A Deep Learning-based Formula Detection Method for PDF Documents

Liangcai Gao, Xiaohan Yi, Yuan Liao, Zhuoren Jiang, Zuoyu Yan, Zhi Tang✉

ICST, Peking University Beijing, China

{glc,chlxyd, liao_yuan, jiangzr,yanzuoyu3,tangzhi}@pku.edu.cn

***Abstract*—In practice, PDF files may be generated by different tools and their character information quality could be different. As a result, the approaches to detecting formulae from PDF documents usually have much different performance on different PDF files. To address this problem, in this paper we combine and refine the Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN) model to detect formulae according to both their character and vision features. Based on the characteristic of PDF documents, we propose a series of strategies to train and optimize deep networks, such as the implicit class down-sampling strategy which can reduce the unbalancedness between formulae and other page elements (e.g., text paragraphs, tables, figures, etc.). The region proposal method is also redesigned to generate moderate formula candidates through combining the bottom-up and top-down layout analysis. The experimental results show that the combination of CNN and RNN can increase the robustness of our proposed detection method. Furthermore, the proposed method outperforms the existing formula detection methods on both a ground-truth dataset and a larger self-built dataset, which would be released and available for research purposes.**

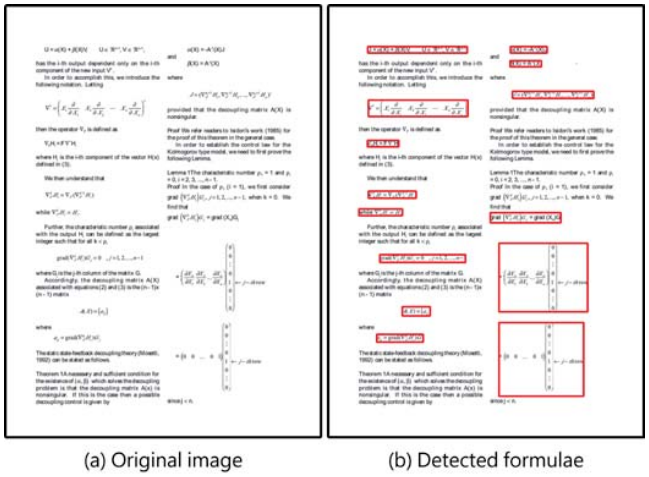
Keywords—formula detection; deep learning; PDF documents

I. Introduction

Up to now, there are massive formulae containing in PDF documents. However, PDF files only contain page rendering information, which makes their formulae not convenient to be transformed, searched and reused. Thus, PDF formula recognition is proposed to convert the formulae in PDF documents to structured formats such as Latex, MathML, which usually contains two sequenced steps: formula detection and structure analysis. The former is to identify whether a page contains formulae and further determines their bounding boxes. The latter is to extract the spatial and grammar structure of formulae. In this paper we focus on the former task, formula detection. Fig. 1 shows an example of formula detection.

This is an annotation

Several PDF formula detection methods have been proposed in the past, these methods are rule-based or learning-based. Rule-based methods firstly extract formula symbols from PDF, then extend formula areas according to the operator domains of these symbols. The idea of learning-based methods is to split document pages into candidate regions at first, then extract region features and classify each region. Despite those existing methods, the formula detection task still faces many challenges. For example, existing formula detection methods mainly rely on character information directly extracted from PDF files. However, character encoding of different PDF version may be different, which could make PDF parsing tools such as PDFBox [1] output inconsistent and unreadable character encoding information from PDF files. Besides, some mathematical symbols might be stored as images in PDF files, not characters. Furthermore, complex symbols may consist of several parts. For instance, "$\sqrt{\ }$" may consists of a character " √ " and a short line. Those complex symbols cannot be directly extracted from PDF files. Therefore, formula detection in PDF documents still needs further research to cope with the difference and complexity of PDF files.



