A Clustering Approach Combining Lines and Text Detection for Table Extraction

Karima Boutalbi[1,2], Visar Sylejmani[2], Pierre Dardouillet[1,2], Olivier Le Van[2], Kave Salamatian[1], Hervé Verjus[1], Faiza Loukil[1], and David Telisson[1]

[1] Savoie-Mont-Blanc University, Annecy, France
{firstname.lastname}@univ-smb.fr
[2] Cegedim-SRH, Lyon, France {firstname.lastname}@cegedim-srh.com

Abstract. Table detection is a crucial step in several document analysis applications as tables are used to present essential information to the reader in a structured manner. In companies that deal with a large amount of data, administrative documents must be processed with reasonable accuracy, and the detection and interpretation of tables are crucial. Table recognition has gained interest in document image analysis, particularly in unconstrained formats (absence of rule lines, unknown information of rows and columns). This problem is challenging due to the variety of table layouts, encoding techniques, and the similarity of tabular regions with non-tabular document elements. In this, paper, we make use of the location, context, and content type, thus it is purely a structure perception approach, not dependent on the language and the quality of the text reading. We evaluate our model on invoice-like documents and the proposed method showed good results for the task of table extraction.

Keywords: Table detection · Table extraction· Document analysis · Table recognition · PDF accessibility

1 Introduction

Many non-editable documents are shared in PDF (Portable Document Format). They are typically not accompanied by tags for annotating the page layout, including the position of the table. One of the major challenges for the analysis and understanding of such documents is how to extract tables from them. Table extraction as a part of table understanding includes two stages: table detection, i.e. recovering the bounding box of a table in a document, and table data extraction by structure recognition, i. e. recovering its rows, columns, and cells.

In our work, we deal with payslip documents, which consist of a header, the principal table (body of the document) that we want to extract, and a footer. Fig.1 represents three templates of payslips with different characteristics. We observe that all tables are separated by columns but there are no horizontal row lines; in addition, the header and footer may also contain tables or have a similarity of tabular regions with non-tabular document elements.

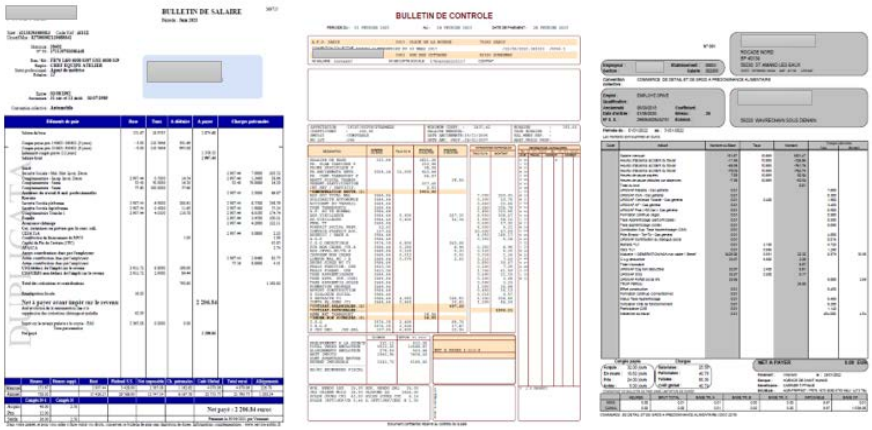


Fig. 1: Examples of payslip templates.

We first detect all the content of the document including the table content, and then we define the boundary of the table as a filter that separates the table content from the header and the footer of the document, assuming that a table and its content have certain properties related to vertical lines and text position.

Thus, rather than converting PDFs to images or HTML and then processing with other methods (e.g., OCR), we propose a method that combines lines and text detection for table extraction. Our approach is purely a perception approach, we make use of the location of the text, not its context and its signification. The preprocessing stage is crucial to the effectiveness of our method; we then perform a clustering algorithm for column classification. In the final stage, the distinctive size and succession of rectangles around the items allow us to separate the table from the rest of the document.

Considering the above contributions, this paper is organized as follows. Section 2 discusses existing papers that have dealt with table detection. Section 3 describes the steps of the adopted method. Section 4 presents the study results. Finally, Section 5 concludes the paper and presents some enhancements.

2 Related work

Many works have been achieved in document table detection. Mikhailov et al. [2] proposed a novel approach for table detection in untagged PDF documents. They first use deep neural networks to predict some table candidates, then they select probable tables from the candidates by verifying their graph representation. Thus, they build a weighted directed graph from text blocks inside a predicted area of a table. Riba et al. [5] proposed a graph-based approach for detecting tables in document images by using Graph Neural Networks (GNNs) to describe the local repetitive structural information of tables in invoice documents. Authors in [1] demonstrate the performance improvement of the proposed approach

for detecting tabular regions from document images. They use Deep learning-based Faster R-CNN, assuming that tables tend to contain more numerical data and hence it applies color coding/coloration as a signal for separate apart numerical and textual data. In [4], the authors tackle the problem of table detection by proposing a bi-modular approach based on the structural information of tables. This structural information includes bounding lines, row/column separators, and space between columns.

