TDeLTA: A Light-Weight and Robust Table Detection Method Based on Learning Text Arrangement

Yang Fan[1], **Xiangping Wu**[1, 2*], **Qingcai Chen**[1, 3*], **Heng Li**[1]
Yan Huang[4], **Zhixiang Cai**[4], **Qitian Wu**[4]
[1]Harbin Institute of Technology (Shenzhen)
[2]The Hong Kong Polytechnic University
[3]Peng Cheng Laboratory
[4]China Mobile Information Technology Co.,Ltd.
yfan@stu.hit.edu.cn, wxpleduole@gmail.com, qingcai.chen@hit.edu.cn, hengli.lh@outlook.com

Abstract

The diversity of tables makes table detection a great challenge, leading to existing models becoming more tedious and complex. Despite achieving high performance, they often overfit to the table style in training set, and suffer from significant performance degradation when encountering out-of-distribution tables in other domains. To tackle this problem, we start from the essence of the table, which is a set of text arranged in rows and columns. Based on this, we propose a novel, light-weighted and robust **T**able **De**tection method based on **L**earning **T**ext **A**rrangement, namely **TDeLTA**. TDeLTA takes the text blocks as input, and then models the arrangement of them with a sequential encoder and an attention module. To locate the tables precisely, we design a text-classification task, classifying the text blocks into 4 categories according to their semantic roles in the tables. Experiments are conducted on both the text blocks parsed from PDF and extracted by open-source OCR tools, respectively. Compared to several state-of-the-art methods, TDeLTA achieves competitive results with only 3.1M model parameters on the large-scale public datasets. Moreover, when faced with the cross-domain data under the 0-shot setting, TDeLTA outperforms baselines by a large margin of nearly 7%, which shows the strong robustness and transferability of the proposed model.

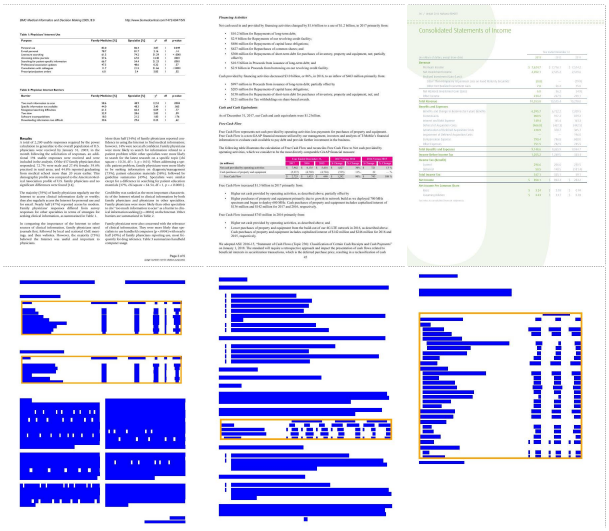

Figure 1: Tables in different styles and their text arrangement. The three images above show three tables in different styles, and the ones below show their text blocks with blue rectangles. The orange boxes denote the location of tables.

Introduction

As a popular means of storing and displaying structured data, tables are widely used in various documents including financial reports, scientific articles, invoices, etc. With the rapid growth of the number of digital documents on the Internet, how to teach machines to understand the tables and utilize massive valuable information in them, has become the focus of researchers. Table detection task aims to detect the boundaries of tables in documents. It can be a difficult problem due to the diversity of table styles. For example, some tables have complete borders between the cells of each row and column, while others may have only partial borders, such as the three-line tables in scientific articles. What's more, the diversity of table contents further complicates this issue.

Research on table detection tasks began in the last century. Early researchers used mostly heuristic-based methods (Gatos et al. 2005; Tupaj, Shi, and Chang 1996; Wang, Haralick, and Phillips 2001), which tend to require a lot of manual work and can hardly generalized on tables in different styles. Recently, with the great success of deep learning technology in computer vision applications (Tian et al. 2019, 2023; Zhang et al. 2020; Peng et al. 2023), many researchers perceive table detection task as a special object-detection problem, taking an document image as input and detecting bounding boxes of tables. These methods only treat tables as visual patterns, overlooking the fact that tables are a form of structured data, containing important information in their content organization. As visual models become larger and more complex, this not only increases the inference time and memory usage, limiting the practicality of this technique in many scenarios, such as mobile applications and large-scale data processing, but also makes the model more sensitive to changes in table visual styles. Although these methods

*Corresponding authors

achieved very high performance on some datasets, such as ICDAR 2013 (Göbel et al. 2013), they often fall into the trap of superficial table styles. Experiments show that the performance of these image-based methods will degrade significantly under the 0-shot setting when encountering the cross-domain tables.

To tackle this problem, we start from the essence of table, a set of text arranged in row and columns. As shown in Figure 1, although tables in the upper images have different styles, we can easily find the pattern of them from the text blocks below and then locate them. Motivated by this, we proposed **TDeLTA**, a light-weighted and robust **T**able **De**tection method based on **L**earning **T**ext **A**rrangement, which relies on the position information of text blocks rather than raw images. The fundamental idea of modeling text arrangement enables TDeLTA to be robust to variations in table styles, and demonstrates strong performance when facing out-of-distribution tables from different domains. Additionally, TDeLTA's simple input of text block positions results in a smaller model size, faster inference speed, and reduced memory usage. We also propose a text-classification task, classifying the text blocks into 4 categories according to their semantic roles, which aids in precise table localization and differentiation between adjacent tables.

