is helpful to include contextual information to determine the label of a region. To this end, our approach incorporates contextual information about pages and ROI bounding boxes, encoded as additional features for the classification and fine-tuning regression stages of the model.

The position, size, and page number of a candidate ROI in a document are valuable contextual clues to the true label of that region. They help disambiguate region classes that are otherwise visually similar, such as abstracts, body text, and figure captions (without context, these all appear simply as isolated blocks of text). Further, if two proposed regions overlap, their size and position can help determine which is more likely to fit a true document region more closely. Additional contextual features are being explored in ongoing work (see Section 6), however these were selected first for their ease of integration and anticipated high impact on detection performance.

Another important consideration is the existence of newer, even more successful detection models like YOLOv3 (Redmon and Farhadi, 2018) and RetinaNet (Lin et al., 2017), which use a single-stage detection paradigm. Our rationale in selecting Faster R-CNN as a baseline model to build upon in this work is that the separate region proposal and classification models make it easier to intercept ROI dimensions from the first model and embed them as additional features (along with page numbers) for input to the classification and bounding-box regression models. We do, however, compare our results to single-stage models and intend to explore these further.

## 4 Novel Labeled Dataset

Existing region-labeled datasets of scientific articles proved too noisy for our visual region detection method. Training the model with the full GROTOAP2 dataset (Tkaczyk et al., 2014), for example, yielded a best overall detection performance of 5.1% mean average precision (mAP) over all 22 labels. The key problems were that many of the regions were far too granular (e.g. each cell of a table might have its own bounding box labeled ‘table’), and many regions were mislabeled or misaligned. The labeling noise was so pronounced that by simply filtering and carefully

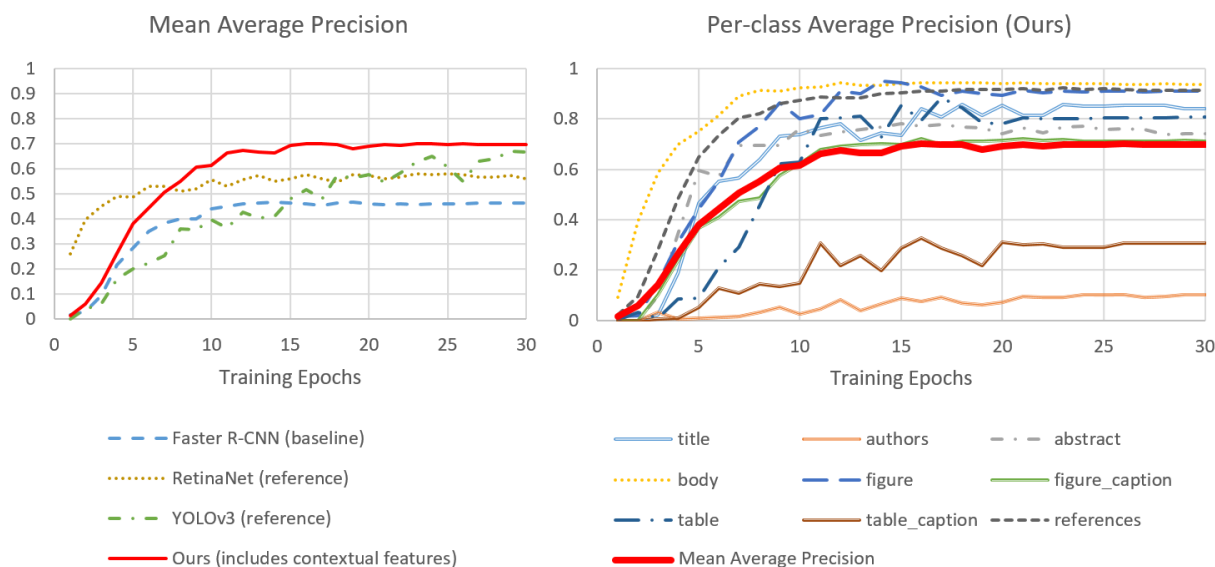


Figure 4: Incorporating page and bounding box context yielded a relative improvement of 51.6% over baseline Faster R-CNN performance (23.9% absolute: 70.3% vs 46.4%). Our model also outperforms reference state-of-the-art single stage models: RetinaNet (58.1% peak mAP) and YOLOv3 (68.9%, after 49 epochs). Small regions (authors, table captions) remain challenging and bring down average performance. All results @ 0.5 IOU.

coalescing bounding boxes for ‘body content’ labels, single-class detection performance was dramatically improved to 72% AP (up from 18% for unfiltered labels). However, the label-cleaning process was very time consuming and this improvement did not generalize to other classes.