Another popular method for extracting tables is proposed in [3] where authors focus on extracting information from tables with various layouts, maintaining their structure, and processing them into textual data from images.

3 Methodology

We have collected 16 payslips templates, with a variety of table structures.We focus on the detection and extraction of data from the table. We use text boxes and columns to define the table.

Definition 1 (text box): A text box is a rectangle surrounding a string, and the position of the box is defined by four points (x_1, y_1, x_2, y_2).

Definition 2 (column): a column is a vertical line surrounding the content of a table and it's defined by four points (x_1,y_1, x_2, y_2) where $x_1 = x_2$.

We can further suppose that every item i is defined with a pair $< b$,$e>$of a start point b of the item and an end point$e$ of the item.

Definition 3 (cluster): Let's S be a set of tokens (text boxes or columns). S is a cluster if $\forall t_k \in S\ \exists t'_k \in S$ such that: $\forall t''_k \notin S | center(\, t_k) - center(\, t'_k) \,| \leq |center(\, t_k - t''_k) \,|$.

As clustering algorithms, we use K-means and DBSCAN (Density-Based Spatial Clustering of Applications with Noise) algorithms which both are based on Euclidean distance.

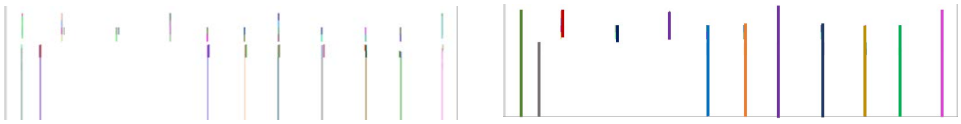
K-means algorithm is an iterative algorithm that splits a dataset into K predefined number of distinct non-overlapping subgroups (clusters) where each data point belongs to only one group. DBSCAN finds core samples of high density and expands clusters from them. It is useful for data that contains clusters of similar density. Also, it is able to find arbitrary-shaped clusters and clusters with noise.

3.1 Data processing

We start by extracting all vertical lines from the document. Due to the encoding techniques, some lines appear to be continuous, but there is a serie of overlapping and non-continuous lines. Also, some lines that are very close to each other appear visually to belong to the same line as shown in Fig.2a. We group together all those lines that visually appear to belong to the same vertical line by using the rules we have defined; 2b illustrates the vertical lines after data preprocessing. Another way to regroup the series of non-continuous lines is vertical line clustering but we rather use a rule-based method since the vertical lines will be our input data to the proposed approach.

3.2 Line detection-based approach

A simple first approach for extracting tables is to find fixed columns surrounding text. We assume that each column belongs to a cluster. Let S be a set of columns' table that we want to extract. We wonder if we can extract tables only by using vertical lines. Since we assume that each column should be separated enough to belong to a different cluster. According to Definition 3, we can partition the entire set of vertical lines into clusters as shown in Fig.2b. The clustering was performed with K-means and DBSCAN, as presented in Definition 2. A column is defined by a start point $y - 1$ and an endpoint y_2. A vertical line is a line that their $x_1 = x_2$. Thus, the input of the clustering algorithm will be a one-dimensional vector where each instance is x_1.



Before regrouping After regrouping

Fig. 2: Vetical lines processing.

3.3 Proposed approach: Clustering using lines and text box detection

The first step of our method consists in detecting a unique position of the text box. Since a text box surrounds a string of characters or numerical values, separated by more than one space. So we did a preprocessing to define all page boxes. Each item is surrounded by a rectangular box that will be our reference for the position of textual data. We extend all boxes according to the right and left surroundings' vertical lines as illustrated in Fig.3, thus all boxes belonging to the same columns will have the same size. Fig.4 shows an example of a payslip after box extension.

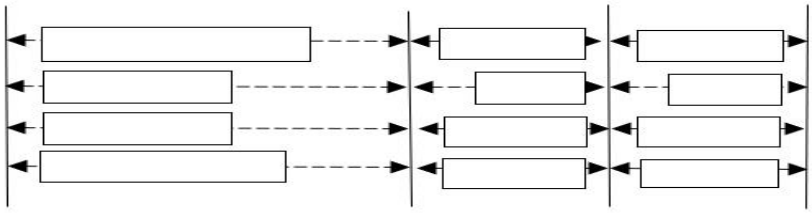


Fig. 3: An illustration of text boxes extension.

After extending boxes to vertical lines each box belonging to the same column will have the same size; the goal of this step is to obtain vertically aligned boxes.

Thus, we perform clustering on all aligned boxes. The input of the clustering algorithm is 2-dimensional data (x_1 and x_2) corresponding to the beginning and the end of each box.

