

DeepTabStR: Deep Learning based Table Structure Recognition

Shoaib Ahmed Siddiqui^{*†}, Imran Ali Fateh^{*†}, Syed Tahseen Raza Rizvi^{*†}, Andreas Dengel^{*†}, Sheraz Ahmed[†]

^{*}TU Kaiserslautern, Kaiserslautern, Germany

[†]German Research Center for Artificial Intelligence (DFKI), Kaiserslautern, Germany

Email: *firstname.lastname@dfki.de*

Abstract—This paper presents a novel method for the analysis of tabular structures in document images using the potential of deformable convolutional networks. In order to assess the suitability of the model to the task of table structure recognition, most of the prior methods have been tested on the smaller ICDAR-13 table structure recognition dataset comprising of just 156 tables. We curated a new image-based table structure recognition dataset, *TabStructDB*², comprising of 1081 tables densely labeled with row and column information. Instead of collecting new images for this purpose, we leveraged the famous Page-Object Detection dataset from ICDAR-17, and added structural information for all the tabular regions present in the dataset. This new publicly available dataset will enable the development of more sophisticated table structure recognition techniques in the future. We performed extensive evaluation on the two datasets (ICDAR-13 and *TabStructDB*) including cross-dataset testing in order to evaluate the efficacy of the proposed approach. We achieved state-of-the-art results with deformable models on ICDAR-13 with an average F-Measure of 92.98% (89.42% for rows and 96.55% for columns) and report baseline results on *TabStructDB* for guiding future research efforts with an F-Measure of 93.72% (91.26% for rows and 95.59% for columns). Despite promising results, structural analysis of tables with arbitrary layouts is still far from achievable at this point.

Keywords-Document Analysis, Table Structure Recognition, Table Understanding, Deformable FPN, Convolutional Neural Networks

I. INTRODUCTION

Significant efforts have been made in the past for automated extraction of information from documents [1], [2], [3], [4], [5]. Tabular structure embedded in documents is one of the most important mediums of communicating important information, specifically for financial and scientific records. This information is usually of high interest. Automatic extraction of such information for the purpose of digitization/processing is highly valuable for organizations [2]. Different methods have been proposed in the past to extract this information automatically from documents ranging from hand-designed heuristics to data-driven methodologies [1], [2].

The complete problem of table understanding is comprised of two sub-problems [6]. The first problem is the detection of the table itself [7], [3], [2]. Once a table is detected, the structure of the table is analyzed [2]. Most of the prior methods for the analysis of tabular structures rely on the availability of born-digital PDFs [1]. This allows the

direct extraction of textual regions. However, this limits the applicability of these systems to only PDFs where many of the documents are scanned or present in the form of images [2].

Training data-driven models for the task of table structure analysis requires access to a large amount of labeled data, specifically for deep models [2]. The largest publicly available dataset for this purpose is ICDAR-13 table structure recognition dataset comprising of 156 tables extracted from 238 pages (67 PDF documents) [6]. The dataset is comprised of labels for cells. Schreiber et al. (2017) [2] translated the dataset to labels for rows and columns in order for the image-based models to be trained. Despite being extensively labeled, the dataset itself is quite small compared to current demands. Segregation into the corresponding train and test sets specifically result in only a small number of images for testing. Schreiber et al. (2017) [2] used only 31 images to test their model.

The ICDAR-17 Page-Object Detection dataset is comprised of 2417 document images and 1081 tabular structures. In order to deal with the problem of data scarcity, we hand labeled all the tabular structures in the ICDAR-17 POD dataset with row and column information for image-based table structure recognition task². The dataset will enable further developments in the domain of table structure recognition, specifically for data-driven approaches.

The problem of table structure recognition (cell identification) can be decomposed into the problem of identification of the corresponding rows and columns. Once the rows and columns are identified, they can be coupled together to identify the cells. However, this simple assumption fails for hierarchical columns (which have a row/column span of more than one). Therefore, both datasets (ICDAR-13 labeled for rows/columns by Schreiber et al. [2] and our custom ICDAR-17) neglect row/column span (hierarchical labels).

