A Table Detection Method for PDF Documents Based on Convolutional Neural Networks

Leipeng Hao, Liangcai Gao*, Xiaohan Yi, Zhi Tang
Institute of Computer Science and Technology
Peking University
Beijing, China
Email: {haoleipeng, glc, chlxyd, tangzhi}@pku.edu.cn

***Abstract*—Because of the better performance of deep learning on many computer vision tasks, researchers in the area of document analysis and recognition begin to adopt this technique into their work. In this paper, we propose a novel method for table detection in PDF documents based on convolutional neutral networks, one of the most popular deep learning models. In the proposed method, some table-like areas are selected first by some loose rules, and then the convolutional networks are built and refined to determine whether the selected areas are tables or not. Besides, the visual features of table areas are directly extracted and utilized through the convolutional networks, while the non-visual information (e.g. characters, rendering instructions) contained in original PDF documents is also taken into consideration to help achieve better recognition results. The primary experimental results show that the approach is effective in table detection.**

***Keywords*—table detection; convolutional neural networks; deep learning; document analysis**

I. Introduction

Tables are widely used in many domains to present and communicate structured information to human readers since tables enable readers to rapidly search, compare and understand facts and draw conclusions. Hence, automatically detecting tables from documents and extracting the information contained in tables are of significant importance in the field of document recognition and analysis and have attracted a lot of research efforts in the past two decades. This paper focuses on the table detection in PDF documents.

PDF (Portable Document Format) has become more and more popular because of its consistency of presentation between different underlying platforms, screens of handheld devices. However, there is little or no structure information in PDF documents, which makes the information extraction and document understanding a challenging problem even though later PDF supports *tagging*.

As far as table detection is concerned, many researches have been carried out on scanned document images and web pages, but no work can handle all the tables well due to the diversity of table layouts and variety of encodings. The existing methods on table recognition still suffer from some limitations. For example, the image-based methods are prone to fail when directly carried out on PDF pages, while most of rule-based methods are hardly able to recognize the tables without ruling lines or the tables that has complex layouts. More often, (flow) charts with intersected vertical and horizontal lines are usually detected as faked tables in prior methods.

Meanwhile, in the recent years, deep learning techniques have greatly improved the results of many computer vision tasks and information processing work. In order to improve table detection performance and make up for the limitations of prior methods, this paper proposes a method of table detection based on deep learning techniques. In more details, according to the knowledge of the diversity of tables, several table-like areas first are selected by some loose rules. An area is chosen according to the loose rules that if it shares similarity with a table area, even the similarity is a little, which means almost all the table areas are detected and a lot of areas that are prone to be detected as tables such as (flow) charts and matrixes are also collected. Then, this paper adopts and refines the convolutional neural networks, one of the popular deep learning models, as the basic element of networks to determine whether the selected areas are tables or not. In this step, the areas that used to be detected as faked tables are discarded. Meanwhile, in the deep learning model, the non-visual information (e.g. the coordinates of characters and the start point and end point of ruling lines) contained in original PDF documents is extracted and taken advantage of to further improve the performance of table detection. In addition, because training the convolutional networks requires a large number of sample images, this paper collects a dataset with more than 7000 PDF pages containing tables with labeled ground-truth. And the dataset would be publicly available for the researching purpose.

The rest of the paper is organized as follows. Section II summarizes the relevant researches on table recognition, especially those carried out directly on PDF documents. Section III describes the framework of the proposed method in detail, including selecting table-like areas, adopted structure of the convolutional neural networks etc. Section IV introduces the dataset and analyses the experimental results. Conclusion and future works are discussed in Section V.

II. Related work

Table detection has attracted a good number of research efforts so far. Zanibbi *et al.*[1] and Silva *et al.*[2] have given a comprehensive survey of table recognition methods. Most of the early researches on table recognition concentrated on

*Liangcai Gao is the corresponding author

978-1-5090-1792-8/16 $31.00 © 2016 IEEE
DOI 10.1109/DAS.2016.23

image-based documents, such s the T-Recs and T-Recs++ systems proposed by Kieninger *et al.*[3][4]. However, those image-based methods don't perform well directly on PDF documents.