(a) Original image (b) Detected formulae

Fig. 1. An example of formula detection. (a) Document page (b) Detected results, formulae in it is detected by our method and annotated with red boxes.

On the other hand, as a new technology, deep learning has achieved outstanding performance in many applications, and many structures of deep networks such as Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN) have been proposed to various tasks. The advantage of deep networks is that the networks can automatically extract appropriate features for given data and tasks, which is quite helpful for researchers, especially those who are lacking in domain specific knowledge. Deep learning has also shown a great ability on document related tasks such as document classification, handwriting recognition, etc.

2379-2140/17 $31.00 © 2017 IEEE
DOI 10.1109/ICDAR.2017.96

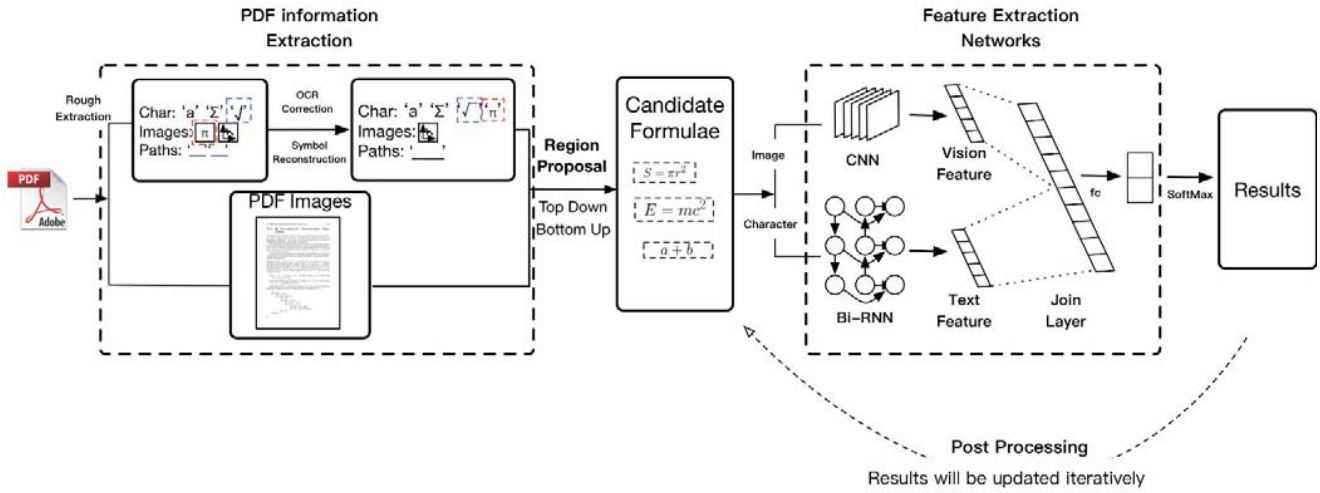


Fig. 2. Method framework

Taking advantages of deep networks, in this paper, we combine a CNN model and a RNN model to utilize both character and visual features on PDF formula detection. Specifically, for a given PDF page, candidate formula regions are firstly generated by both bottom-up and top-down strategies, then the feature extraction networks and post-processing rules are recursively used to rate and modify these candidates. The feature extraction networks combine both CNN and RNN to separately process image flow and text flow. In addition, we also create a large dataset to test the proposed PDF formula detection methods. The dataset contains more than 1,000 PDF pages with more than 22,000 manual labeled formulae. This dataset would be released for research purposes.

II. Related Work

In this section, we introduce the current research on formula detection and deep neural networks.