In contrast to the Fully-Convolutional Network (FCN) based formulation presented by Schreiber et al. (2017) [2], we treat the problem of row/column identification in a tabular structure as that of object detection where the document can be considered analogous to scene and row/column can be considered as analogous to objects. With an FCN based approach, the system heavily relies on post-processing in order to filter out the regions where no ruling lines or textual content is present [2]. On the other hand, object detection

based approach can directly regress for the bounding boxes without having to rely on sophisticated post-processing techniques. The proposed approach, DeepTabStR (Deep Table Structure Recognizer), leverages the potential of deformable convolution operation to tackle the structure recognition problem [8], [7]. We also test the proposed approach on a new large dataset that we self-labeled (TabStructDB) along with the publicly available ICDAR-13 dataset [6] in order to identify the efficacy of the proposed approach in real-world. In particular, the contributions of this paper are as follows:

- We introduced a new dataset for table structure recognition comprising of 1081 tabular regions densely labeled with row and column information called as TabStructDB². This dataset will enable the development of more sophisticated methods for the task of table structure recognition in the future.
- We formulated the problem of table structure recognition as an object detection problem and leverage the potential of deformable convolution operation for this task.
- We performed an exhaustive evaluation on the two different datasets including cross-dataset evaluation and achieved state-of-the-art structure recognition results on the publicly available ICDAR-13 dataset (considering the metrics for rows and columns) while also setting a baseline for TabStructDB.

The rest of the paper is structured as follows. We first provide a brief recapitulation of the previous work in the direction of table structure analysis in Section II. We then provide details regarding the datasets including the one we curated ourselves in Section III. We describe the proposed approach (DeepTabStR) in detail in Section ???. Finally, we present our results in Section V along with a brief discussion followed by the concluding remarks in Section VI.

II. LITERATURE REVIEW

Despite a vast amount of literature on the topic of document analysis [1], [9], [2], [10], [3], [4], [5], [7], and a range of methods proposed for the task of table detection [2], [3], [7], there have been only a modest number of attempts for the full table understanding problem which is much more challenging as compared to the prior. Kieninger and Dengel (1999) [1], who are the pioneers of the work on table structure analysis, proposed the T-Recs system which initially grouped words into columns by estimating their horizontal ruling lines, These horizontal ruling lines were then divided into cells based on the column margins. Wang et al. (2004) [9] proposed a system which was similar to the X-Y cut algorithm. The probabilities were computed based on data, hence, making the system data-driven.

One of the major benchmarks for the task of table structure recognition was established by the table structure recognition competition organized in ICDAR-13 [6]. The participants were required to detect the cells present in a

table which includes information regarding its location (row and column), content, as well as the row and column span. Based on this information, an adjacency list was formulated which was in turn used for computing cell-level statistics. Computation of the adjacency list relies heavily on perfect extraction of the textual content. This perfect extraction is almost impossible for image-based systems, therefore, adjacency list based cell metrics inherently penalizes image-based recognition systems. The only image-based system in the competition achieved significantly poor scores as compared to the rest of the participants due to additional errors incurred through OCR along with the increase in the complexity of the task itself [6].

Klampfl et al. (2014) [11] presented an unsupervised learning approach along with a combination of hand-crafted heuristics for the detection of tables along with the corresponding analysis of the tabular structure in PDF documents. Kasar et al. (2015) [12] introduced a table information extraction system which leveraged a query-based approach to selectively extract information from tabular structures. The users were required to provide a query-pattern which was transformed into an attributed relational graph. The generated graph was matched with similar graphs in the document using a fast graph matching technique to retrieve other similar graphs, providing the desired information from the table.

Shigarov et al. (2016) [10] explored the problem of table structure analysis by providing analyzing different algorithms, suitable thresholds and their rule bases to achieve reasonable performance. They made heavy use of meta-data available in born-digital PDFs such as font, font-size, their corresponding bounding boxes etc. They also developed custom heuristics for the extraction of relevant information from the tabular structures. Rastan et al. (2019) recently introduced the TEXUS framework [13], which recognizes table structure in a layout independent manner. The system is limited to born-digital PDFs.

All these methods were specifically developed leveraging the meta-data available in born-digital PDFs. Since DeepTabStR directly operates over images, all prior approaches are not directly comparable to our system. One of the recent image-based deep learning methods for table structure analysis has been proposed by Schreiber et al. (2017) [2]. They utilized the power of Fully-Convolutional Network (FCN) designed for the task of semantic segmentation for the identification of the corresponding rows and columns in a tabular structure. They detected the corresponding bounding boxes from the table using contour detection on the generated segmentation masks. On the other hand, we directly treat it as an object detection problem [8], [14], [15] allowing us to directly regress for the coordinates of the rows and columns instead of going through an intermediate representation and then post-processing it.

III. DATASETS

A. ICDAR-13 Table Structure Recognition Dataset¹

ICDAR-13 dataset is comprised of 67 PDF files. These PDF files contain a total of 238 pages, along with 156 tabular structures [6]. Following the work of Schreiber et al. (2017) [2], we converted the cell-based annotations to the corresponding annotations for rows and columns. We used the same train and test split as used by Schreiber et al. (2017) [2] in order to enable a direct comparison against their approach.