As far as we know, the *pdf2table* system, proposed by Burcu Yildiz *et al.*[5], is the first relevant research that deals with PDF documents directly. Their work is based on the data returned by the *pdftohtml* (http://pdftohtml.sourceforge.net) tool, which returns text chunks and their absolute coordinates in the PDF document. The *pdf2table* consists of two components: table detection and table decomposition. Firstly, it merges text segments into lines. Then it determines a portion of text elements as a table only by means of the knowledge of the absolute coordinates of the text elements. At last, it returns the identified table headers, the spanning behavior of the headers and the assigning of data cells. Oro *et al.*[6] propose a similar table recognition approach which classifies text lines into three categories: text lines, table lines and unknown lines according to the number of segments in the lines. After the classification process, the table lines and unknown lines are combined to form tables. But these methods are all based on the assumption that the pages are of single-column.

Liu *et al.*[7] develop a table search engine system called *TableSeer*. It craws scientific PDF documents online, such as DBLP and CiteSeer, finds out the documents with tables, recognizes the table regions, extracts the table contents and indexes them. Their approach of table detection is mainly based on merging sparse lines which are similar to the methods of Burcu Yildiz[5]. However, *TableSeer*, as a searching system, depends too much on the precision of table detection. What's more, it makes the assumption that all the tables in the PDF documents have a caption, discarding those without a caption and leading to a low recall rate.

The methods described above purely take the content layout features into consideration and are based on the observation that table lines contain more than one text segments. On one hand, text line segmentation, as well as spanning cells detection, is sensitive to predefined thresholds. On the other hand, two or more tables on the same page and irregular tables cannot be handled very well by those methods. As a result, some table regions are under-segmented or over-segmented. From the essence of tables, the graphic ruling lines should also be treated as importantly as content layout to spot table regions. Hassan *et al.*[8] detect tables in PDF documents utilizing both ruling lines and content layout. However, their method utilizes these two sources separately. And lots of false positive tables are detected because the detected graphic lines are not verified first in their method. Similarly, Fang *et al.*[9] propose a method via both visual separators and tabular structures of contents. The separators refer to not only graphic lines but also white spaces to handle unruled tables. This method detects page columns in the first place to assist table detection in multi-column pages, and achieves a satisfactory accuracy rate.

Deep structured learning has emerged as a new area of machine learning. The techniques developed from deep learning have been utilized in a wide range of signal and information processing tasks during the past several years, such as the hand-written characters recognition and picture classification. For example, LeNet, proposed by LeCun *et al.*[10], has greatly improved the performance of hand-written character recognition. Wang *et al.*[11] propose a method based on auto-encoder to recognize handwritten Chinese characters which outperforms traditional methods using hand-crafted features and convolutional neural networks. The table detection method based on deep learning has not been proposed yet. Since deep learning has greatly enhanced the results of many computer vision tasks, this paper attempts an adoption of deep learning on the task of table detection.

III. Proposed Method

The proposed method consists of three main procedures: table-like areas proposal, convolutional neural networks and adding information contained in original PDF documents. This section will discuss each procedure in detail. The work flow of the proposed method is illustrated in Fig. 1.

A. Table-like areas proposal

This method deals with tables in three categories: tables with horizontal and vertical rule lines, tables with horizontal rule lines and tables with no rule lines. The table-like areas proposal procedure begins with examining horizontal and vertical lines on the page from top to bottom. In order to obtain the lines from the PDF, we use PDF parser to generate graphic objects for each PDF instruction, each with its respective coordinates and attributes. For different categories of tables, different strategies are applied.

Tables with horizontal and vertical rule lines. When horizontal lines are crossed (or almost crossed) by vertical lines, this may represent a tabular grid. This method collects all the regions that contain vertical lines intersecting with a set of horizontal lines as candidate table areas. Apparently, some areas that correspond to graphic objects such as (flow) charts and boxes are selected as well. Those non-table areas will be discarded by convolutional neural networks next.

Tables with horizontal rule lines. When horizontal lines delineate rows in a table, the horizontal lines are practically always the entire width of the table, so we attempt to check whether the lines delineate table rows or not. First, this method collects the line objects not contained in the areas that are proposed as horizontally and vertically ruled tables, and then some pre-processing are performed to merge any dotted or touching lines that have been written in parts to the document page. Second, to ensure that the lines represent ruling lines of a table, not the underline of headings, or separators for headers or footers on the page, this method applies the trained convolutional networks to do the checking and generate a boolean matrix, $G[N][N]$, where N denotes the number of the lines. If $G[i][j]$ is true, the region from line i to line j represents a table.

