

PdfTable: A Unified Toolkit for Deep Learning-Based Table Extraction

Lei Sheng^{1*} and Shuai-Shuai Xu²

¹* Automated institute, Wuhan University of Technology, 122 Luoshi Road, Wuhan, 430070, Hubei, China.

²School of Software, University of Science and Technology of China, No.96, JinZhai Road Baohe District, Hefei, 230026, Anhui, China.

*Corresponding author(s). E-mail(s): xuanfeng1992@whut.edu.cn;
Contributing authors: sa517432@mail.ustc.edu.cn;

Abstract

Currently, a substantial volume of document data exists in an unstructured format, encompassing Portable Document Format (PDF) files and images. Extracting information from these documents presents formidable challenges due to diverse table styles, complex forms, and the inclusion of different languages. Several open-source toolkits, such as Camelot, Plumb a PDF (pdfnumber), and Paddle Paddle Structure V2 (PP-StructureV2), have been developed to facilitate table extraction from PDFs or images. However, each toolkit has its limitations. Camelot and pdfnumber can solely extract tables from digital PDFs and cannot handle image-based PDFs and pictures. On the other hand, PP-StructureV2 can comprehensively extract image-based PDFs and tables from pictures. Nevertheless, it lacks the ability to differentiate between diverse application scenarios, such as wired tables and wireless tables, digital PDFs, and image-based PDFs. To address these issues, we have introduced the PDF table extraction (PdfTable) toolkit. This toolkit integrates numerous open-source models, including seven table recognition models, four Optical character recognition (OCR) recognition tools, and three layout analysis models. By refining the PDF table extraction process, PdfTable achieves adaptability across various application scenarios. We substantiate the efficacy of the PdfTable toolkit through verification on a self-labeled wired table dataset and the open-source wireless Publicly Table Reconnition Dataset (PubTabNet). The PdfTable code will available on Github: https://github.com/CycloneBoy/pdf_table.

Keywords: Intelligent document analysis, Table structure recognition, Portable Document Format file table extraction, Information Extraction

1 Introduction

Portable Document Format (PDF¹), a file format used to present documents in a hardware- and software-independent manner. It finds widespread application across various domains, including academic papers and financial report documents. With the rapid development of document digitization, the automated extraction of information from PDFs has gained paramount significance. Consequently, several tools have emerged to facilitate the conversion of PDFs into easily parseable HTML formats. However, the intricate structures and diverse styles of tables, coupled with the potential inclusion of different languages in table contents, present persistent challenges in table structure recognition(TSR) during the parsing of PDF documents. In response to this challenge, diverse methods have been proposed to address the complexities associated with TSR.

Several toolkits are available to directly extract tables from PDFs, including Tabula², Camelot³ and pdfnumber⁴. Tabula, Camelot, and pdfnumber primarily employ rule-based methods for table extraction in digital PDFs, demonstrating inaccuracies when confronted with tables featuring complex cross-line and cross-column styles. With the rapid development of deep learning, early researchers proposed models such as DeepDeSRT[1] ,TableNet[2] and SEM[3] to address table extraction challenges in image-based documents. However, due to a scarcity of extensively annotated datasets, the outcomes were less than satisfactory. Recent years have witnessed the introduction of diverse TSR datasets, such as SciTSR[4], TableBank[5], PubtabNet[6] , PubTables-1M[7], and WTW[8]. Models like CascadeTabNet[9], EDD[6], LGPMA[10], TSRFormer[11], Cycle-CenterNet[8], LORE[12], etc., trained on these datasets have demonstrated proficient table parsing results. Despite successful parsing in generalized table scenarios, these models encounter challenges when applied to real-world scenarios.

While the existing table parsing algorithm performs admirably, there remains a deficiency in open-source tools designed for end-to-end PDF table extraction to address diverse table extraction tasks in practical applications. Baidu’s recent open source PP-StructureV2[13] toolkit, employing the SLANet table structure recognition model in conjunction with PaddleOCR[14], has garnered widespread user appreciation for achieving end-to-end table recognition and extraction. Nevertheless, there are notable areas for optimization: (1) the end-to-end table extract process lacks sufficient subdivision, such as the differentiation between wired and wireless tables, and the extraction of text from digital PDFs versus image-based PDFs; (2) Each functional module in the recognition process supports a limited number of models, for instance: two layout analysis models, three table recognition models and one OCR text recognition model; (3) Open source table recognition models commonly employ different frameworks and dependent environments, posing challenges for debugging and reproducibility within a unified environment.

¹<https://en.wikipedia.org/wiki/PDF>

²<https://github.com/tabulapdf/tabula>

³<https://github.com/camelot-dev/camelot>

⁴<https://github.com/jsvine/pdfplumber>

In addressing the aforementioned challenges, we present a novel end-to-end PDF extraction table toolkit called PdfTable. Initially, we partition the table recognition process into distinct modules, including data preprocessing, layout analysis, table structure recognition, table content extraction and upper-layer application. Then different open source algorithms and toolkits are integrated for different modules. Diverse open-source algorithms and toolkits are integrated for each module, with uniform coding implemented in Pytorch[15] to streamline debugging and model integration. Presently, the toolkit encompasses seven table structure recognition algorithms, three layout analysis algorithms, and four mainstream OCR recognition tools. Subsequently, we conduct end-to-end integration and optimization of the table recognition process, ultimately enabling the batch conversion of both digital and scanned PDF documents into HTML or WORD formats. PdfTable facilitates the direct extraction of PDF tables into Excel and supports numerous languages. To validate the toolkit's effectiveness, we annotated a small table dataset within the Chinese financial domain, comprising both digital and scanned PDFs. PdfTable demonstrated commendable performance on this dataset, affirming the efficacy of the toolkit. Concurrently, we evaluated the integration of four wireless table models on the PubtabNet[6] wireless table dataset, with results attesting to the correctness of the model integration.

