# ICDAR 2019 Competition on Table Detection and Recognition (cTDaR)

Liangcai Gao
Yilun Huang
*ICST*
*Peking University, China*
*glc@pku.edu.cn*
*huangyilun@pku.edu.cn*

Hervé Déjean
Jean-Luc Meunier
*Naver Labs Europe*
*Meylan, France*
*firstname.lastname@naverlabs.com*

Qinqin Yan
Yu Fang
*State Key Laboratory of*
*Digital Publishing Technology*
*Founder Group Co. LTD., China*
*{yan.qq, fangyu}@founder.com.cn*

Florian Kleber
*Computer Vision Lab*
*TU Wien*
*1040 Vienna, Austria*
*kleber@cvl.tuwien.ac.at*

Eva Lang
*Archiv des Bistums Passau*
*Passau, Germany*
*eva.lang@ieee.org*

***Abstract*—The cTDaR competition aims at benchmarking state-of-the-art table detection (TRACK A) and table recognition (TRACK B) methods. In particular, we wish to investigate and compare general methods that can reliably and robustly identify the table regions within a document image on the one hand, and the table structure on the other hand. Due to the presence of hand-drawn tables and handwritten text, the methods must be robust against various noise conditions, interfering annotations, and variations of the tables. Two new challenging datasets were created to test the behaviour of state-of-the-art table detection and recognition systems on real world data. One dataset consists of modern documents, while the other consists of archival documents with presence of hand-drawn tables and handwritten text. The evaluation scheme is adapted from the ICDAR 2013 Table competition. We received results of Track A from 11 teams and results of Track B from 2 teams. Results for Track A are very good for the top participants. The winner and his runner-up are very close while using very different approaches. Track B was more challenging and only one participant was able to produce good results.**

***Keywords*-table detection; table recognition; modern tables; archival documents;**

## I. Introduction

Table analysis is considered as an open research topic in the document analysis community and is essential for understanding of structured documents. The goal of this competition is to evaluate the performance of state-of-the-art methods for table detection (TRACK A) and table recognition (TRACK B). There are 2 subtracks in the latter, which provide different a prior information to participants.

For the current cTDaR 2019 competition, two new datasets consisting of modern printed documents and archival documents are presented. This is, to the best of our knowledge, the first dataset which contains historical documents with handwritten and printed tables. The datasets are described in detail in Section II.

The last table competition which dealt with table location and table structure recognition was ICDAR 2013 Table Competition [1]. Digital-born PDF documents were used as competition dataset. ICDAR 2017 Page Object Detection (POD) competition focused on the detection of specific page objects comprising the detection of tables [2]. Compared with the previous competitions, the cTDaR competition focuses on table detection and recognition on both modern and archival documents.

This paper is organized as follows: First, in Section II an overview of the dataset is given, including the GT format. Section III presents methods of participants while Section IV describes the evaluation protocols. The results of this competition are presented in Section V and finally, conclusions are drawn in Section VI.

## II. Dataset

Two new datasets consisting of modern and archival documents have been prepared for cTDaR 2019. The historical dataset contains contributions from more than 23 institutions around the world. The images show a great variety of tables from hand-drawn accounting books to stock exchange lists and train timetables, from record books to prisoner lists, simple tabular prints in books, production census and many more. The modern dataset comes from different kinds of PDF documents such as scientific journals, forms, financial statements, etc. The dataset contains of Chinese and English documents with various formats, including document images and born-digital format. The annotated contents contain the table entities and cell entities in a document, while we do not deal with nested tables.

For TRACK A, document images containing one or several tables are provided. For TRACK B there are two subtracks: the first subtrack (B.1) provides the table region and only the table structure recognition needs to be performed. The second subtrack (B.2) provides no a priori information. That is to say, both table region detection and table structure recognition have to be done. An example of a fully annotated document from the historical dataset containing a handwritten table is shown in Figure 1.