This method intends to search for large integrated table areas between those horizontal lines. In order to settle this problem,

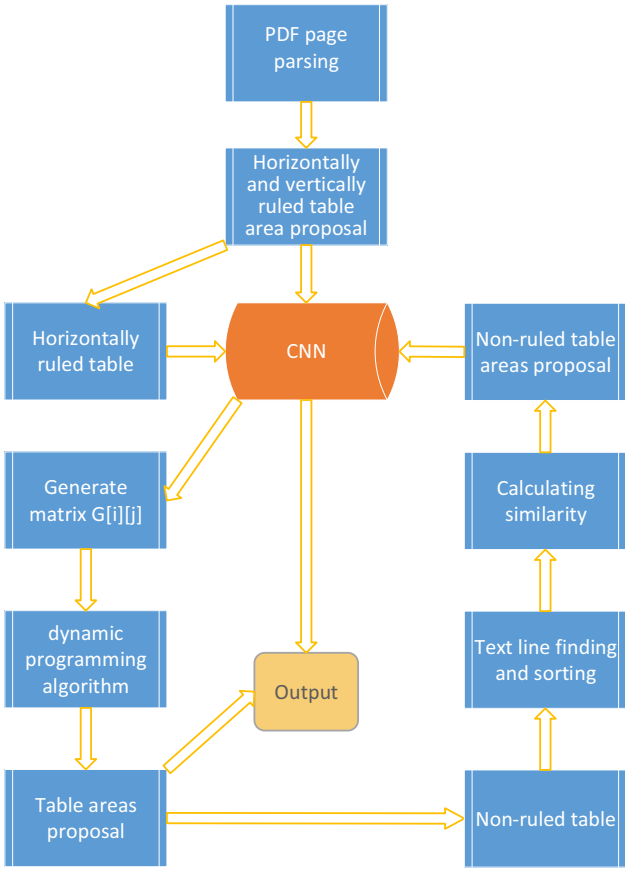

Fig. 1. The work flow of the proposed method

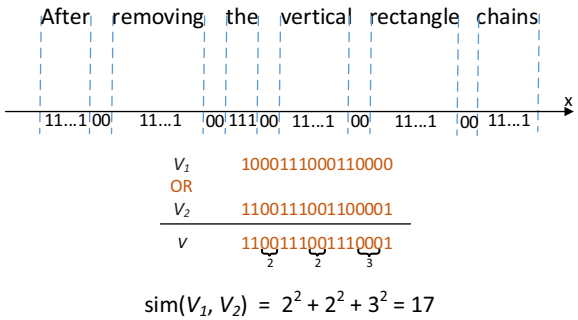

Fig. 2. Example of generating binarized vector and similarity calculation

a dynamic programming algorithm is applied on the horizontal lines. Let $Val(T)$ denote the value of table T which contains m horizontal lines. We define

$$Val(T) \triangleq m^2$$

Apparently, the larger the value of a table is, the larger and more complete the table area is. Let $maxval[i][j]$ denote the max value that if we search for j tables between the first line and the i-th line. Then

$$maxval[i][j] = max(maxval[i-1][j], maxval[k][j-1] + (i-k)^2) \quad (if \quad G[i][k] == true)$$

At the same time, this method uses $route[i][j]$ to mark which the value of $maxval[i][j]$ is from. At the end of algorithm, this method selects the max value from $maxval[N][1]$ to $maxval[N][N]$ and calculates the table areas with horizontal rule lines in the document page according to matrix $route[i][j]$.

Tables with no rule lines. Given the fact that all the tables with no rule lines in our pre-training dataset are within one single column and non-ruled tables are scarce, this method only deals with the non-ruled tables within one column and this step begins with column detection and separation.

As shown in Fig. 1, a line-finding algorithm is carried out in every column. The horizontal text lines in a table area are practically share the same alignment. Furthermore, the gaps in a horizontal table line are much larger than that in a text line of paragraphs. To detect the table area with no rule lines, for text each line, this method calculates its vertical projection and generates a binarized vector. The n-th bit of the vector is 1 if the projection of the corresponding text line is not 0 in x-coordinate. Fig. 2 shows how to generate a binarized vector corresponding to a text line.