Extensive experiments are conducted on two large-scale benchmark datasets (i.e. *PubTables-1M* and *FinTabNet*). It is demonstrated that TDeLTA can achieve competitive performance with only 3.1M parameters compared to the state-of-the-art methods. Moreover, under the 0-shot setting, TDeLTA outperforms these strong baselines by a large margin of nearly 7%, shows the best robustness and transferability when facing out-of-distribution tables in different domains.

The main contributions of this paper are as follows:

- Different from existing image-based methods, we proposed TDeLTA for table detection by learning text arrangement, which takes text blocks as input and can effectively handle out-of-distribution tables in various styles.
- To enable precise table localization and help distinguish adjacent tables, we propose a text-classification task, which classifies text blocks into 4 categories according to their semantic roles in tables.
- Extensive experiments are conducted on two large-scale benchmark datasets demonstrating the effectiveness and strong robustness of TDeLTA. It is also proven that text arrangement can essentially help the model overcome interference caused by table styles.

Related Work

Traditional Methods

Heuristic-based methods are among the earliest approaches for table detection task. These methods usually employed different visual cues like horizontal and vertical borderlines (Wang, Haralick, and Phillips 2001; Seo, Koo, and Cho 2015), key words (Tupaj, Shi, and Chang 1996), white space features (Pyreddy and Croft 1997; Akmal-Jahan and Ragel 2014), etc. to detect tables. These heuristic-based methods usually require extensive manual efforts to design rules and tune hyper-parameters, and can only work well on documents with uniform layouts.

Deep Learning Based Methods

With the rapid development of deep learning techniques and the emergence of large-scale table datasets, many deep learning-based methods have been proposed and achieved much better performance, which can be roughly divided into two categories: top-down methods and bottom-up methods.

Top-down methods usually treat table detection as a object detection problem and adapt state-of-the-art objection detection frameworks to predict table boundaries. Hao et al. and Yi et al. used R-CNN based model for table detection, however, the traditional region proposal generation still relied on heuristic rules and handcrafted features. Then, Vo et al. and Gilani et al. adopted Fast R-CNN and Faster R-CNN to detect tables, and Gilani et al. further proposed to use image transformation techniques, such as coloration and dilation, to enhance input documents. Huang et al. used a Yolo-based method. Saha, Mondal, and Jawahar and Prasad et al. introduced Mask R-CNN and Cascade Mask R-CNN to table detection task, respectively. Siddiqui et al. incorporated deformable convolution and deformable RoI Pooling operations to enhance the robustness of the model when facing geometric transformation problems. Ma et al. proposed CornerNet as a new region proposal network to generate higher quality table proposals for Faster R-CNN.

Bottom-up methods tend to group pixels or page object, such as word and text-line, into table regions. Some researchers view table detection as a semantic segmentation problem. Yang et al. tried to solve this problem in a multi-modal manner. They leveraged both visual features from images and linguistic features from text content to predict a pixel-level segmentation mask. Kavasidis et al. proposed a saliency-based FCN network performing multi-scale reasoning on visual cues followed by a fully-connected conditional random field (CRF) for localizing tables and charts in digital/digitized documents. Other researchers focused on page objects. Riba et al. and Holecek et al. both took each document as a graph, where each node represent a text-block and each represent a neighbouring relationship between two nodes. Holecek et al. used the coordinates of text-block and the number of characters in each text as the features of nodes. In addition to the geometrical features of the text position, Riba et al. also considered the texual features and image features. They both leveraged graph neural networks to encode the text blocks. However, these complex models together with multi-modal features failed to perform well on public datasets.

Methodology

Overview

Unlike previous works, we transform the table detection task into the classification of text blocks. As shown in Figure 2, we first extract the text blocks from the documents via OCR

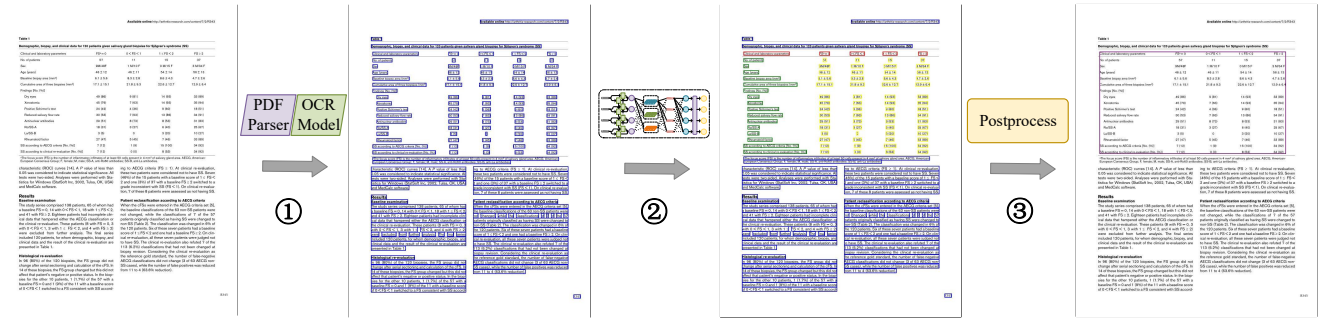

Figure 2: The flowchart of our method. First, we use a PDF parser or a OCR tool to extract text blocks from document pages. Then, we use TDeLTA to classify the text blocks into four categories. At last, we generate the table location with the classification results by a post-processing algorithm

tools or PDF parsers, depending on the type of input files. Then, TDeLTA learns the arrangement among text blocks, and classifies them into four categories, including row/column headers, content cells, and text outside tables. Consequently, the boundaries of the tables are generated with the classification results by a post-processing algorithm.