In summary, our primary contributions can be outlined as follows:

1. Introduction of PdfTable, an end-to-end deep learning-based PDF table extraction toolkit, supporting the extraction of tables from both digital and scanned PDFs, encompassing wired and wireless table extraction.
2. Integration of numerous open-source algorithms into our toolkit, encompassing seven table structure recognition models, four mainstream OCR recognition tools, and three layout analysis models. This integration provides users with a straightforward and user-friendly API.
3. Conducted experiments to validate the efficacy of the PdfTable toolkit, utilizing a self-labeled small Chinese financial field wired table dataset and the wireless table dataset PubtabNet[6]. The experimental results unequivocally demonstrate the effectiveness and correctness of our toolkit.

2 Related Work

2.1 Document Layout Analysis

Document layout analysis is a basic pre-processing task for modern document understanding and digitization. It mainly divides documents into different regions, such as pictures, tables, text and formulas, which can be regarded as a sub-task of object detection. Presently mainstream methods include object detection-based models, segmentation-based models and GNN-based methods[16]. DeepDeSRT[1] pioneered the use of Faster R-CNN[17] for table detection, achieving commendable results. With more and more layout analysis datasets ,such as: PubLayNet[18], TableBank[5], etc. and different object detection models such as Mask R-CNN[19], YOLO[20], DETR[21], etc. proposed, the task of layout analysis has been further developed.

Layout-parser[22] is a unified toolkit for document image analysis based on deep learning, providing rich pre-trained models and user-friendly APIs. PP-StructureV2[13] also provides a variety of English and Chinese layout analysis models trained on PP-YOLOv2[23] and PP-PicoDet[24] models.

2.2 Table Structure Recognition

Historically, early approaches to table recognition primarily employed rule-based and statistical machine learning methods, often limited by their dependence on the rigid rectangular layout of tables. Consequently, these methods could only effectively handle straightforward table structures or tables embedded in PDFs. In recent years, the landscape has shifted towards deep learning-based methods, demonstrating substantial improvements in accuracy compared to traditional approaches. Broadly categorized, these contemporary methods fall into three main groups: boundary extraction-based methods, image-to-markup generated methods, and graph-based methods.

Boundary extraction based methods These methods employ object detection or semantic segmentation algorithms to initially identify the rows and columns of the table. Subsequently, the cells of the table are determined through cross-combination of the identified rows and columns. DeepDeSRT[1] and TableNet[2] leverage Fully Convolutional Network (FCN)-based semantic segmentation model for TSR analysis. However, the basic FCN faces challenges in accurately recognizing numerous blank tables due to its limited receptive field. To address this limitation, subsequent researchers proposed enhancements [3, 9, 25]. SEM[3] stands out by integrating visual and textual information through three independent modules—splitter, embedder, and merge—enabling the extraction of both simple and complex tables. RobusTabNet[11] proposed a new method of splitting and merging TSR using spatial Convolutional Neural Network (CNN) module, which can effectively identify tables with a large number of blanks and distortions.

Image-to-markup generation based methods These methods transform the table recognition task into an image-to-markup generation task, directly generating markup (HTML or LaTeX) to represent the table structure. Leveraging a substantial volume of labeled table data extracted from existing PDFs, web pages, or LaTeX papers through rules or semi-supervised methods, researchers have proposed many benchmark datasets TABLE2LATEX[26], Tablebank[5], PubtabNet[6]. Additionally, they have organized related competitions, including ICDAR2019[26], ICDAR2021[27], significantly fostering the rapid development of TSR. TableMaster[28] directly predicts HTML and text box regression based on MASTER[29], achieving the best results on the PubtabNet benchmark dataset. SLANet[13] uses PP-LCNet[30] and a series of optimization strategies to make model inference efficient on the CPU. MTL-TabNet[31] proposes an end-to-end TSR model that uses a multi-task learning method to directly solve table structure recognition and table content recognition with one model. OTSL[32] proposes a novel method for marking tables, utilizing only five tokens to represent the table structure, thereby reducing the inference time of the Image2seq method by approximately half while enhancing model accuracy. Only five tokens can

be used to represent the table structure, which can shorten the inference time of the Image2seq method by about half while improving model accuracy.

Graph based methods These methods treat table cells or cell contents as nodes in a graph, employing graph neural networks(GNN) to predict whether these nodes belong to the same group. GraphTSR[4] takes table cells as input, and then uses GNN to predict the relationship between table cells to predict the table structure, achieving good results on the SciTSR[4] data set. TGRNet[33] proposes an end-to-end table graph reconstruction network to perform table structure recognition by simultaneously predicting the physical and logical positions of table cells.

2.3 Optical Character Recognition

Table content recognition is also a crucial phase in the table recognition process. Tables in digital PDFs can directly read text coordinates and content, while scanned PDFs usually require an OCR model to extract text. OCR is currently divided into two primary tasks: text detection and text recognition, each optimized independently. Additionally, there are also end-to-end recognition models. Since OCR has a wide range of applications, it has received widespread attention from researchers and industries, leading to the proposal of numerous models. Notably, the DB[34] detection model and CRNN[35] recognition model stand out as widely adopted combinations. Several readily available open-source toolkits (such as PaddleOCR[14], EasyOCR⁵, TesseractOCR⁶, MMOCR⁷, and duguangOCR⁸) and commercial APIs (Amazon Texttract⁹, Google Document ai¹⁰, BaiDu OCR¹¹) offer fundamental OCR capabilities. The majority of these open-source toolkits provide the latest OCR algorithms and pre-trained models, facilitating convenient direct use or fine-tuning. Due to the distinct nature of document OCR, it is readily identifiable, and existing open-source toolkits can effectively fulfill most requirements.