2379-2140/19/$31.00 ©2019 IEEE
DOI 10.1109/ICDAR.2019.00243

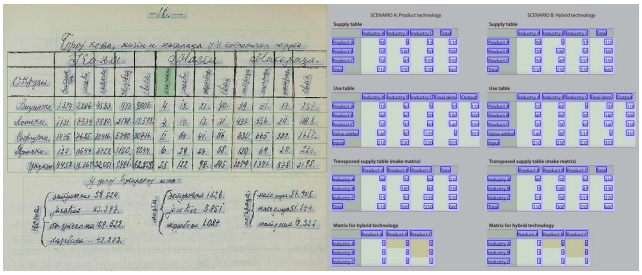

Figure 1. Example documents showing a handwritten table and a modern table.

| | Historical Dataset | | Modern Dataset | |
|---|---|---|---|---|
| | train | test | train | test |
| Track A | 600 | 199 | 600 | 240 |
| Track B.1 | 600 | 150 | - | - |
| Track B.2 | - | 150 | - | 100 |

Table I
NUMBER OF TRAINING AND TEST IMAGES OF THE CTDAR HISTORICAL AND MODERN DATASET.

The number of training and test images is summarized in Table I. For the modern dataset no training images are provided for Track B. The historical dataset provides the same images for Track B.1 and Track B.2.

For the annotation of the dataset, we use an similar notation derived from ICDAR 2013 Table Competition format (see [1]), creating a single XML file to store the structures. Each table element corresponds to a table, which contains a single Coords element with a points attribute to indicate the coordinates of the bounding polygon with N vertices. Each table element also contains a list of cell elements. For each cell element the attributes start-row, start-col, end-row and end-col denotes its position in the table. The element Coords for the cell element denotes the coordinates of the bounding polygon of this cell box, and content is the text within this cell (optional for submission). A minimal sample of a XML is shown in Listing 1.

The difference to Gobel et al. [1] is the Coords tag which defines a table/cell as a polygon specified by a list of coordinates. For B.1 the table and its coordinates is given together with the input image. For the modern dataset, the convex hull of the content describes a cell region. For the historical dataset, it is requested that the output region of a cell is the cell boundary. This is necessary due to the characteristics of handwritten text, which is often overlapping with different cells. Figure 1 shows an example image of an annotated modern document.

The full dataset can be found under [3] or [4] .

## III. METHODS AND PARTICIPANTS

The competition was organized as follows: teams could download the training set along with GT and the images of the test set. For the evaluation teams submitted the resulting XMLs (one per image) which were evaluated using the provided evaluation tool.

In total, 11 teams submitted their results for TRACK A and two teams submitted for TRACK B. A short method description provided by the participating teams is given below. They are listed in alphabetical order.

Listing 1. Minimal sample of a XML.

```
<?xml version="1.0" encoding="UTF-8"?>
<document filename="filename.jpg">
      <table>
            <Coords points="180,160 4354,160
                ↪ 4354,3287 180,3287"/>
               <cell start-row='0' start-
                   ↪ col='0' end-row='1'
                   ↪ end-col='2'>
                     <Coords points="
                         ↪ 180,160
                         ↪ 177,456
                         ↪ 614,456
                         ↪ 615,163"/>
               </cell>
               ...
      </table>
...
</document>
```

### A. ABC Fintech (Track A)

**Chen Chen, Cui Chao, Jin Song, Guo Meng and Yang ManYe**
ABC Fintech
*chchen@abcft.com*

To solve the problem of table detection, we use segmentation model as the main method to extract and locate tables, all training samples include: table simulation (LaTex) and collection and annotation of a large number of document reports. Then, based on the results of the segmentation model, we used rules to repair and segment the tables regions.

We enhanced by latex simulation and data (different resolution, blur, rotation, sharpening and so on) to provide training data model.

