Automatic Metadata Information Extraction from Scientific Literature using Deep Neural Networks

Huichen Yang and William Hsu

Kansas State University, Manhattan, Kansas, USA

ABSTRACT

We present a novel computer vision-based deep learning approach for metadata extraction as both a central component of and an ancillary aid to structured information extraction from scientific literature which has various formats. The number of scientific publications is growing rapidly, but existing methods cannot combine the techniques of layout extraction and text recognition efficiently because of the various formats used by scientific literature publishers. In this paper, we introduce an end-to-end trainable neural network for segmenting and labeling the main regions of scientific documents, while simultaneously recognizing text from the detected regions. The proposed framework combines object detection techniques based on Recurrent Convolutional Neural Network (RCNN) for scientific document layout detection with Convolutional Recurrent Neural Network (CRNN) for text recognition. We also contribute a novel data set of main region annotations for scientific literature metadata information extraction to complement the limited availability of high-quality data set. The final outputs of the network are the text content (payload) and the corresponding labels of the major regions. Our results show that our model outperforms state-of-the-field baselines.

Keywords: Metadata Information Extraction, Document Layout Detection, Text Recognition, Transfer Learning, Deep Learning

1. INTRODUCTION

This paper addresses the task of simultaneous layout and free text recognition using a hybrid deep learning architecture that combines mask and cascade variants of Recurrent Convolutional Neural Networks (RCNN) and Convolutional Recurrent Neural Network (CRNN). We are motivated by the fact that the research community has expanded dramatically and the number of published scientific reports across scientific fields grows enormously each year. Research shows that the volume of daily publication doubles every 15 days.[1] Scientific literature thus includes much valuable information for researchers to help them extract key insights[2] and potential methods[3] in their respective research area. The time required to select and systematically read this increasing body of scientific literature presents a major challenge to researchers. Natural Language Processing (NLP) technology provides an efficient way for scientists and researchers to gain key insights from published articles, particularly by helping to find relevant papers related to their research area.[4] These efficient methods must, however, be trained using large corpora. Reconstructing this large corpus automatically is an ideal task for scientific literature-related NLP.

The digital documents have been most used in scientific publications, such as Portable Document Format (PDF) documents which are not machine-readable. The information of text, image, or table from these digital documents must be extracted for further processing. Some existing tools can help extract text information from digital scientific literature. For instance, PyMuPDF[5] or Tesseract Optical Character Recognition (OCR) engine[6] can help extract plain text from PDF documents. These tools, however, do not provide a comprehensive solution that combines layout segmentation and text recognition to produce structured metadata for the scientific literature such as a paper's title, authors, abstract or bibliographic references. CERMINE,[7] an open-source system for extracting structured metadata information from scientific articles, can generate an XML file that includes labels for each region of an article as well as content text, but it is not flexible enough to adapt to various formats of scientific literature, e.g., it cannot process three-column PDF scientific documents. Thus, a robust comprehensive framework is urgently needed.

To address the limitations mentioned above, we present an end-to-end learning framework for neural networks, as designed for the task of comprehensive metadata information extraction from scientific literature. In this work,

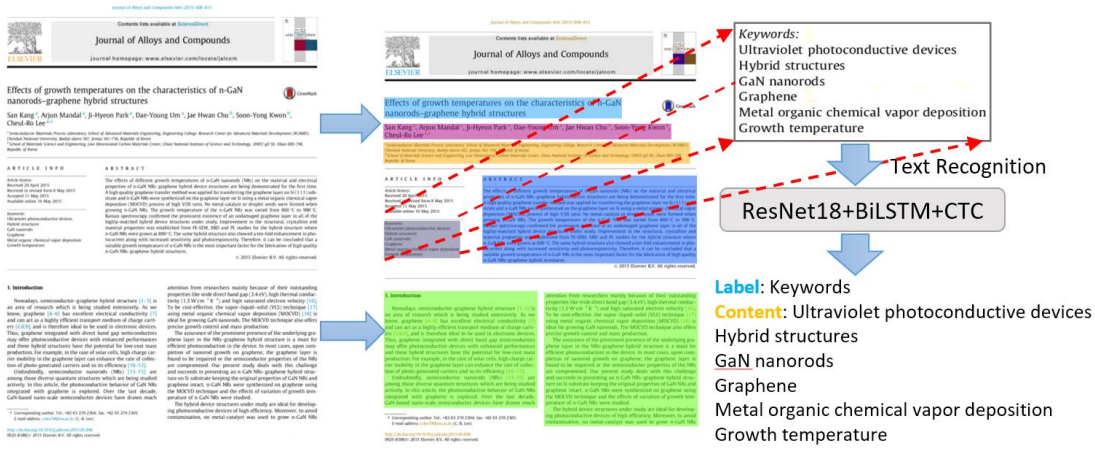

Figure 1. Example of metadata information extraction from a scanned scientific literature with end-to-end framework.