A. Formula Detection

The research of formula structure extraction can date back to 1968 [2]. For the formula detection task, there are also many efforts having been made in the past. Traditional formula detection methods are rule based or learning based. Most of the rule-based methods firstly identify specific mathematical characters, such as "+", "Σ", etc., and then attempt to extend formula area according to the operator domains of these symbols [3-7]. Kacem et al. [8] utilized a fuzzy logic model to extract mathematical symbols, then merged their operation region to form the complete formula areas. Inoue et al. [9] designed rules to separate isolated formulae form text lines, according to the observation that the text line usually contains more plain text words. The rule based methods usually contain a set of parameters, which can hardly be adaptive to various documents and formulae. Currently, the main idea of learning based methods is to split document pages into text lines at first, then classify each text line and combine the classification results by rules. Chowdhury et al. [10] firstly extracted lines in documents, then used a decision tree to filter formula lines according to candidate layout features. Lin et al. [11] utilized Support Vector Machine (SVM) to classify each text line to identify formula lines. The main differences between those methods are the features and learning models adopted in them.

There are also many works on the basis of extracted formulae. [12] constructed a formula retrieval engine based on the formulae extracted from Wikipedia. [13] utilized the deep learning technical to convert formula images to structure format descriptions. Detecting formulae in document is one precondition of these researches.

B. Deep Networks

In the past few years, deep networks have been widely researched and achieved good performance on various tasks. In object detection field. Ren et al. proposed the Faster-RCNN [14], which achieved the mean Average Precisions (mAPs) of 69% on VOC 2007. On machine translation, Cho et al. [15] and Sutskever et al. [16] proposed a framework called RNN Encoder-Decoder. This framework encoded a sentence of one language to an intermediate representation and then decoded it to a target language. In document modeling, Tang et al. [17] proposed a novel model, which first learned sentence representations with Long Short Term Memory (LSTM) and then encoded them into document representations using gated RNN. Besides, deep learning has also been widely used on document recognition related tasks. Sharma et al. [18] used the deep networks to automatically spot and recognize document words. Afzal [19] employed CNN to classify documents.

III. Framework

The framework of our method is shown in Fig. 2. In the framework, the input is a PDF file, and the formulae appearing in their pages will be detected and output by two sequential parts, namely candidate formula region generation and formula identification.

In the first part, the page rendering information in PDF files is firstly parsed and several steps are executed to check and correct the information. Then, both top-down and bottom-up layout analysis methods are utilized to generate candidate formula regions. The output of this step is the text, graph and image information stream of each candidate region, which is the input of the following process.

Then in the formula identification part. Feature extraction networks are trained to extract the features of candidates, and the actual class of each candidate is decided according to these features. Then a series of post-processing rules are designed to adjust and refine incomplete formula areas, which would be detailed in Section. 6. The classifying and post-processing will be executed alternately, till no formula region is modified.

IV. Candidate Formula Region Generation

This section introduces the candidate formula generation methods used in the above framework. At first, the page rendering information in PDF files is extracted. Then on the basis of PDF information, this paper utilizes two different methods to generate candidate formula regions.

A. PDF Information Extraction

Rendering information such as mathematical characters are useful for PDF formula detection. However, character information directly extracted from PDF files might be imprecise. Therefore, a correction process is necessary to fix the extraction results. In our method, PDF information is extracted and corrected by the following three steps.

1) Original Extraction

In the original extraction step, we use the extraction tool provided by Founder Corporation to extract PDF information. The tool is very similar to PDFBox [1] and is developed according to the PDF specification [20]. The extracted PDF information includes characters, images and paths, with their bounding boxes, encodings, fonts and other attributes. The extracted information is split into different information streams for the further processing: character stream, graph (path) stream and image stream.

2) OCR Based Character Correction

The primary information could be imprecise. For instance, some characters may be falsely encoded, or may be stored as character images in PDF. Thus an OCR system is used to correct the primary information. Especially, if a small image is recognized as a character, it will be added to the character list.

3) Symbol Reconstruction

Some complex symbols in PDF files could be stored in multiple parts, such as the "$\sqrt{\ }$", which might be the combination of a character "√" and a sort line "—". Besides, some operators in formula might be words, such as "log", "max", "sin", etc. In the symbol reconstruction step, those particular mathematical operators are recognized by a series of rules, for example, the adjacent 'l', 'o', 'g' would be replaced by symbol 'log'. These recognized symbols will be added into character stream with corresponding character encodings.