Model Architecture

As shown in Figure 3, TDeLTA consists of two modules, namely the Feature Extraction Module and the Contextual Attention Module.

Feature Extraction Module First, we get the position information of all text blocks from the document pages, represented by $\boldsymbol{D} = [bbox^1, bbox^2, ..., bbox^n]$, where n denotes the number of text blocks and $bbox^i$ denotes the coordinates of the i-th text blocks. Specifically, $bbox^i = [x_1^i, y_1^i, x_2^i, y_2^i]$, where (x_1^i, y_1^i) represents the coordinates of the upper-left corner of the i-th text block and (x_2^i, y_2^i) represents the coordinates of the lower-right corner.

Then, we calculate the coordinates of the midpoint, represented by (x_3^i, y_3^i), together with its width w and height h, as follows.

$$\begin{cases} x_3^i = (x_1^i + x_2^i)/2 \\ y_3^i = (y_1^i + y_2^i)/2 \\ w = x_2^i - x_1^i \\ h = y_2^i - y_1^i \end{cases} \tag{1}$$

After that, we normalize the coordinates as follows.

$$\begin{cases} \hat{x}_k^i = x_k^i / w_D & k \in \{1, 2, 3\} \\ \hat{y}_k^i = y_k^i / h_D & k \in \{1, 2, 3\} \\ \hat{w} = w / w_D \\ \hat{h} = h / h_D \end{cases} \tag{2}$$

where (w_D, h_D) denotes the width and height of the whole document page. Finally, we can get the model input consisting of 8-dimensional features $[\hat{x}_1^i, \hat{y}_1^i, \hat{x}_2^i, \hat{y}_2^i, \hat{x}_3^i, \hat{y}_3^i, \hat{w}, \hat{h}]$ from the coordinates of the i-th text block.

Contextual Attention Module The contextual attention module is used to learn the text arrangement, and classify text blocks into 4 pre-defined categories. It has four components, which are the embedding layer, the BiLSTM encoder (Huang, Xu, and Yu 2015), the multi-head attention (Vaswani et al. 2017) and the linear classifier.

Embedding Layer Given an input tensor $\mathbf{I} \in \mathbb{R}^{n \times 8}$, we adopt a linear layer to map it to a high-dimensional embedding space as follows:

$$\mathbf{E} = W_e \cdot \mathbf{I} + b_e \tag{3}$$

BiLSTM Encoder LSTM, as a classic model for sequence encoding, has been widely applied in various tasks. Bidirectional LSTM can learn temporal information from both directions simultaneously. Therefore, we have chosen it as the sequence encoder for TDeLTA. In the subsequent sections, we will provide detailed comparisons with other commonly used encoders.

Give an input $\mathbf{E} \in \mathbb{R}^{n \times l_I}$, we adopt N_L-layer bidirectional long short-term memory network (BiLSTM) to learn the spatial arrangement among text blocks.

$$\mathbf{H} = \mathrm{BiLSTM}(\mathbf{E}) \tag{4}$$

Thus, we can get the vector representation $\mathbf{H} \in \mathbb{R}^{n \times 2l_h}$ that contains spatial arrangement information, where l_h indicates the hidden size of BiLSTM in a single direction.

Multi-head Attention The multi-head attention mechanism has been proven to be effective in mining the connections among tokens in the input sequence on various tasks. Therefore, we use it to further learn the two-dimensional spatial arrangement among text blocks in document pages.

Then, we use the multi-head attention mechanism to further learn the global arrangement information of the text block.

$$\mathbf{A} = \mathrm{MHA}(\mathbf{H}) \tag{5}$$

where MHA denotes the multi-head attention mechanism.

Linear Classifier Since we get the arrangement-aware vector representation $\mathbf{A} \in \mathbb{R}^{n \times l_A}$, where l_A denotes the hidden size of the output of multi-head attention layer, a linear classifier layer is used to classify the text blocks as follows:

$$\hat{Y} = \mathrm{argmax}(\mathrm{softmax}(\mathrm{W_{cls}} \cdot \mathbf{A} + \mathrm{b_{cls}})) \tag{6}$$