our novel contribution integrates an object detection model based on Cascade R-CNN[8] and Mask R-CNN[9] that can be used to detect main region layout in scientific literature and text recognition model based on CRNN.[10] The output gives the labels of corresponding regions, such as title, author, and abstract, and the content text of each main region (Figure 1). We also extract images and tables without processing the content into text. We train the framework separately, and the experiment results demonstrate that the applied transfer learning with fine-tuning in a pre-trained model works well even for tasks that lack large, annotated corpus. Meanwhile, we create a new annotated data set for scientific literature layout detection tasks. The data set will be available at: https://github.com/huichentt/scientific_literature_regions.

2. RELATED WORK

Several existing approaches can extract metadata information from published scientific literature automatically. Such methods usually fall into rule-based or machine learning-based categories.

These rule-based approaches rely on hand-crafted templates and strategies specifying how to extract desired information from a document. For example, Constantin et al.[11] develop PDFX, a rule-based system to extract logical structure from scholarly literature published in PDF format. Huynh et al.[12] introduce a GATE framework, based on a predefined rule for automated metadata extraction from scientific papers. The rule-based methods are commonly used for document layout analysis and require third party tools like OCR to handle the post-processing task for text recognition.

Machine learning-based approaches can be divided into the supervised machine learning approaches and the unsupervised machine learning approaches. Supervised machine learning approaches generally use a feature-based classification model with a labeled data set. Lopez et al.[13] developed GROBID, a system that uses Conditional Random Fields (CRFs) to extract metadata information from PDF documents for analyzing scientific text. Unsupervised machine learning approaches generally use clustering algorithms with a non-annotation data set. Tsai et al.[14] use an unsupervised bootstrapping algorithm to identify, categorize, and cluster scientific concepts from the literature. Moreover, deep learning approaches have been heavily researched for their ability to extract metadata from scientific literature because of the growing number of publications in diverse domains. Yang et al.[15] consider scientific literature layout detection as an object detection task with transfer learning. Prasad et al. developed Neural ParsCit,[16] a system for extracting layout and bibliographic metadata from a research document using a Long Short-Term Memory (LSTM) network.

3. METHODOLOGY

We treat metadata information extraction from scientific literature as detection and recognition tasks, which our system performs in two independent stages whose results are then integrated within our end-to-end framework.

In the first stage, the detection model identifies the potential key regions of the scientific literature in the rectangular boxes as well as the corresponding text blocks. In the second stage, the recognition model recognizes and transcribes the words from the regions that have been detected. This two-stage process has two main advantages: the flexibility of training process and the independent recognition of different languages. We describe our approach in depth in the following section.

3.1 Scientific Literature Layout and Text Detection

In scientific literature, metadata elements or attributes such as title, author, abstract, etc., are considered major regions which can be delimited using different bounding boxes within a scanned document image. We consider the problem of scientific literature layout detection as one of object detection due to the similar characteristics between the underlying formal tasks. Besides the layout of major regions in scientific literature, lines of text also need to be detected in the text recognition stage. Unlike the scene text detection task, text in scientific literature does not involve complex backgrounds or irregular fonts. Therefore, we perform the text line detection using the same object detection model. Our experiment shows the feasibility of this approach.

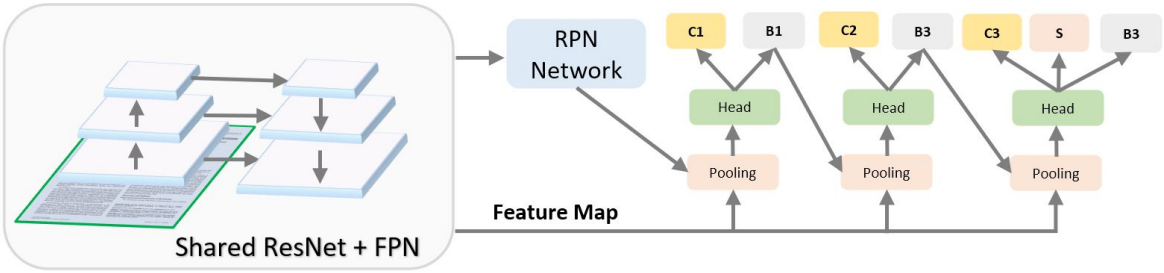

Figure 2. Architecture of Cascade Mask R-CNN. The segmentation branch is added to each cascade stage. "C" is classification, "S" is segmentation branch, and "B" is bounding box.