The correction of original PDF information increases the veracity of formula elements, and relieves the influence of wrong and missing characters on PDF formula detection task.

B. Candidate Formula Generation

To detect formulae in PDF documents, formula or formula-like candidate regions should be firstly proposed from document pages before the classification process. In traditional region proposal methods, *Sliding-Window* is usually used to generate candidate regions, but the large difference between formula scales make it hard to use sliding window to generate precise candidate formula regions. To obtain candidate regions with high recall, we use both top-down and bottom-up layout analysis methods to generate candidate formula regions from document pages.

1) Top-down candidate formula generation

The top-down layout analysis generally starts from the whole page, and separates pages into individual regions. Our top-down method is built upon the XY cut algorithm [21], which is usually used to segment document pages. The algorithm cuts a page into new parts on horizontal and vertical direction recursively according the white space of the page. And its time complexity is $O(W*H)$, where W and H are the width and height of a page. For PDF documents, there is a faster way to implement XY cut for PDF document. After the step of PDF information extraction, the bounding boxes of characters, images and paths are obtained. We create and maintain two segment trees [22] respectively for X-axis and Y-axis, which memorize the projection height of each coordinate. Then for each object extracted from PDF files, we insert it into the two trees and update the tree values. Because the cost of insert and update operation is $O(log(n))$ for segment tree, so the time complexity of the faster XY cut is $O(N*(log(W)+log(H)))$, where N is the number of extracted objects in the page, which is generally less than 1,000 in a document page.

2) Bottom-up candidate formula generation

The top-down methods are difficult to process the irregular layout such as the wrap or surrounding layout, while the bottom-up methods can process this situation well [23]. Thus in this paper, the bottom-up strategy is also used to generate more candidate regions to improve the recall.

Our bottom-up method is inspired by [23]. Firstly, the extracted objects in the step of PDF information extraction can be regarded as low level Connected Components (CCs). Then these CCs are incrementally regrouped in horizontal and vertical direction by different rules. In the horizontal direction, the CCs are merged according to their distances and scales, because the components in one line are usually in a similar height and width. In the vertical direction, since the task is formula detection, the merge rules are designed on the basis of formula symbols. For example, for a CCs group contains symbol "Σ", "∏", etc., or this group contains a line in the top or bottom which might be a fraction line, the adjacent CCs in vertical direction such as "i=1" should to be merged in this group.

V. Feature Extraction Networks

In the framework of our method, feature extraction networks are used to extract both visual features and character features of input formula candidates. This section introduces the detailed structures of the networks, as well as some optimizing strategies.

A. Feature Extraction Networks

The feature extraction networks consist of two sub networks. One is a RNN, which is used to extract the features of sequential character information. The other is a CNN, which is utilized to extract the visual features. A joint layer is used to connect these two networks. Then the joined features are sent to a classifier to determine the category of a candidate region.

TABLE. I STRUCTURES OF OUR USED CNN

Part Index	Layer Type	Layer Params
1	Input Layer	None
2	Convolution Layer	96 of 11x11 Conv kernels
	Max-Pooling Layer	Pooling size: 3x3
	Convolution Layer	256 of 5x5 Conv kernels
	Max-Pooling Layer	Pooling size: 3x3
	Convolution Layer	384 of 3x3 Conv kernels
	Convolution Layer	384 of 3x3 Conv kernels
	Convolution Layer	256 of 3x3 Conv kernels
	Spatial Pooling Layer	**Pooling size: 7x7**
3	Full Connection Layer	4096 of Neurons
	Full Connection Layer	4096 of Neurons
The network output 4096 dimensional vision features		

1) Extraction of charecter feature

Character information is useful for detecting formulae. Traditional methods generally use the manually designed character features to decide whether a candidate is a formula, which is limited to the domain knowledge and the relevant threshold parameters are difficult to determine. Thus in our method, the character features are extracted by RNN, which is good at extracting sequence features. The detailed structures of our model are described as follows.