### B. AIRL (Track A)

**Sheik Faisal Rashid and Masooma Zia**
Al-Khawarizmi Institute of Computer Science, University of Engineering and Technology, Lahore, Pakistan
*masooma.zia@kics.edu.pk*

This work proposes a novel learning based methodology for the recognition of table contents in heterogeneous document images. Textual contents of documents are classified as table or non-table elements using a pre-trained neural network model. Those contents are then merged together, forming rows and then rows are combined to provide table

region. This bottom-up approach, starts at cells level boundary and expands to table boundary. Predicted table boundary is drawn to get visual results. Moreover, the information of cell, row and table coordinates help to form an HTML output of the table structure.

### C. applica robots (Track A)

**Pawet Dyda, Anna Wroblewska, Przemyslaw Lipka and Tomasz Stanislawek**
Applica.ai
*tomasz.stanislawek@applica.ai*

To resolve table detection problem we have built model with three main building blocks: images preprocessing, building machine learning model and finally postprocessing generated results for some simple heuristic. In the preprocessing phase for all images (both archive and modern) we clean edges (1.5% of image width and height), do otsu thresholding and next L1 distance morphological transformation. To build machine learning model based on preprocessed images we used Faster RCNN algorithm with VGG16 architecture, which turned out to be the best from all available methods. On the final phrase of our solution we removed detected tables which meet the following criteria: there is a table that has a more then 50% of overlap and greater score; table contain more then 2 subtables with greater score.

### D. Cinners table (Track A)

**Tran Tuan Anh, Pham Qui Luan, Nguyen Quoc Thang and Tran Minh Quan**
R&D department, Cinnamon AI labs, Viet Nam
*tommy@cinnamon.is*

Our method includes principles of operation and steps for the training procedure and the prediction procedure:

1) Data preparation: Since we have 1200 labeled samples, we splitted the train-validation set as 7-3 in ratio according to 840 images for the training and 360 images for the validation. We have applied the offline data augumentation by the use of rotation and horizontal/vertical flip images, that procedure increase the data up to over 50,000 samples.
2) Build model: We use the default setup of unet model.
3) Training Setup: Loss function: Cross Entropy Loss; Optimizer: Adam; GPU: K80 12GB VRAM; Epoch: 9; Batch size: 5; Image size: 512 x 512 x 3; Normalization: Mean is [0.485, 0.456, 0.406] Standard deviation is [0.229, 0.224, 0.225]

Prediction:

1) Rotated image generation: The predicted image be rotated for 4 times (10 degree for each), then, we have 5 samples to be predicted.
2) Prediction and region refinement: We use the trained model to predict the 5 rotated samples. Re-rotated each sample and get the median as the final result without any postprocessing. Based on the boundary of the regions, we extracted the rectagular shape of that regions.

### E. Clova AI (Track A)

**Kim Seonghyeon**
NAVER
*kim.seonghyeon@navercorp.com*

Our team used Faster R-CNN with ImageNet pretrained ResNet-50 and Feature Pyramid Networks with stride 4, 8, 16, 32, and have used Non Maximum Suppression with lower thresholds (0.5). Image was resized to shorter edge size is 800, and long edge size to be 1333. Model is trained with Stochastic Gradient Descent with momentum 0.9 and weight decay 1e-4 and batch size was 4. Learning rate warmup was used with linear schedule from 0.006 to 0.02 during 500 iterations, and learning rate decayed by 10 at 8th, 11th epochs. Model trained total 12 epochs.

### F. HCL IDORAN (Track A and B)

**Dhev, Bhaskar, Sarath, Dinesh and Gokul**
HCL
*menugulab@hcl.com*

Pre processing: The input image is converted into gray scale and then it is converted into binary image.