Our novel neural architecture combines Cascade R-CNN and Mask R-CNN within our scientific literature layout and text detection framework. Cascade R-CNN is an extension of the Faster R-CNN[17] which is a classic two-stage object detection network. The first stage of Faster R-CNN uses region proposal network that takes the feature map as input and generates the hypotheses of object proposals. These hypotheses would be processed by a region-of-interesting network to generate the detection head in the second stage. Each hypothesis is generated based on a final classification score and a bounding box. Unlike Faster R-CNN with single R-CNN network, Cascade R-CNN extends the Faster R-CNN to multi-stage with cascaded regressor. The different stages have different heads $\{H_1, H_2, ..., H_N\}$ (where N represents the number of heads) and each of the heads is assigned to a specific IoU (Intersection over Union) threshold which defines the quality of a detector to produce the bounding boxes that are loosely aligned with (small threshold) or tightly aligned with (large threshold) their ground truth. The multiple specialized regressors $\{f_T, f_{T-1}, ..., f_1\}$ (where T represents the total number of cascade stages) are optimized for the distributions that are resampled for the different heads. This resampling procedure of cascaded regression provides good positive samples for the next stage. Those features of Cascade R-CNN show better performance than Faster R-CNN. On the other hand, Mask R-CNN is another state-of-the-art object detection network and is modified based on Faster R-CNN. Compared with Faster R-CNN, the Mask R-CNN introduces ROIAlign to replace the ROI Pooling layer of Faster R-CNN to resolve the pixel-to-pixel misalignment issue between network inputs and outputs. Moreover, Mask R-CNN adds a segmentation branch for instance segmentation. The final loss function is composed of three parts: classification, bounding box regression, and mask loss.

To integrate Cascade R-CNN and Mask R-CNN, we use ResNet-50[18] with a Feature Pyramid Network (FPN)[19] as the backbone network for feature extraction from input images. The RPN network generates the proposal regions which are then fed into ROI pooling layer. The network head takes ROI pooling as input and generates three predictions: classification (C), mask (S), and bounding box regression (B). The output of one stage is used as input for the next stage (Figure 2). Therefore, the current bounding box distribution is generated

by the previous regressor to optimize the current regressor. The final loss function combines classification and location (bounding box regression) loss functions at different stages (1).

$$\mathcal{L} = \sum_{i=0}^{2}(\mathcal{L}_{\text{cls\_S}_i} + \mathcal{L}_{\text{loc\_S}_i}) + \mathcal{L}_{\text{cls\_FPN}} + \mathcal{L}_{\text{loc\_FPN}} + \mathcal{L}_{\text{mask}} \tag{1}$$

In the equation above, cross-entropy loss is used as the classification loss function, and smooth L1 is used as the bounding box regression loss function. The total loss function is calculated at three cascade stages (denoted by i ranging from 0 to 2), FPN network, and mask branch. We choose GIoU (Generalized Intersection over Union) (2)[20] instead of IoU as increasing thresholds over stages. The GIoU handles the case of non-overlapping bounding boxes that is inapplicable for IoU.

$$GIoU = IoU - \frac{|C_{ab} - U|}{|C_{ab}|} \tag{2}$$

In the equation above, C represents the smallest enclosing convex object of a and b where a and b are two arbitrary convex shapes. U represents the area of C that does not belong to either a or b. The bounding box regressor trained for a certain GIoU threshold tends to produce bounding boxes of higher GIoU threshold. The segmentation branch has been added to the last cascade stage. Furthermore, the anchor box aspect ratio is another important thing to be considered in object detection tasks since different objects have different aspect ratios. For instance, the aspect ratio box of reference is different from the aspect ratio box of text block in scientific literature layout detection tasks. Our experiment results demonstrate that selecting appropriate aspect ratio brings a higher accuracy level.