The text lines are sorted according to their vertical position from top to bottom. Then a measure of similarity is calculated between adjacent text lines. Fig. 2 also shows the process of calculating the similarity. First, a vector is generated by calculating *or* operation bit by bit between two text line vectors. Second, this method counts the lengths of consecutive 0 in the new vector. The measure of similarity is the sum of the square of the lengths counted in last step.

Apparently, the similarity between table text lines from a same table is much larger because they have the same alignment and larger overlapped gap in the horizontal direction, while the similarity between paragraph text lines is smaller. After the calculation of text line similarity, this method selects the consecutive text lines that has roughly equal and larger similarity as non-ruled table areas.

The candidate table areas of all the three categories are put as input into the adopted and refined convolutional neural networks, a judgement will be made at the output of convolutional networks whether the input area is a table or not.

B. Convolutional neural networks

Convolutional Networks are trainable multi-stage architectures. The input and output of each stage are sets of arrays called feature maps[10]. Each feature map at the output represents a particular feature extracted through a specific filter on the input. The feature maps in higher stages are usually smaller than that in lower stages, representing more abstract

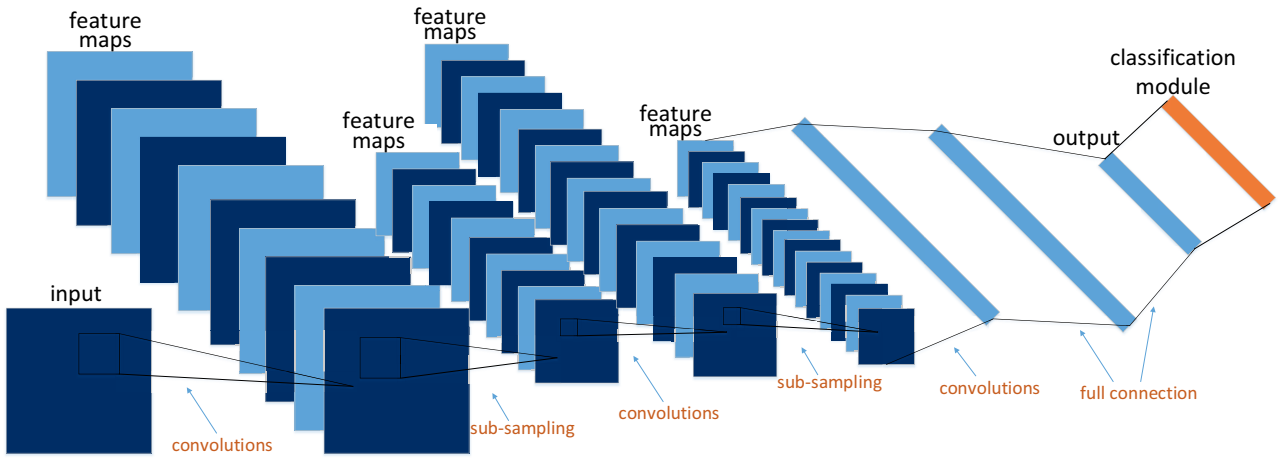

Fig. 4. The architecture of the convolutional neural networks

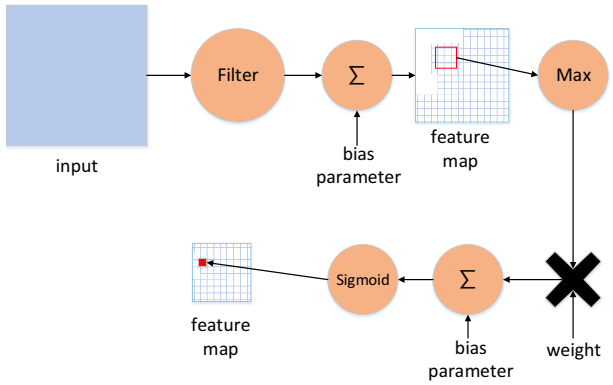

Fig. 3. The operation process of 3-layer stage in convolutional networks

features. Basically, each stage is composed of three layers: a filter bank layer, a non-linearity layer and a feature pooling layer. Fig. 3 shows the detailed operations on a feature map in such 3-layer stage. The architecture is similar to LeNet by LeCun[10] but with some improvements.