Therefore, a novel dataset was created. We rendered PDF articles to JPEG image sets (using the ImageMagick package, at 72dpi), and used an open source utility (Tzutalin, 2015) to manually annotate regions. Version 1 of this dataset consists of region annotations for 100 scientific articles sampled from the PMC Open Access set. The collection will be available at <https://github.com/cxsoto/article-regions>, and includes scripts to download and render the original article PDFs to images, as well as to convert the annotations to various formats. The default format is PASCAL VOC. Nine labeled region classes are included in the annotations:

- *Title*. Includes subtitle, if present.
- *Authors*. Author names only, where possible (i.e. no affiliations, etc.).
- *Abstract*. Abstract text only, where possible.
- *Body*. All main article text, including section headers. Contiguous, where possible.
- *Figure*. Any labeled figures (i.e. no journal logos, etc.).
- *Figure Caption*. The caption text for a figure.

- *Table*. Including only the tabular contents, where possible. Includes adjacent notes or commentary if short and table-aligned.
- *Table Caption*. Main table caption as well as paragraph-form table commentary that follows some tables.
- *References*. Full bibliography, not including post-references notes (e.g. author bios, journal marketing, etc.).

The bounding boxes for these regions were created to be consistent and tight-fitting, with typical padding of 2-4 pixels. In general, the bounding boxes do not overlap. Each page of each article is labeled and processed separately. Due to labeling precision, annotation took approximately 20 to 60 seconds per image, depending on the contents. However, as this process does not involve specialized knowledge, it can be readily crowd-sourced.

For version 2 of the dataset, we intend to include labels for Equations, Sub-Figures, and Author Affiliations. It is expected to cover approximately 1000 articles from more varied sources, including pre-prints from arXiv and similar repositories.

## 5 Implementation and Experiments

Using the novel labeled dataset described in Section 4, a baseline model was trained using a standard Faster R-CNN implementation (Yang et al., 2017). The model was trained using a single



NVIDIA P100 GPU for 30 epochs on 600 images, and tested on the remaining 222 in 5 randomized sessions, using a ResNet-101 base network pre-trained on ImageNet (Russakovsky et al., 2015), with a batch size of 8, Adam optimizer (Kingma and Ba, 2014), and a starting learning rate of 0.0001, with decay of 0.1 every 5 epochs. Standard anchor scales of [8, 16, 32] and anchor ratios of [0.5, 1.0, 2.0] were used. At a intersection-over-union (IOU) threshold of 0.5, the model achieved a mean average precision (mAP) of 46.38% on all nine region labels, with peak class performance on ‘body’ regions (87.49%) and lowest performance on ‘authors’ (1.22%).

Contextual information was incorporated into the classification and regression stages of the model for article page information and proposed ROI bounding boxes. Page context consisted of the page number of the current image in its article, and the number of pages in the article, both normalized to the average page length of articles in the dataset (8.22). Bounding box context consisted of the position and size of the proposed region of interest, normalized to the dimensions of the image. Both were appended to the pooled feature vectors for batched ROIs sent to the classification and bounding-box regression models. The input size of these models were increased accordingly. Retraining the model in the same manner as the baseline yielded a mean average precision of 70.3%, with peak class-performance of 93.58% on ‘body’ regions and a low performance of 10.34% on ‘authors’. The second-lowest performance was on ‘table captions’ (30.8%).

Figure 4 shows per-class performance over 30 training epochs, as well as comparative performance against the baseline Faster R-CNN model and reference model implementations of YOLOv3 (Linder-Noren, 2018) and RetinaNet (Henon, 2018). Most models plateaued early on this small dataset, except YOLOv3 which peaking at 68.9% after 49 epochs (beyond the figure bounds, but still below our model’s results). Processing time for our model averaged 0.65 seconds per article. By contrast, CERMINE averaged 9.4 seconds per article on the same set of articles, on the same machine.

With the exception of two small region classes (authors and table captions), detection performance quickly reaches a rather high plateau (83.63% not including these two classes), espe-

cially considering the model has been trained on less than 100 labeled articles. The size and placement of these region classes makes them difficult to localize and distinguish from other classes. The ongoing work described in Section 6 focuses on incorporating additional contextual features that are expected to improve overall performance, and especially target problematic classes like these.

## 6 Discussion and Ongoing Work

Visual layout detection in documents benefits directly from advances in state-of-the-art object detection techniques, yet is also well suited for optimizations that exploit the structured nature of documents. Adding even very simple contextual information (page numbers and region-of-interest position and size) yielded a relative performance improvement over the baseline of over 50% – while also outperforming state-of-the-art single stage detectors. Unsurprisingly, the position of a region in a scientific article is strongly correlated with its label. Similarly, other contextual information may be correlated with a region’s label and placement.

In ongoing work, additional contextual features are being explored, including ROI oversampling (i.e. looking at a region’s surroundings), random feature map sampling (i.e. looking at patterns in a whole page or whole article), and all-region positional information (i.e. the position of other predicted ROIs in an article). Performance improvements are expected as these features are incorporated, and as additional labeled data is created. Alternative detection frameworks are also being explored.

The value of this work extends beyond text extraction. Visual region detection can serve as a precursor to numerous existing information extraction techniques or adaptations of them, including parsing of reference (Lammey, 2015), tables (Rastan et al., 2015), and equations (Smithies et al., 2001), as well as data extraction from figures (Tummers, 2006). And so it is valuable to all these efforts to be able to accurately and quickly segment document pages into regions of interest.