3.2 Text Recognition

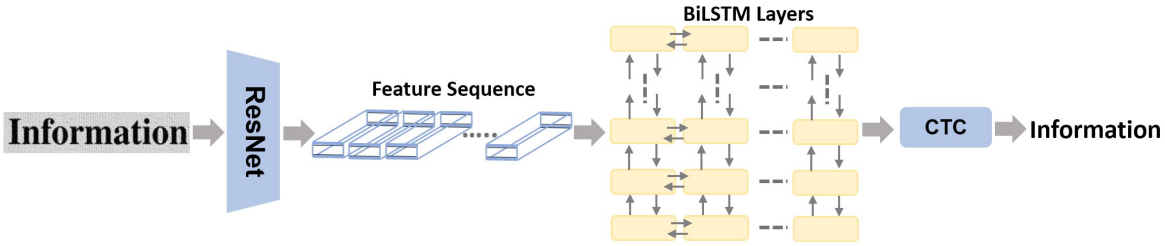

Figure 3. Architecture of Text Recognition model.

We propose a text recognition model based on an CRNN architecture which outputs a sequence of characters (Figure 3). The model uses Convolutional Neural Network (CNN) to extract visual features from input images. We use ResNet18 as backbone for visual feature extraction. Then the feature vector will be used to predict the character label distribution of each frame through the sequence modeling stage - a multi-layer Bidirectional Long-Short Term Memory (BiLSTM) network[21] which is used for sequence feature extraction. The BiLSTM layer will generate a number of layer-specific hidden state sequences, h^1, h^2, ..., h^N, where N is the number of layers in the BiLSTM. The encoder layer is followed by a softmax to produce a posterior probability matrix $y = (y^1, y^2, ..., y^T)$ where T is the length of the sequence. The last transcription layer is implemented by Connectionist Temporal Classification (CTC) loss.[22] CTC allows us to find the optimal label with the highest conditional probability by using dynamic programming to compute all potential alignment paths for predicting the probability distribution of all characters of the alphabet at each position in the image.

$$\mathcal{L} = \log \sum_{c \in A(s)} \prod_{t=1}^{T} p(c_t | h_t^N) \tag{3}$$

In the above loss function, $c_t \in C$ is a label set, consisting of all the possible output labels and a "blank" symbol. $A(\mathrm{s})$ includes all possible paths of sequence $\mathbf{s}$, where $\mathbf{s} = \mathrm{s}_1, \mathrm{s}_2, ..., \mathrm{s}_\mathrm{L}$. The alignments between those predictions could include duplicate or blank characters. For instance, the word "STUDY" could be processed into "S-T-UU–DD-Y" such that there are duplicate and blank characters. Assuming that the optimal labeling is from the most probable path, we can find the most probable path based on the most likely character at each position of the sequence. The duplicate characters will be removed then if there is no blank character between them. The loss function is used to jointly learn all the model parameters.

4. EXPERIMENT

In this section we describe the data set that we use to train the models, together with implementation details for training process.

4.1 Data Set

We use two data sets to train the framework independently.

Scientific Literature Layout and Text detection

Due to the limited availability of training corpora, we developed a new corpus by annotating the composite corpus (synthesis data set) from Yang et al.[15] This experimental test bed combines three existing data sets: region-labeled articles,[23] ICDAR-2013,[24] and GROTOAP;[25] however, it does not include text block-related annotation. Therefore, we labeled text blocks using 110 images from 20 scientific literature in PDF format. The final data set has 1660 images from 383 scanned scientific articles in PDF format, and it includes 11 labels corresponding to main regions.

- ***Title***: the title of scientific literature.
- ***Authors***: the authors' names.
- ***Address***: the affiliation information of authors, including authors' addresses, email, etc.
- ***Abstract***: an abstract section.
- ***Keyword***: the selected keywords.
- ***Body***: the main block of articles.
- ***Figure***: all figures but excluding logos or icons from publishers.
- ***Table***: the tabular contents.
- ***Caption***: the captions for both figures and tables.
- ***Reference***: the bibliography information, excluding post-references notes.
- ***Text***: text block.

Text Recognition

We use a combined corpus of data sets to train the network for text recognition. Three different data sets are integrated into the composite data set. The first one is English2k which is the sub-data set of SCUT_FORU_DB.[26] It has 1715 natural images for text detection and recognition from the Flickr website. We crop 7136 images of characters from this data set since we only focus on text recognition tasks. The second and third data sets are generated by Text Recognition Data Generator, which is an open-source tool to generate synthetic data for text recognition. We use three categories to produce the composite data randomly for generating:

- 2000 images of letters and numbers, containing 3, 4, 5, 6, 7 characters, respectively.