The adopted RNN model is two stacked layers of Bi-LSTM. At first, each character in the character stream is encoded to a one-hot vector. The length of the vector is the number of different character encodings. For the vector of each character, the index which represents its encoding is 1, others are 0. For each formula, its character vector sequences are input into the first Bi-LSTM layer, with each vector corresponding to a unit cell. The output of the first layer will be input to the second Bi-LSTM layer and the final 10 units from the second layer are combined as the character features of input formula. The unit is set to output a 512-dimension feature vector, so the length of character features is 5,120.

2) Extraction of visual feature

Besides the character features, formulae also contain particular visual characteristics. Traditional methods for PDF formula detection seldom consider the visual information. Since CNN has shown its great power on various vision tasks, we introduce the CNN to extract the visual features of formulae.

The CNN used in our method is modified from Alexnet [24], which is the champion network of the ILSVRC 2012, and is also widely used in many document-related tasks [18], [19]. The architecture of our CNN consists of three parts, as shown in Table I.

As Table I shows, the first part is the input layer. The image data is structured to an input batch, then the input batch is sent to the data layer for training. The input layer is followed by a series of convolution layers and pooling layers. A convolution layer contains hundreds of convolution kernels, which can extract different features from the input images and generate the feature maps; The pooling layer is the down-sampling of the input images, which can reduce the computation and increase the robustness of the networks. One characteristic of formulae is that their scales are various, with a large range of aspect ratios. Traditional CNN requires the input be in the same size, which may be not suitable for various formula scales. Thus we replace the last max pooling layer by the SPatial Pooling (SPP) layer [25]. The SPP layer can overcome the problem of fixed input by using a fixed-scale down-sampling, so the size of the input image can be changeable. Rectified Linear Units (ReLU) is utilized as the activation function to increase the nonlinearity of our CNNs. The second part will output 256 of 7*7 feature maps. The third part is the full connection layer, which is a double-layer of multi layered perceptron. This layer will receive the feature maps from the former part and output a 4096-dimension image feature vector.

3) Networks merging

Since the image features and character features are extracted by CNN and RNN, we add a joint layer to connect these features. Then an extra full connection layer is used to map the features to a 2-dimention vector, which is sent to the final Softmax classifier.

B. Training

1) Implicit class down-sampling of trainig data

For the detection task, one common problem is that the number of background is much more than that of foreground. Unbalanced data makes the classifier tend to classify data to the background side and reduces the accuracy. Thus some methods perform random down-sampling process on the negative data in order to reduce the unbalancedness. However, most of the areas in document pages are texts, thus a large number of the negative data are text areas. This would reduce the networks classification ability for minority page objects such as formulae, figures and tables. Therefore, in the training process, we propose the implicit class down-sampling method to balance training data.

Specifically, one positive data list and several implicit negative data lists are created at first. The training data is manually labeled with several kinds of regions, such as formula regions, paragraph regions, table regions, etc. The formula regions are added into positive data list while others are added into their corresponding negative data lists. Next, the region proposal method mentioned in the former section is used to generate candidate regions. For each region, if its Intersect Over Union (IOU) with one ground truth formula region is higher than a threshold (*IOU_THRSHOLD*), this region is added to the positive data list, otherwise the data will be added to a negative data list. Finally, in the training process, each training batch will load data from those lists averagely. The proportion of positive data is set to 40% in order to guarantee the number of positive data.

The implicit class down-sampling balances the quantity of positive data and negative data, as well as guaranteeing the diversity of negative data. This could help the classifier learn more robust features.