A typical convolutional networks consist of one or more such 3-layer stages mentioned above, followed by more than one fully-connected layers and a classification module. The architecture of the convolutional networks adopted in this paper is illustrated in Fig. 4, which consists of 7 layers: three convolutional layers, two sub-sampling layers, and tow fully-connected layers.

Each input image is down-sampled to $76 \times 76 \times 3$. The first convolutional layer consists of 100 trainable filter kernels of size 5×5 with stride of 1, followed by a max-pooling layer with sampling stride of size 2. The second convolutional layer has 200 filter kernels of size 7×7 with stride of 1, followed by a max-pooling layer with sampling stride of 3. The third convolutional layer consists of 300 filters of size 5×5 with stride of 1, not followed by a sub-sampling layer. The fifth and sixth layer are fully-connected layers with 1024 and 2048 nodes respectively.

The convolution operation is formulated as

$$y_{Kj}^{l} = f\left(\left(\sum_{M_i} x_i^{l-1} * K_{ij}^{l}\right) + b_j^{l}\right)$$

where y_{Kj}^{l} is the j-th convolutional result of the j-th layer and x_i^{l-1} is the i-th output feature map of $(l-1)$-th layer. K_{ij}^{l} is the convolutional kernel between i-th input feature map and the j-th output feature map of l-th layer. b_j^{l} is the bias of j-the output layer. $*$ denotes the convolutional operation and $f()$ denotes the activation function. This method takes the *hyperbolic tangent* as the activation function. So the convolution operation is actually expressed as

$$y_{Kj}^{l} = \tanh\left(\left(\sum_{M_i} x_i^{l-1} * K_{ij}^{l}\right) + b_j^{l}\right)$$

The max-pooling function is

$$y_j^{l} = \max_{k \in (s \times s)} \left\{x_{j(k)}^{l}\right\}$$

where y_j^{l} is the pooling output of the j-th feature map of the l-th layer. $s \times s$ denotes the max pooling region. Thus the max pooling output is the maximum value of convolution map in the region.

The fully connected layer takes the function

$$y_j^{l} = \sum_i y_i^{l-1} \cdot \omega_{i,j}^{l} + b_j^{l}$$

where y_j^{l} denotes the j-th node in the l-th layer, $\omega_{i,j}^{l}$ denotes the weights between y_j^{l-1} and y_j^{l} , and b_j^{l} is the bias of the l-th layer.

The proposed method takes SoftMax as the classification module. The networks are then trained under a log loss (or cross-entropy) regime, giving a non-linear variant of multinomial logistic regression. Since the function maps a vector and a specific index i to a real value, the derivative needs to take the index into account:

$$\frac{\partial}{\partial q_k}(q,i) = \sigma(q,i)(\delta_{ik} - \sigma(q,k))$$

Here, the Kronecker delta is used for simplicity. The process of training is terminated when the loss function converges.

C. Adding information of original PDF Documents

The convolutional networks take the image of the proposed areas as input. There is much information contained in the original PDF documents that could also be utilized, such as the coordinates of the characters and the start point and end point of a line in tables. In this paper, those features are extracted from PDF pages and transformed into vectors in order to improve the results further.

PDF documents are described by low-level structural objects such as a group of characters, lines, curves, images, etc., and associated style attributes such as font, color, stroke, fill, and shape, etc.[12]. To parse those low-level objects, this paper utilizes the PDF parser provided by Founder Corporation, which is developed according to the PDF specification[13]. Both text objects and graph objects are parsed. For text objects, the character attributes such as font and the bounding-box of the characters are taken into consideration. This method also takes the graphic objects including drawing and clipping instructions into account.

Based on the architecture shown in Fig. 4, there are two layers we can put the information vector in: the input layer and the output layer. First, this paper transforms the text and graphic objects features extracted from the proposed area in the PDF page into line vectors. Second, the line vector can be put into either the input layer or the output layer of convolutional networks. For the input layer, enlarge the line vector to the same dimension of input image, and catenate it to the edge of the image; for the output layer, catenate the vector to the end of output vector of convolutional networks after normalization and the enlarged vector will be connected to the classification module. This paper attempts both approaches mentioned above and adding the vector into the input layer achieves better results.

In summary, the framework of the proposed method is as follows: First, this method trains the convolutional neural networks with the training dataset, and selects some table-like areas based on the loose rules. Then the trained convolution networks tests the collected areas and makes a judgement whether the area is a table or not, after which some post processing is carried out to output the detection results of table area on a document page. Furthermore, the information contained in the PDF documents are extracted and added to the input layer and output layer of convolutional networks respectively to improve the performance.

