# Rethinking Semantic Segmentation for Table Structure Recognition in Documents

Shoaib Ahmed Siddiqui*[†], Pervaiz Iqbal Khan*[†], Andreas Dengel*[†], Sheraz Ahmed[†]
**TU Kaiserslautern, Kaiserslautern, Germany*
[†]*German Research Center for Artificial Intelligence (DFKI), Kaiserslautern, Germany*
*Email: firstname.lastname@dfki.de*

***Abstract*—Based on the recent advancements in the domain of semantic segmentation, Fully-Convolutional Networks (FCN) have been successfully applied for the task of table structure recognition in the past [1]. We analyze the efficacy of semantic segmentation networks for this purpose and simplify the problem by proposing prediction tiling based on the consistency assumption which holds for tabular structures. For an image of dimensions $H \times W$, we predict a single column for the rows ($\hat{y}_{row} \in H$) and a predict a single row for the columns ($\hat{y}_{col} \in W$). We use a dual-headed architecture where initial feature maps (from the encoder-decoder model) are shared while the last two layers generate class specific (row/column) predictions. This allows us to generate predictions using a single model for both rows and columns simultaneously, where previous methods relied on two separate models for inference. With the proposed method, we were able to achieve state-of-the-art results on ICDAR-13 image-based table structure recognition dataset with an average F-Measure of 92.39% (91.90% and 92.88% F-Measure for rows and columns respectively). The obtained results advocate that constraining the problem space in the case of FCN by imposing valid constraints can lead to significant performance gains.**

***Keywords*-Document Analysis, Table Structure Recognition, Table Understanding, Fully-Convolutional Networks (FCN)**

## I. Introduction

Documents are ubiquitous in this age of knowledge and information. Albeit the recent advances in technology and upgrades in communication mediums, paper-based documents are still prevalent [1]. The ability to automatically process the information embedded in these documents is of high interest. Numerous efforts have been made in the past to automatically extract the relevant information from documents [1], [2], [3], [4], [5]. One particular entity that is very commonly encountered in documents is a tabular structure. These tabular structures convey some of the most important information in a very concise form. These structures are extremely prevalent in domains like finance, administration, research, and even archival documents [3]. Therefore, automated extraction of these tabular regions can be useful in a wide range of applications [1], [2], [3].

Significant efforts have been made in the past to extract this tabular information from documents through an automated process [1], [2], [3], [6], [7]. The problem of successful table understanding can be decomposed into two sub-problems. The first sub-problem is to detect the tabular regions in a document image. Numerous methods have dealt with this problem in the past [1], [2], [3] and achieved near perfect detection rates on publicly available datasets. Once a tabular region is successfully detected, the next task is to understand the layout of the table which ultimately includes the identification of different cells present in a table [7]. This problem of cell region detection can be further decomposed into row and column identification which can ultimately be combined together for the discovery of the corresponding cells in the table [1]. Most of the prior methods leverage born-digital documents (PDFs) for the task of table understanding [6]. Minor efforts have also been made on performing analysis of tabular regions directly on images [1]. We approach the problem of table structure recognition by directly operating over images instead of digital-born PDFs.

In order to tackle this task, we propose a new formulation where semantic segmentation systems are constrained using prediction tiling framework in order to successfully extract the structure out of sparse tabular regions. Since the method directly operates over images, this enhances the applicability of the system (to both PDFs and images), where even born-digital documents can be easily converted to images. Schreiber et al. [1] proposed a table recognition system where they computed precision and recall based on the rows and columns instead of using the cell-level information from the ICDAR-2013 table competition dataset [7]. We compare our method using these image-based row and column-level statistics [1]. In particular, the contributions of this paper are as follows:

- A unified architecture for the detection of both rows and columns simultaneously using Fully-Convolutional Networks.
- Introduction of a novel prediction tiling framework which significantly reduces the complexity of the problem and improves segmentation performance based on the consistency assumption of tabular structures.
- Benchmarking of the results obtained by the proposed prediction tiling approach using the evaluation metrics used by Schreiber et al. (2017) [1] on the publicly available ICDAR-13 table structure recognition dataset [7].