2.4 PDF To HTML

The conversion of PDF to machine-readable HTML format holds significant implications. For instance, it can enhance accessibility for individuals who are blind or visually impaired[36] and contribute to the improved retrieval and dissemination of academic papers[37, 38]. While several off-the-shelf systems exist for direct PDF-to-HTML conversion, they exhibit limitations[39–41]. Notably, [37, 38] lacks table parsing functionality, converting PDF tables into images for display only. [42] relies on hand-designed rules for table extraction, demonstrating poor generalization. TableParser[41] is a model trained based on a weakly supervised dataset constructed from spreadsheets and cannot parse wireless tables or deformed tables. Pdf2htmlEX¹² exclusively converts digital PDFs, failing to convert tables into HTML format. The end-to-end

⁵<https://github.com/JaideAI/EasyOCR>

⁶<https://github.com/tesseract-ocr/tesseract>

⁷<https://github.com/open-mmlab/mmocr>

⁸<https://github.com/AlibabaResearch/AdvancedLiterateMachinery>

⁹<https://aws.amazon.com/texttract/>

¹⁰<https://cloud.google.com/document-ai>

¹¹<https://ai.baidu.com/tech/ocr>

¹²<https://github.com/pdf2htmlEX/pdf2htmlEX>

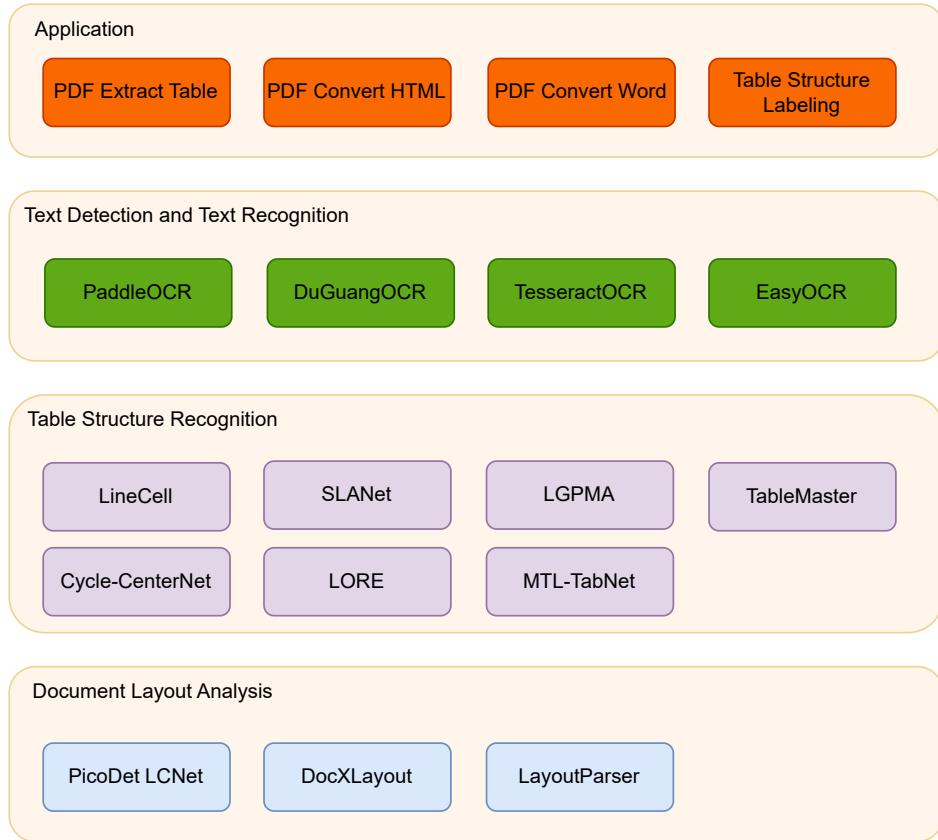


Fig. 1 System overview of PdfTable

model proposed by Nougat[43], based on a visual Transformer, excels in converting academic papers into LaTeX format. However, its end-to-end nature necessitates data collection for retraining when dealing with PDFs in different languages or structures, imposing certain limitations on its versatility. Despite the effectiveness of these systems in specific scenarios, a unified PDF-to-HTML conversion tool supporting diverse languages and document types remains elusive.

3 Design and implementation of PdfTable library

3.1 System Overview

The system overview of PdfTable is illustrated in Figure 1. The core of the entire system is to provide table parsing related algorithms, which is mainly composed of four modules. The layout analysis module locates tables and images; the table structure recognition module parses table structures; text detection and recognitionc module identifies textual content; the application module primarily handles the conversion of