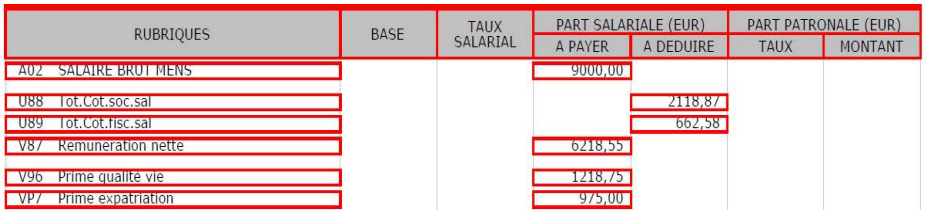


Fig. 4: Example of payslip after text boxes extension.

4 Results

We compare the two approaches, the clustering of vertical lines and the second one combining lines and text position. To evaluate our method we used two measures, namely *WRCTC* (well-recognized columns over the total number of columns) and *WPTT* (well-partitioned tables over the total number of tables).

WRCTC: represents the rate of the well-recognized columns over the total number of columns per table.

WPTT: represents the rate of well-partitioned tables over the total number of tables.

We observe that K-means is more efficient with the first approach even if is more time-consuming due to the number of cluster research in terms of the number of columns recognition with 93% and this method aims to recognize 75% of the tables in the collected documents. In the second approach, we observe that we obtain almost the same result using K-means, but the results are considered improved in terms of the number of column recognition with 97% and 81% of well table extraction. In addition, with the proposed approach, the time consumption is very low compared with our other experiments and it takes 0.0032 seconds for table extraction per document. Fig.5 shows an example of obtained results using the proposed approach, we can see that the partitioning of columns is well defined, thus we can separate the table from the page's header and footer on one hand, on the other hand, it allows us to extract the table's columns and cells.

Table 1: Obtained results comparison.

Method	**Algoritm**	**WRCTC**	**WPTT**	**Time(S)**
Line	K-means	0.93	0.75	0.4313
Line	DBSCAN	0.54	0.25	0.0030
Proposed method	K-means	0.94	0.75	1.1688
Proposed method	DBSCAN	0.97	0.81	0.0032

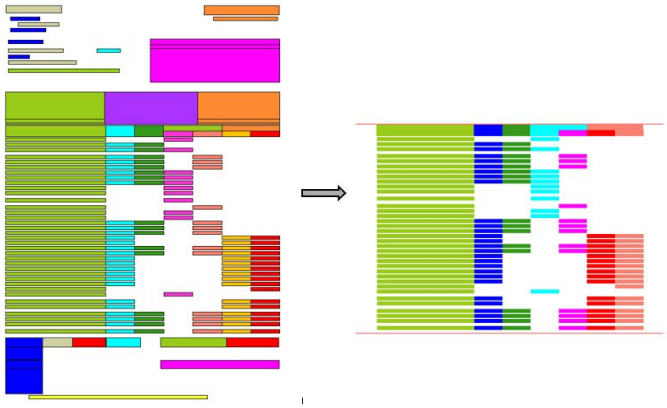


Fig. 5: Result example of the proposed table extraction approach.

5 Conclusion

We have proposed a method for table extraction combining vertical lines extracted from PDF documents and text box location. Our method does not use the context or the text content of the document, we only make use of the location. Our approach has been tested on real data of our company that need to extract information from payslips' tables. This will be useful for a lot of applications, such as deducing the calculation modes, triggers remuneration, frequencies, etc. Our problem is difficult due to the similarity of tabular regions with non-tabular document elements. However, we have obtained an accuracy of 81% over 16 extremely different payslip templates.

References

1. Arif, S., Shafait, F.: Table detection in document images using foreground and background features. In: 2018 Digital Image Computing: Techniques and Applications (DICTA). pp. 1–8 (2018). https://doi.org/10.1109/DICTA.2018.8615795
2. Mikhailov, A., Shigarov, A., Rozhkov, E., Cherepanov, I.: On graph-based verification for pdf table detection. In: 2020 Ivannikov Ispras Open Conference (ISPRAS). pp. 91–95 (2020). https://doi.org/10.1109/ISPRAS51486.2020.00020
3. Nidhi, Saluja, K., Mahajan, A., Jadhav, A., Aggarwal, N., Chaurasia, D., Ghosh, D.: Table detection and extraction using opencv and novel optimization methods. In: 2021 International Conference on Computational Performance Evaluation (ComPE). pp. 755–760 (2021). https://doi.org/10.1109/ComPE53109.2021.9752204
4. Ranka, V., Patil, S., Patni, S., Raut, T., Mehrotra, K., Gupta, M.K.: Automatic table detection and retention from scanned document images via analysis of structural information. In: 2017 Fourth International Conference on Image Information Processing (ICIIP). pp. 1–6 (2017). https://doi.org/10.1109/ICIIP.2017.8313719
5. Riba, P., Dutta, A., Goldmann, L., Fornés, A., Ramos, O., Lladós, J.: Table detection in invoice documents by graph neural networks. In: 2019 International Conference on Document Analysis and Recognition (ICDAR). pp. 122–127 (2019). https://doi.org/10.1109/ICDAR.2019.00028