- 2000 images of letters only, containing 3 to 7 characters, respectively.
- 5000 images of letters, numbers and symbols, containing 5 characters.

The final combined corpus for text recognition consists of 30136 images. We split them into 24000 images for training and the rest are used for testing.

4.2 Implementation Details

All models are trained and tested on a single NVIDIA 2080 Ti GPU. The framework of scientific literature layout detection and text detection is implemented on Pytorch using Detectron2[27] and fine-tuned with pre-trained weight on MS COCO data set[28] (37 epochs). It has been trained with a batch size of 4 for 50 epochs. The SGD (Stochastic Gradient Descent) is used as an optimization algorithm. We use 0.002 as the initial learning rate which decays by 0.1 after every 20 epochs. The Cascade Mask R-CNN model is trained with GIoU threshold values of 0.5, 0.6, and 0.7, with 5 anchor scales from 32 to 512, and with 4 anchor aspect ratios: [0.1, 0.5, 1.0, 2.0]. The rest of models are trained by standard aspect ratios with [0.5, 1.0, 2.0]. We also use YOLOv3[29] which is trained based on DarkNet53 network that was pre-trained on MS COCO data set.

Compared to the detection model which processes the images with sizes close to 612 $\times$ 729 pixels, the text recognition model processes much smaller input images at 32 x 280 pixels. Thus, we use ResNet18 as CNN layer to extract the feature from images. Our experiment indicates that using ResNet18 increases the accuracy by 7% compared with VGG16.[30] To consider the robustness of the recognition model, we add some scientific literature-related symbols, such as $\pm$, °C, $\neq$, to the dictionary to train the model. The text recognition model has been trained from scratch with 50 epochs.

5. EVALUATION

We evaluate our end-to-end framework from two different perspectives. First, we use the same data set to train and test other different configuration models to compare with our Cascade Mask R-CNN model for scientific literature layout and text detection. The detection results may affect the performance of the final framework, because the text recognition model relies on the layout and text detection results. Second, we manually determine ground truth from 10 scientific articles for metadata extraction evaluation. It includes the main regions of scientific literature that are consistent with our training data set. Given the various layouts from different publishers, this ground truth is collected from different publishers for robustness testing.

Scientific Literature Layout and Text Detection Evaluation

We use MS COCO evaluation metrics for layout and text detection testing. COCO provides 12 indicators to evaluate the performance of object detector. We use 7 of these for our detector evaluation. The Faster R-CNN with ResNet50_FPN is baseline, and we compare it with different other object detection models using different backbones. Our Cascade Mask R-CNN model achieves better performance than others for scientific literature layout and text detection tasks (Table 1).

Table 1. Overall comparison of bounding box results for scientific literature layout and text detection.

Model	Backbone	mAP	AP50	AP75	APs	APm	APl	AR
Faster R-CNN (**baseline**)	ResNet50_FPN	75.79	92.51	84.24	59.18	63.82	76.71	78.95
Faster R-CNN$_{\text{GIoU}}$	ResNet50_FPN	76.24	93.52	85.77	62.78	63.67	76.33	81.31
Faster R-CNN	VoVNetV2-39	76.52	93.23	86.50	72.32	61.63	74.66	77.28
YOLOV3	DarkNet53	55.36	73.65	64.10	-	-	-	53.26
Mask R-CNN	ResNet50_FPN	77.25	92.22	85.60	64.38	65.23	77.65	83.96
Cascade Mask R-CNN (**ours**)	ResNet50_FPN	**79.92**	**94.36**	**88.30**	70.75	**69.84**	**81.26**	**88.60**

Likewise, for each label, our Cascade Mask R-CNN model also outperforms the baseline Faster R-CNN (Figure 4). More examples of layout and text detection are shown in Figure 5.