The rest of the paper is structured as follows. We first provide a brief overview of the previous work in the direction of table structure recognition in Section II. We then

2379-2140/19/$31.00 ©2019 IEEE
DOI 10.1109/ICDAR.2019.00225

Authorized licensed use limited to: Xian Jiaotong University. Downloaded on May 19,2021 at 10:26:24 UTC from IEEE Xplore. Restrictions apply.

provide details regarding the publicly available ICDAR-13 dataset that we used for the corresponding experimentation in Section III. We describe the proposed approach with all the details in Section IV. Finally, we present our results in Section V before presenting the concluding remarks in Section VI.

## II. Literature Review

There has been only a limited number of attempts to tackle the problem of table structure recognition due to its higher complexity as compared to the table detection task [1], [2], [6]. T-Recs system proposed by Kieninger and Dengel (1999) [6] was one of the first systems to tackle this problem. The system initially grouped words into columns by computing their horizontal ruling lines which were subsequently divided into cells based on column margins.

Another system proposed by Wang et al. (2004) [8] relied on probability optimization to tackle the table structure understanding problem. Their approach was similar to the X-Y cut algorithm. Their system relied on probabilities computed from the data itself, hence, resulting in a data-driven aspect.

A table structure recognition competition was organized in ICDAR-13 [7] where the task was to detect cell-level information from tabular structures. The systems were evaluated based on cell-level metrics where the metrics were computed based on the adjacency list. Since the system compared actual textual content instead of their location, this disabled a fair comparison with pure image-based table structure analysis systems where perfect extraction of text is almost impossible. All the entries in the competition (except one) made extensive use of the PDF meta-data. The pure image-based system achieved poor cell-level metrics indicating that the proposed metrics were not suitable for the assessment of image-based analysis systems [7].

Kasar et al. (2015) [9] presented a graph matching system which constructed attributed relational graphs based on user queries. This graph was then matched with the graphs present in the document via a fast graph matching algorithm to retrieve the relevant information. Shigarov et al. (2016) [10] provided a detailed analysis of different algorithms, thresholds and rule bases that can be utilized for the structure recognition task. Their approach relied heavily on external sources of information i.e. PDF meta-data comprising of information regarding font, font-size, bounding boxes etc. as well as custom heuristics for the utilization of the extracted information. A recent framework, TEXUS [11], has been introduced by Rastan et al. (2019) capable of extracting the structure information as well as the content from tabular structures embedded in born-digital PDFs.

Since all of the prior approaches rely on digital-born PDFs, they are not comparable to our system which directly operates over images, making it much more generally applicable in the real world. Schreiber et al. (2017) [1] made a recent attempt for the incorporation of deep learning based techniques for the task of table structure recognition where they used two different fully-convolutional networks for the identification of the corresponding rows and columns. We similarly investigate the formulation of table structure recognition where it is treated as a semantic segmentation problem. However, there are many significant differences between the two approaches. We propose a different problem formulation, use a more powerful base model, different batch size, use the original image instead of a stretched version, and introduce the boundary class instead of ignoring it. We highlight all these different aspects in Section IV.

## III. Dataset

We use the largest publicly available table structure recognition dataset for the evaluation of our model. The dataset was released as part of a table competition conducted during ICDAR-13 [7]. ICDAR-13 dataset is comprised of 67 PDF files containing a total of 238 pages. From these 238 pages, a total of 156 tables were extracted. The dataset originally contains annotations for cells. In order to make our results comparable with previous data-driven state-of-the-art approaches [1], we converted the cells into rows and columns and trained our model based on this information. Similarly, we follow an identical data split (with the same files segregated into train and test sets) as the one defined by Schreiber et al. (2017) [1] comprising of 125 training and 31 test table images.