TABLE I
EXPERIMENTAL RESULTS AND COMPARISON WITH OTHER METHODS

Participant methods	Precision	Recall	**F1-measure**
proposed method	0.9846	0.8366	**0.9046**
proposed method(X)	0.9724	0.9215	**0.9463**
Silva	0.9292	0.9831	**0.9554**
Nitro	0.9397	0.9323	**0.9360**
Nurminen	0.9210	0.9077	**0.9143**
Yildiz	0.6399	0.8530	**0.7313**
Stoffel	0.7536	0.6991	**0.7253**

IV. EXPERIMENT AND RESULT

A. Dataset

This paper collects over 7000 pages of PDF documents that contain tables and labels the table areas. The pages are from both English and Chinese e-Books, conferences and journals. The dataset shows good variety in table styles, from horizontally and vertically ruled tables to horizontally ruled and non-ruled tables, from horizontal tables to vertical tables, from inside-column tables to span-column tables. We choose 5000 table areas randomly and convert them to images to train the convolutional networks. Another 5000 images of other objects of the document pages (e.g. paragraphs, charts, matrixes) are selected as well. Both the table area and non-table area images are cut out from the pages manually and the 10000 images constitute the training dataset.

The test dataset this method uses is the dataset of table competition of ICDAR 2013[14]. It consists of 156 tables overall. Non-ruled tables, tables with complex header structures and small tables, with fewer than five rows are very common in this dataset, which used to cause difficulties for most of the proposed methods.

B. Results and analysis

The proposed method is compared with some academic participant methods of the table competition of ICDAR 2013. Their experimental results are shown in literature[14].

This paper use three performance metrics: precision (the percentage of the objects that are in fact true), recall (the percentage of the true objects that the method finds) and F1-measure. The experimental results are shown in Table I. Fig. 5 shows some detected table areas examples of the proposed method. As shown in Table I (proposed method(X) means the results after adding the information extracted from original PDF pages into the input layer of the convolutional networks), the proposed method surpasses most of the academic participant methods of table competition of ICDAR 2013 in F1-measure metric. Unlike some other methods, our method is not adapted specifically for the competition dataset. Also, the recall and F1-measure metrics are all notably improved after adding the information extracted from PDF pages, indicating the information of original PDF document makes a good contribution. The proposed method is fairly effective, while the examples illustrated in Fig. 5 also shows some limitations of this method: some selected and detected as true table areas include extra region that belongs to other object in the page.