Metadata information extraction evaluation

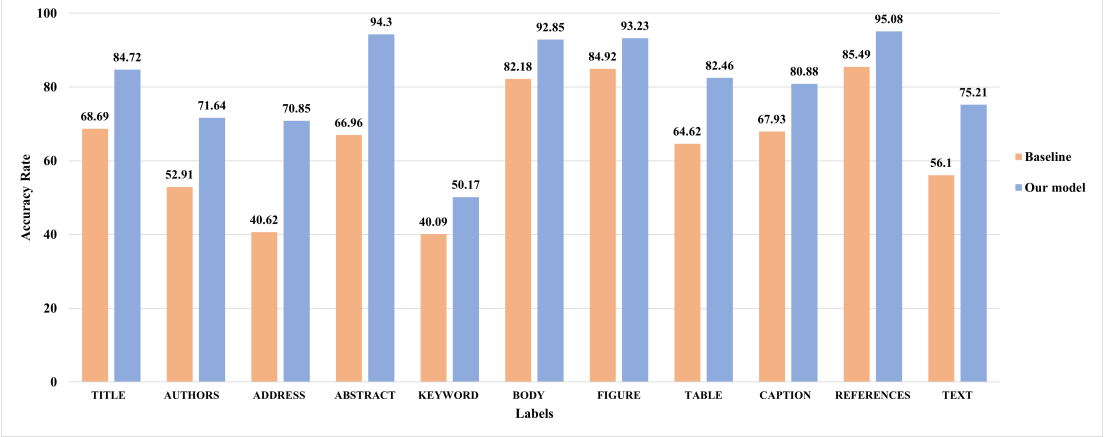

Figure 4. Detection results comparison with baseline by labels at 0.5 IoU.

We use precision, recall and F1 to evaluate the performance of metadata information extraction.

$$Precision = \frac{N_C}{N_T}, \quad Recall = \frac{N_C}{N_G} \tag{4}$$

where N_C is the number of words extracted correctly by our framework, N_T is the total number of words extracted, and N_G is the number of words in ground truth. To be clear, the extracted word is considered correct if all characters match the word of ground truth. In addition, we consider the subscript or superscript character as a regular part of the word and ignore the different font styles. Moreover, the metadata information does not include any text from images or tables. We conduct these three tests for each label and compare them with CERMINE (Table 2).

Table 2. The evaluation results comparison between our approach and CERMINE.

Label Name	**Precision(%)**	**Recall (%)**	**F1 (%)**	**Precision(%)**	**Recall (%)**	**F1 (%)**
	Our approach			CERMINE		
Title	90.92	88.73	89.81	91.67	89.27	90.45
Authors	**82.57**	**80.36**	**81.45**	74.75	67.37	70.87
Address	**87.22**	**79.69**	**83.29**	87.16	79.29	83.04
Abstract	**81.26**	**78.59**	**79.90**	75.18	75.17	75.17
Keyword	**92.93**	**91.56**	**92.24**	77.84	64.84	70.75
Body	75.23	77.49	76.34	83.56	79.79	81.63
Caption	**79.16**	**77.86**	**78.50**	22.50	22.50	22.50
Reference	**62.73**	58.52	60.55	59.77	62.55	61.13

The metadata information extraction results from text recognition model heavily depend on the training data set. Our ground truth in this evaluation is collected from scientific papers, particularly related to biology and materials science. Some domain-specific characters such as unit symbols in chemistry actions need to be considered to add for further improvement. Moreover, evaluation results are also affected by resolution of input images. Cropping clear images from layout and text detection stage is a necessary preprocessing step.

6. CONCLUSION

In this paper, we have introduced an end-to-end framework for metadata information extraction from scientific literature. The framework integrates two models: scientific literature layout and text detection based on Cascade Mask R-CNN, and text recognition based on CRNN. Such integration enables us to both input scanned scientific