One of the problems with ICDAR-13 dataset is only the marking for cells containing the textual content, hence, resulting in smaller rows and columns. This scheme is not useful in the real world, since we would like to extract cell level information based on the intersection of rows and columns. Therefore, we manually fine-tuned the labels from the ICDAR-13 dataset to include these empty regions[1]. Since our system works on the consistency assumption, this type of annotation is essential to successfully train our model.

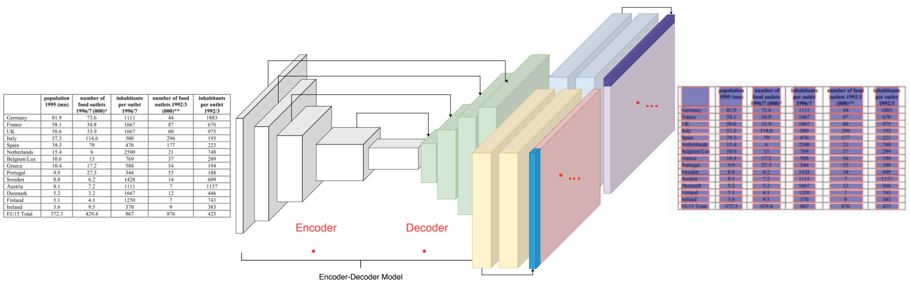
## IV. Method

We formulate the problem of table structure recognition as a semantic-segmentation problem. Semantic segmentation deals with the task of pixel-wise prediction where each pixel is classified to a particular class. Deep learning has been employed at the core of state-of-the-art semantic segmentation systems [12], [13]. Since the ultimate goal is to discover cell-level information, we split this problem into two sub-problems where the first one deals with the segmentation of rows while the second one deals with the segmentation of columns.

Fig. 1 visualizes the complete system pipeline comprising of several sub-components. The first component is the

[1] Fine-tuned ICDAR-13 dataset: https://bit.ly/2NhZHCr

Authorized licensed use limited to: Xian Jiaotong University. Downloaded on May 19,2021 at 10:26:24 UTC from IEEE Xplore. Restrictions apply.



|  | population 1995 (mn) | number of food outlets 1996/7 (000)* | inhabitants per outlet 1996/7 | number of food outlets 1992/3 (000)** | inhabitants per outlet 1992/3 |
|---|---|---|---|---|---|
| Germany | 81.9 | 73.6 | 1111 | 44 | 1883 |
| France | 58.1 | 34.8 | 1667 | 87 | 670 |
| UK | 58.6 | 33.9 | 1667 | 60 | 975 |
| Italy | 57.3 | 114.6 | 500 | 296 | 193 |
| Spain | 39.3 | 79 | 476 | 177 | 223 |
| Netherlands | 15.4 | 6 | 2500 | 21 | 748 |
| Belgium/Lux | 10.6 | 13 | 769 | 37 | 289 |
| Greece | 10.4 | 17.2 | 588 | 54 | 194 |
| Portugal | 9.9 | 27.3 | 344 | 53 | 188 |
| Sweden | 8.8 | 6.2 | 1428 | 14 | 609 |
| Austria | 8.1 | 7.2 | 1111 | 7 | 1157 |
| Denmark | 5.2 | 3.2 | 1667 | 12 | 446 |
| Finland | 5.1 | 4.1 | 1250 | 7 | 743 |
| Ireland | 3.6 | 9.5 | 370 | 9 | 383 |
| EU15 Total | 372.3 | 429.4 | 867 | 876 | 425 |

Figure 1: The proposed semantic segmentation based table structure recognition pipeline

encoder-decoder model which is responsible for extracting the relevant features from the image. Our encoder-decoder model is inspired by the FCN architecture [12]. These features from the decoder are passed onto the prediction tiling module. The prediction tiling module first pools the features, followed by a couple of convolutional layers before tiling the final predictions to obtain the segmentation masks. These masks are post-processed to obtain cell-level information based on their intersection. We will now cover each of the components in detail.

