QuickTable, an ultra lightweight application to extract tables in PDF

Xiang Fei (Team Leader)
Student ID: 12009041
xiangfei@link.cuhk.edu.cn

Yuantao Zhang
Student ID: 120090358
yuantaozhang@link.cuhk.edu.cn

Yuzhou Cheng
Student ID: 119010047
yuzhoucheng@link.cuhk.edu.cn

Abstract

This paper proposes a new pipeline to extract tables of interest in PDF files, and develops an ultra lightweight application named QuickTable accordingly. Most of the previous research only focused on one or two tasks of table recognition and there is little research on finding tables of interest. The developed QuickTable uses the proposed pipeline based on PP-Picodet, SLANet, PPOCRv3, Text Segmentation and Cosine Similarity Analysis, which allows users to upload PDF files from mobile devices and enter keywords to get tables of interest. In addition, we have trained models in both Chinese and English so that users can upload files in different languages. Experiments show that the proposed pipeline is lightweight and outperforms previous approaches, demonstrating the effectiveness of our method.

More resources can be found in https://github.com/EdgarFx/QuickTable.

1. Introduction

Table recognition is a booming hot topic and its real application becomes the focus of research. The main tasks for table recognition can be divided into three parts.

1. *Table detection* This procedure mainly find the position of tables in given documents
2. *Table structure recognition* This procedure finds table cells and their positions in table images.
3. *Table text extraction* This procedure extract text information from table images.

To apply table recognition efficiently to extract tables from given documents in real industry, several difficulties occurs. In our work, we specifically focus on table extraction from PDF files.

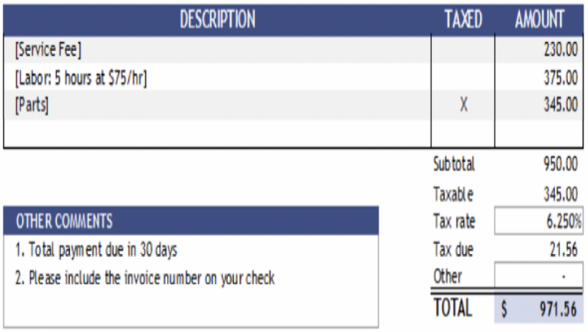

Figure 1. Example PDF with ambiguous tables. Whether the main invoice table should be one or two tables is uncertain

First, there is great diversity of table formatting among different subjects and fields, which may raise the difficulty of table detection. For example, forms of invoice may be different from tables on scientific papers. Moreover, tables may be ambiguous in some cases. [45]The Figure 1 above shows an example.

Second, Blank blocks and wireless tables raise the difficulties in table structure recognition. The white spaces in tables can easily be classified to the wrong table cell, and this has great influence on the accuracy of table structure recognition [34].

Besides, the above-mentioned three tasks of table recognition are separated in many previous works; some models about table text extraction (e.g. Tesseract OCR) are trained on English datasets, their performance on Chinese PDFs is poor; Previous models used (such as R-CNN) is heavyweight, the inference time of those models may be long.

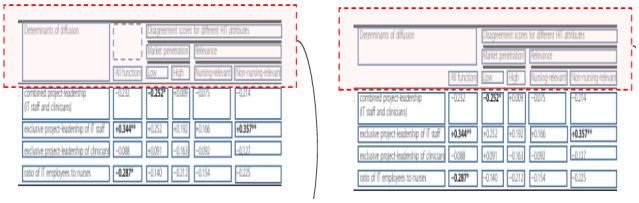

Figure 2. An example that white spaces can be classified into wrong table cells. The left is the ground-truth of aligned bounding boxes and the right is a wrong classification

This will heavily limit their applications on mobile devices.

Motivated by above mentioned difficulties, we propose QuickTable, a ultra ligheweight application to extract tables in PDF. QuickTable consists of a pipeline using the combination of three state-of-the-art models with respect to the three tasks in table recognition. The proposed pipeline can deal with the difficulties efficienctly. The contribution of QuickTable is four-fold:

1. QuickTable uses models which are light-weight. The accuracy is guaranteed while the model parameters are reduced and the inference speed is increased
2. QuickTable combines the tasks mentioned before together
3. QuickTable performs well on both Chinese PDFs and English PDFs.
4. QuickTable encapsulates the proposed pipeline into backend APIs and users can use the application via smartphones or desktops.

2. Related Work

2.1. Table detection

The task of table detection is included in the field of object detection. The state-of-the-art object detection models mainly includes two branches: one-stage detectors and two-stage detectors [27]. A large amount of one-stage detectors including YOLOv2 [35], YOLOv3 [36], YOLOv4 [2], RetinaNet [26], RefineDet [50], EfficentDet [43], FreeAnchor [51], and two-stage detectors including faster R-CNN [37], FPN [25], Cascade R-CNN [3], Trident-Net [23] are proposed to promote the growth of state-of-the-art performance in object detection continuously.

YOLO YOLO series detectors [2, 35, 36] have been widely used in practice, due to their excellent effectiveness and efficiency. To be more specific, YOLOv4 [2] discusses a large number of tricks including many "bag of freebies" which not increase the infer time, and several "bag of specials" that increase the inference cost by a small amount but can significantly improve the accuracy of object detection. YOLOv4 greatly improves the effectiveness and efficiency of the YOLOv3 [36].

PP-YOLO and PP-YOLOv2 Variants of YOLOv3 can also be used in table detection like PP-YOLO [30], PP-YOLOv2 [18]. Specifically, PP-YOLO first replaces the backbone to ResNet50-vd [15]. After that a total of 10 tricks which can improve the performance of YOLOv3 almost without losing efficiency are added to YOLOv3 such as Deformable Conv [8], SSLD [6], CoordConv [28], DropBlock [12], SPP [14] and so on. PP-YOLOv2 adds a bunch of refinements that almost not increase the infer time to improve the overall performance of the PP-YOLO.