2) Training details

The parameters of CNN and RNN are initialized with Xavier. Stochastic Gradient Descent(SGD) is utilized to train the network. In each iteration, the probability of each class from the network is compared with the ground truth, and the

differentiation is used to adjust the network layer by layer. There are still two parameters need to be adjusted, namely batch-size and learning rate. The batch-size affects the result less, so is simply set to 64. The adjustment of learning rate is based on the training process. If the average loss decreases slowly, the learning rate may be too low. If the average loss fluctuates severely, the learning rate may need to be decreased. After adjustment, the learning rate in the experiments is 0.001.

VI. Post-Processing

After classification, the category and probability of each candidate region is given. But these regions may overlap with each other, so further process is needed to obtain the final results.

Non Maximum Suppression (NMS) is used in the post-processing as the first step to filter overlapped regions according to their confidence. Then we implement a correction process for each formula to ensure its integrity. In detail, each mathematic symbol in formulae is checked, if the operands of this symbol are missing, its possible parts will be searched and added to this formula to form a new formula region. This new formula region will be sent to the deep classifier to redefine its category. If it is still classified as formula, then the new formula region will replace the old one. The correction process will be executed recursively, till no new formula regions are generated.

VII. Experiments

A. Dataset

Two datasets are used for comparison on PDF formula detection. One is the public Marmot dataset, the other is a larger dataset collected by us.

1) Marmot dataset

The public Marmot dataset [26] consists of 400 document pages and contains 1574 isolated formulae. The bounding boxes of formulae have been labeled manually. This dataset is fully used as the test set.

2) Our dataset

The Marmot dataset is too small to be used for deep-learning based methods. Besides, the dataset is lack of document pages without formulae. Thus, we collected a larger dataset to train deep networks well.

The dataset consists of more than 1,000 scientific papers collected from *CiteSeer*[1]. We selected more than 12,000 document pages from those papers. The dataset is also various in page layouts and formula styles.

The formulae in the dataset is annotated manually. This dataset contains more than 22,000 formulae. As mentioned in Sec 5.2, the implicit class down-sampling requires the label of other categories. We also labeled texts, figures and tables in the dataset. 80% of the data is used for training the networks, and 20% is used for testing.

B. Evaluation Metrics

Single Match: IOU measurement is used to estimate whether a detected formula is correct or not. Concretely, we calculate the IOUs between a detected formula and all the formulae in ground truth, if one is larger than the *IOU_THRSHOLD*, the detected formula is considered correct and matches the corresponding ground truth, then the ground truth formula is removed to avoid matching again. Considering the importance of formulae integrality, the *IOU_THRSHOLD* is set to 0.8 in the experiments.

Holistic Evaluation: The F1 metric is also used for evaluating the performance of PDF formula detection methods. We first calculate the number of detected formulae ($N_{detecte}$), the number of formulae in ground truths (N_{gt}), and their matching number (N_{Match}). Then the F1 value is calculated as follows:

$$F1 = \frac{2 * N_{match}}{N_{detecte} + N_{gt}} \quad (1)$$

C. Network Combination Results

As mentioned in Section 5, the feature extraction networks combine a CNN and a RNN to simultaneously extract text and visual features. We compare the combined networks with single RNN and CNN model. Table II shows the results.

TABLE. II Recognition Accuracy of Different Models

Model	Accuracy of Different Dataset (%)	
	Marmot	Ours
CNN	93.8	92.4
RNN	92.9	87.5
Combined	**95.2**	**94.7**

As table II shows, the performance of the combined feature extraction networks is better than any of the single RNN or CNN. Fig. 3 shows two samples of the classification results. As shown in Fig. 3(a), the combined networks can recognize formulae which resemble text-line and with less structure information. Besides, in Fig. 3(b), even the character information of this formula is missing, the networks can still correctly recognize it.