### A. *Encoder*

The encoder is responsible for condensing meaningful information from the input signal in a comprehensive way while suppressing irrelevant details. We use Inception ResNet v2 [14] as our encoder (which is one of the most powerful image classification models to date), responsible for extracting the global information from the image. We only use the base model comprising of the convolutional layers transforming it into an FCN capable of processing arbitrary-sized images. To leverage the benefits of transfer learning, we use a pretrained model on ImageNet [15] which enables reuse of already learned features. Using such a deep base model pretrained on a large dataset enables the network to extract highly abstract features from the input. The encoder can be represented as:

$$Z = \Psi(x) \tag{1}$$

where $x$ represents the input, $\Psi$ represents the encoder and $Z$ represents the encoded features produced by the encoder.

### B. *Decoder*

The decoder is attached to the output produced by the encoder ($Z$) in order to transform it to the intermediate/output space which is of the same dimensions as the input. The decoder is comprised of transposed convolutional layers followed by convolutional layers with $3 \times 3$ kernel and 256 filters each. We use batch-normalization layers after every convolutional layer to stabilize gradient propagation along with leaky ReLU as the activation function (leak rate of 0.2). We use the features from the encoder and combine it with the feature maps from the decoder (skip-connections) at four different stride-levels (16, 8, 4, 2) until a full-sized image is obtained. We tested two different formulations for these skip-connections. In the first formulation, we stacked the feature maps together, while in the second formulation, we performed element-wise addition. We found stacking to be a much more efficient solution encouraging feature reuse which is consistent with the findings of [16] (DenseNet). The decoder can be represented as:

$$A = \Phi(Z) \tag{2}$$

where $Z$ represents the output from the encoder (Eq. 1), $\Phi$ represents the decoder and $A$ represents the intermediate representation. This intermediate representation can also be the output of the system as in the case of Schreiber et al. (2017) [1]. However, in our case, this intermediate representation is consumed by the prediction tiling module (Section IV-C).

### C. *Prediction Tiling*

Predicting complete segmentation mask for both rows and columns is a difficult task. We make a consistency assumption on the table i.e. all the rows start from the starting point of the first column and end at the ending point of the last column. Similarly, all the columns start from the starting point of the first row and span until the ending point of the last row. This simple assumption holds for all normal tabular structures. However, this assumption is violated in the case of hierarchical tables requiring extra post-processing to merge back the overly-segmented rows/columns.

Based on this simple assumption, we reduce the load on each of the output prediction heads. For an image of size $H \times W$, we simplify the prediction from the two heads. The encoder-decoder model produces an intermediate representation which is of the same size as the input. This intermediate representation is composed of the features extracted by the system which are deemed as useful for the task at hand i.e. segmentation of rows and columns in a tabular structure. This operation can be represented as:

$$A = \Phi\big(\Psi(x)\big) \tag{3}$$

Once this intermediate representation is computed, we reduce the available information based on the consistency assumption. This reduction for rows can be represented as:

Authorized licensed use limited to: Xian Jiaotong University. Downloaded on May 19,2021 at 10:26:24 UTC from IEEE Xplore. Restrictions apply.

$$A_{row}[i,:] = \frac{1}{W}\sum_{j=1}^{W} A[i,j,:] \qquad \forall i = \{1,...,H\} \tag{4}$$

where $A_{row} \in \mathbb{R}^{H\times C}$ ($C$ represents the number of channels) indicates the features for rows. The first index of the activation tensor ($A \in \mathbb{R}^{H\times W\times C}$) is $i$, which spans the height of the image, while the second index $j$ spans the width. Similarly, for columns, the reduction operation can be represented as:

$$A_{col}[j,:] = \frac{1}{H}\sum_{i=1}^{H} A[i,j,:] \qquad \forall j = \{1,...,W\} \tag{5}$$