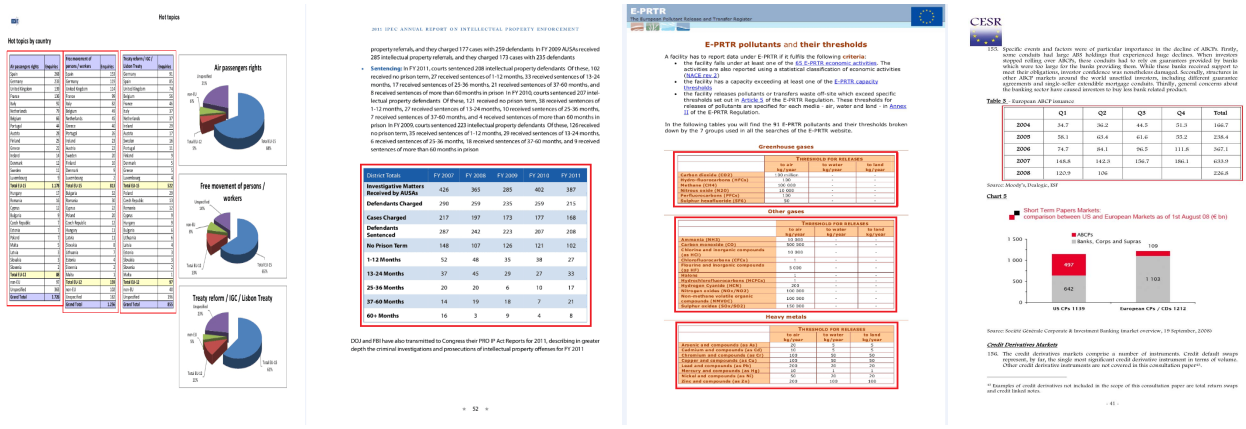

Fig. 5. Experimental example illustrations

In fact, the loose rules this method uses to collect table-like areas may cause three limitations: (a) proposing some areas that contains intersected lines such as figures and (flow) charts; (b) merging horizontally ruled tables that are in the same page and quite close to each other; (c) the selected area contains extra region that belongs to other objects in the page. As described in Section IV, the convolutional neural networks can discard the areas of (a). The proposed method applies effective post process (by defining the value of table and dynamic programming algorithm) to separate the tables in (b). However, the propose method still suffers from limitation (c). We will deal with it in future work.

V. Conclusion

This paper proposes a table detection method by combining loose rules to collect table-like areas and convolutional neural networks to determine whether the chosen areas are table or not. Experimental results show that the proposed method is effective, and the information contained in original PDF pages makes a good contribution to the performance, indicating that the information is valuable and cannot be ignored.

The proposed method still suffers from some limitations such as the detected table area contains extra region that belongs other objects in the page which will be dealt with next. Also, table structure recognition and understanding will be carried out in the future.

Acknowledgment

This work is supported by the projects of National Natural Science Foundation of China (No. 61573028), the Natural Science Foundation of Beijing (No. 4142023) and the Beijing Nova Program(XX2015B010). We also thank the anonymous reviewers for their valuable comments.

References

[1] T. M. Breuel, "Two geometric algorithms for layout analysis," in *Document analysis systems v*. Springer, 2002, pp. 188–199.

[2] A. C. e Silva, A. M. Jorge, and L. Torgo, "Design of an end-to-end method to extract information from tables," *International Journal of Document Analysis and Recognition (IJDAR)*, vol. 8, no. 2-3, pp. 144–171, 2006.

[3] T. Kieninger and A. Dengel, "A paper-to-html table converting system," in *Proceedings of Document Analysis Systems (DAS)*, vol. 98, 1998.

[4] ——, "Applying the t-recs table recognition system to the business letter domain," in *Document Analysis and Recognition, 2001. Proceedings. Sixth International Conference on*. IEEE, 2001, pp. 518–522.

[5] B. Yildiz, K. Kaiser, and S. Miksch, "pdf2table: A method to extract table information from pdf files," in *IICAI*, 2005, pp. 1773–1785.

[6] E. Oro and M. Ruffolo, "Pdf-trex: An approach for recognizing and extracting tables from pdf documents," in *Document Analysis and Recognition, 2009. ICDAR'09. 10th International Conference on*. IEEE, 2009, pp. 906–910.

[7] Y. Liu, K. Bai, P. Mitra, and C. L. Giles, "Tableseer: automatic table metadata extraction and searching in digital libraries," in *Proceedings of the 7th ACM/IEEE-CS joint conference on Digital libraries*. ACM, 2007, pp. 91–100.

[8] T. Hassan and R. Baumgartner, "Table recognition and understanding from pdf files," in *Document Analysis and Recognition, 2007. ICDAR 2007. Ninth International Conference on*, vol. 2. IEEE, 2007, pp. 1143–1147.

[9] J. Fang, L. Gao, K. Bai, R. Qiu, X. Tao, and Z. Tang, "A table detection method for multipage pdf documents via visual seperators and tabular structures," in *Document Analysis and Recognition (ICDAR), 2011 International Conference on*. IEEE, 2011, pp. 779–783.

[10] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[11] M. Wang, Y. Chen, and X. Wang, "Recognition of handwritten characters in chinese legal amounts by stacked autoencoders," in *Pattern Recognition (ICPR), 2014 22nd International Conference on*. IEEE, 2014, pp. 3002–3007.

[12] J. Fang, X. Tao, Z. Tang, R. Qiu, and Y. Liu, "Dataset, ground-truth and performance metrics for table detection evaluation," in *Document Analysis Systems (DAS), 2012 10th IAPR International Workshop on*. IEEE, 2012, pp. 445–449.

[13] PDF Reference 1.7.

[14] R. Girshick, J. Donahue, T. Darrell, and J.Malik, "ICDAR 2013 Table Competition," in *Document Analysis and Recognition (ICDAR), 2013 12th International Conference on*. IEEE, 2013, pp. 1449–1453.