In overall, compared to two-stage detectors, one-stage detectors often have much lower model size and inference time but may suffer from low accuracy. PP-YOLO [30] and PP-YOLOv2 [18] highly increase the accuracy without increasing inference time in a large scale. In our proposed QuickTable, we select to use PP-PicoDet [49], which is more lightweight and achieves superior performance compared to PP-YOLOv2.

2.2. Table structure recognition

Traditional table recognition researches mainly worked with hand-crafted features and heuristic rules [19, 21]. These methods are mostly applied to simple table structures. With the great success of deep neural network in computer vision field, works began to focus on the image-based table with more general structures [33,39]. The previous table structure recognition methods can be divided into two types: global-object-based methods and local-object-based methods. Global-object-based methods mainly focus on the characteristics of global table components and mostly started from row/column or grid boundaries detection. For example, works of [39,41,42] obtain the rows and columns regions using the detection or segmentation models and then intersect these two regions to obtain the grids of cells. Local-object-based methods begin from the smallest fundamental element, cells. In this category, some methods [4, 22, 24]treat the detected boxes as nodes in a graph and attempt to predict the relations based on techniques of Graph Neural Networks [38].

LGPMA To further improve the accuracy and deal with blank blocks efficiently, methods like LGPMA [34] are developed. LGPMA compromises the advantages of both global and local features. Based on the local detection results, this method integrates the global information to refine the detected bounding boxes and provide a straightforward guide for empty cell division.

RARE and TableRec-RARE For light-weight purpose, an end-to-end Table structure recognition method TableRec-RARE has been developed, based on the text recognition algorithm RARE [40]. RARE uses the connec-

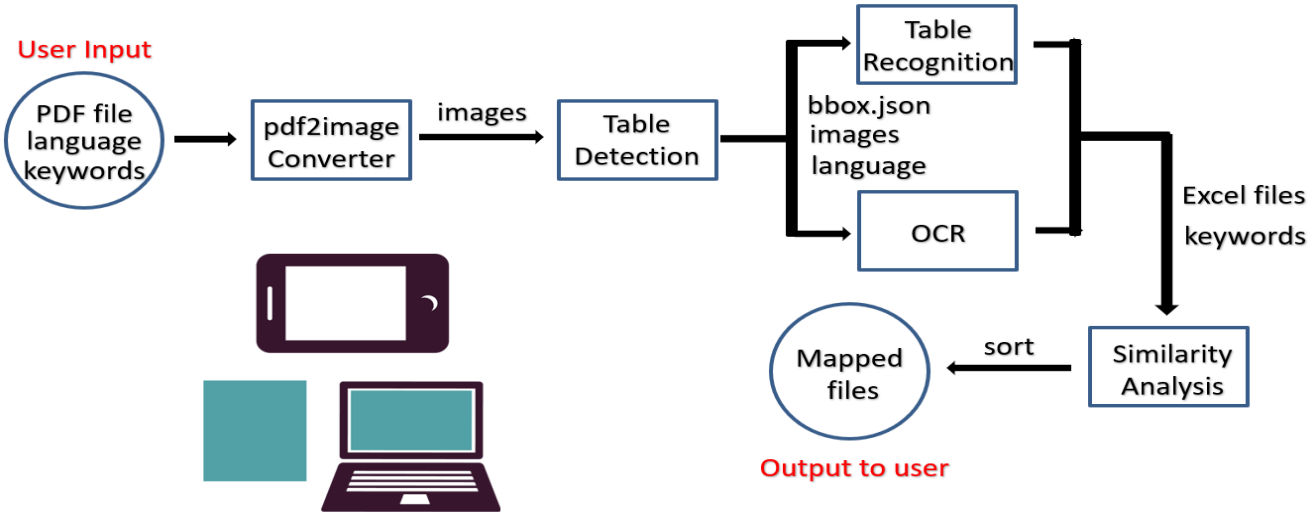

Figure 3. Pipeline of QuickTable. The user chooses pdf files, and also inputs the language (Chinese or English) and keywords from mobile devices. Then, through Converter, Table Detection, Table Recognition, Table Text Extraction and Similarity Analysis, the user can get the mapped tables in excel format.

tion of a spatial transformer network and an attention-based sequence recognizer. It can deal with the irregular text problem in an elegant way and return the table structure and cell coordinates. TableRec-RARE uses DB model to retrieve the The coordinates of single-line text and RARE to predict the the table structure and cell coordinates. The recognition result of the cell is combined by recognition result of the single line and the coordinates of the cell. The cell recognition result and the table structure together construct the html string of the table. In QuickTable, we choose to use SLANet (Structure Location Alignment Network) which has some improvements on the TableRec-RARE.

2.3. Table Text Extraction

Table text extraction belongs to the field of scene text extraction. Traditional methods of scene text recognition first performed detection to generate multiple candidates of character locations, then applied a character classifier for recognition. Wang et al. [44] used Random Ferns and HOG features to detect characters and then found an optimal configuration of a particular word via a pictorial structure. Mishra et al. [32] detected character candidates using sliding windows and integrated both bottom-up and top-down cues in a unified Conditional Random Field (CRF) model.

To avoid the impact of inaccurate character detector, some researches in scene text recognition focused on the mapping from the entire image to word string directly. Almazán et al. [1] embedded word images and word labels into a common Euclidean space and the embedding vectors were used to match images and labels. Jaderberg et al. [20] constructed two CNNs to classify character at each position in the word and detect the N-grams contained within the word separately, following a CRF model to combile their representations.

PP-OCR For the light-weight purpose and requirements of great performance on Chinese documents, the series of PP-OCR is developed [10, 11]. PP-OCR [11] fully tapped the ability of text detection algorithm DB and text recognition algorithm CRNN, and adopted 19 effective strategies from 8 aspects, including backbone network selection, prediction head design, data augmentation, learning rate transformation strategy, regularization parameter selection, pretraining model use, and model automatic clipping quantification. PP-OCRv2 [10] utilizes methods like Collaborative mutual learning(CML),and achieve higher accuracy than PP-OCR without increasing the model size. In QuickTable, we adopt to use PP-OCRv3, which is further upgraded on the basis of PP-OCRv2.