## Acknowledgments

This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Biological and Environmental Research, DE-SC0012704 and BNL LDRD #17-018.

## References

- Hannah Bast and Claudius Korzen. 2017. A benchmark and evaluation for text extraction from pdf. In *Proceedings of the 17th ACM/IEEE Joint Conference on Digital Libraries*, pages 99–108. IEEE Press.
- Deng Cai, Shipeng Yu, Ji-Rong Wen, and Wei-Ying Ma. 2003. Vips: a vision-based page segmentation algorithm. *Microsoft Technical Report, MSR-TR-2003-79*.
- Kai Chen, Mathias Seuret, Marcus Liwicki, Jean Hennebert, and Rolf Ingold. 2015a. Page segmentation of historical document images with convolutional autoencoders. In *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, pages 1011–1015. IEEE.
- Kai Chen, Mathias Seuret, Hao Wei, Marcus Liwicki, Jean Hennebert, and Rolf Ingold. 2015b. Ground truth model, tool, and dataset for layout analysis of historical documents. In *Document Recognition and Retrieval XXII*, volume 9402, page 940204. International Society for Optics and Photonics.
- Sébastien Eskenazi, Petra Gomez-Krämer, and Jean-Marc Ogier. 2017. A comprehensive survey of mostly textual document segmentation algorithms since 2008. *Pattern Recognition*, 64:1–14.
- Angelika Garz, Mathias Seuret, Fotini Simistira, Andreas Fischer, and Rolf Ingold. 2016. Creating ground truth for historical manuscripts with document graphs and scribbling interaction. In *2016 12th IAPR Workshop on Document Analysis Systems (DAS)*, pages 126–131. IEEE.
- Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.
- Yann Henon. 2018. Pytorch implementation of retinanet object detection. <https://github.com/yhenon/pytorch-retinanet>.
- Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Rachael Lammey. 2015. Crossref text and data mining services. *Science Editing*, 2(1):22–27.
- Thomas Lang, Markus Diem, and Florian Kleber. 2018. Physical layout analysis of partly annotated newspaper images. In *Proceedings of the 23rd Computer Vision Winter Workshop*.
- Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988.
- Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer.
- Erik Linder-Noren. 2018. A minimal pytorch implementation of yolov3. <https://github.com/eriklindernoren/PyTorch-YOLOv3>.
- Dario Augusto Borges Oliveira and Matheus Palhares Viana. 2017. Fast cnn-based document layout analysis. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1173–1180.
- Roya Rastan, Hye-Young Paik, and John Shepherd. 2015. Texus: a task-based approach for table extraction and understanding. In *Proceedings of the 2015 ACM Symposium on Document Engineering*, pages 25–34. ACM.
- Joseph Redmon and Ali Farhadi. 2018. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*.
- 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*, pages 91–99.
- Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. 2015. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252.
- Mayank Singh, Barnopriyo Barua, Priyank Palod, Manvi Garg, Sidhartha Satapathy, Samuel Bushi, Kumar Ayush, Krishna Sai Rohith, Tulasi Gamidi, Pawan Goyal, et al. 2016. Ocr++: a robust framework for information extraction from scholarly articles. *arXiv preprint arXiv:1609.06423*.
- Pranaydeep Singh, Srikrishna Varadarajan, Ankit Narayan Singh, and Muktabh Mayank Srivastava. 2018. Multidomain document layout understanding using few shot object detection. *arXiv preprint arXiv:1808.07330*.
- Steve Smithies, Kevin Novins, and James Arvo. 2001. Equation entry and editing via handwriting and gesture recognition. *Behaviour & information technology*, 20(1):53–67.
- Ruihua Song, Haifeng Liu, Ji-Rong Wen, and Wei-Ying Ma. 2004. Learning block importance models for web pages. In *Proceedings of the 13th international conference on World Wide Web*, pages 203–211. ACM.
- Dominika Tkaczyk, Pawel Szostek, and Lukasz Boliowski. 2014. Grotoap2-the methodology of creating a large ground truth dataset of scientific articles. *D-Lib Magazine*, 20(11/12).

- Dominika Tkaczyk, Paweł Szostek, Mateusz Fedoryszak, Piotr Jan Dendek, and Łukasz Bolikowski. 2015. Cermine: automatic extraction of structured metadata from scientific literature. *International Journal on Document Analysis and Recognition (IJ-DAR)*, 18(4):317–335.
- Tuan Anh Tran, In Seop Na, and Soo-Hyung Kim. 2015. Separation of text and non-text in document layout analysis using a recursive filter. *TIIS*, 9(10):4072–4091.
- B. Tummers. 2006. DataThief III: A program to extract (reverse engineer) data points from a graph. <https://datathief.org>, accessed: 2019-05-01.
- Tzutalin. 2015. Labelimg. <https://github.com/tzutalin/labelImg>.
- Jianwei Yang, Jiasen Lu, Dhruv Batra, and Devi Parikh. 2017. A faster pytorch implementation of faster r-cnn. <https://github.com/jwyang/faster-rcnn.pytorch>.