For table ROI detection we have followed two types of the methods:

1) Line based table detection: Horizontal and vertical ruling lines are identified using Hough transform. Then the horizontal and vertical lines are combined to from closed lines. The closed lines, thus found is contour analysed to get bounding rectangles and table region of interest is identified. Using morphological operation text is removed from the roi and table cells are identified using horizontal and vertical projection.
2) Structure based table detection: A morphological operation is performed on the text regions to get the text blobs. Top and bottom of first and last table-lines in every set of separated table-lines is taken as upper and lower bound of individual tables in a page of document. Minimum and maximum left and right is taken respectively to get broader bounds. For table structure cell coordinates, midpoint between successive rows as well as columns will be calculated from the text-line rectangles.

Finally, the table regions and table cell coordinates are written in xml.

### G. Lenovo Ocean (Track A)

**Li Hui, Zhang ChenDi, Lv WangJun, Wang LuYan and Wu YaQiang**
Lenovo Research
*lihuid@lenovo.com*

The basic idea of this method is to use one network to generate two result images, table edge segmentation and table region(mask) segmentation, After that, two result images are processed in post-process stage to determine the number and location of the tables. So, we use a modified HED as the network to generated table edge segmentation, but replace the VGG16 backbone with the ResNet101 to get better representational ability, we also add side-output layers after several selected convolution layers to predict the table edge segmentation result; another branch is created from these convolution layers connected with deconvolution layers to predict the table mask segmentation result. The network was trained with augmented(blur, scale, flip, random noise) dataset based on given 1200 training images. The post-process stage contains follow steps:

1) Get the table count and rough position with filtering small area and truncating small connections from table mask segmentation result.
2) Get all possible lines from the table edge segmentation result with traditional line detection method.
3) Combine above two steps results find edge line for each table to get refined final table position result.

### *H. NLPR PAL (Track A and B)*

**Xiao-Hui Li, Fei Yin and Cheng-Lin Liu**
National Laboratory of Pattern Recognition (NLPR), Institute of Automation of Chinese Academy of Sciences (CASIA), University of Chinese Academy of Sciences (UCAS)
*xiaohui.li@nlpr.ia.ac.cn*

Before table detection and recognition, we use some simple heuristic rules based on image size and ratio of white pixels (pixels whose intensities are greater than 250) to automatically classify the documents into archive documents and modern documents. Because the sizes and gray level distributions of archive documents and modern documents are very different, our simple rules can easily separate these two kinds of documents without making any errors. For table detection in historical documents, we use Fully Convolutional Network (FCN) to classify image pixels into two categories: table and background, then table regions are extracted with Connected Component Analysis (CCA). To enlarge the receptive field of our network, we resize the original image to the same short edge of 512 pixels and apply atrous convolution. To avoid wrong merging of adjacent table regions, we transform the original single label map into a set of label maps containing gradually shrunken regions through distance transformation and multi-level thresholding. After obtaining the multi-task output of probability maps, we sum them into one single probability map on which watershed transformation is performed to get table region separators. These region separators are used in the CCA process. For table detection in modern documents, we conduct the similar pixel labeling task using FCN as in historical documents. However, we don't assume all the pixels inside table regions to be table class. In contrast, we only predict the pixels inside table text line regions and the horizontal and vertical gaps between adjacent text line regions. After that, table regions are extracted and refined with Connected Component Analysis (CCA). For table recognition in historical documents, we first extract the guiding lines and junction points of tables using FCN, then broken lines are repaired with junction information. Once the guiding lines of tables are extracted and repaired, cells can be extracted through CCA. To analysis the row range and column range of each cell, we first find the horizontal and vertical neighbors of each cell based on which an adjacency graph is built, then cell adjacent information is propagated horizontally and vertically to infer the row ranges and column ranges of cells, respectively. For table recognition in modern documents, we just treat the table text line regions as cells because there is no training data for this task. The inference of row ranges and column ranges of cells are the same as that in historical documents.

### *I. Table Fan (Track A)*

**Yibo Li**
Individual
*li597383845@gmail.com*

We use the YoloV3 model and separate the dataset into archive documents dataset and modern dataset.