3. Method

This section presents the proposed pipeline of QuickTable. And then, advanced methods used in Table Detection, Table Recognition and Table Text Extraction modules are introduced. In addition, we consider the similarity analysis method. Last, technologies related to the implementation of the application are illustrated.

3.1. Overview

Our proposed pipeline for QuickTable allows the user to send the request which contains the pdf file that he wants to analyze, the language used in the uploaded file, and the

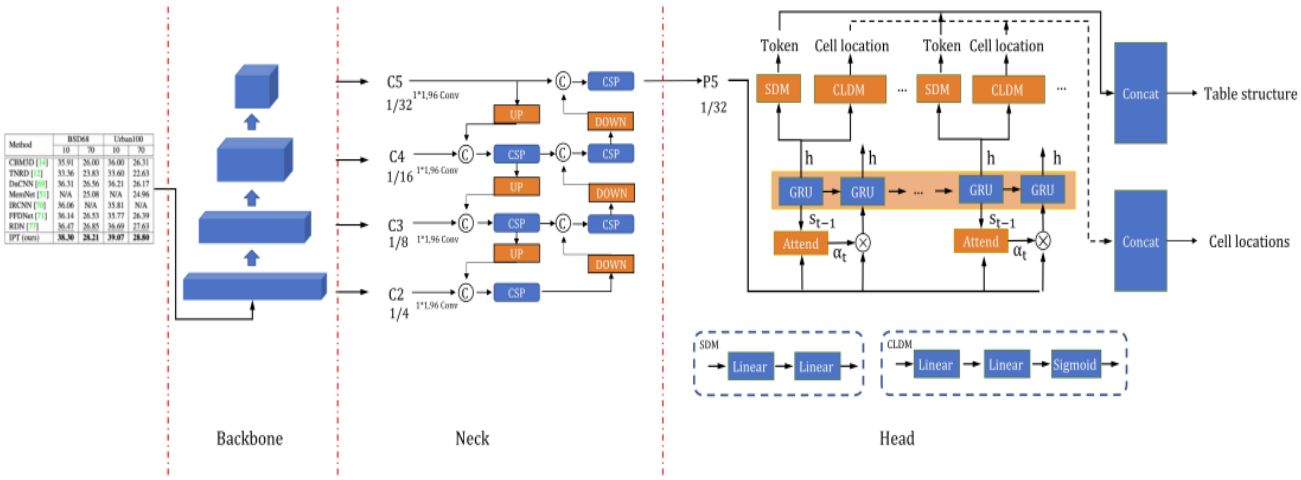

Figure 4. Architecture of our proposed SLANet, where C represent concat operation.

keywords used to filter the tables that the user is interested in from the developed front-end interface. And then, The back-end system will perform table recognition and similarity analysis. After that, the back-end will also sort the matched table files according to the similarity and send the response to the user through the server. This pipeline is mainly composed of five modules: pdf2image Converter, Table Detection, Table Recognition, OCR and Similarity Analysis.

In the pdf2image Converter, each page of the input pdf file will be converted into one image. And then, all these images will be sent to the Table Detection Module, which is used to find all tables with their positions in the images. The generated json file and the images can be used to obtain the table images. After that, in Table Recognition and OCR module, these table images These table images will be parsed into HTML code containing text content according to the language the user has input, and then converted into excel files. In addition, the Similarity Analysis module can score and sort each table excel file according to the keywords entered by the user, and return these files to the user in order.

3.2. Table Detection

Table Detection refers to finding table areas in ducument images. In PP-Structure proposed by paddle, the object detection algorithm PP-YOLOv2 [18] is adopted as the table detector. In the proposed pipeline, we use a more lightweight detector PP-PicoDet [49], which achieves superior performance on mobile devices. In addition, the image scale is adjusted for the table detection scene, and use a knowledge distillation algorithm named FGD [47] to further improve the model accuracy.

PP-PicoDet: A better real-time object detector on mobile devices. PaddleDetection proposed a new family of realtime object detectors, named PP-PicoDet, which achieves superior performance on mobile devices. PP-PicoDet adopts the CSP structure to constructure CSP-PAN as the neck, SimOTA as label assignment strategy, PP-LCNet as the backbone, and an improved detection One-shot Neural Architecture Search(NAS) is proposed to find the optimal architecture automatically for object detection. We replace PPYOLOv2 adopted by PP-Structure with PP-PicoDet, and adjust the input scale from 640*640 to 800*608, which is more suitable for document images. With 1.0x configuration, the accuracy is comparable to PP-YOLOv2, and the CPU inference speed is 11 times faster.

FGD: Focal and Global Knowledge Distillation. FGD [47], a knowledge distillation algorithm for object detection, takes into account local and global feature maps, combining focal distillation and global distillation. Focal distillation separates the foreground and background of the image, forcing the student to focus on the teacher's critical pixels and channels. Global distillation rebuilds the relation between different pixels and transfers it from teachers to students, compensating for missing global information in focal distillation. Based on the FGD distillation strategy, the student model (LCNet1.0x based PP-PicoDet) gets 0.5% mAP improvement with the knowledge from the teacher model (LCNet2.5x based PP-PicoDet). Finally the student model is only 0.2% lower than the teacher model on mAP, but 100% faster.

3.3. Table Strucuture Recognition

In recent years, many Table Structure Recognition algorithms based on deep learning have been proposed. In PP-Structure, an end-to-end Table Structure Recognition algorithm TableRec-RARE has been used, based on the text

recognition algorithm RARE [40]. The model output is an HTML representation of a table structure, which can be easily converted into Excel files. In our pipeline, we refer to PP-Structurev2 and choose an efficient Table Strucutre Recognition algorithm named SLANet (Structure Location Alignment Network). Compared with TableRec-RARE, SLANet has been upgraded in terms of model structure and loss. Figure 4 shows the network structure of SLANet.