B. TabStructDB²

In ICDAR-17, a Page-Object Detection (POD) competition was organized where the task was to identify page objects in documents which includes tables, figures and equations in documents [16]. The dataset was composed of 2417 images in total, where 1600 images were used for training, while the rest of the 817 images were used for testing. We are introducing a new table structure recognition dataset, TabStructDB², where we labeled each tabular region present in the ICDAR-17 POD dataset with table structure information comprising of the row and column information. In contrast to the annotations generated by Schreiber et al. (2017) [2], we label the complete row regardless of the textual region for consistency. We also ignore hierarchical labels where multiple columns are nested and only mark one at the finest level. This is consistent with ICDAR-13 image-based table structure recognition dataset generated by Schreiber et al. (2017) [2] where row/column span (hierarchical labels) were also discarded. The training set of ICDAR-17 POD is comprised of 1600 images containing 731 tabular regions while the test set comprised of 817 images containing 350 tabular regions. We kept the same dataset split for consistency. Therefore, our dataset is composed of 731 training and 350 test table images.

It is important to mention that we found several cases in ICDAR-17 datasets where clear tables were left out. We marked some of them where a similar table was marked on the same page of the document but left out the ones where there were no markings on a particular page to minimize deviation from the original dataset. We found quite a large number of errors in the annotations provided for ICDAR-17 POD competition, as is common in any large dataset.

IV. METHOD

Object detection has achieved amazing advances in recent years [8], [15], [14]. With the synergy in the task of identification of objects in natural scene images and identification of rows and columns in a tabular structure, we propose the use of object detection models for the task of table structure recognition.

¹ICDAR-13 dataset is publicly available at: <https://bit.ly/2RLgFYu>

²TabStructDB is publicly available at: <https://bit.ly/2XonOEx>

Conventional convolution operation has a fixed receptive field. This fixed receptive field is problematic for layers on top of the feature hierarchy where features can be present at arbitrary scales along with arbitrary transformations. Siddiqui et al. [7] showed the effectiveness of the deformable convolutional network for the task of table detection. Therefore, we employ the deformable model family for the task of table structure recognition. We will now dive deep into the different components within the overall system pipeline presented in Fig. 1.

A. Deformable Convolution

Convolutional neural networks are based on the primitive convolution operation which operates as a sliding window over the input. This enables the convolution operation to share parameters at different locations in the image, hence, making it parameter efficient. The conventional 2-D convolution operation can be represented mathematically as:

$$(F * I)(i, j) = \sum_{m=-K}^K \sum_{n=-K}^K F(m, n) \times I(i - m, j - n) \quad \forall i = 1, \dots, H, \forall j = 1, \dots, W \quad (1)$$

where $*$ is used to denote the convolution operation, F denotes the filter which is learned using the data, I denotes the image, K denotes a value which is computed as $\lfloor |F|/2 \rfloor$ ($|F|$ denotes the size of the filter), H denotes the image height, W denotes the image width and i, j represents the location where the convolution operation is performed. Since the convolution operation takes a fixed window of size $|F| \times |F|$ into account, it fails to compensate for objects occurring at different scales along with different transformations. In order to deal with this issue of fixed receptive field, Dai et al. (2017) [8] proposed the deformable convolution operation. The deformable convolution operation uses extra offsets instead of using a fixed grid which allows the layer to adapt itself. These offsets are computed based on another set of convolutional layers, hence making them learnable. Since they are computed for every input, the generated offsets are also conditioned on the input allowing the network to adjust its receptive field based on the location and object in view. The deformable 2-D convolution operation can be expressed mathematically as:

$$(F \circ I)(i, j) = \sum_{m=-K}^K \sum_{n=-K}^K F(i, j) \times I(i - m + \delta_{i,j,m,n}^{vertical}, j - n + \delta_{i,j,m,n}^{horizontal}) \quad \forall i = 1, \dots, H, \forall j = 1, \dots, W \quad (2)$$

where \circ denotes the deformable convolution operation while all the mutual parameters are the same as Eq. 1. $\delta_{i,j,m,n}^{vertical}$