where $A_{col} \in \mathbb{R}^{W\times C}$ indicates the features for the column. Once this reduced representation is computed, we apply the type specific head for both rows and columns. These heads are represented as $\Phi_{row} : \mathbb{R}^{H\times C} \mapsto \mathbb{R}^{H\times K}$ and $\Phi_{col} : \mathbb{R}^{W\times C} \mapsto \mathbb{R}^{W\times K}$ for rows and columns respectively where $K$ represents the number of output classes ($K = 3$ in our case). This final representation of the output can be represented as:

$$\hat{y}_{row} = \Phi_{row}\big(A_{row}\big) \tag{6}$$

$$\hat{y}_{col} = \Phi_{col}\big(A_{col}\big) \tag{7}$$

where both classification heads are comprised of three convolutional layers (Conv-1D) including batch-norm, leaky ReLU and feature concatenation. The first two layers use a kernel size of 3 and 64 filters while the last layer uses a kernel size of 1 and $K$ filters. Since the required segmentation mask is of size $H \times W$ and the computed representations are 2D instead of 3D, we tile (repeat and stack) the predictions together to convert the fixed representation to full size ($\mathbb{R}^{H\times W\times K}$). This tiling operation can be represented as:

$$\hat{y}_{row} = \begin{bmatrix} \hat{y}_{row}[1,:] & \hat{y}_{row}[1,:] & \dots & \hat{y}_{row}[1,:] \\ \hat{y}_{row}[2,:] & \hat{y}_{row}[2,:] & \dots & \hat{y}_{row}[2,:] \\ \vdots & \vdots & \ddots & \vdots \\ \hat{y}_{row}[H,:] & \hat{y}_{row}[H,:] & \dots & \hat{y}_{row}[H,:] \end{bmatrix}$$

$$\hat{y}_{col} = \begin{bmatrix} \hat{y}_{col}[1,:] & \hat{y}_{col}[2,:] & \dots & \hat{y}_{col}[W,:] \\ \hat{y}_{col}[1,:] & \hat{y}_{col}[2,:] & \dots & \hat{y}_{col}[W,:] \\ \vdots & \vdots & \ddots & \vdots \\ \hat{y}_{col}[1,:] & \hat{y}_{col}[2,:] & \dots & \hat{y}_{col}[W,:] \end{bmatrix}$$

With this tiling, the size of the output predictions matches the size of the input ($\hat{y}_{row} \in \mathbb{R}^{H\times W\times K}$ and $\hat{y}_{col} \in \mathbb{R}^{H\times W\times K}$). Instead of having two classes for each head ($\Phi_{row}$ and $\Phi_{col}$) i.e. background and row/column, we use three different classes i.e. background, row/column and boundary ($K = 3$). The boundary is the pixels in between the row/column and the background. In the usual semantic segmentation literature, these pixels are ignored [12], however, in our case, these pixels are the most important ones as they act as separators between the distinct regions which are extremely close to each other like rows. Therefore, we assign 5 times the weight of normal pixels to correctly identify these pixels since they enable differentiation between different rows/columns.