PP-LCNet: CPU-friendly Lightweight Backbone. PPLCNet [5] is a lightweight CPU network based on the MKLDNN acceleration strategy, which achieves better performance on multiple tasks than lightweight models such as ShuffleNetV2 [31], MobileNetV3 [16], and GhostNet [13]. Additionally, pre-trained weights trained by SSLD [7] on ImageNet are used for Table Recognition model training process for higher accuracy.

CSP-PAN: Lightweight Multi-level Feature Fusion Module. Fusion of the features extracted by the backbone network can effectively alleviate problems brought by scale changes in complex scenes. In the early days, the FPN [25] module was proposed and used for feature fusion, but its feature fusion process was one-way (from high-level to low-level), which was not sufficient. CSP-PAN [49] is improved based on PAN. While ensuring more sufficient feature fusion, strategies such as CSP block and depthwise separable convolution are used to reduce the computational cost. In SLANet, we reduce the output channels of CSP-PAN from 128 to 96 in order to reduce the model size.

SLAHead: Structure and Location Alignment Module. In the TableRec-RARE head, output of each step is concatenated and fed into SDM (Structure Decode Module) and CLDM (Cell Location Decode Module) to generate all cell tokens and coordinates, which ignores the one-to-one correspondence between cell token and coordinates. Therefore, we propose the SLAHead to align cell token and coordinates. In SLAHead, output of each step is fed into SDM and CLDM to get the token and coordinates of the current step, the token and coordinates of all steps are concatenated to get the HTML table representation and coordinates of all cells.

Merge Token. In TableRec-RARE, we use two separate tokens <td>and </td>to represent a non-cross-row-column cell, which limits the network's ability to handle tables with a large number of cells. Inspired by TableMaster [48], we regard <td>and </td>as one token - <td></td>in SLANet.

3.4. Table Text Extraction

For table text extraction, we choose to use PP-OCRv3, which is further upgraded on the basis of PP-OCRv2 [10]. The detection module is still optimized based on the DB algorithm. And the recognition module no longer uses CRNN, but replaces it with the latest text recognition algorithm SVTR included in IJCAI 2022. Fig. 5 shows the framework of PP-OCRv3. The strategies in the green boxes are the same as PP-OCRv2, while those in the pink boxes are the newly added strategies.

LK-PAN: A PAN module with large receptive field. LKPAN (Large Kernel PAN) is a lightweight PAN [29] module with larger receptive field. The main idea is to increase the convolution kernel size in the path augmentation of the PAN module from 3×3 to 9×9, which can improve the receptive field of each pixel of the feature map, making it easier to detect text in large fonts and text with extreme aspect ratios.

DML: Deep Mutual Learning for Teacher Model. DML(Deep Mutual Learning) [46] can effectively improve the accuracy of the text detection model by learning from each other with two models with the same structure. The DML strategy is adopted in the teacher model training to improve the Hmean of the teacher model as much as possible.

RSE-FPN: A FPN module with residual attention mechanism. RSE-FPN (Residual Squeeze-and-Excitation FPN) introduces residual attention mechanism by replacing the convolution layers in FPN with RSEConv, to improve the representation ability of the feature map. RSEConv consists of two parts: Squeeze-and-Excitation(SE) block [17] and the residual structure. At first, we tried to add only SE blocks, which turned out not as effective as expected. Considering the number of channels of the lightweight FPN of PP-OCRv2 is relatively small, the SE module may suppress some channels containing important features. The introduction of residual structure in RSEConv can alleviate the above problems and improve the text detection performance.

SVTR-LCNet: Lightweight Text Recognition Network. SVTR-LCNet is a lightweight text recognition network fusing Transformer based network SVTR [9] and lightweight CNN-based network PP-LCNet [5]. Specifically, we adopt a tiny version of SVTR, named SVTR-Tiny. However, SVTR-Tiny is 10 times slower than the recognizer of PP-OCRv2 based on CRNN on CPU with MKLDNN enabled due to the limited model structure supported by the MKLDNN acceleration library, which is not practical enough. The main structure in SVTR-Tiny is Mix Block, which is proved to be the most time-consuming module through analysis, so we optimize the structure in three steps to speed up and ensure the effectiveness of the model.

GTC: Guided Training of CTC by Attention. Connectionist Temporal Classification (CTC) and attention mechanism are two main approaches used in recent scene text recognition works. Compared with attention-based methods, CTC decoder can achieve a much faster prediction speed, but lower accuracy. To obtain an efficient and effective model, this strategy uses an attention module to guide