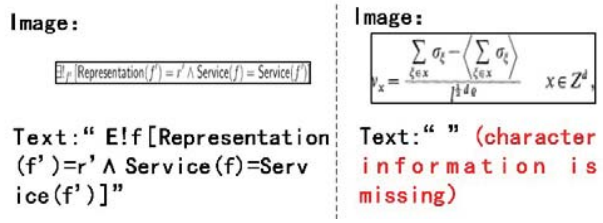
D. Formula detection results

Lin et al [27] compared and summarized several existing formula detection methods on the Marmot dataset. We implement their method and compare it with ours. Besides, to compare with the general deep-learning based object detection methods, the method proposed in the literature [14] with Alexnet network is also conducted. Table III shows the results.

TABLE. III Formula Detection Results

Methods	F1 of Different Dataset (%)	
	Marmot	Ours
Comprehensive[27]	68.1	55.3
Faster RCNN[14]	14.3	11.2
Ours+Only RNN	90.9	82.4
Ours+Only CNN	91.5	85.6
Ours+Combined	**93.4**	**88.7**

[1] http://citeseer.ist.psu.edu/index



(a) A sample which is recognized as background by CNN, but correctly classified by the

(b) A sample which is recognized as background by RNN, but correctly classified by the

Fig. 3. Samples of network conbination results

Traditional methods including [27] mainly rely on the character information. Thus, their performance may be very different in different files. For the object detection method [14], a large number of formula candidates generated by its Region Proposal Networks (RPNs) are not precise enough. This results in that most of its detected formulae have low IOU value. As shown in Table 3, when the IOU_THRSHOLD is 0.8 in the experiments, the F1 of Faster-RCNN is below 15%. However, when the threshold is reduced to 0.6, the best F1 of Faster-RCNN can reach 56%. The combination of CNN and RNN achieve the better performance. The good classification ability also ensures the effect of post-processing. In the correction process, much wrong recombination can be identified by the classifier.

VIII. Conclusion

This paper proposes a deep learning-based method to detect formulae from PDF documents. Comparing with traditional PDF formula detection methods, our system could relieve the influence of imprecise PDF information on formula detection, with the help of the region proposal method, which combines top-down and bottom-up layout analysis, and uses the feature extraction networks on both character information and visual information of formulae in PDF files. The experimental results show that the proposed method outperforms the existing PDF formula detection methods on the public Marmot dataset, as well as a large dataset collected by us. In the future, we will recheck our dataset and release it, and explore formula structure recognition.

Acknowledgement

This work is supported by the projects of National Natural Science Foundation of China (No. 61472014), the Beijing Nova Program (XX2015B010) and the China Postdoctoral Science Foundation (No. 2016M590019), which is also a research achievement of Key Laboratory of Science, Technology and Standard in Press Industry (Key Laboratory of Intelligent Press Media Technology). We also thank the anonymous reviewers for their valuable comments.

References

[1] https://pdfbox.apache.org/index.html

[2] Anderson, Robert H. "Syntax-directed recognition of hand-printed two-dimensional mathematics." Symposium on Interactive Systems for Experimental Applied Mathematics: Proceedings of the Association for Computing Machinery Inc. Symposium. ACM, 1967.

[3] Fateman, Richard J., et al. "Optical Character Recognition and Parsing of Typeset Mathematics1." Journal of Visual Communication and Image Representation 7.1: 2-15, 1996.

[4] Toumit, J-Y, et al. "A hierarchical and recursive model of mathematical expressions for automatic reading of mathematical documents." Document Analysis and Recognition, 1999. ICDAR'99. Proceedings of the Fifth International Conference on. IEEE, 1999.

[5] Garain, Utpal, and B. B. Chaudhuri. "A syntactic approach for processing mathematical expressions in printed documents." Pattern Recognition, 2000. Proceedings. 15th International Conference on. Vol. 4. IEEE, 2000.