### D. Post-Processing

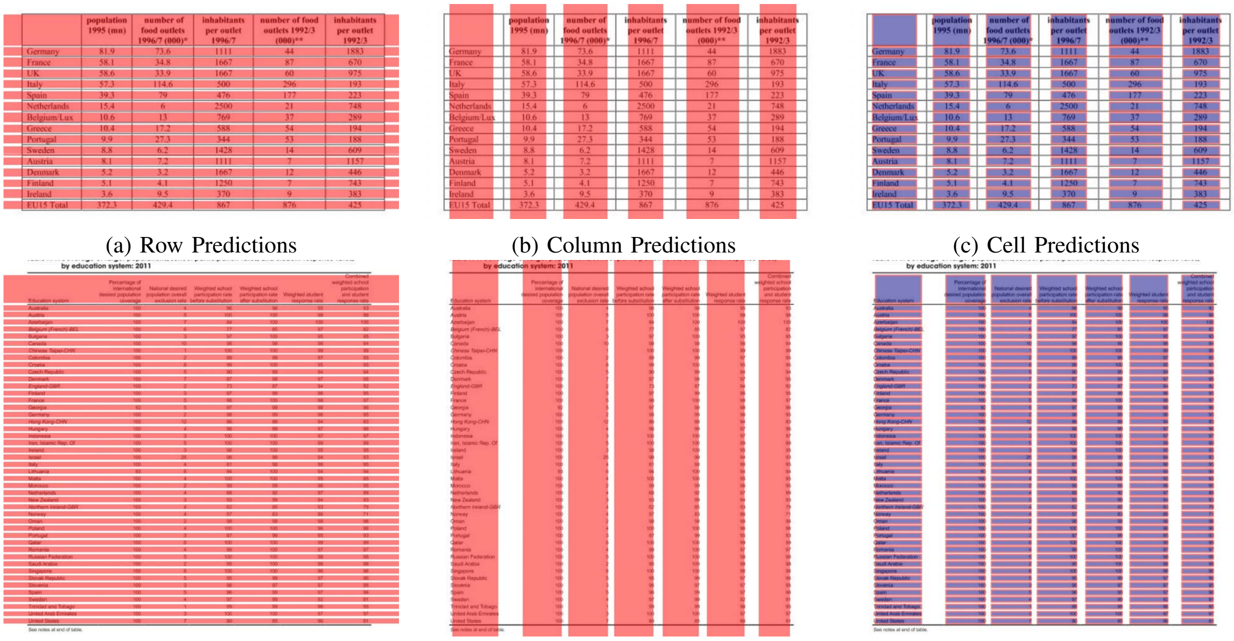
Our post-processing is similar to the one used by Schreiber et al. (2017) [1]. We first perform morphological operations of closing and then opening with a square kernel of 3 × 3. Closing fills up the small holes in the mask while opening removes small objects. In order to extract the region boundaries for row/column, we then perform contour detection. Since we have perfect bounding boxes going from one end to the other of the corresponding row/column, we convert the contour to a bounding box. Once we have the bounding boxes, we perform simple filtering based on the size of the bounding box. We discard all detected bounding boxes whose height is less than 1% of the corresponding image height for rows or whose width is less than 2% of the corresponding image width for columns. The detected bounding-box coordinates for rows are further fine-tuned by setting the starting point for all rows to the starting point of the first column and the end point of all the rows to the end point of the last column. Similarly, we fine-tune the annotations for columns.

## V. Evaluation

In order to compare our method with the state-of-the-art approaches, we use the same dataset along with the same data-split and evaluation metrics. We evaluated the proposed approach on the publicly available ICDAR-13 dataset. We follow the same evaluation protocol as used by Schreiber et al. [1]. We use an Intersection-over-Union (IoU) threshold of 0.5 for the detections. There are two possibilities to compute the statistics. The first one focuses on per-document averages where the precision, recall, and F-Measures are computed for every document separately and then averaged over all the documents. The second approach counts the number of true positives, false positives, and false negatives over the entire dataset which are then, in turn, used to compute the precision, recall, and F-Measure. The first approach avoids bias towards a single document in the system. The second approach takes all the elements into account. Therefore, we report metrics based on both these strategies for the sake of completion.

The results from the method are presented in Table I. Although there are numerous different methods reported on the ICDAR-13 dataset from the ICDAR-13 table structure detection competition, however, as our system is based on document images, this evades the possibility of a fair comparison with PDF-based structure analysis systems benchmarked using the cell-level statistics. These cell-based metrics are computed based on the adjacency list which relies on perfect extraction of text from the document. Adding an

Authorized licensed use limited to: Xian Jiaotong University. Downloaded on May 19,2021 at 10:26:24 UTC from IEEE Xplore. Restrictions apply.