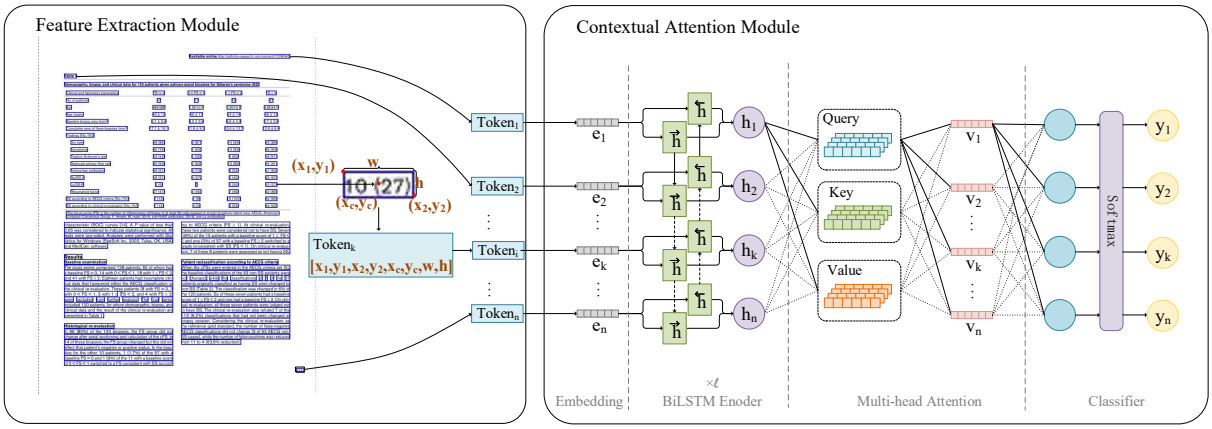

Figure 3: Model Architecture of TDeLTA.

The text blocks in document pages are now classified into four categories.

Loss Function As we treat it as a classification task, we choose to use Cross Entropy loss as the loss function of TDeLTA.

$$\mathcal{L} = -\frac{1}{n}\sum_{i=1}^{n} y^i log(\hat{y^i}) + (1-y^i) log(1-\hat{y^i}) \qquad (7)$$

Post-processing Algorithm

After getting the classification results of text blocks, we employ a post-processing algorithm to generate table boundaries. The post-processing algorithm consists of two steps: 1) aggregating the neighboring text classified as being inside tables (including three categories) to form preliminary results., and 2) generating the top and left boundaries of the tables based on row headers and column headers, and then splitting the preliminary results to generate the final detected boundaries. We designed the second step to handle cases where multiple tables are adjacent.

Experiments

Datasets

PubTables-1M[1] (2021) contains 460,589 fully-annotated pages for training; 57,591 for validation; and 57,125 for testing. This dataset focuses on tables in scientific domain, as they choose the PMCOA corpus to be their data source, which consists of millions of publicly available scientific articles. The authors generated the labels automatically by matching the text content from PDF documents and XML documents.

FinTabNet[2] (2020) contains 48,001 fully-annotated pages for training; 5,943 for validation; and 5,903 for testing. This dataset focuses on complex tables from the annual reports of the S&P 500 companies. Financial tables often have very different styles compared with those in scientific documents, like fewer borderlines and more diverse background colors. Therefore, we report results on the test set of FinTabNet under 0-shot setting to verify the robustness and transfer ability of TDeLTA when facing out-of-distribution tables.

Metrics

Like objection detection tasks, we take Intersection-over-Union(IoU) as our evaluation metrics, where a detection bounding box is considered as a positive result when the proportion of its intersection area with ground truth is larger than the threshold. All results in this experiment were calculated by the open-source Python package Pycocotools.

Baselines

We compare TDeLTA with several strong baselines including Table-DETR (Smock, Pesala, and Abraham 2021), CascadeTabNet (Prasad et al. 2020), DiT (Li et al. 2022) and YoLov7 (Wang, Bochkovskiy, and Liao 2022). Table-DETR is a variant of DETR (Carion et al. 2020) on table-related tasks, and the best baseline reported in the original paper of Pubtables-1M. Here we use its officially published weights. CascadeTabNet is a widely-used method based on Cascade R-CNN (Cai and Vasconcelos 2018) and have achieved great performance in many table detection benchmarks. YoLov7 is one of the state-of-the-arts in object detection task. We trained these two baseline models on Pubtables-1M by ourselves. DiT, LayoutLMv3 and StrucTexTv2 are all document-oriented visual backbone pretrained on large-scale unlabeled document images. We fine-tuned DiT on PubTables-1M and utilized the officially released weights fine-tuned on PubLayNet for the other two models, since both datasets derive from the PubMed corpus, we assume they have similar data distributions. Therefore, for LayoutLMv3 and StrucTexTv2, we only conduct comparisons under the 0-shot setting on the FinTabNet dataset.

Implementation Details

For BiLSTM layers, we set the hidden size l_h to 128 and the number of layers N_L to 8. For the multi-head attention layer,

[1]https://github.com/microsoft/table-transformer
[2]https://developer.ibm.com/exchanges/data/all/fintabnet/

Methods	IoU@0.5			IoU@0.6			Avg. (IoU@0.5-0.95)			#Parm (M)
	P	R	F1	P	R	F1	P	R	F1	
Table-DETR (2021)	**99.50**	**99.97**	**99.74**	**99.50**	**99.96**	**99.73**	97.08	98.57	97.82	28.9
CascadeTabNet (2020)	99.00	99.92	99.46	99.00	99.91	99.45	**98.99**	**99.83**	**99.41**	82.6
DiT (2022)	98.99	99.93	99.47	98.99	99.92	99.46	98.96	99.81	99.38	113.2
YoLov7 (2022)	99.00	99.90	99.45	99.00	99.88	99.44	98.77	99.52	99.14	36.9
TDeLTA	98.53	99.71	99.12	98.47	99.65	99.05	98.02	99.39	98.70	3.1
w/ Transformer	98.49	99.53	99.01	98.41	99.45	98.92	97.80	99.11	98.45	4.7
w/ LSTM	97.84	99.13	98.48	96.67	98.92	97.78	94.62	97.66	96.11	4.2
w/ BiGRU	98.26	99.39	98.82	98.18	99.32	98.75	96.49	98.46	97.47	2.4

Table 1: Performance comparison on PubTables-1M. We report results of TDeLTA with different encoders.

we set the number of heads N_h to 4 and the output size l_A to 128 which equals to l_h. Ablation study of hyperparameters can be found in the following section.