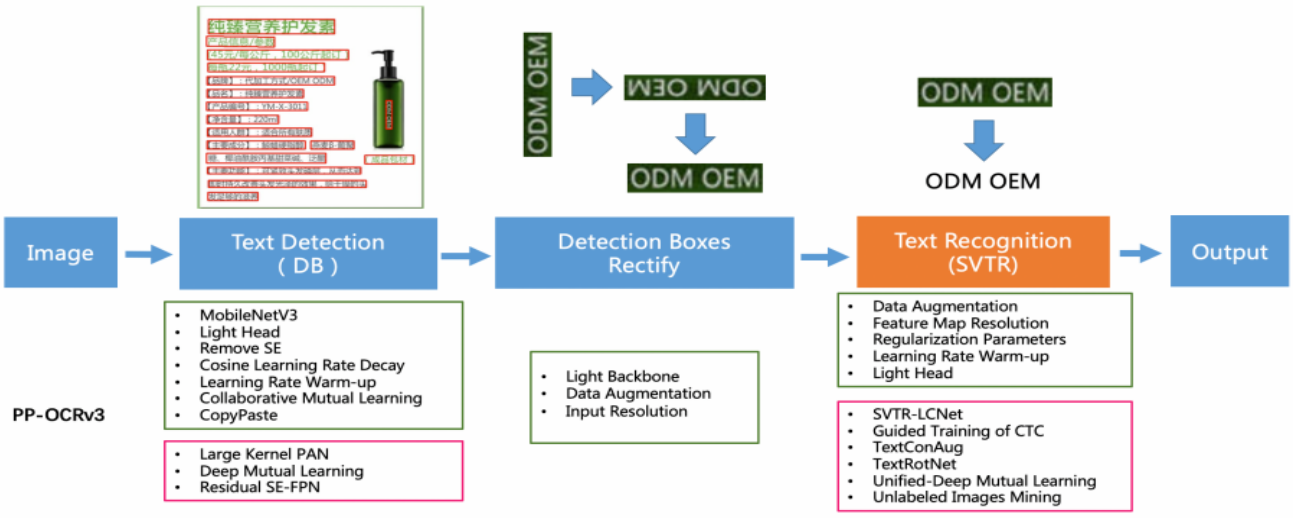

Figure 5. Framework of PP-OCRv3. Strategies in the green boxes are the same as PP-OCRv2. Strategies in the pink boxes are the newly added ones in the PP-OCRv3.

the training of CTC to fuse multiple features, which is effective for the improvement of accuracy. As the attention module is completely removed during prediction, no more time cost is added in the inference process.

TextConAug: Data Augmentation for Mining Text Context Information. TextConAug is a data augmentation strategy for mining textual context information. The main idea comes from the paper ConCLR [46], in which the author proposed data augmentation strategy ConAug to concat 2 different images in a batch to form new images and perform self-supervised comparative learning. We apply this method to supervised learning tasks, and design TextConAug which can enrich the context information of training data and improve the diversity of training data.

TextRotNet: Self-Supervised Pre-trained Model. TextRotNet is a pre-trained model trained with a large amount of unlabeled text line data in a self-supervised manner. This strategy uses this model to initialize the weights of SVTR-LCNet, helping the text recognition model to converge better.

U-DML: Unified-Deep Mutual Learning. U-DML is a strategy proposed in PP-OCRv2 which is very effective to improve the accuracy without increasing model size. In PPOCRv3, for two different structures SVTR-LCNet and attention module, the feature map of PP-LCNet, the output of the SVTR module and the output of the Attention module between them are simultaneously supervised and trained.

UIM: Unlabeled Images Mining. UIM is a simple unlabeled data mining strategy. The main idea is to use a highprecision text recognition model to predict unlabeled images to obtain pseudo-labels, and select samples with high prediction confidence as training data for training lightweight models.

3.5. Similarity Analysis

After getting the excel files, we need to match the contents of excel files to the keywords passed by the users. We first need to do preprocess to those keywords. We adopt to use the package Jieba in python to do the partition of the keywords. Then we search the partition of keywords in all cells of an excel file using a N-grams fashion and calculate the cosine similarity of the searched string and the target string. Note that the N here is the length of the partition. If the searching process reaches the end of file, we sum up the cosine similarity of all cells together and therefore, for an excel file, we can get a value representing the total cosine similarity. Finally, we sort the excel files according to the value in a descending order.

3.6. Back-end and Front-end Development

We provide a website for users to get access to the data in the database interactively. The front-end is built based on the React frame and the back-end is built based on the Django frame with an embedded Django-SQL database. Besides, we use yarn from Node.js to manage our exogenous libraries. We decide not to use a server considering the cost. However, we note that the front-end is run locally at ”localhost:3000” while the front-end is run at ”localhost:8000”, which are two different ports. Under these circumstances, we cannot conduct the front-end and back-end interactions. Thus, to ensure that our website could be run locally, we adopt the Nginx to redirect the root URL of the front-end as well as the back-end to the same port. For example, we set the redirected port as 4000 in Fig. 7.

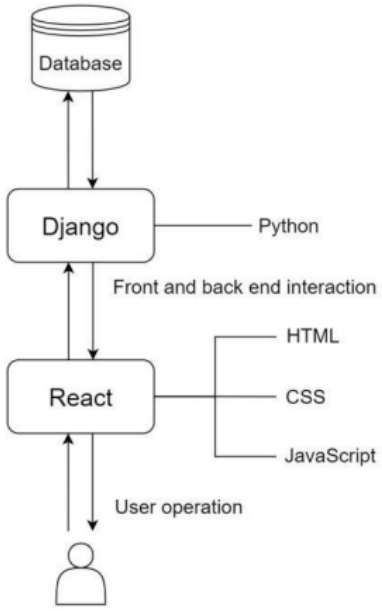

Figure 6. UI Structure

```
# necessary
map $http_upgrade $connection_upgrade {
        default upgrade;
        '' close;
}

server {
    listen 4000;
    server_name localhost, 127.0.0.1;

    location / {
        autoindex on;
        proxy_pass  http://localhost:3000;
    }

    # redirect
    location ^~ /ws {
            # for backend
            proxy_pass http://localhost:8000;
            # parameters
            proxy_read_timeout 300s;
            proxy_send_timeout 300s;
            proxy_set_header  Host $http_host;
            proxy_set_header  X-Real-IP  $remote_addr;
            proxy_set_header  X-Forwarded-For $proxy_add_x_forwarded_for;
            proxy_set_header  X-Forwarded-Proto $scheme;
            proxy_http_version 1.1;
            proxy_set_header Upgrade $http_upgrade;
            proxy_set_header Connection $connection_upgrade;
    }
}
```

Figure 7. Nginx Configuration

4. Experiments

4.1. Experiments Setup

Datasets. For Table Detection, experiments are carried out on PubLayNet dataset. PubLayNet is a large-scale dataset of document images, which contains 335,703 training, 11,245 validation and 11,405 testing images. Document layout elements such as text, title, list, table and figure are covered. MAP(Mean Average Precision) is used to evaluate the model performance. To verify the strategy generalization, we also carry out experiments on CDLA dataset, which is a Chinese layout analysis dataset and covers document elements such ad text, title, figure, figure caption, table, table caption, header, footer, reference, equation. The dataset contains 6,000 annotated images (5,000 for training and 1,000 for validation).