(a) Row Predictions (b) Column Predictions (c) Cell Predictions

(d) Row Predictions (e) Column Predictions (f) Cell Predictions

Figure 2: Correctly Recognized Tabular Structures

| Model | Averaging Type | Row | | | Column | | | Average | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | Precision | Recall | F-Measure | Precision | Recall | F-Measure | Precision | Recall | F-Measure |
| Schreiber et al. [1] | Document | - | - | - | - | - | - | **0.9593** | 0.8736 | 0.9144 |
| Proposed system | | 0.9233 | 0.9203 | 0.9190 | 0.9281 | 0.9341 | 0.9288 | 0.9257 | **0.9272** | **0.9239** |
| Proposed system | Complete | 0.9532 | 0.9418 | 0.9475 | 0.9157 | 0.9261 | 0.9209 | 0.9344 | 0.9340 | 0.9342 |

Table I: Results on the ICDAR-13 table dataset

OCR system on top of the detection system results in an accumulation of error from the OCR end, disabling any fair comparison strategy [7].

It is evident from the table that the proposed prediction tiling based system outperforms the previous image-based state-of-the-art table structure recognition system [1]. Although their system achieved better precision, but significantly lacked in terms of recall. The results from the proposed approach are consistent and homogeneous. Some correct examples are visualized in Fig. 2. It can be observed from the detection results that the generated layout is fairly consistent and accurate.

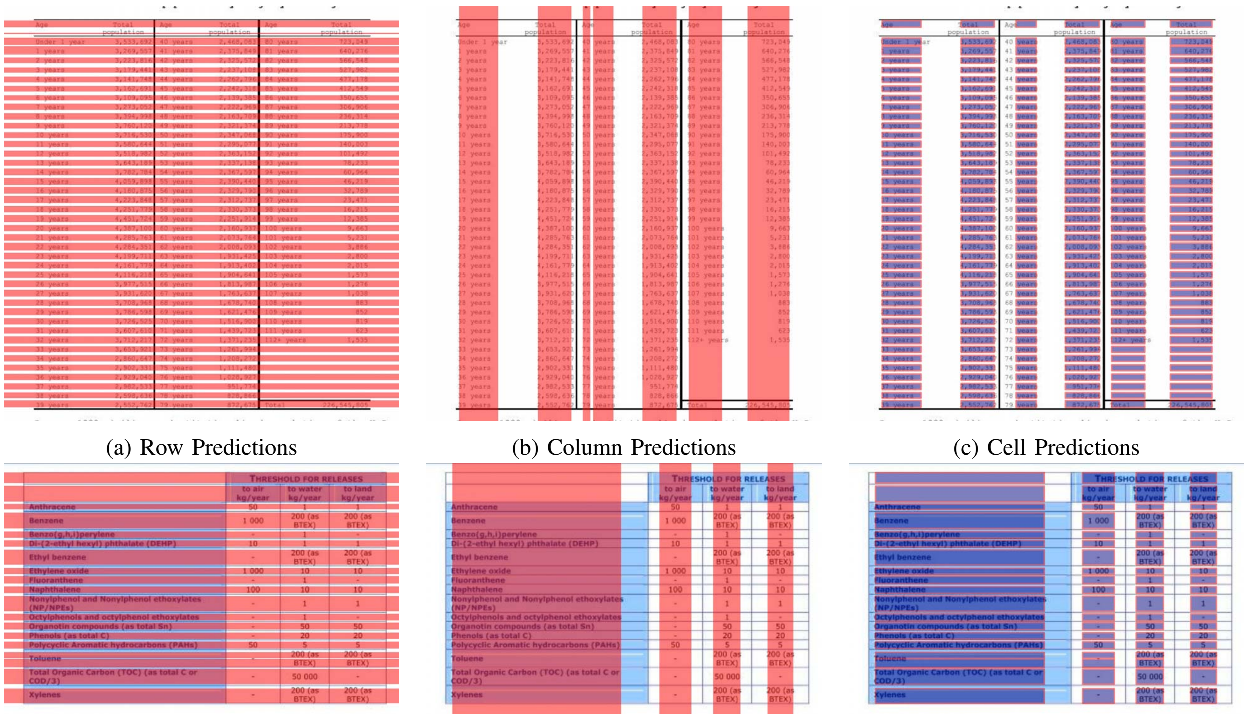
Sample incorrect detection results are visualized in Fig. 3. In the figure on the top (Fig. 3a), the system was unable to detect the multi-line row, resulting in an IoU of less than 0.5. This resulted in a false negative as well as a false positive. The system was only able to partially detect one column, however, the IoU was still greater than 0.5 in that case. For the figure on the bottom (Fig. 3d), the system incorrectly segregated a row which was instead a multi-line row. The next row, however, which is also a multi-line row, was correctly recognized since the text in the next columns is center aligned. These issues can be resolved by adding sophisticated post-processing techniques on the output, which lies beyond the scope of our work. The scope of this work is only to highlight the efficacy of the prediction-tiling based framework.