[6] Chang, Tzu-Yuan, Yusuke Takiguchi, and Minoru Okada. "Physical structure segmentation with projection profile for mathematic formulae and graphics in academic paper images." ICDAR 2007. Ninth International Conference on. Vol. 2. IEEE, 2007.

[7] Garain, Utpal. "Identification of mathematical expressions in document images." Document Analysis and Recognition, 2009. ICDAR'09. 10th International Conference on. IEEE, 2009.

[8] Kacem, A., Belaïd, A., Ben Ahmed, M.: Automatic extraction of printed mathematical formulae using fuzzy logic and propagation of context. Int. J. Document Anal. Recognit. 4(2), 97–108, 2001.

[9] Inoue, K., Miyazaki, R., Suzuki, M.: Optical recognition of printed mathematical documents. In: Proceedings of the Third Asian Technology Conference on Mathematics, pp. 280–289, 1998.

[10] Chowdhury, S.P., Mandal, S., Das, A.K., Chanda, B.: Automated segmentation of math-zones from document images. In: 7th International Conference on Document Analysis and Recognition (ICDAR), vol. 2, pp. 755–759 (2003)

[11] Liu, Y., Bai, K., Gao, L.: An efficient pre-processing method to identify logical components from PDF documents. Adv. Knowl. Discov. Data Min. pp. 500–511, 2011.

[12] Wang, Yuehan, et al. "WikiMirs 3.0: a hybrid MIR system based on the context, structure and importance of formulae in a document." Proceedings of the JCDL. ACM, 2015.

[13] Deng Y, Kanervisto A, Rush A M. What You Get Is What You See: A Visual Markup Decompiler[J]. arXiv preprint arXiv:1609.04938, 2016.

[14] Ren, S., et al. "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. " IEEE Transactions on Pattern Analysis & Machine Intelligence: 1-1, 2016.

[15] Tang, Duyu, Bing Qin, and Ting Liu. "Document Modeling with Gated Recurrent Neural Network for Sentiment Classification." EMNLP. 2015.

[16] Bahdanau D, Cho K, Bengio Y. Neural machine translation by jointly learning to align and translate[J]. arXiv preprint arXiv:1409.0473, 2014.

[17] Sutskever, Ilya, Oriol Vinyals, and Quoc V. Le. "Sequence to sequence learning with neural networks." Advances in neural information processing systems. 2014.

[18] Sharma, Arjun. "Adapting off-the-shelf CNNs for word spotting & recognition." Document Analysis and Recognition (ICDAR), 2015 13th International Conference on. IEEE, 2015.

[19] Afzal, Muhammad Zeshan, et al. "Deepdocclassifier: Document classification with deep Convolutional Neural Network." ICDAR, 2015 13th International Conference on. IEEE, 2015.

[20] http://www.adobe.com/cn/devnet/pdf/pdf_reference.html

[21] Ha, Jaekyu, et al. "Recursive XY cut using bounding boxes of connected components." Document Analysis and Recognition, 1995., Proceedings of the Third International Conference on. Vol. 2. IEEE, 1995.

[22] https://en.wikipedia.org/wiki/Segment_tree

[23] Lazzara, Guillaume, et al. "The SCRIBO module of the Olena platform: a free software framework for document image analysis." ICDAR, 2011 International Conference on. IEEE, 2011.

[24] Krizhevsky, Alex, I. Sutskever, and G. E. Hinton. "ImageNet Classification with Deep Convolutional Neural Networks." Advances in Neural Information Processing Systems 25.2, 2012.

[25] He, K., et al. "Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. " IEEE Transactions on Pattern Analysis & Machine Intelligence 37.9:1904-16, 2015.

[26] http://www.icst.pku.edu.cn/cpdp/data/marmot_data.html.

[27] Lin, Xiaoyan, et al. "Mathematical formula identification and performance evaluation in PDF documents." International Journal on Document Analysis and Recognition (IJDAR) 17.3 : 239-255, 2014