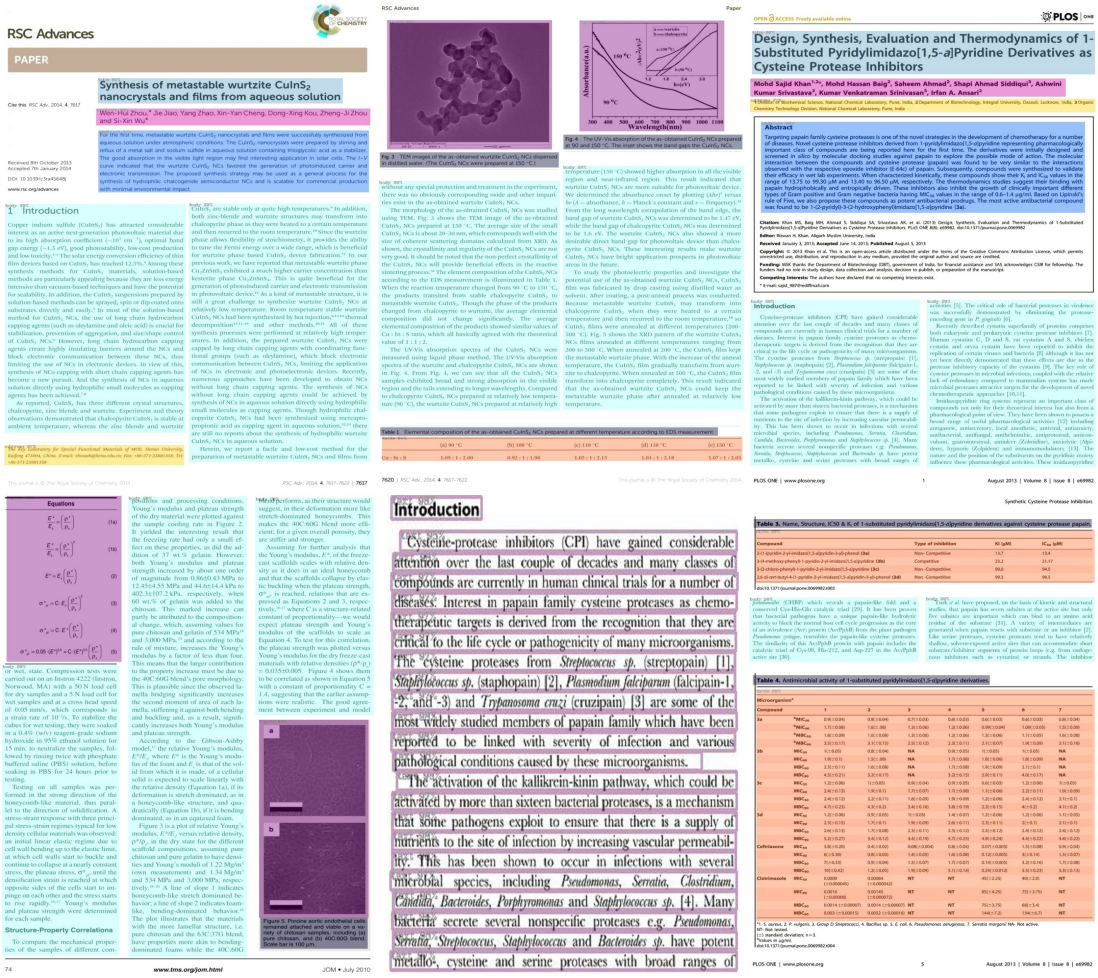

Figure 5. Layout and text detection results with corresponding labels.

article images and output the text with corresponding labels at once. This framework has flexibility for training to adopt diverse layouts of scientific literature from different publishers. Meanwhile, it can recognize different languages independently. In addition, we create a novel data set for scientific literature layout and text detection tasks. Our proposed Cascade Mask R-CNN model provides a benchmark with this data set.

In future work, we will examine the application of this end-to-end framework to the development of a large corpus for a range of domain-specific NLP tasks that currently lack such corpora, such as information extraction for knowledge base population, question answering, and dynamic search and indexing in the materials science domain. As we have seen from this work, there is a potential trade-off between the model performance and its accuracy that requires further attention, such as balancing between reducing the number of parameters of the model and maintaining/improving the accuracy.

REFERENCES

[1] Torres-Salinas, D., "Daily growth rate of scientific production on covid-19. analysis in databases and open access repositories," *arXiv preprint arXiv:2004.06721* (2020).

[2] Kim, E., Huang, K., Saunders, A., McCallum, A., Ceder, G., and Olivetti, E., "Materials synthesis insights from scientific literature via text extraction and machine learning," *Chemistry of Materials* **29**(21), 9436–9444 (2017).

[3] Yang, H., Aguirre, C. A., Maria, F., Christensen, D., Bobadilla, L., Davich, E., Roth, J., Luo, L., Theis, Y., Lam, A., et al., "Pipelines for procedural information extraction from scientific literature: towards recipes using machine learning and data science," in [*2019 International conference on document analysis and recognition workshops (ICDARW)*], **2**, 41–46, IEEE (2019).

[4] Huo, H., Rong, Z., Kononova, O., Sun, W., Botari, T., He, T., Tshitoyan, V., and Ceder, G., "Semi-supervised machine-learning classification of materials synthesis procedures," *npj Computational Materials* **5**(1), 1–7 (2019).

[5] Liu, R. and McKie, J. X., "PyMuPDF." May 24, 2018 http://pymupdf.readthedocs.io/en/latest/.

[6] Kay, A., "Tesseract: an open-source optical character recognition engine," *Linux Journal* **2007**(159), 2 (2007).