## VI. Conclusion

We formulated the problem of table structure recognition as a semantic segmentation problem. Since pixel-wise segmentation of document images is hard, due to the presence of sparse regions, specifically in cases where there are no ruling lines, we introduce prediction tiling which significantly reduces the complexity of the problem and encourages consistent results. The obtained results outperformed the state-of-the-art image-based table structure analysis system on the publicly available ICDAR-13 table structure recognition dataset.

The proposed system over-segments tabular structures where the row/column span is larger than one. Future work can be directed towards heuristics or even a separate model to merge back these overly segmented regions since we don't use any meta-data from the PDF which enables direct extraction of textual regions. Integrating text detection systems on top to filter out false positives can also be beneficial. Alongside, different input representations can be merged

Authorized licensed use limited to: Xian Jiaotong University. Downloaded on May 19,2021 at 10:26:24 UTC from IEEE Xplore. Restrictions apply.



(a) Row Predictions (b) Column Predictions (c) Cell Predictions

(d) Row Predictions (e) Column Predictions (f) Cell Predictions

Figure 3: Incorrectly Recognized Tabular Structures

into the system in order to provide the system with extra contextual information to further improve the results [2].

## References

[1] 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.

[2] A. Gilani, S. R. Qasim, M. I. Malik, and F. Shafait, "Table detection using deep learning," in *ICDAR*, 2017, pp. 771–776.

[3] 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.

[4] 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.

[5] S. Ahmed, M. Liwicki, M. Weber, and A. Dengel, "Improved automatic analysis of architectural floor plans," in *ICDAR*. IEEE, 2011, pp. 864–869.

[6] T. Kieninger and A. Dengel, "The T-Recs Table Recognition and Analysis System," in *Document Analysis Systems*, Berlin, 1999, pp. 255–270.

[7] M. Gbel, T. Hassan, E. Oro, and G. Orsi, "Icdar 2013 table competition," in *ICDAR*, Aug 2013, pp. 1449–1453.

[8] 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.

[9] T. Kasar, T. K. Bhowmik, and A. Belad, "Table information extraction and structure recognition using query patterns," in *ICDAR*, Aug 2015, pp. 1086–1090.

[10] A. Shigarov, A. Mikhailov, and A. Altaev, "Configurable Table Structure Recognition in Untagged PDF documents," in *ACM Symposium on Doc. Engg.*, 2016, pp. 119–122.

[11] 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.

[12] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *IEEE CVPR*, 2015, pp. 3431–3440.

[13] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *ICCV*. IEEE, 2017, pp. 2980–2988.

[14] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *31st AAAI Conference on AI*, 2017.

[15] 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.

[16] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *IEEE CVPR*, 2017, pp. 4700–4708.

Authorized licensed use limited to: Xian Jiaotong University. Downloaded on May 19,2021 at 10:26:24 UTC from IEEE Xplore. Restrictions apply.