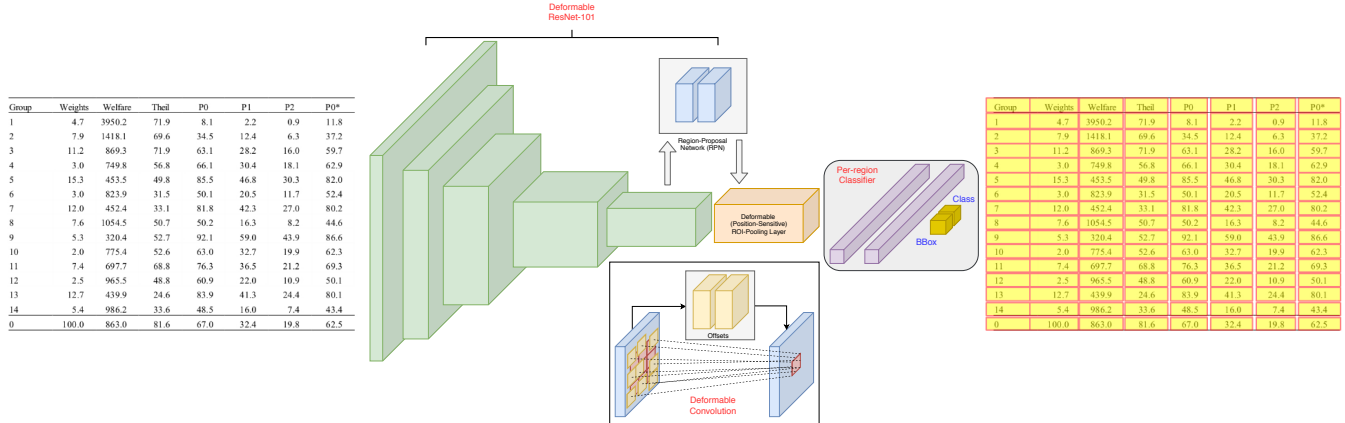


Figure 1: DeepTabStR: The proposed system pipeline

and $\delta_{i,j,m,n}^{horizontal}$ denotes the corresponding offsets generated. Since these offsets are generated by the convolutional layers, hence, they can be fractional and are implemented via bilinear interpolation [8]. Since there are separate convolutional layers for the generation of these offsets, the number of parameters in the network increases.

The per-region classification head at the end of the network (Fig. 1) expects a fixed-size input since it is comprised of fully-connected layers. On the other hand, the generated region-proposals vary in terms of size based on different bounding boxes. Girshik et al. (2015) [14] introduced ROI-pooling layer which was able to convert these region proposals into fixed feature volume while retaining differentiability. ROI-pooling is a core component for all region based detection methods [8], [14], [17], [18].

Let F be the feature map obtained from the network, (i_0, j_0) be the top-left corner of the ROI, and $w \times h$ be the ROI size, the conventional ROI-pooling layer converts the ROI to a fixed size of volume of $k \times k$. This operation can be represented mathematically as:

$$ROI_Pool(F, m, n) = \sum F(i_0 + i, j_0 + j) / n_{m,n},$$

$$\forall i = \{1, \dots, h \lfloor m \times (h/k) \rfloor \leq i < \lceil (m+1) \times (h/k) \rceil\},$$

$$\forall j = \{1, \dots, w \lfloor m \times (w/k) \rfloor \leq j < \lceil (m+1) \times (w/k) \rceil\}$$
(3)

where $n_{m,n}$ is the number of pixels in the bin (m, n) . If there are C input feature maps, the overall output from the layer will be $k \times k \times C$ which will be fed to the classification head. Just like its convolutional counterpart, deformable ROI-pooling has a fixed receptive field. In order to also enable the ROI-pooling layer to deal with objects at arbitrary scales, Dai et al. (2017) [8] similarly equipped the ROI-pooling layer with deformable property by again introducing the additional offsets. This can be represented as:

$$DeformableROI_Pool(F, m, n) =$$

$$\sum F(i_0 + i + \delta_{m,n}^{vertical}, j_0 + j + \delta_{m,n}^{horizontal}) / n_{m,n},$$

$$\forall i = \{1, \dots, h \lfloor m \times (h/k) \rfloor \leq i < \lceil (m+1) \times (h/k) \rceil\},$$

$$\forall j = \{1, \dots, w \lfloor m \times (w/k) \rfloor \leq j < \lceil (m+1) \times (w/k) \rceil\}$$
(4)

where all mutual parameters are the same as Eq. 3. $\delta_{i,j}^{vertical}$ defines the vertical offset while $\delta_{i,j}^{horizontal}$ defines the horizontal offset. These offsets are again fractional and implemented via bilinear interpolation.

B. Deformable Models

We evaluated the complete family of deformable models which includes Faster R-CNN, R-FCN and FPN [8]. In all deformable models, a deformable base model is used which was a ResNet-101 in our case. The model is pretrained on ImageNet dataset comprising of 1.2 million images [19]. This transforms the network into a generic feature extractor. Since deformable convolution is a memory intensive operation, only three layers in the network were transformed into their deformable variant. In the case of FPN, an additional fourth layer was also transformed to further improve the quality of the extracted features. All the deformable layers were located on top of the feature hierarchy as deformable convolution operation is mainly useful where detection of complete objects is desired [8]. In order to leverage a non-deformable pretrained ResNet-101 ImageNet model [20], the offsets are initialized with zero and adapted during training. Zero offsets translate to the regular convolutional grid, making it directly equivalent to the non-deformable variant.