### *J. Table Radar (Track A)*

**Fengjun Guo and Zhuyan Zhang**
CCi Intelligence Co., Ltd.
*zhuyan_zhang@intsig.net*

Preprocess: Train a classifier to classify modern and history samples. Process: Train two faster-rcnn models for modern and history samples respectively to detect tables. Post Process:

1) Merge the regions whose overlapped areas are larger than defined threshold.
2) Detect lines in candidate table regions. If the detected line extends over table border, extend the table region accordingly.(only for modern samples)

### *K. TJNU202-2 (Track A)*

**Yuanping Zhu and Ningning Sun**
Tianjin Normal University
*zhuyuanping@tjnu.edu.cn*

There exists modern documents and achieve documents in the task. We use different methods to process the two datasets.

1) Modern documents: Firstly, coarse table detection is implemented through Faster R-CNN network. Secondly, corner locating is implemented through RPN

and refined through Fast R-CNN network. Corner grouping and filtering are implemented through a post-processing algorithm. Therefore, unreliable corners are filtered. Thirdly, table boundaries are adjusted and refined via reliable corner groups, which improves the precision of table boundary locating.

2) Archive documents: Firstly, we made data augmentation for achieve documents. Secondly, we trained them on faster rcnn network. Thirdly, table detection is implemented.

## IV. Performance Evaluation

The ICDAR 2019 cTDaR evaluates two aspects of table analysis: table detection and recognition. We choose the metric Intersection over Union (IoU) to evaluate the performance of table region detection, and apply a cell's adjacency relation-based table structure evaluation (based on Gobel et al. [1]) to evaluate that of table recognition. Based on these metrics, overall performances of various algorithms can be compared with each other.

The following subsections describe the evaluation in detail. The evaluation tools are also provided on the competition website http://sac.founderit.com.

### A. Table detection (Track A)

IoU is calculated to tell if a table region is correctly detected. It's used to measure the overlapping of the detected polygons:

$$IoU = \frac{area(GTP \cap DTP)}{area(GTP \cup DTP)} \tag{1}$$

where *GTP* defines the Ground Truth Polygon of the table region and *DTP* defines the Detected Table Polygon. IoU has a range from 0 to 1, where 1 suggests the best possible segmentation. When evaluating, different threshold values of IoU will be used to determine if a region is considered as being detected correctly.

Then, the precision and recall values are computed from a method's ranked output. Recall is defined as the proportion of all true positive examples ranked above a given rank. Precision is the proportion of all examples above that rank which are from the positive class. Furthermore, F1 score will be computed as the harmonic average of recall and precision value. Precision, recall and F1 scores are calculated with IoU threshold of 0.6, 0.7, 0.8 and 0.9 respectively.

### B. Table recognition (Track B)

This track is evaluated by comparing the structure of a table that is defined as a matrix of cells. For each cell, participants are required to return the coordinates of a polygon defining the cell (historical documents) or a polygon defining the convex hull of the cell's contents (modern documents). Additionally, participants must provide the start/end column/row information for each cell.

We propose the following metric: cell's adjacency relation-based table structure evaluation (based on Gobel et al. [5]). When comparing predicted and GT table structures, the following procedure is processed: we apply a table matching under IoU at 0.8 to map the table regions, for each pair of matched table region, we map a groundtruth cell to a predicted cell with the highest $IoU$ and $IoU \geq \sigma$, where $\sigma$ is IoU threshold. After identifying valid predicted cells with this mapping, we generate a list of adjacency relations between valid cells and their nearest neighbors in horizontal and vertical directions, where neighbors are reasoned from the column/row information of cells. Blank cells are not represented in the relation grid. No adjacency relations are generated between blank cells or a blank cell and a non-blank cell. So participants are required not to submit blank cells in their results. This 1-D list can be compared with groundtruth list by calculating precision and recall of adjacency relations between cells. For any two adjacency relations from predicted results and groundtruth, if corresponding cells are identical and directions match, the predicted relation is marked as a true positive; otherwise it is marked as a false positive.