For each sequence encoder compared, we set the hidden size to 128 and the number of layers to 8, except for LSTM, which has a hidden size of 256. This ensures that the size of each model is similar for a fair comparison.

All the weights are initialized with a uniform distribution. During training, we employ an Adam (Kingma and Ba 2014) optimizer for 300 epochs using a cosine decay scheduler with 10% warmup steps. A batch size of 256 and an initial learning rate of 0.001 are used. We implement our approach based on PyTorch and conduct experiments on Nvidia Tesla v100 32G GPUs.

Experimental Results

Results on PubTables-1M As shown in Table 1, TDeLTA and all the strong baselines achieve very high performance on Pubtables-1M, with the average F1 scores around 99%. This is because the documents in Pubtables-1M come from a single source, and the table styles are relatively uniform, making it simpler for these strong baselines. TDeLTA achieves competitive results while the number of parameters is just a fraction of other methods.

We also compare the BiLSTM encoder with other sequential encoders. (i) LSTM exhibits the poorest performance among the encoders, primarily due to its unidirectional nature. (ii) BiGRU achieves a slight lower performance with just 77% of the parameters (2.4M/3.1M). (iii) Transformer relies solely on positional embedding to learn sequential information. BiLSTM, by requiring tokens to be inputted in a specific order, exhibits heightened sensitivity to sequential features. Furthermore, we have incorporated a multi-head attention module in TDeLTA, allowing other sequential encoders to harness the advantages of Transformer.

Results on FinTabNet To better verify the robustness of TDeLTA on cross-domain data, we directly test these models on the test set of FinTabNet without fine-tuning. Results are listed in Table 2. Among all the baselines, it is reasonable that DiT and LayoutLMv3 show the best robustness since they have the most parameters and are pre-trained on large-scale unlabeled document images. Although CascadeTabNet achieves the best result on PubTables-1M, its performance decreased severely, and Table-DETR and YoLov7 have also exhibited similar trends. The decline in performance demonstrates that these models excessively focus on the style of tables and overfit to the data in the training set, resulting in poor performance when faced with out-of-distribution data. In contrast, TDeLTA achieves the best performance with the fewest parameters and without using any external data, and the gap becomes larger as the IoU threshold increased, with an average F1 score surpassing DiT by more than 7%. This demonstrates that TDeLTA can effectively learn the intrinsic features of tables and exhibits robustness across different table styles.

Considering that in many cases only document images can be obtained without the annotations of text blocks, to demonstrate the practicality of TDeLTA, we utilize the lightweight open-source OCR tool PaddleOCR[3] to perform text detection on the images from FinTabNet, thereby getting model input from raw images. As we can see in Table 2, under this setting, TDeLTA's performance only experienced a minor decrease, while still significantly outperforming baseline models. This demonstrates the robustness of TDeLTA, as it does not rely on high-quality text position annotations and can be effectively applied in various scenarios. At lower IoU threshold values, it can be observed that TDeLTA performs better when using OCR results as input slightly. This could be attributed to that some trivial text was missed by the OCR tool, which may introduce interference to the model.

Ablation Study

To further study the influence of model scale on the performance of TDeLTA, we conduct ablation studies on the number of layers and the hidden size of the BiLSTM encoder.

Ablation study on the number of layers. We first keep the hidden size of BiLSTM as 128 and set the number of BiLSTM layers to 2, 4, 6, 8, 12, separately. Results are listed in Table 3. We can observe that as the number of layers increases, the performance of TDeLTA gradually improves on both datasets, and it stabilizes when approaching 8 layers. However, when the number of layers further increases to 12, the performance on the PubTables-1M slightly decreases. This indicates that increasing the number of layers can enhance TDeLTA's fitting ability and reach saturation around 8 layers. Furthermore, it can be seen that when the number of