The deformable Faster R-CNN adapts the original faster R-CNN framework [14] by using deformable ROI-pooling instead of the conventional ROI-pooling layer along with

the deformable ResNet-101 as the base model. Similarly, deformable R-FCN adapts the R-FCN framework [17] by equipping the model with deformable position-sensitive ROI-pooling layer instead of the normal position-sensitive ROI-pooling layer along with the deformable base model.

In order to further improve performance in detection of objects that are present at different scales, a common practice is to perform detection at several different scales and aggregate the predictions from all these different scales. In order to deal with this issue, Feature-Pyramid Networks (FPN) were proposed [18]. FPN uses a top-down pathway along with a bottom-up pathway. This enables the network to detect objects at multiple scales without multiple forward passes through the network. Deformable FPN uses deformable position-sensitive ROI-pooling layer and a deformable base model [8].

There are two distinct possibilities of training the model i.e. a separate model for both rows and columns or a single combined model. We trained both separate as well as combined models in order to establish a clear difference between the two approaches for this task.

V. EVALUATION

We evaluated DeepTabStR on the two available datasets i.e. ICDAR-13 table structure recognition dataset and TabStructDB described in detail in Section III. In addition to testing on the corresponding test set of a particular dataset, we also perform cross-dataset testing in order to get a real hint regarding the generalization capabilities of our models. The results from all the models including cross-dataset testing are presented in Table I.

We report the document averages where we first compute the precision, recall and F-Measure for every document separately followed by averaging over the entire dataset. This scheme avoids large influence originating from only one of the documents containing a large number of rows/columns by averaging over the entire dataset. The same evaluation scheme was used for the ICDAR-13 table structure recognition competition and Schreiber et al. (2017) [6], [2].

A. ICDAR-13

It is important to understand that the official ICDAR-13 dataset is labeled with cell-level information. Hence, the metrics reported in the original competition were cell-level metrics. On the other hand, we report metrics on rows and columns, which is a completely different direction. Therefore, methods operating on rows/columns including ours and others [2], cannot be compared with the entries from the competition.

We, therefore, compare our method with the only other image-based model presented by Schreiber et al. [2]. We used the same train/test split as theirs, making a direct comparison possible. The results from the comparative study are presented in Table II. It is evident from the table

that the proposed DeepTabStR based on deformable FPN comprehensively outperforms the previous approach.

For cross-dataset testing, we trained the model on TabStructDB and evaluated it on the entire as well as only the test set of ICDAR-13 dataset and vice versa. Training the combined model or separately for both classes resulted in a negligible difference in performance (~ 0.70 F-Measure) indicating that there is still a vital gap in the generalization capabilities of the system.

The first row in Fig. 2 and Fig. 3 presents a sample correct and incorrect detection from the ICDAR-13 dataset. In the case of incorrect detection, the system was unable to detect the two rows containing only the title as the network learned to separate out rows based on the complete line instead of just a single word. The presence of these single words is also quite common in the case of multi-line rows.

B. TabStructDB

We have reported baseline results on the TabStructDB. Training and testing the model on TabStructDB resulted in a high F-Measure (~ 0.93). However, if we tested the model on the ICDAR-13 dataset and evaluated on the complete TabStructDB, there was a very significant drop in performance (~ 0.70 F-Measure). A significant drop in performance during cross-dataset can be in part because of the different labeling schemes. ICDAR-13 is labeled with only the textual regions while TabStructDB is marked with the complete row/column regardless of the textual content.

A sample correct and incorrect detection is visualized in the second row of Fig. 2 and Fig. 3 respectively. It is clear from the incorrect detection that the system had a hard time telling apart a multi-line row as the system learned to segment rows without relying on the presence of ruling lines. Confusion in the case of multi-line rows was prevalent throughout the dataset.

VI. CONCLUSION

DeepTabStR leveraged the power of deformable convolution to achieve table structure recognition in documents. We also introduced a new image-based table structure recognition dataset (TabStructDB) comprising of 1081 tables densely labeled with information regarding the rows and columns of the table. We presented an exhaustive evaluation on the publicly available ICDAR-13 table structure recognition dataset along with the newly proposed TabStructDB. We achieved state-of-the-art results on the ICDAR-13 dataset with an average F-Measure of 92.98% and report baseline results on TabStructDB with an F-Measure of 93.72%. The obtained results advocate that DeepTabStR was indeed able to successfully segment the rows and columns in a wide range of documents.