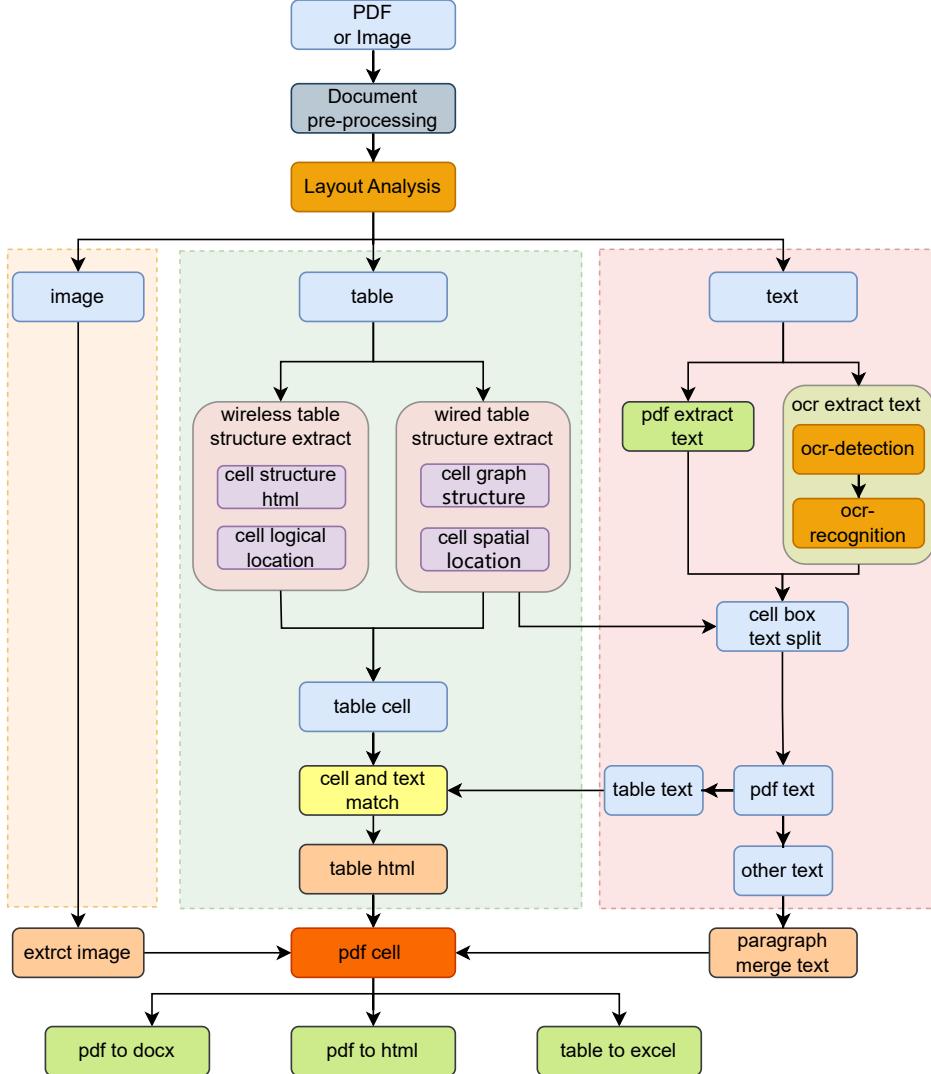


Fig. 2 table processing pipeline

all recognition results into various types. We have standardized the algorithm interface for each module, allowing flexible switching based on the model name to facilitate user utilization. Since different algorithms rely on different environments and frameworks, we use the Pytorch[15] framework to reconstruct part of the model, eliminating unnecessary dependency packages.

3.2 PdfTable Parse Pipeline

The framework of table recognition is illustrated in Figure 2. Firstly, the input image or PDF document is preprocessed through the document preprocessing module,

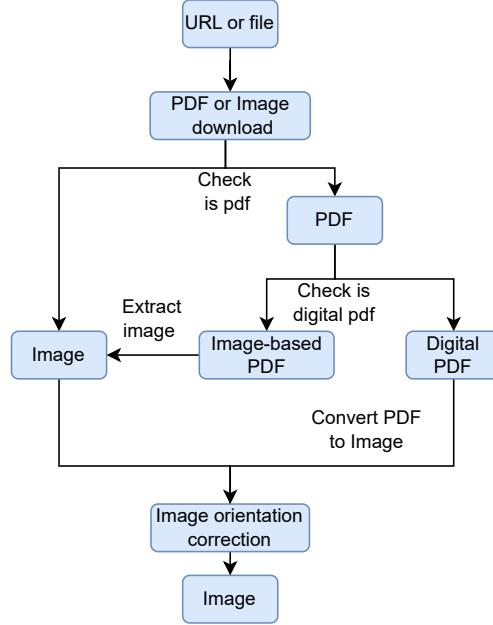


Fig. 3 table preprocess

such as network file download, PDF conversion to image, image orientation correction, etc. Subsequently, the layout analysis module divides the image into distinct regions (e.g., pictures, tables, and text) to facilitate subsequent individual processing. The image area is sent to the image extraction module for extraction. The table area distinguishes whether it is a wired table or a wireless table through rules, and then extracts the table structure through the TSR algorithm. For the text area, extraction is performed based on the document type. In the case of digital PDFs, text is directly extracted from the PDF, while OCR is employed for scanned PDFs or images to identify the corresponding text. Next, the text in the table area is matched with the table structure to generate table HTML, and other text is consolidated into paragraphs through the paragraph merging module. Ultimately, the recognized pictures, tables, and text paragraphs are output into distinct files according to the specified output format requirements.

3.3 Module Design

3.3.1 Input preprocessing

The input preprocessing module primarily preprocesses PDFs or images to facilitate extraction of subsequent algorithm models. The processing flow chart is shown in Figure 3. Initially, the determination is made regarding the necessity of downloading the input file. If the input file is a PDF, it is requisite to split the PDF file into individual pages and convert them into images. For digital PDF, the Ghostscript¹³ tool

¹³<https://www.ghostscript.com/>

is used to convert it into images, whereas for image PDFs, direct image extraction is conducted. Since the current document processing algorithm mainly processes documents with a 0-degree orientation, the extraction outcome for rotated document information is suboptimal. Consequently, the orientation of the input document must be rectified before the subsequent processing stages. The image orientation correction module incorporates the document orientation classification algorithm ¹⁴ (output categories: 0,90,180,270) and the text orientation classification algorithm (output categories: 0,180) to execute rotation correction based on the document orientation (rotation directions: 0,90,180,270). Concurrently, rules are applied for small-angle rotation correction (rotation angle: -45 - 45 degrees) on documents tilted at a slight angle, ultimately aligning the image for processing to roughly 0 degrees. The pre-processing module has significantly enhanced our recognition efficacy on irregular documents.

3.3.2 Layout analysis module

The task of layout analysis is to divide the areas in the document image according to categories (e.g.,text, images, tables, formulas). Currently, mainstream object detection-based models have demonstrated commendable performance across various benchmark datasets. In PdfTable, we have incorporated two lightweight layout analysis models, PP-picodet[24] and DocxLayout¹⁵, and the LayoutParser[22] toolkit. PP-picodet is a lightweight target detection backbone model based on PaddleDetection¹⁶, and ppstructure[13] extends its capabilities to Chinese and English layout analysis models and table detection models. We convert them into a pytorch model. DocxLayout is a layout analysis model based on the DLA-34[44] backbone network provided by Alibaba Research. The LayoutParser[22] toolkit integrates a variety of layout analysis models based on different datasets. To enhance usability, we provide a standardized interface for invoking different models.

3.3.3 Table Structure Recognition

Table borders are usually used for visual display of table structures and can also be used as an important basis for identifying table structures. The current mainstream method divides tables into two categories (wired tables and wireless tables), and designs TSR algorithms for processing different types of tables. In the TSR processing flow of PdfTable, an initial rule-based method is employed to categorize tables as either wired or wireless, followed by the application of specific algorithms to identify each table type. PdfTable currently integrates seven TSR algorithms, offering flexibility in configuration and utilization.