[3]https://github.com/PaddlePaddle/PaddleOCR

Methods	IoU@0.5			IoU@0.6			Avg. (IoU@0.5-0.95)			#Parm (M)
	P	R	F1	P	R	F1	P	R	F1	
Table-DETR (2021)	53.20	79.43	63.72	48.75	72.60	58.33	38.81	59.46	46.96	28.9
CascadeTabNet (2020)	47.05	51.72	49.27	44.15	49.78	46.80	37.23	43.37	40.06	82.6
DiT (2022)	81.77	89.61	85.51	78.62	87.01	82.60	69.11	78.37	73.45	113.2
YoLov7 (2022)	64.00	80.59	71.34	62.62	79.66	70.12	53.14	69.96	60.40	36.9
LayoutLMv3* (2022)	78.49	88.82	83.34	76.22	87.27	81.37	65.02	77.90	70.88	133.0
StrucTexTv2* (2023)	64.81	92.19	76.11	60.70	89.23	72.25	43.90	68.82	53.61	50.2
PDF Parser										
TDeLTA	**82.82**	**92.00**	**87.17**	**80.86**	**90.61**	**85.46**	**74.88**	**86.93**	**80.46**	3.1
w/ Transformer	74.32	86.42	79.92	71.47	84.86	77.59	65.07	80.55	71.99	4.7
w/ LSTM	66.29	81.74	73.21	61.73	78.87	69.25	50.27	70.25	58.60	4.2
w/ BiGRU	78.42	88.71	83.25	75.08	86.85	80.54	64.26	79.77	71.18	2.4
Open-source OCR Tool										
TDeLTA	**83.46**	**91.60**	**87.34**	**81.71**	**90.52**	**85.89**	**72.72**	**84.81**	**78.30**	3.1
w/ Transformer	76.43	87.66	81.66	73.47	85.93	79.21	63.58	79.09	70.49	4.7
w/ LSTM	74.67	85.64	79.78	69.83	82.76	75.75	55.78	72.63	63.10	4.2
w/ BiGRU	80.08	89.24	84.40	75.54	86.99	80.86	62.40	77.76	69.24	2.4

Table 2: Performance comparison on FinTabNet under 0-shot setting. We report results of TDeLTA that applying different techniques to extract text blocks (i.e. PDF parsers and open-source OCR tools). * indicates fine-tuned on PubLayNet.

N_L	PubTables-1M		FinTabNet		#Parm (M)
	AP	AR	AP	AR	
2	97.23	98.88	67.32	81.94	0.7
4	97.40	98.97	70.91	84.07	1.5
6	97.69	99.17	72.49	84.54	2.3
8	**98.02**	**99.39**	74.89	86.93	3.1
12	97.91	99.20	**76.08**	**87.10**	4.7

Table 3: Ablation study on the number of layers of BiLSTM

l_h	PubTables-1M		FinTabNet		#Parm (M)
	AP	AR	AP	AR	
16	88.54	94.36	41.66	62.69	0.051
32	96.08	98.37	71.81	84.18	0.199
64	97.87	99.21	73.50	85.70	0.783
128	**98.02**	**99.39**	**74.89**	**86.93**	3.1

Table 4: Ablation study on the hidden size of BiLSTM

layers is 4 and the number of parameters is 1.5M, the performance on FinTabNet already surpasses all baseline models. We set the number of layers to 8.

Ablation study on the hidden size. We fixed the number of layers to 8 and set the hidden size to 16, 32, 64, 128 respectively to conduct ablation experiments. Results are listed in Table 4. We can see that as the hidden size increases, the performance of TDeLTA gradually improves on both datasets. When the hidden size approaches 128, the performance improvement becomes more gradual. From Table 4, it can be observed that when the hidden size is set to 32, the model with only 119k parameters already outperforms all basline models on FinTabNet. This proves the strong transferability and stability of TDeLTA. We set the hidden size to 128.

Visualization of Attention

For a deeper look, we visualized the attention scores for two sampled documents. As shown in Figure 4, the first column represents the original inputs. The text block in blue denotes the current step, and each subfigure displays the attention scores of all other text blocks with respect to the blue one, indicating the level of attention the blue text block pays to others in the document. As we can see, different categories of text blocks focus on global information differently. The row headers is the forefront part of the table, which pays more attention to the delineation area; The column headers determines themselves based on the information from row headers; content cells are more concerned with neighboring text arranged in rows and columns; text outside the tables are of little concern for the information within the table. This demonstrates that TDeLTA has learned the semantic roles of each text within the table and their interdependencies, and can utilize them to obtain representations for different texts, indicating that TDeLTA truely understands the tables.

Case Study

We sample three documents with tables in different styles from the test set of FinTabNet and the results are shown in Figure 5. Each row in the figure represents a sampled document. Columns 1-4 show the results of baseline models, respectively. Columns 5-6 represent TDeLTA's classification and detection results. The detection boxes in red represent the bad cases, while those in green are correct results.

Figure 5 (a) shows that the background pattern of the image can seriously affect the detection results of the image-based baselines. And the horizontal line above the table also misleads them. In Figure 5 (b), there are some texts

Figure 4: Visualization of attention scores. The first column represents the original inputs. The text block in blue denotes the current step, and each subfigure displays the attention scores of all other text blocks with respect to the blue one.

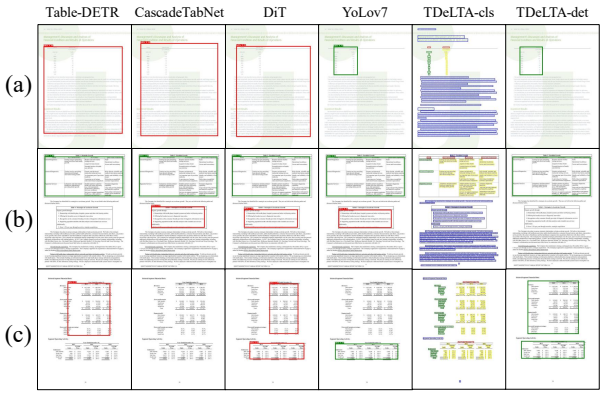

Figure 5: Experimental results on FinTabNet under 0-shot setting. The red boxes represent the bad cases, while the green ones denote correct detection.