Image-based ICDAR-13 and TabStructDB neglect the row/column span in the table. An important extension of DeepTabStR could be to directly regress for cells along

Training Dataset	Testing Dataset	Model	Training Method	Row			Column			Average F-Measure	
				Precision	Recall	F-Measure	Precision	Recall	F-Measure		
ICDAR-13 (Training set)	ICDAR-13 (Test Set)	Deformable FPN	Combined	0.8845	0.8945	0.8861	0.7858	0.4715	0.4922	0.6892	
			Separate	0.8949	0.8986	0.8942	0.9688	0.9630	0.9655	0.9298	
		Deformable Faster R-CNN	Combined	0.6651	0.1032	0.1510	0.9568	0.8335	0.8648	0.5079	
			Separate	0.8817	0.4097	0.4531	0.9520	0.9477	0.9497	0.7014	
		Deformable RFCN	Combined	0.8506	0.7564	0.7873	0.9156	0.8589	0.8810	0.8341	
			Separate	0.8835	0.8374	0.8568	0.9441	0.9624	0.9562	0.9065	
		TabStructDB (Complete)	Deformable FPN	Combined	0.5734	0.5934	0.5739	0.6152	0.5247	0.5095	0.5417
				Separate	0.6239	0.6781	0.6433	0.7665	0.7556	0.7498	0.6966
			Deformable Faster R-CNN	Combined	0.6563	0.1668	0.2119	0.7872	0.6803	0.7056	0.4587
				Separate	0.5545	0.2785	0.4531	0.7681	0.7489	0.7533	0.6032
			Deformable RFCN	Combined	0.5207	0.4063	0.4285	0.7950	0.6017	0.6428	0.5356
				Separate	0.6888	0.6303	0.6366	0.7308	0.7245	0.7132	0.6749
	TabStructDB (Test Set)	Deformable FPN	Combined	0.5777	0.5994	0.5789	0.6421	0.5682	0.5464	0.5626	
			Separate	0.6533	0.6935	0.6562	0.7641	0.7580	0.7457	0.7009	
		Deformable Faster R-CNN	Combined	0.6411	0.1669	0.2091	0.7976	0.6813	0.7088	0.4589	
			Separate	0.5492	0.2622	0.3009	0.7687	0.7462	0.7501	0.5255	
		Deformable RFCN	Combined	0.5110	0.4028	0.4246	0.7717	0.5675	0.6118	0.5182	
			Separate	0.7073	0.6401	0.6488	0.7201	0.7053	0.6934	0.6711	
	TabStructDB (Training set)	ICDAR-13 (Complete)	Deformable FPN	Combined	0.7793	0.7450	0.7618	0.7592	0.7331	0.7427	0.7522
				Separate	0.7808	0.7626	0.7699	0.7496	0.7416	0.7439	0.7569
			Deformable Faster R-CNN	Combined	0.6634	0.2163	0.2536	0.7340	0.7204	0.7227	0.4881
				Separate	0.6048	0.5507	0.5660	0.7378	0.7518	0.7422	0.6541
			Deformable RFCN	Combined	0.1504	0.2455	0.1719	0.5013	0.4304	0.4053	0.2886
				Separate	0.6954	0.6133	0.6273	0.6937	0.6644	0.6736	0.6504
ICDAR-13 (Test Set)			Deformable FPN	Combined	0.7096	0.6803	0.6924	0.7424	0.7015	0.7166	0.7045
				Separate	0.7303	0.7120	0.7196	0.7262	0.7238	0.7244	0.7220
			Deformable Faster R-CNN	Combined	0.6182	0.1520	0.1920	0.6939	0.6580	0.6721	0.4321
				Separate	0.5279	0.4625	0.4818	0.6701	0.6768	0.6705	0.5761
			Deformable RFCN	Combined	0.1418	0.2310	0.1719	0.4214	0.4265	0.3883	0.2801
				Separate	0.6024	0.5660	0.5764	0.6842	0.6494	0.6605	0.6184
TabStructDB (Test Set)		Deformable FPN	Combined	0.9081	0.9426	0.9180	0.9465	0.9368	0.9350	0.9265	
			Separate	0.9093	0.9404	0.9186	0.9560	0.9628	0.9559	0.9372	
		Deformable Faster R-CNN	Combined	0.8986	0.5416	0.5917	0.9508	0.9452	0.9420	0.7669	
			Separate	0.8921	0.9125	0.8945	0.9585	0.9682	0.9594	0.9269	
		Deformable RFCN	Combined	0.1383	0.2799	0.1764	0.5190	0.5007	0.4318	0.3041	
			Separate	0.8470	0.7795	0.7843	0.9616	0.9611	0.9562	0.8702	