Wired Table Since wired tables have the obvious feature of borders, algorithms can be used to directly identify the borders of the table and then restore the table structure through post-processing. Traditional methods effectively handle most straightforward table scenes. We refer to camelot¹⁷ and Multi-TypeTD-TSR[45] to

¹⁴<https://www.ghostscript.com/>

¹⁵<https://github.com/AlibabaResearch/AdvancedLiterateMachinery/tree/main/DocumentUnderstanding/DocXLayout>

¹⁶<https://github.com/PaddlePaddle/PaddleDetection>

¹⁷<https://github.com/camelot-dev/camelot>

implement the LineCell algorithm for extracting table cells based on OpenCV[46]. Firstly, we extract horizontal and vertical line segments, table areas, and intersections of line segments through a series of operations such as binarization, erosion, expansion, and contour search. Then we use line segment intersections and line segments to construct table cells. Finally, we use line segment relationships to merge across rows and columns cells. Despite their efficacy in simple scenes, traditional methods exhibit limited generalization due to their dependence on manually set rules. In contrast, contemporary approaches leverage deep learning techniques to identify table edges or cells. Therefore, PdfTable also integrates two latest TSR algorithms, CycleCenterNet[8] and LORE[12]. They adopt different methods to simultaneously predict the logical structure and physical structure of table cells, and then restore the structure of the table through simple post-processing operations, which can identify wired tables in real-world scenarios.

Wireless Table Wireless tables distinguished by the absence of table borders, present a more challenging identification task compared to wired tables. Presently, various methods employ image-to-sequence generation techniques, directly generating tags and text borders to represent the table structure. Subsequently, the table is reconstructed by aligning table cells with their respective content positions. We implemented four such algorithms: SLANet[13], LGPMA[10], TableMaster[28], and MTL-TabNet[31]. The LORE[12] algorithm, while initially designed for recognizing wired tables, exhibits the capability to recognize wireless tables as well. This is achieved by predicting both the logical structure of the table and the physical borders of the table text.

3.3.4 Text Extraction

In order to completely restore the table, it is also necessary to extract the content in the table cells (mainly text). The process of extracting table content in PdfTable is illustrated in the right part of Figure 2, which is mainly divided into PDF text extraction and OCR text extraction. For digital PDF sources, the existing toolkit pdfminer.six¹⁸ is utilized to directly extract text coordinates and content; otherwise, an OCR toolkit is employed. To accommodate multiple languages and diverse business scenarios, PdfTable integrates several mainstream OCR toolkits, including PaddleOCR[14], EasyOCR¹⁹, TesseractOCR²⁰ and duguangOCR²¹. Combined with the table structure extracted previously, we split the text boxes across cells, and then match them with the table structure to generate the final table HTML. Text outside the table is merged into paragraphs to facilitate subsequent processing.

3.3.5 Application

To address diverse application scenarios, we summarize the extracted table structure, text content and images and uniformly represent them into a PdfCell structure with coordinate positions and content. This approach facilitates the generation of

¹⁸<https://github.com/pdfminer/pdfminer.six>

¹⁹<https://github.com/JaideAI/EasyOCR>

²⁰<https://github.com/tesseract-ocr/tesseract>

²¹<https://github.com/AlibabaResearch/AdvancedLiterateMachinery>

diverse output formats. Currently, the applications implemented in PdfTable include: PDF to HTML, PDF to DOCX, and table to Excel. In the future, we will implement more different applications.

4 Experiments

The primary objective of PdfTable is to streamline the extraction of tables from diverse PDF formats. The extraction of tables from PDFs poses challenges in practical applications due to variations in PDF types (digital or image-based), table categories (wired or wireless), and the presence of text in multiple languages (English, Chinese or other languages). A singular model proves insufficient for accommodating all business scenarios. PdfTable overcomes this limitation by integrating multiple models and allowing the selection of appropriate models for combined extraction based on distinct business types. Given that the extraction effectiveness of PdfTable depends on the chosen models and specific application contexts, a comprehensive evaluation is challenging. Initial assessments were conducted on a common application scenario involving Chinese wired tables to validate the effectiveness of the PdfTable toolkit. Additionally, for wireless table recognition, an evaluation on the PubTabNet[6] dataset was undertaken to verify the correctness of the integrated TSR algorithm.

Table 1 Statistics of the datasets that we use in experiments.

Test Sets	Page	Table
Digital PDF	2,589	3,709
Image-based PDF	2,192	2,956
PubTabNet	9,115	9,115

4.1 Datasets and evaluation metrics

To assess PdfTable’s capability in extracting wired tables, we curated a dataset comprising Chinese financial documents tables, encompassing both digital and image-based PDFs. This dataset comprises 4,781 pages and encompasses 6,665 tables. For the evaluation of wireless table extraction, we utilized the validation set from the extensively employed PubTabNet[6] dataset. The details of the data set are shown in Table 1.

We employ metrics such as Accuracy, Precision, Recall, F1-scores and TEDS-Struct[12, 13, 47] for evaluating table structure recognition. A table is deemed correctly recognized in Precision calculation when all its cells are accurately identified. TEDS-Struct is a modified variant of the tree edit distance-based similarity (TEDS)[6] metric, which disregards the text content within table cells and exclusively evaluates the table structure.

Table 2 Performance on the financial reporting dataset.

Dataset	Methods	Precision (%)	Recall (%)	F1 (%)	TEDS-Struct (%)
Digital PDF	LineCell	98.5	98.2	98.4	99.5
	LORE[12]	90.5	87.7	89.1	97.2
	LORE*[12]	95.2	93.2	94.2	98.4
Image-based PDF	LineCell	83.9	84.7	84.2	94.7
	LORE[12]	80.5	77.1	78.9	92.8
	LORE*[12]	86.3	83.4	84.8	95.3

Best results are in **bold**. The “*” indicates that the model first uses the layout analysis model to identify the table area, and then identifies the table structure separately in the table area. No “*” means that the model directly recognizes the table structure of the entire PDF image.