Similar to track A, precision, recall and F1 scores are calculated with IoU threshold of 0.6, 0.7, 0.8 and 0.9 respectively.

In the end, the final ranks of teams are decided by the weighted average F1 (WAvg. F1) value of the whole dataset for each track. The WAvg. F1 value is defined as:

$$WAvg.F1 = \frac{\sum_{i=1}^{4} IoU_i \cdot F1@IoU_i}{\sum_{i=1}^{4} IoU_i}$$

which shows that the weight of each F1 value is the corresponding IoU threshold. We think results with higher IoUs are more important than those with lower IoUs, so we use IoU threshold as the weight of each F1 value to get a definitive performance score for convenient comparison.

## V. Results

### A. Track A : Table Detection

The detailed evaluation results for Track-A on the combined dataset are given in Table II, and the results on the archival and modern datasets are given in Table III.

The two first participants (TableRadar, NLPR_PAL) achieve a very good performance, with a weighted average of F1 higher than 93%. It is interesting to point out that they use different approaches, one based on Region Proposal, the other based on pixel-level categorization. Considering dataset types (archival and modern) separately, the ranking is almost the same, except for ABC Fintech which can be explained by their additional training materiel. Most of the competitors preferred to apply a classification step

| Rank | Team | IoU = 0.8 | | IoU = 0.9 | | WAvg. F1 |
|---|---|---|---|---|---|---|
| | | P | R | P | R | |
| 1 | TableRadar | **0.95** | **0.94** | **0.90** | **0.89** | **0.94** |
| 2 | NLPR-PAL | 0.93 | 0.93 | 0.86 | 0.86 | 0.93 |
| 3 | Lenovo Ocean | 0.88 | 0.86 | 0.82 | 0.81 | 0.87 |
| 4 | ABC Fintech | 0.84 | 0.75 | 0.76 | 0.68 | 0.78 |
| 5 | Applica-robots | 0.82 | 0.82 | 0.54 | 0.54 | 0.77 |
| 6 | Table Fan | 0.81 | 0.79 | 0.58 | 0.57 | 0.77 |
| 7 | TJNU202-2 | 0.86 | 0.81 | 0.48 | 0.46 | 0.76 |
| 8 | Cinners-table | 0.70 | 0.63 | 0.62 | 0.56 | 0.67 |
| 9 | Clova AI | 0.48 | 0.67 | 0.22 | 0.31 | 0.56 |
| 10 | HCL IDORAN | 0.33 | 0.41 | 0.20 | 0.25 | 0.36 |
| 11 | AIRL | 0.08 | 0.10 | 0.04 | 0.05 | 0.11 |

Table II
RESULTS FOR THE COMBINED DATASET, TRACK A (DUE TO LIMITED SPACE, ONLY DATA WITH IOU @ 0.8 AND 0.9 IS PRESENTED)

| Rank | Team | WAvg. F1 (Archival) | Team | WAvg. F1 (Modern) |
|---|---|---|---|---|
| 1 | TableRadar | **0.94** | Table Radar | **0.95** |
| 2 | NLPR-PAL | 0.93 | NLPR-PAL | 0.92 |
| 3 | Lenovo Ocean | 0.91 | ABC Fintech | 0.89 |
| 4 | TJNU202-2 | 0.85 | Lenovo Ocean | 0.85 |
| 5 | Applica-robots | 0.82 | Table Fan | 0.85 |
| 6 | Cinners-table | 0.64 | Applica-robots | 0.74 |
| 7 | Table Fan | 0.63 | TJNU202-2 | 0.71 |
| 8 | ABC Fintech | 0.58 | Cinners-table | 0.68 |
| 9 | Clova AI | 0.58 | Clova AI | 0.55 |
| 10 | HCL IDORAN | 0.35 | HCL IDORAN | 0.38 |
| 11 | AIRL | 0.17 | AIRL | 0.05 |

Table III
RESULTS FOR THE ARCHIVAL AND MODERN DATASET, TRACK A

(modern/archival) as the first step in their workflow. Also, it is worth noticing that for most teams, there exists a rapid drop in precision and recall when IoU is changed from 0.8 to 0.9.