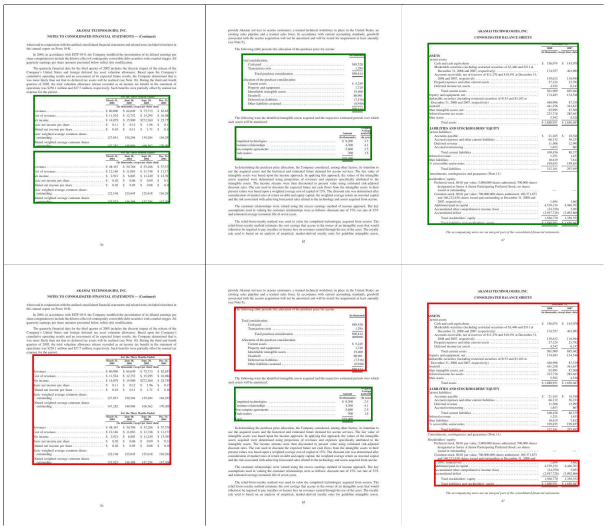

Figure 6: Bad cases of TDeLTA from the test set of FinTabNet. The images above exhibit the ground truth, and the ones below shows the detection result.

surrounded by a rectangular box, and three of these baselines incorrectly detect it as a table, which shows that they learned to detect rectangles, not tables. Figure 5 (c) illustrates the poor performance of these baseline models when faced with borderless tables. All these cases emphasizes that these image-based methods do not truly understand tables but rather overfit to shallow visual features. The results in column 5-6 show that TDeLTA can be robust to variations in table styles and detect correctly.

Error Analysis

As show in Figure 6, the major error types are summarized as follows: a) When two adjacent tables have identical formats, they may be detected as one because their combination also meets the criteria of aligning text in rows and columns; b) When there is a separate short text above the table, the model may mistakenly interpret it as the title and include it within the scope of the table; c) When a table contains long text content, it could be incorrectly divided because the model misinterprets these long sentences as paragraphs between tables. These cases indicate that although learning text arrangement can effectively improve the model's robustness, it is still insufficient to fully address the diversity of tables. Combining multiple modalities of information will help, such as textual content and visual cues.

Conclusion

In this work, we focus on improving the stability of table detection models in the 0-shot setting when dealing with tables of different styles. We propose a light-weight and robust table detection method called TDeLTA, which learns text arrangement, classifies text blocks, and generates table boundaries. Experimental results show that TDeLTA, with only 3.1M parameters, achieves performance comparable to the state-of-the-arts, and shows the best robustness when faced with the cross-domain data under the 0-shot setting.

Acknowledgements

We thank the reviewers for their thoughtful and constructive comments. This work was supported by the National Key R&D Program of China (2022ZD0116002), Science and Technology Planning Project of Shenzhen (KCXFZ20230731093001002), Natural Science Foundation of China (62276075, 62102118), the projects of Shenzhen (GXWD20231129115148001, BA11409025), the project of Research on Large Model Methods for Understanding Complex Document Images, and China Mobile Information Technology Co.,Ltd. It was also supported by Music Style Transfer for Anxiety Therapy (P0042962), Artificial Intelligence and Robotics (AIR) Group: Artifical Intelligence Art (P0033738).

References

Akmal-Jahan, M. A. C.; and Ragel, R. G. 2014. Locating Tables in Scanned Documents for Reconstructing and Republishing (ICIAfS14). *CoRR*, abs/1412.7689.

Cai, Z.; and Vasconcelos, N. 2018. Cascade r-cnn: Delving into high quality object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 6154–6162.

Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; and Zagoruyko, S. 2020. End-to-end object detection with transformers. In *European conference on computer vision*, 213–229. Springer.

Gatos, B.; Danatsas, D.; Pratikakis, I.; and Perantonis, S. J. 2005. Automatic Table Detection in Document Images. In *International Conference on Advances in Pattern Recognition*.

Gilani, A.; Qasim, S. R.; Malik, M. I.; and Shafait, F. 2017. Table Detection Using Deep Learning. *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, 01: 771–776.

Göbel, M. C.; Hassan, T.; Oro, E.; and Orsi, G. 2013. ICDAR 2013 Table Competition. *2013 12th International Conference on Document Analysis and Recognition*, 1449–1453.

Hao, L.; Gao, L.; Yi, X.; and Tang, Z. 2016. A Table Detection Method for PDF Documents Based on Convolutional Neural Networks. *2016 12th IAPR Workshop on Document Analysis Systems (DAS)*, 287–292.

Holecek, M.; Hoskovec, A.; Baudis, P.; and Klinger, P. 2019. Table Understanding in Structured Documents. *2019 International Conference on Document Analysis and Recognition Workshops (ICDARW)*, 5: 158–164.

Huang, Y.; Lv, T.; Cui, L.; Lu, Y.; and Wei, F. 2022. LayoutLMv3: Pre-training for Document AI with Unified Text and Image Masking. *Proceedings of the 30th ACM International Conference on Multimedia*.

Huang, Y.; Yan, Q.; Li, Y.; Chen, Y.; Wang, X.; Gao, L.; and Tang, Z. 2019. A YOLO-Based Table Detection Method. *2019 International Conference on Document Analysis and Recognition (ICDAR)*, 813–818.