Table I: Cross-dataset testing results

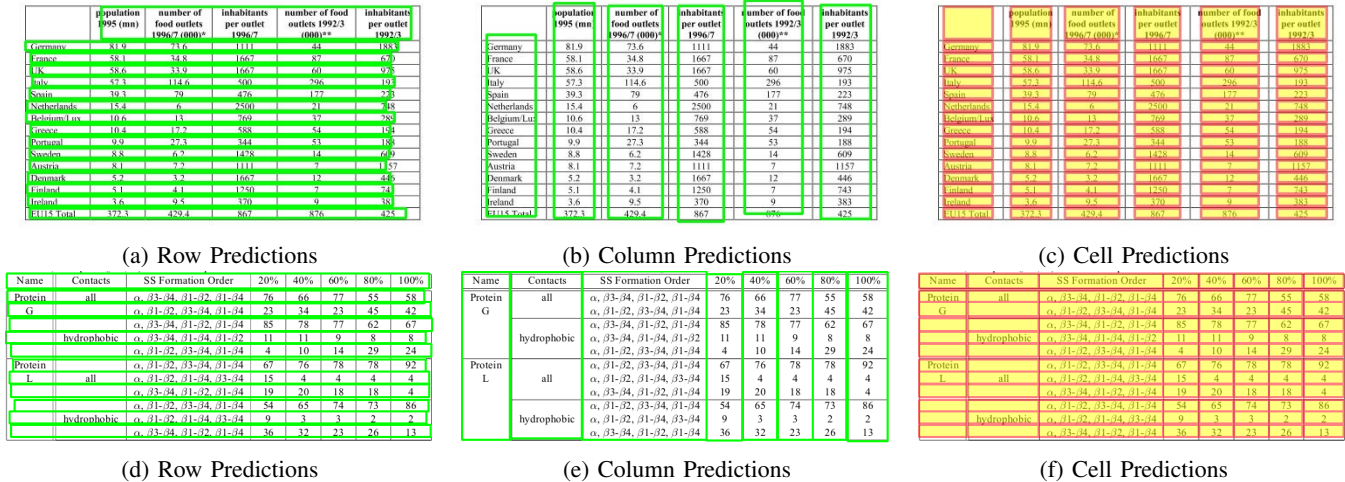


Figure 2: Correctly Recognized Tabular Structures

with the information regarding row/col span which cannot be achieved with the current system. An alternate direction could be to directly generate a textual description of the tabular region instead of detecting the cells independently and post-processing it to merge back the overly-segmented regions or discovering the row/column span [21].

REFERENCES

- [1] T. Kieninger and A. Dengel, "The T-Recs Table Recognition and Analysis System," in *Document Analysis Systems*, ser. Lecture Notes in Computer Science, S.-W. Lee and Y. Nakano, Eds. Berlin: Springer, 1999, pp. 255–270.
- [2] S. Schreiber, S. Agne, I. Wolf, A. Dengel, and S. Ahmed, "Deepdesrt: Deep learning for detection and structure recognition of tables in document images," in *ICDAR*, vol. 1. IEEE, 2017, pp. 1162–1167.
- [3] A. Gilani, S. R. Qasim, M. I. Malik, and F. Shafait, "Table detection using deep learning," in *ICDAR*, 2017, pp. 771–776.
- [4] S. Ahmed, M. Liwicki, M. Weber, and A. Dengel, "Improved automatic analysis of architectural floor plans," in *2011 ICDAR*. IEEE, 2011, pp. 864–869.
- [5] J. Younas, M. Z. Afzal, M. I. Malik, F. Shafait, P. Lukowicz, and S. Ahmed, "D-star: A generic method for stamp segmentation from document images," in *ICDAR*, 2017, pp. 248–253.
- [6] M. Gbel, T. Hassan, E. Oro, and G. Orsi, "Icdar 2013 table