For Table Recognition, we conduct experiments on PubTabNet dataset to verify the effectiveness of the proposed SLANet. PubTabNet contains 500,777 training, 9,115 validation, and 9,138 testing images generated by matching the XML and PDF representations of scientific articles. Since the annotations of the testing set are not released, we only report results on the validation set. A new Tree-Edit-Distance-based Similarity (TEDS) metric for table recognition task is proposed in this work, which can identify both table structure recognition and OCR errors. However, taking OCR errors into account may cause unfair comparison because of different OCR models. Some recent works have proposed a modified TEDS metric named TEDS-Struct to evaluate table structure recognition accuracy only by ignoring OCR errors. We use accuracy, TEDS and this modified metric to evaluate our approach on this dataset.

Implementation Details. For Table Detection model, we use Momentum with momentum of 0.9 and weight decay 4e5. Cosine decay learning rate scheduling strategy is adopted with learning rate of 0.4. The batch size and epoch num are set as 24 and 70 on 8*32G V100 GPU devices.

For Table Recognition model, we use Adam optimizer, the initial learning rate is set to 0.001 and adjusted to 0.0001 and 0.00005 after 50 and 60 epochs. The batch size and epoch num are set as 48 and 100 on 4*32G V100 GPU devices.

4.2. Table Detection

Ablation experiments on PubLayNet are shown in Table 1. PP-YOLOv2 is used for Table Detection in PP-Structure. PP-PicoDet-LCNet2.5x is much more efficient than PPY-OLOv2, but mAP is reduced by 1.1%. By adjusting the input image scale, mAP can be improved by 1.7%, which is higher than baseline. To get a more lightweight model, we train 1.0x model with FGD, using the previous 2.5x model as the teacher model. The final mAP exceeds the baseline by 0.4% with the inference speed increasing by 11 times, and the model storage is reduced by 95%.

To verify the generalization of these strategies, we also conduct ablation experiments on the Chinese Layout Analysis dataset CDLA, and the results are shown in Table 2. It can be found that the performance of layout analysis in both Chinese and English scenarios can be significantly improved.

We also compare the optimized PP-PicoDet with open source method layout-parser, which is based on Detectron2. As can be seen from Table 3, PP-PicoDet outperforms layout-parser by a large margin on both mAP and inference speed.

Strategy	mAP (%)	Speed (ms)	Model Size(M)
PP-YOLOv2(640*640)	93.6	512	221
PP-PicoDet-LCNet2.5x(640*640)	92.5	53.2	29.7
PP-PicoDet-LCNet2.5x(800*608)	94.2	83.1	29.7
PP-PicoDet-LCNet1.0x(800*608)	93.5	41.2	9.7
PP-PicoDet-LCNet1.0x(800*608) + FGD	94.0	41.2	9.7

Table 1: Ablation experiments on PubLayNet dataset. **LCNet** refers to the backbone used in PP-PicoDet. The inference speed is tested on CPU.

Strategy	mAP (%)
PP-YOLOv2	84.7
PP-PicoDet-LCNet2.5x(800*608)	87.8
PP-PicoDet-LCNet1.0x(800*608)	84.5
PP-PicoDet-LCNet1.0x(800*608) + FGD	86.8

Table 2: Ablation experiments on CDLA Dataset. LCNet refers to the backbone used in PP-PicoDet. The inference speed is tested on CPU.

Strategy	mAP (%)	Speed (ms)
layoutparser(Detectron2)	88.98	2900.0
PP-StructureV2(PP-PicoDet)	94.00	41.2

Table 3: Comparison with different methods on PubLayNet dataset.

4.3. Table Recognition

Table 4 shows the ablation experiments of optimization strategies for SLANet. The baseline model is TableRecRARE which is proposed in PP Structure. It can be found that the accuracy can be improved from 71.73% to 74.71% by replacing the MobileNetV3 based backbone with PPLCNet, without increasing the inference time. Using CSPPAN, the accuracy can be further improved to 75.68%, and the inference time is reduced by 70ms due to the reduction of the number of feature maps entering the head. Subsequently, we use SLAHead to align the structure and location of cells, which improves the accuracy from 75.68% to 77.7%, but the model inference time cost increases from 708ms to 766ms due to the repeated execution of SDM and CLDM. During the previous training processes, the maximum number of tokens can be recognized is set to 500, so images with a token length greater than 500 will not participate in the calculation of the accuracy, but will participate in the calculation of TEDS. After merging tokens that appear in pairs, a HTML string of more tokens can be recognized. Almost all validation sets will participate in the calculation, so the accuracy is reduced slightly, but the TEDS is increased from 94.85% to 95.89%.

We compare our choosed SLANet with several stateof-the-art methods on PubTabNet dataset. Table 5 shows the results of SLANet and some state-of-the-art methods on PubTabNet such as EDD TableMaster and LGPMA. As can be seen from the table, SLANet is optimal for model size and inference time while maintaining competitive results.

Strategy	Acc (%)	TEDS (%)	Speed (ms)	Model Size(M)
TableRec-RARE	71.73	93.88	779	6.8
+PP-LCNet	74.71	94.37	778	8.7
+CSP-PAN	75.68	94.72	708	9.3
+SLAHead	77.7	94.85	766	9.2
+MergeToken	76.31	**95.89**	766	9.2

Table 4: Ablation experiments of SLANet on PubTabNet Dataset. The prediction speed is tested on CPU.

Methods	Acc (%)	TEDS (%)	TEDS-Struct (%)	Inference time (ms)	Model Size(M)
EDD	-	88.3	-	-	-
TableMaster	77.90	96.12	-	2144	253
LGPMA	65.74	94.70	96.70	-	177
TableRec-RARE	71.73	93.88	-	779	6.8
SLANet	76.31	95.89	**97.01**	**766**	9.2

Table 5: Compare with state-of-the-art methods on PubTabNet dataset.