Huang, Z.; Xu, W.; and Yu, K. 2015. Bidirectional LSTM-CRF Models for Sequence Tagging. *ArXiv*, abs/1508.01991.

Kavasidis, I.; Palazzo, S.; Spampinato, C.; Pino, C.; Giordano, D.; Giuffrida, D.; and Messina, P. 2018. A Saliency-based Convolutional Neural Network for Table and Chart Detection in Digitized Documents. *ArXiv*, abs/1804.06236.

Kingma, D. P.; and Ba, J. 2014. Adam: A Method for Stochastic Optimization. *CoRR*, abs/1412.6980.

Li, J.; Xu, Y.; Lv, T.; Cui, L.; Zhang, C.; and Wei, F. 2022. DiT: Self-supervised Pre-training for Document Image Transformer. *Proceedings of the 30th ACM International Conference on Multimedia*.

Ma, C.; Lin, W.; Sun, L.; and Huo, Q. 2022. Robust Table Detection and Structure Recognition from Heterogeneous Document Images. *Pattern Recognit.*, 133: 109006.

Peng, B.; Tian, Z.; Wu, X.; Wang, C.; Liu, S.; Su, J.; and Jia, J. 2023. Hierarchical Dense Correlation Distillation for Few-Shot Segmentation.

Prasad, D.; Gadpal, A.; Kapadni, K.; Visave, M.; and Sultanpure, K. A. 2020. CascadeTabNet: An approach for end to end table detection and structure recognition from image-based documents. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2439–2447.

Pyreddy, P.; and Croft, W. B. 1997. TINTI: A System for Retrieval in Text Tables TITLE2:.

Riba, P.; Goldmann, L.; Terrades, O. R.; Rusticus, D.; Fornés, A.; and Lladós, J. 2022. Table detection in business document images by message passing networks. *Pattern Recognit.*, 127: 108641.

Saha, R.; Mondal, A.; and Jawahar, C. V. 2019. Graphical Object Detection in Document Images. *2019 International Conference on Document Analysis and Recognition (ICDAR)*, 51–58.

Seo, W.; Koo, H. I.; and Cho, N. I. 2015. Junction-based table detection in camera-captured document images. *Int. J. Document Anal. Recognit.*, 18(1): 47–57.

Siddiqui, S. A.; Malik, M. I.; Agne, S.; Dengel, A. R.; and Ahmed, S. 2018. DeCNT: Deep Deformable CNN for Table Detection. *IEEE Access*, 6: 74151–74161.

Smock, B.; Pesala, R.; and Abraham, R. 2021. PubTables-1M: Towards comprehensive table extraction from unstructured documents. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 4624–4632.

Tian, Z.; Chen, P.; Lai, X.; Jiang, L.; Liu, S.; Zhao, H.; Yu, B.; Yang, M.; and Jia, J. 2023. Adaptive Perspective Distillation for Semantic Segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 45(2): 1372–1387.

Tian, Z.; Shu, M.; Lyu, P.; Li, R.; Zhou, C.; Shen, X.; and Jia, J. 2019. Learning Shape-Aware Embedding for Scene Text Detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Tupaj, S.; Shi, Z.; and Chang, D. H. 1996. Extracting Tabular Information From Text Files.

Vaswani, A.; Shazeer, N. M.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is All you Need. *ArXiv*, abs/1706.03762.

Vo, N. D.; Nguyen, K.-D.; Nguyen, T. V.; and Nguyen, K. 2018. Ensemble of Deep Object Detectors for Page Object Detection. *Proceedings of the 12th International Conference on Ubiquitous Information Management and Communication*.

Wang, C.-Y.; Bochkovskiy, A.; and Liao, H.-Y. M. 2022. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *ArXiv*, abs/2207.02696.

Wang, Y.; Haralick, R. M.; and Phillips, I. T. 2001. Automatic table ground truth generation and a background-analysis-based table structure extraction method. *Proceedings of Sixth International Conference on Document Analysis and Recognition*, 528–532.

Yang, X.; Yumer, E.; Asente, P.; Kraley, M.; Kifer, D.; and Giles, C. L. 2017. Learning to Extract Semantic Structure from Documents Using Multimodal Fully Convolutional Neural Networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 4342–4351.

Yi, X.; Gao, L.; Liao, Y.; Zhang, X.; Liu, R.; and Jiang, Z. 2017. CNN Based Page Object Detection in Document Images. *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, 01: 230–235.

Yu, Y.; Li, Y.; Zhang, C.; Zhang, X.; Guo, Z.; Qin, X.; Yao, K.; Han, J.; Ding, E.; and Wang, J. 2023. StrucTexTv2: Masked Visual-Textual Prediction for Document Image Pre-training. *ArXiv*, abs/2303.00289.

Zhang, L.; Chen, Q.; Hu, B.; and Jiang, S. 2020. Text-guided neural image inpainting. In *Proceedings of the 28th ACM international conference on multimedia*, 1302–1310.

Zheng, X.; Burdick, D.; Popa, L.; and Wang, N. X. R. 2020. Global Table Extractor (GTE): A Framework for Joint Table Identification and Cell Structure Recognition Using Visual Context. *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 697–706.