4.2 Experimental results

Wired table result Table 2 presents the evaluation outcomes for the Chinese financial documents table dataset. We choose the LineCell model implemented in this paper and the latest LORE[12] model for comparison. Additionally, we investigated the impact of employing the layout analysis model for table area identification in the LORE[12] model. Analyzing the experimental results, we can find:

Table 3 Compare with state-of-the-art methods on PubTabNet dataset.

Methods	Acc (%)	TEDS (%)	TEDS-Struct (%)	Inference time(ms)	Model Size(M)
TableMaster[28]	77.90	96.12	-	2144	253
TableMaster*[28]	78.60	-	97.56	2764	260
LGPMA[10]	65.74	94.70	96.70	-	177
LGPMA*[10]	65.30	-	96.68	345	177
SLANet[13]	76.31	95.89	97.01	766	9.2
SLANet*[13]	76.03	-	97.33	798	9.2
MTL-TabNet[31]	-	96.67	97.88	-	289
MTL-TabNet*[31]	79.10	-	98.48	4520	289

“*” denotes the results of our assessment. Model size refers to the actual physical size of the model. Regarding the inference time of some models, we quote from the SLANet[13] paper.

The evaluation metrics exhibit superior performance on digital PDFs compared to image-based PDFs, with the F1-score demonstrating an 11.2% increase on digital PDFs. This suggests that table extraction is more challenging in image-based PDFs, highlighting substantial room for improvement. The LineCell model outperforms the LORE[12] model by 4.2% in F1-score on digital PDFs, while registering a 0.6% lower F1-score on image-based PDFs. This indicates that the traditional LineCell model still has certain advantages in identifying wired tables within PDFs. Notably, the LineCell model achieves an F1-score of 98.4% on digital PDFs and 84.2% on image-based PDFs, showcasing its effective identification of wired tables in PDF documents.

By comparing the results of whether the LORE[12] model first uses the layout analysis model, it is evident that employing the layout analysis model before table structure recognition enhances the F1 score by 5.5% and the TEDS-Struct score by 1.85%. This underscores the effectiveness of incorporating the layout analysis model for predictive processing, resulting in a notable improvement in table recognition accuracy.

Wireless table result The PdfTable toolkit incorporates diverse models for wireless table structure recognition. To assess the accuracy of algorithm integration, we conducted evaluations on the [6] dataset. The experimental results are shown in Table 3. It can be found from the experimental results:

By comparing the Acc and TEDS-Struct metric of the four models, the maximum difference between our evaluation results and the original paper results is 0.7%, falling within the acceptable margin of error. This preliminary validation underscores the accuracy of the algorithm integration.

From the perspective of inference speed and Acc metric, SLANet[13] exhibits distinct advantages compared to other models. It achieves an Acc metric of 76% with an average inference time of 798 ms. TableMaster[28] and MTL-TabNet[31] can attain higher Acc, their average inference times are considerably slower. Notably, the MTL-TabNet[31] model achieves the best results, but the average inference time is as high as 4520 ms.

The LORE[12] model can also support wireless table recognition, and the TEDS metric reaches 98.1% on the PubTabNet[6] dataset. However, there are currently problems with the integration in PdfTable, and the experimental results have not been entirely replicated. Future optimizations are planned.

4.3 Qualitative Assessment

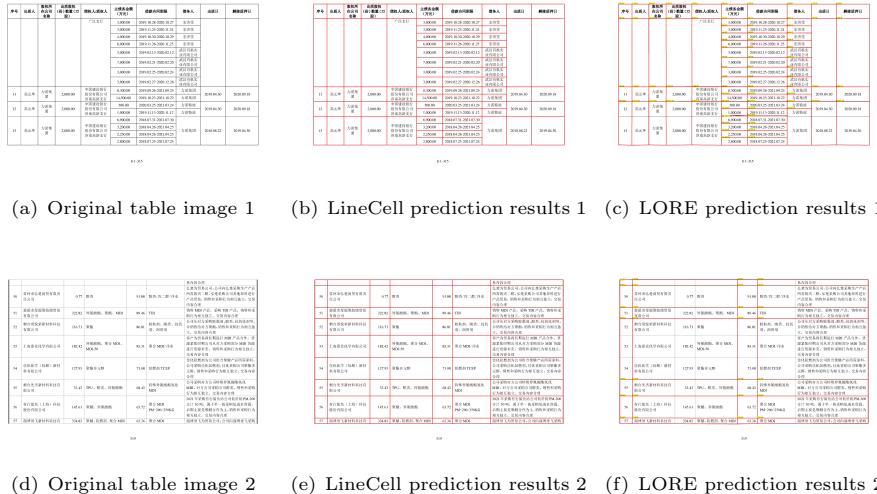


Fig. 4 Qualitative results of LineCell and LORE on digital PDF. The red border represents the identified cell.

The qualitative results in Figure 4 show that in some cases, the LORE[12] model predicts that some cells in the table cannot be accurately identified, whereas LineCell can accurately identify all cells.

5 Conclusion and future work

In this paper, we introduce a novel end-to-end PDF table extraction toolkit, PdfTable, designed for seamless table extraction from both digital and image-based PDFs. The toolkit integrates various existing models, including those for layout analysis, table structure recognition, OCR detection, and OCR recognition. This integration allows for flexible combinations to adapt to diverse application scenarios. To validate the efficacy of the PdfTable toolkit, we annotated a small dataset of wired tables. Concurrently, we evaluated the wireless table recognition model on the PubTabNet[6] dataset, confirming the accuracy of the algorithm integration. In the future, we will optimize this toolkit from the following aspects: 1. Developing new algorithms to differentiate wired tables from wireless tables; 2. Incorporating the ability to fine-tune integrated models, such as table recognition models; 3. Enhancing the toolkit's capacity to recognize wired tables in image-based PDFs.

6 Declarations

Conflict of interest The authors have no conflicts of interest to declare that are relevant to the content of this article.

Ethics approval This article has never been submitted to more than one journal for simultaneous consideration. This article is original.

Data Availability The datasets analysed during the current study are available in the <https://github.com/ibm-aur-nlp/PubTabNet>.

Code availability Code and data used in this paper are publicly available at https://github.com/CycloneBoy/pdf_table.