[7] Tkaczyk, D., Szostek, P., Fedoryszak, M., Dendek, P. J., and Bolikowski, Ł., "Cermine: automatic extraction of structured metadata from scientific literature," *International Journal on Document Analysis and Recognition (IJDAR)* **18**(4), 317–335 (2015).

[8] Cai, Z. and Vasconcelos, N., "Cascade r-cnn: High quality object detection and instance segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2019).

[9] He, K., Gkioxari, G., Dollár, P., and Girshick, R., "Mask r-cnn," in [*Proceedings of the IEEE international conference on computer vision*], 2961–2969 (2017).

[10] Shi, B., Bai, X., and Yao, C., "An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition," *IEEE transactions on pattern analysis and machine intelligence* **39**(11), 2298–2304 (2016).

[11] Constantin, A., Pettifer, S., and Voronkov, A., "Pdfx: fully-automated pdf-to-xml conversion of scientific literature," in [*Proceedings of the 2013 ACM symposium on Document engineering*], 177–180 (2013).

[12] Huynh, T. and Hoang, K., "Gate framework based metadata extraction from scientific papers," in [*2010 International Conference on Education and Management Technology*], 188–191, IEEE (2010).

[13] Lopez, P., "Grobid: Combining automatic bibliographic data recognition and term extraction for scholarship publications," in [*International conference on theory and practice of digital libraries*], 473–474, Springer (2009).

[14] Tsai, C.-T., Kundu, G., and Roth, D., "Concept-based analysis of scientific literature," in [*Proceedings of the 22nd ACM international conference on information & knowledge management*], 1733–1738 (2013).

[15] Yang, H. and Hsu, W. H., "Vision-based layout detection from scientific literature using recurrent convolutional neural networks," in [*2020 25th International Conference on Pattern Recognition (ICPR)*], 6455–6462, IEEE (2021).

[16] Prasad, A., Kaur, M., and Kan, M.-Y., "Neural parscit: a deep learning-based reference string parser," *International journal on digital libraries* **19**(4), 323–337 (2018).

[17] Ren, S., He, K., Girshick, R., and Sun, J., "Faster r-cnn: towards real-time object detection with region proposal networks," *IEEE transactions on pattern analysis and machine intelligence* **39**(6), 1137–1149 (2016).

[18] He, K., Zhang, X., Ren, S., and Sun, J., "Deep residual learning for image recognition," in [*Proceedings of the IEEE conference on computer vision and pattern recognition*], 770–778 (2016).

[19] Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., and Belongie, S., "Feature pyramid networks for object detection," in [*Proceedings of the IEEE conference on computer vision and pattern recognition*], 2117–2125 (2017).

[20] Rezatofighi, H., Tsoi, N., Gwak, J., Sadeghian, A., Reid, I., and Savarese, S., "Generalized intersection over union: A metric and a loss for bounding box regression," in [*Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*], 658–666 (2019).

[21] Hochreiter, S. and Schmidhuber, J., "Long short-term memory," *Neural computation* **9**(8), 1735–1780 (1997).

[22] Graves, A., Fernández, S., Gomez, F., and Schmidhuber, J., "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in [*Proceedings of the 23rd international conference on Machine learning*], 369–376 (2006).

[23] Soto, C. and Yoo, S., "Visual detection with context for document layout analysis," in [*Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*], 3464–3470 (2019).

[24] Göbel, M., Hassan, T., Oro, E., and Orsi, G., "Icdar 2013 table competition," in [*2013 12th International Conference on Document Analysis and Recognition*], 1449–1453, IEEE (2013).

[25] Tkaczyk, D., Czeczko, A., Rusek, K., Bolikowski, L., and Bogacewicz, R., "Grotoap: ground truth for open access publications," in [*Proceedings of the 12th ACM/IEEE-CS joint conference on Digital Libraries*], 381–382 (2012).

[26] Zhong, Z., Jin, L., and Huang, S., "Deeptext: A new approach for text proposal generation and text detection in natural images," in [*2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*], 1208–1212, IEEE (2017).

[27] Wu, Y., Kirillov, A., Massa, F., Lo, W.-Y., and Girshick, R., "Detectron2." https://github.com/facebookresearch/detectron2 (2019).

[28] Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L., "Microsoft coco: Common objects in context," in [*European conference on computer vision*], 740–755, Springer (2014).

[29] Redmon, J. and Farhadi, A., "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767* (2018).

[30] Simonyan, K. and Zisserman, A., "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556* (2014).