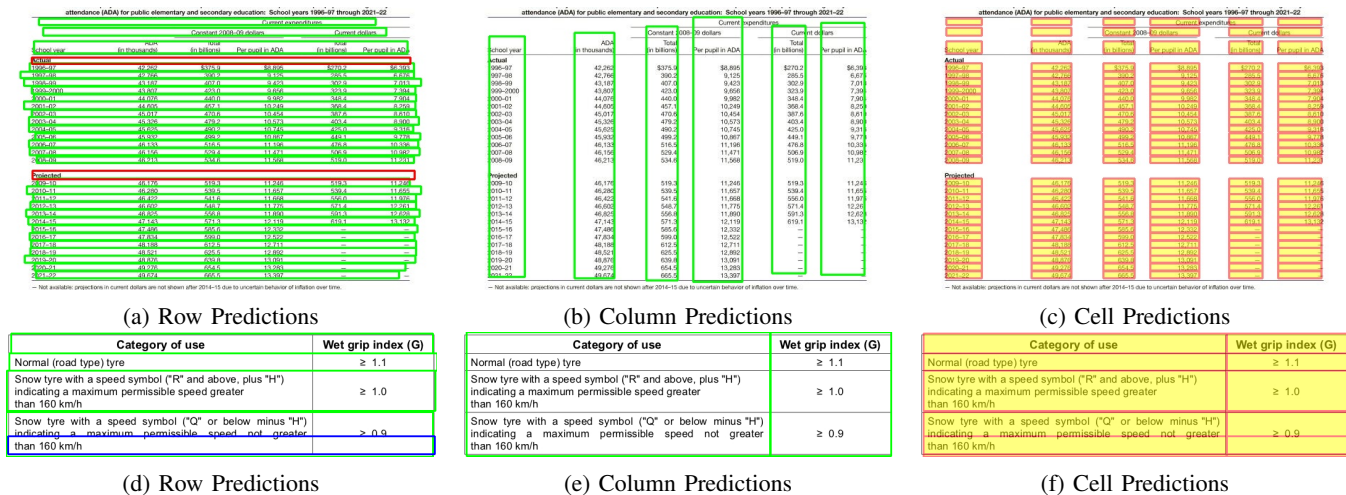


Figure 3: Incorrectly Recognized Tabular Structures. Green color depicts true positives, blue color depicts false positives and red color depicts false negatives for both rows and columns. Cell-level information is not color-coded.

Model	Row			Column			Average		
	Precision	Recall	F-Measure	Precision	Recall	F-Measure	Precision	Recall	F-Measure
Schreiber et al. [2]	-	-	-	-	-	-	0.9593	0.8736	0.9144
DeepTabStR (Proposed)	0.9688	0.9630	0.9655	0.8845	0.8945	0.8861	0.9319	0.9308	0.9298

Table II: Results on the ICDAR-13 table dataset

- competition,” in *2013 12th ICDAR*, Aug 2013, pp. 1449–1453.
- [7] S. A. Siddiqui, M. I. Malik, S. Agne, A. Dengel, and S. Ahmed, “Decnt: Deep deformable cnn for table detection,” *IEEE Access*, vol. 6, pp. 74 151–74 161, 2018.
- [8] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei, “Deformable convolutional networks,” *CoRR*, *abs/1703.06211*, vol. 1, no. 2, p. 3, 2017.
- [9] Y. Wang, I. T. Phillips, and R. M. Haralick, “Table structure understanding and its performance evaluation,” *Pattern Recognition*, vol. 37, no. 7, pp. 1479–1497, 2004.
- [10] A. Shigarov, A. Mikhailov, and A. Altaev, “Configurable Table Structure Recognition in Untagged PDF documents,” in *2016 ACM Symposium on Document Engineering, DocEng 2016*, 2016, pp. 119–122.
- [11] S. Klampfl, K. Jack, and R. Kern, “A comparison of two unsupervised table recognition methods from digital scientific articles,” *D-Lib Magazine*, vol. 20, no. 11, p. 7, 2014.
- [12] T. Kasar, T. K. Bhowmik, and A. Belad, “Table information extraction and structure recognition using query patterns,” in *2015 13th ICDAR*, Aug 2015, pp. 1086–1090.
- [13] R. Rastan, H.-Y. Paik, and J. Shepherd, “Texus: A unified framework for extracting and understanding tables in pdf documents,” *Information Processing & Management*, vol. 56, no. 3, pp. 895–918, 2019.
- [14] S. Ren, K. He, R. Girshick, and J. Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks,” in *28th NIPS*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds., 2015, pp. 91–99.
- [15] K. He, G. Gkioxari, P. Dollár, and R. Girshick, “Mask r-cnn,” in *ICCV*. IEEE, 2017, pp. 2980–2988.
- [16] L. Gao, X. Yi, Z. Jiang, L. Hao, and Z. Tang, “ICDAR2017 competition on page object detection,” in *ICDAR*, 2017, pp. 1417–1422.
- [17] J. Dai, Y. Li, K. He, and J. Sun, “R-fcn: Object detection via region-based fully convolutional networks,” in *NIPS*, 2016, pp. 379–387.
- [18] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, “Feature pyramid networks for object detection,” in *IEEE CVPR*, 2017, pp. 2117–2125.
- [19] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, “ImageNet Large Scale Visual Recognition Challenge,” *IJCV*, vol. 115, no. 3, pp. 211–252, 2015.
- [20] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
- [21] M. Li, L. Cui, S. Huang, F. Wei, M. Zhou, and Z. Li, “Tablebank: Table benchmark for image-based table detection and recognition,” 2019.