### B. Track B : Table Recognition

We had only two submissions for tasks B1 and B2. NLPR-PAL again using a pixel-categorization method clearly outperforms HCL IDORAN. As expected results for B1 (the table region is given) are better than those for B2 which are nevertheless competitive (weighted avg. F1:45.35 (B2) compared to 48.46 (B1)). There is a significant difference between archival and modern documents for B2, certainly due to the lack of training data for modern documents.

Results for Track B by team NLPR-PAL are shown in Table IV. Table V reports the results for Track B of team HCL IDORAN.

| Track \ IoU | IoU = 0.6 | | IoU = 0.7 | | WAvg. F1 |
|---|---|---|---|---|---|
| | P | R | P | R | |
| B1 | 0.76 | 0.83 | 0.69 | 0.75 | 0.48 |
| B2 - combined | 0.68 | 0.74 | 0.62 | 0.67 | 0.45 |
| B2 - archival | 0.72 | 0.77 | 0.65 | 0.70 | 0.47 |
| B2 - modern | 0.32 | 0.42 | 0.27 | 0.35 | 0.20 |

Table IV
RESULTS FOR TEAM NLPR-PAL, TRACK B

| Track \ IoU | IoU = 0.6 | | IoU = 0.7 | | WAvg. F1 |
|---|---|---|---|---|---|
| | P | R | P | R | |
| B1 | 0.25 | 0.05 | 0.23 | 0.04 | 0.06 |
| B2 - combined | 0.09 | 0.02 | 0.08 | 0.01 | 0.02 |
| B2 - archival | 0.12 | 0.02 | 0.11 | 0.01 | 0.02 |
| B2 - modern | 1E-3 | 1E-3 | 1E-3 | 7E-4 | 3E-4 |

Table V
RESULTS FOR TEAM HCL IDORAN, TRACK B

## VI. CONCLUSION

In this paper the ICDAR 2019 Competition on Table Detection and Recognition has been presented. The submitted methods have been evaluated on modern and historical datasets. It has been shown that state-of-the-art methods for table detection achieve an weighted average F1 higher than 90%. Contrary, table recognition reaches only an weighted average F1 of 48% (B1) resp. 45% (B2) which shows, that there is clearly room for improvement (although only 2 methods have been submitted).


## ACKNOWLEDGMENT

This work has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 674943 (project READ). We would also like to thank all archives who contributed to the historical dataset.


## REFERENCES


[1] M. Göbel, T. Hassan, E. Oro, and G. Orsi, "Icdar 2013 table competition," in *2013 12th International Conference on Document Analysis and Recognition*, Aug 2013, pp. 1449–1453.

[2] L. Gao, X. Yi, Z. Jiang, L. Hao, and Z. Tang, "Icdar2017 competition on page object detection," in *14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, vol. 01, Nov 2017, pp. 1417–1422.

[3] H. Déjean, J.-L. Meunier, L. Gao, Y. Huang, Y. Fang, F. Kleber, and E.-M. Lang, "ICDAR 2019 Competition on Table Detection and Recognition (cTDaR)," Apr. 2019, http://sac.founderit.com/. [Online]. Available: https://doi.org/10.5281/zenodo.2649216

[4] cndplab founder, "Icdar2019_ctdar," 2019. [Online]. Available: https://github.com/cndplab-founder/ICDAR2019_cTDaR

[5] M. Göbel, T. Hassan, E. Oro, and G. Orsi, "A methodology for evaluating algorithms for table understanding in pdf documents," in *Proceedings of the 2012 ACM symposium on Document engineering*. ACM, 2012, pp. 45–48.