5. Conclusion

In this paper, we propose a pipeline to implement the table extraction from pdf files with interested keywords in different languages, so as to implement an application named QuickTable. For table detection. table sturecture recognition, table text extraction, we use some of the latest models to integrated a more robust and comprehensive structural transformation system. Experiments demonstrate the used structure outperforms PP-Structure on all subtasks (Table Detection and Table Recognition) in terms of speed and accuracy. The corresponding ablation experiments are also provided.

Limitations The interaction design of the developed application can be improved. Besides, The speed and user experience of the product can be improved by further optimizing the data flow. Last but not least, the currently used text similarity analysis method is relatively traditional, and may be improved to a learning model with a faster inference time to achieve better results.

References

[1] Jon Almazán, Albert Gordo, Alicia Fornés, and Ernest Valveny. Word spotting and recognition with embedded attributes. *IEEE transactions on pattern analysis and machine intelligence*, 36(12):2552–2566, 2014. 3

[2] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020. 2

[3] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6154–6162, 2018. 2

[4] Zewen Chi, Heyan Huang, Heng-Da Xu, Houjin Yu, Wanxuan Yin, and Xian-Ling Mao. Complicated table structure recognition. *arXiv preprint arXiv:1908.04729*, 2019. 2

[5] Cheng Cui, Tingquan Gao, Shengyu Wei, Yuning Du, Ruoyu Guo, Shuilong Dong, Bin Lu, Ying Zhou, Xueying Lv, Qiwen Liu, Xiaoguang Hu, Dianhai Yu, and Yanjun Ma. Pp-lcnet: A lightweight CPU convolutional neural network. *CoRR*, abs/2109.15099, 2021. 5

[6] Cheng Cui, Ruoyu Guo, Yuning Du, Dongliang He, Fu Li, Zewu Wu, Qiwen Liu, Shilei Wen, Jizhou Huang, Xiaoguang Hu, et al. Beyond self-supervision: A simple yet effective network distillation alternative to improve backbones. *arXiv preprint arXiv:2103.05959*, 2021. 2

[7] Cheng Cui, Ruoyu Guo, Yuning Du, Dongliang He, Fu Li, Zewu Wu, Qiwen Liu, Shilei Wen, Jizhou Huang, Xiaoguang Hu, et al. Beyond self-supervision: A simple yet effective network distillation alternative to improve backbones. *arXiv preprint arXiv:2103.05959*, 2021. 5

[8] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 764–773, 2017. 2

[9] Yongkun Du, Zhineng Chen, Caiyan Jia, Xiaoting Yin, Tianlun Zheng, Chenxia Li, Yuning Du, and Yu-Gang Jiang. Svtr: Scene text recognition with a single visual model, 2022. 5

[10] Yuning Du, Chenxia Li, Ruoyu Guo, Cheng Cui, Weiwei Liu, Jun Zhou, Bin Lu, Yehua Yang, Qiwen Liu, Xiaoguang Hu, Dianhai Yu, and Yanjun Ma. Pp-ocrv2: Bag of tricks for ultra lightweight ocr system, 2021. 3, 5

[11] Yuning Du, Chenxia Li, Ruoyu Guo, Xiaoting Yin, Weiwei Liu, Jun Zhou, Yifan Bai, Zilin Yu, Yehua Yang, Qingqing Dang, et al. Pp-ocr: A practical ultra lightweight ocr system. *arXiv preprint arXiv:2009.09941*, 2020. 3

[12] Golnaz Ghiasi, Tsung-Yi Lin, and Quoc V Le. Dropblock: A regularization method for convolutional networks. *Advances in neural information processing systems*, 31, 2018. 2

[13] Kai Han, Yunhe Wang, Qi Tian, Jianyuan Guo, Chunjing Xu, and Chang Xu. Ghostnet: More features from cheap operations. *CoRR*, abs/1911.11907, 2019. 5

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 37(9):1904–1916, 2015. 2

[15] Tong He, Zhi Zhang, Hang Zhang, Zhongyue Zhang, Junyuan Xie, and Mu Li. Bag of tricks for image classification with convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 558–567, 2019. 2

[16] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, Quoc V. Le, and Hartwig Adam. Searching for mobilenetv3. *CoRR*, abs/1905.02244, 2019. 5

[17] Jie Hu, Li Shen, Samuel Albanie, Gang Sun, and Enhua Wu. Squeeze-and-excitation networks, 2017. 5

[18] Xin Huang, Xinxin Wang, Wenyu Lv, Xiaying Bai, Xiang Long, Kaipeng Deng, Qingqing Dang, Shumin Han, Qiwen Liu, Xiaoguang Hu, Dianhai Yu, Yanjun Ma, and Osamu Yoshie. Pp-yolov2: A practical object detector. *CoRR*, abs/2104.10419, 2021. 2, 4

[19] Katsuhiko Itonori. Table structure recognition based on textblock arrangement and ruled line position. In *Proceedings of 2nd International Conference on Document Analysis and Recognition (ICDAR'93)*, pages 765–768. IEEE, 1993. 2

[20] Max Jaderberg, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep structured output learning for unconstrained text recognition. *arXiv preprint arXiv:1412.5903*, 2014. 3

[21] Thomas G Kieninger. Table structure recognition based on robust block segmentation. In *Document Recognition V*, volume 3305, pages 22–32. SPIE, 1998. 2

[22] Elvis Koci, Maik Thiele, Wolfgang Lehner, and Oscar Romero. Table recognition in spreadsheets via a graph representation. In *2018 13th IAPR International Workshop on Document Analysis Systems (DAS)*, pages 139–144. IEEE, 2018. 2

[23] Yanghao Li, Yuntao Chen, Naiyan Wang, and Zhaoxiang Zhang. Scale-aware trident networks for object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6054–6063, 2019. 2