References

- [1] Schreiber, S., Agne, S., Wolf, I., Dengel, A., Ahmed, S.: Deepdesrt: Deep learning for detection and structure recognition of tables in document images. In: 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), vol. 01, pp. 1162–1167 (2017). <https://doi.org/10.1109/ICDAR.2017.192>
- [2] Paliwal, S., D, V., Rahul, R., Sharma, M., Vig, L.: TableNet: Deep Learning model for end-to-end Table detection and Tabular data extraction from Scanned Document Images. arXiv. <https://doi.org/10.48550/arXiv.2001.01469> . <http://arxiv.org/abs/2001.01469> Accessed 2023-05-23
- [3] Zhang, Z., Zhang, J., Du, J.: Split, embed and merge: An accurate table structure recognizer. arXiv. <http://arxiv.org/abs/2107.05214> Accessed 2023-07-28

- [4] Chi, Z., Huang, H., Xu, H.-D., Yu, H., Yin, W., Mao, X.-L.: Complicated table structure recognition. arXiv preprint arXiv:1908.04729 (2019)
- [5] Li, M., Cui, L., Huang, S., Wei, F., Zhou, M., Li, Z.: Tablebank: Table benchmark for image-based table detection and recognition. In: Proceedings of the Twelfth Language Resources and Evaluation Conference, pp. 1918–1925 (2020)
- [6] Zhong, X., ShafieiBavani, E., Jimeno Yepes, A.: Image-based table recognition: Data, model, and evaluation. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) Computer Vision – ECCV 2020, pp. 564–580. Springer, Cham (2020)
- [7] Smock, B., Pesala, R., Abraham, R.: Pubtables-1m: Towards comprehensive table extraction from unstructured documents. 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 4624–4632 (2021)
- [8] Long, R., Wang, W., Xue, N., Gao, F., Yang, Z., Wang, Y., Xia, G.-S.: Parsing table structures in the wild. In: 2021 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 924–932 (2021). <https://doi.org/10.1109/ICCV48922.2021.00098>
- [9] Prasad, D., Gadpal, A., Kapadni, K., Visave, M., Sultanpure, K.: CascadeTabNet: An approach for end to end table detection and structure recognition from image-based documents. arXiv. version: 2. <https://doi.org/10.48550/arXiv.2004.12629> . <http://arxiv.org/abs/2004.12629> Accessed 2023-08-21
- [10] Qiao, L., Li, Z., Cheng, Z., Zhang, P., Pu, S., Niu, Y., Ren, W., Tan, W., Wu, F.: Lgpma: Complicated table structure recognition with local and global pyramid mask alignment. In: Lladós, J., Lopresti, D., Uchida, S. (eds.) Document Analysis and Recognition – ICDAR 2021, pp. 99–114. Springer, Cham (2021)
- [11] Ma, C., Lin, W., Sun, L., Huo, Q.: Robust table detection and structure recognition from heterogeneous document images. Pattern Recognition **133**, 109006 (2023) <https://doi.org/10.1016/j.patcog.2022.109006>
- [12] Xing, H., Gao, F., Long, R., Bu, J., Zheng, Q., Li, L., Yao, C., Yu, Z.: LORE: logical location regression network for table structure recognition. In: Williams, B., Chen, Y., Neville, J. (eds.) Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023, pp. 2992–3000. AAAI Press, ??? (2023). <https://doi.org/10.1609/aaai.v37i3.25402> . <https://doi.org/10.1609/aaai.v37i3.25402>
- [13] Li, C., Guo, R., Zhou, J., An, M., Du, Y., Zhu, L., Liu, Y., Hu, X., Yu, D.: Pp-structurev2: A stronger document analysis system. arXiv preprint arXiv:2210.05391 (2022)

- [14] Du, Y., Li, C., Guo, R., Yin, X., Liu, W., Zhou, J., Bai, Y., Yu, Z., Yang, Y., Dang, Q., et al.: Pp-ocr: A practical ultra lightweight ocr system. arXiv preprint arXiv:2009.09941 (2020)
- [15] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al.: Pytorch: An imperative style, high-performance deep learning library. Advances in neural information processing systems **32** (2019)
- [16] Wei, S., Xu, N.: PARAGRAPH2GRAPH: A GNN-based framework for layout paragraph analysis. arXiv. <http://arxiv.org/abs/2304.11810> Accessed 2023-08-17
- [17] Girshick, R.: Fast r-cnn. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1440–1448 (2015)
- [18] Zhong, X., Tang, J., Jimeno Yepes, A.: Publaynet: Largest dataset ever for document layout analysis. In: 2019 International Conference on Document Analysis and Recognition (ICDAR), pp. 1015–1022 (2019). <https://doi.org/10.1109/ICDAR.2019.00166>
- [19] He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2961–2969 (2017)
- [20] Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 779–788 (2016)
- [21] Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: European Conference on Computer Vision, pp. 213–229 (2020). Springer
- [22] Shen, Z., Zhang, R., Dell, M., Lee, B.C.G., Carlson, J., Li, W.: Layoutparser: A unified toolkit for deep learning based document image analysis. In: Lladós, J., Lopresti, D., Uchida, S. (eds.) Document Analysis and Recognition – ICDAR 2021, pp. 131–146. Springer, Cham (2021)
- [23] Authors, P.: PaddleDetection, Object detection and instance segmentation toolkit based on PaddlePaddle. <https://github.com/PaddlePaddle/PaddleDetection> (2019)
- [24] Yu, G., Chang, Q., Lv, W., Xu, C., Cui, C., Ji, W., Dang, Q., Deng, K., Wang, G., Du, Y., et al.: Pp-picodet: A better real-time object detector on mobile devices. arXiv preprint arXiv:2111.00902 (2021)
- [25] Siddiqui, S.A., Fateh, I.A., Rizvi, S.T.R., Dengel, A., Ahmed, S.: Deeptabstr: Deep learning based table structure recognition. In: 2019 International Conference on Document Analysis and Recognition (ICDAR), pp. 1403–1409 (2019). <https://doi.org/10.1109/ICDAR48324.2019.9010700>