[24] Yiren Li, Zheng Huang, Junchi Yan, Yi Zhou, Fan Ye, and Xianhui Liu. Gfte: graph-based financial table extraction. In *International Conference on Pattern Recognition*, pages 644–658. Springer, 2021. 2

[25] Tsung-Yi Lin, Piotr Dollár, Ross B. Girshick, Kaiming He, Bharath Hariharan, and Serge J. Belongie. Feature pyramid networks for object detection. *CoRR*, abs/1612.03144, 2016. 2, 5

[26] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017. 2

[27] Li Liu, Wanli Ouyang, Xiaogang Wang, Paul Fieguth, Jie Chen, Xinwang Liu, and Matti Pietikäinen. Deep learning for generic object detection: A survey. *International journal of computer vision*, 128(2):261–318, 2020. 2

[28] Rosanne Liu, Joel Lehman, Piero Molino, Felipe Petroski Such, Eric Frank, Alex Sergeev, and Jason Yosinski. An intriguing failing of convolutional neural networks and the coordconv solution. *Advances in neural information processing systems*, 31, 2018. 2

[29] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. Path aggregation network for instance segmentation, 2018. 5

[30] Xiang Long, Kaipeng Deng, Guanzhong Wang, Yang Zhang, Qingqing Dang, Yuan Gao, Hui Shen, Jianguo Ren, Shumin Han, Errui Ding, et al. Pp-yolo: An effective and efficient implementation of object detector. *arXiv preprint arXiv:2007.12099*, 2020. 2

[31] Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. Shufflenet V2: practical guidelines for efficient CNN architecture design. *CoRR*, abs/1807.11164, 2018. 5

[32] Anand Mishra, Karteek Alahari, and CV Jawahar. Top-down and bottom-up cues for scene text recognition. In *2012 IEEE conference on computer vision and pattern recognition*, pages 2687–2694. IEEE, 2012. 3

[33] Kyosuke Nishida, Kugatsu Sadamitsu, Ryuichiro Higashinaka, and Yoshihiro Matsuo. Understanding the semantic structures of tables with a hybrid deep neural network architecture. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017. 2

[34] Liang Qiao, Zaisheng Li, Zhanzhan Cheng, Peng Zhang, Shiliang Pu, Yi Niu, Wenqi Ren, Wenming Tan, and Fei Wu. Lgpma: Complicated table structure recognition with local and global pyramid mask alignment. In *International Conference on Document Analysis and Recognition*, pages 99–114. Springer, 2021. 1, 2

[35] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7263–7271, 2017. 2

[36] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018. 2

[37] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015. 2

[38] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE transactions on neural networks*, 20(1):61–80, 2008. 2

[39] Sebastian Schreiber, Stefan Agne, Ivo Wolf, Andreas Dengel, and Sheraz Ahmed. 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)*, volume 1, pages 1162–1167. IEEE, 2017. 2

[40] Baoguang Shi, Xinggang Wang, Pengyuan Lyu, Cong Yao, and Xiang Bai. Robust scene text recognition with automatic rectification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4168–4176, 2016. 2, 5

[41] Shoaib Ahmed Siddiqui, Imran Ali Fateh, Syed Tahseen Raza Rizvi, Andreas Dengel, and Sheraz Ahmed. Deeptabstr: Deep learning based table structure recognition. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 1403–1409. IEEE, 2019. 2

[42] Shoaib Ahmed Siddiqui, Pervaiz Iqbal Khan, Andreas Dengel, and Sheraz Ahmed. Rethinking semantic segmentation for table structure recognition in documents. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 1397–1402. IEEE, 2019. 2

[43] Mingxing Tan, Ruoming Pang, and Quoc V Le. Efficientdet: Scalable and efficient object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10781–10790, 2020. 2

[44] Kai Wang, Boris Babenko, and Serge Belongie. End-to-end scene text recognition. In *2011 International conference on computer vision*, pages 1457–1464. IEEE, 2011. 3

[45] Nancy Xin Ru Wang, Douglas Burdick, and Yunyao Li. Tablelab: An interactive table extraction system with adaptive deep learning. In *26th International Conference on Intelligent User Interfaces-Companion*, pages 87–89, 2021. 1

[46] Zhenbo Xu, Wei Yang, Ajin Meng, Nanxue Lu, Huan Huang, Changchun Ying, and Liusheng Huang. Towards end-to-end license plate detection and recognition: A large dataset and baseline. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision – ECCV 2018*, pages 261–277, Cham, 2018. Springer International Publishing. 5, 6

[47] Zhendong Yang, Zhe Li, Xiaohu Jiang, Yuan Gong, Zehuan Yuan, Danpei Zhao, and Chun Yuan. Focal and global knowledge distillation for detectors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4643–4652, 2022. 4

[48] Jiaquan Ye, Xianbiao Qi, Yelin He, Yihao Chen, Dengyi Gu, Peng Gao, and Rong Xiao. Pingan-vcgroup's solution for ICDAR 2021 competition on scientific literature parsing task B: table recognition to HTML. *CoRR*, abs/2105.01848, 2021. 5

[49] Guanghua Yu, Qinyao Chang, Wenyu Lv, Chang Xu, Cheng Cui, Wei Ji, Qingqing Dang, Kaipeng Deng, Guanzhong Wang, Yuning Du, Baohua Lai, Qiwen Liu, Xiaoguang Hu, Dianhai Yu, and Yanjun Ma. Pp-picodet: A better real-time object detector on mobile devices. *CoRR*, abs/2111.00902, 2021. 2, 4, 5

[50] Shifeng Zhang, Longyin Wen, Xiao Bian, Zhen Lei, and Stan Z Li. Single-shot refinement neural network for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4203–4212, 2018. 2

[51] Xiaosong Zhang, Fang Wan, Chang Liu, Rongrong Ji, and Qixiang Ye. Freeanchor: Learning to match anchors for visual object detection. *Advances in neural information processing systems*, 32, 2019. 2