<https://doi.org/10.1109/ICDAR.2019.00226>

- [26] Deng, Y., Rosenberg, D., Mann, G.: Challenges in end-to-end neural scientific table recognition. In: 2019 International Conference on Document Analysis and Recognition (ICDAR), pp. 894–901 (2019). <https://doi.org/10.1109/ICDAR.2019.00148>
- [27] ICDAR 2021 Competition on Scientific Literature Parsing. <https://arxiv.org/abs/2106.14616> Accessed 2023-08-27
- [28] Ye, J., Qi, X., He, Y., Chen, Y., Gu, D., Gao, P., Xiao, R.: PingAn-VCGroup's Solution for ICDAR 2021 Competition on Scientific Literature Parsing Task B: Table Recognition to HTML (2021)
- [29] Lu, N., Yu, W., Qi, X., Chen, Y., Gong, P., Xiao, R., Bai, X.: Master: Multi-aspect non-local network for scene text recognition. Pattern Recognition **117**, 107980 (2021) <https://doi.org/10.1016/j.patcog.2021.107980>
- [30] Cui, C., Gao, T., Wei, S., Du, Y., Guo, R., Dong, S., Lu, B., Zhou, Y., Lv, X., Liu, Q., Hu, X., Yu, D., Ma, Y.: PP-LCNet: A Lightweight CPU Convolutional Neural Network. <https://arxiv.org/abs/2109.15099v1> Accessed 2023-09-27
- [31] Ly, N.T., Takasu, A.: An end-to-end multi-task learning model for image-based table recognition, 626–634 (2023) <https://doi.org/10.5220/0011685000003417>
- [32] Lysak, M., Nassar, A., Livathinos, N., Auer, C., Staar, P.: Optimized table tokenization for table structure recognition. In: Fink, G.A., Jain, R., Kise, K., Zanibbi, R. (eds.) Document Analysis and Recognition - ICDAR 2023, pp. 37–50. Springer, Cham (2023)
- [33] Xue, W., Yu, B., Wang, W., Tao, D., Li, Q.: Tgrnet: A table graph reconstruction network for table structure recognition. In: 2021 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 1275–1284. IEEE Computer Society, Los Alamitos, CA, USA (2021). <https://doi.org/10.1109/ICCV48922.2021.00133>. <https://doi.ieeecomputersociety.org/10.1109/ICCV48922.2021.00133>
- [34] Liao, M., Wan, Z., Yao, C., Chen, K., Bai, X.: Real-time scene text detection with differentiable binarization. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 11474–11481 (2020)
- [35] Shi, B., Bai, X., 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)
- [36] Fayyaz, N., Khusro, S., Imranuddin: Enhancing accessibility for the blind and visually impaired: Presenting semantic information in PDF tables **35**(7), 101617

<https://doi.org/10.1016/j.jksuci.2023.101617> . Accessed 2023-07-18

- [37] Wang, L.L., Cachola, I., Bragg, J., Cheng, E.Y.-Y., Haupt, C., Latzke, M., Kuehl, B., Zuylen, M.N., Wagner, L., Weld, D.: SciA11y: Converting scientific papers to accessible HTML. In: Proceedings of the 23rd International ACM SIGACCESS Conference on Computers and Accessibility. ASSETS '21, pp. 1–4. Association for Computing Machinery. <https://doi.org/10.1145/3441852.3476545> . <https://dl.acm.org/doi/10.1145/3441852.3476545> Accessed 2023-05-10
- [38] Ahuja, A.: Analyzing and navigating electronic theses and dissertations. Accepted: 2023-07-22T08:00:15Z Artwork Medium: ETD Interview Medium: ETD Publisher: Virginia Tech. Accessed 2023-07-30
- [39] Shigarov, A., Altaev, A., Mikhailov, A., Paramonov, V., Cherkashin, E.: TabbyPDF: Web-based system for PDF table extraction. In: Damaševičius, R., Vasiljević, G. (eds.) Information and Software Technologies. Communications in Computer and Information Science, pp. 257–269. Springer. https://doi.org/10.1007/978-3-319-99972-2_20
- [40] PR, N., Krishnamoorthy, H., Srivatsan, K., Goyal, A., Santhiappan, S.: DEXTER: An end-to-end system to extract table contents from electronic medical health documents. arXiv. <http://arxiv.org/abs/2207.06823> Accessed 2023-09-18
- [41] Rao, S.X., Rausch, J., Egger, P., Zhang, C.: TableParser: Automatic Table Parsing with Weak Supervision from Spreadsheets. arXiv. version: 1. <https://doi.org/10.48550/arXiv.2201.01654> . <http://arxiv.org/abs/2201.01654> Accessed 2023-08-30
- [42] Namysl, M., Esser, A.M., Behnke, S., Köhler, J.: Flexible Table Recognition and Semantic Interpretation System. arXiv. <http://arxiv.org/abs/2105.11879> Accessed 2023-08-03
- [43] Blecher, L., Cucurull, G., Scialom, T., Stojnic, R.: Nougat: Neural Optical Understanding for Academic Documents. arXiv. <https://doi.org/10.48550/arXiv.2308.13418> . <http://arxiv.org/abs/2308.13418> Accessed 2023-08-31
- [44] Yu, F., Wang, D., Shelhamer, E., Darrell, T.: Deep layer aggregation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2403–2412 (2018)
- [45] Fischer, P., Smajic, A., Abrami, G., Mehler, A.: Multi-type-td-tsr – extracting tables from document images using a multi-stage pipeline for table detection and table structure recognition: From ocr to structured table representations. In: Edelkamp, S., Möller, R., Rueckert, E. (eds.) KI 2021: Advances in Artificial Intelligence, pp. 95–108. Springer, Cham (2021)
- [46] Bradski, G.: The opencv library. Dr. Dobb's Journal: Software Tools for the Professional Programmer **25**(11), 120–123 (2000)

- [47] Raja, S., Mondal, A., Jawahar, C.V.: Table Structure Recognition using Top-Down and Bottom-Up Cues. arXiv. <https://doi.org/10.48550/arXiv.2010.04565> . <http://arxiv.org/abs/2010.04565> Accessed 2023-09-12