DeepDeSRT: Deep Learning for Detection and Structure Recognition of Tables in Document Images

Sebastian Schreiber*‡, Stefan Agne‡, Ivo Wolf*, Andreas Dengel†‡, Sheraz Ahmed‡
*Mannheim University of Applied Sciences, Germany
†Kaiserslautern University of Technology, Germany
‡German Research Center for Artificial Intelligence (DFKI), Kaiserslautern, Germany
Email: firstname.lastname@dfki.de

***Abstract*—This paper presents a novel end-to-end system for table understanding in document images called DeepDeSRT. In particular, the contribution of DeepDeSRT is two-fold. First, it presents a deep learning-based solution for table detection in document images. Secondly, it proposes a novel deep learning-based approach for table structure recognition, i.e. identifying rows, columns, and cell positions in the detected tables. In contrast to existing rule-based methods, which rely on heuristics or additional PDF metadata (like, for example, print instructions, character bounding boxes, or line segments), the presented system is data-driven and does not need any heuristics or metadata to detect as well as to recognize tabular structures in document images. Furthermore, in contrast to most existing table detection and structure recognition methods, which are applicable only to PDFs, DeepDeSRT processes document images, which makes it equally suitable for born-digital PDFs (as they can automatically be converted into images) as well as even harder problems, e.g. scanned documents. To gauge the performance of DeepDeSRT, the system is evaluated on the publicly available ICDAR 2013 table competition dataset containing 67 documents with 238 pages overall. Evaluation results reveal that DeepDeSRT outperforms state-of-the-art methods for table detection and structure recognition and achieves F1-measures of 96.77% and 91.44% for table detection and structure recognition, respectively. Additionally, DeepDeSRT is evaluated on a closed dataset from a real use case of a major European aviation company comprising documents which are highly unlike those in ICDAR 2013. Tested on a randomly selected sample from this dataset, DeepDeSRT achieves high detection accuracy for tables which demonstrates the sound generalization capabilities of our system.**

I. Introduction

Processing tables embedded in digital documents is as old as the analysis of structured documents itself [1]. Despite the multitude of methods already available for detecting tables in document images and decomposing them into their structural building blocks [2]–[5], these tasks still prove to be difficult even for modern document processing systems.

The problem of table detection is extremely challenging due to the high degree of intra-class variability. This means it is hard to give a formal definition of what a table looks like because of different layouts, the erratic use of ruling lines for table or structure delineation, or simply because of very diverse table contents [1]. In addition, there is often a significant degree of inter-class similarity to other objects potentially present in documents, e.g. graphics, code listings, or flow charts [3]. This makes it especially hard to hand-craft a set of good features for describing tabular structures. Because of the ongoing use of paper documents, particularly in commercial and corporate environments, and the abundance of tabular data within, document processing pipelines depend on highly accurate table understanding mechanisms.

There are already some approaches available for detecting and decomposing tables but these systems generally rely on ad-hoc heuristics and additional metadata extracted for example from PDF files. Extraction of tables from PDFs does mitigate some of the complexities of working with raw images due to the metadata available during processing. The problem is much harder when detection and structure recognition need to be performed on raw images. Therefore, We propose a more systematic solution, which is independent of brittle support mechanisms.

This paper presents a novel end-to-end system for table detection and structure recognition in document images called DeepDeSRT. The presented method is data driven, based on deep learning, and hence does not require any heuristics or rules to detect tables and to recognize their structure. This approach makes DeepDeSRT applicable to both, images as well as born-digital documents (e.g. PDFs, Word documents, and web pages, as they can be converted to images).

Usually, deep learning-based solutions require lots of labeled training data, which in our case is not available. To solve this problem, DeepDeSRT uses the concept of transfer learning and domain adaptation for both table detection and table structure recognition. In particular the contributions of DeepDeSRT are the following:

- We present a deep learning-based solution for table detection, where the domain of general purpose object detectors is adapted to the highly different realm of document images. Transfer learning is performed by carefully fine-tuning a pre-trained model of *Faster R-CNN* by Ren et al. [6] for the detection of tables in documents.
- Furthermore, we present a deep learning-based solution for table structure recognition (i.e. the identification of

rows, columns, and cells) where again the general purpose domain is adapted and transfer leaning is performed by augmenting and fine-tuning an FCN semantic segmentation model by Shelhamer et al. [7] pre-trained on Pascal VOC 2011 [8].
- We present another proof for the efficacy of fine-tuning deep neural networks even when source and target domains are highly dissimilar and the target training set is rather small.

II. Related Work

Several works have been published on the topic of table understanding and there are comprehensive surveys available describing and summarizing the state-of-the-art in the field [1]–[5]. For the sake of brevity, we will hence focus on very recent work only as well as methods utilizing machine learning techniques and will leave the discussion of traditional approaches which primarily exploit visual clues, heuristics, and formal table templates to the aforementioned surveys.

A. Table Detection

Cesarini et al. were one of the first to apply machine learning techniques to the table detection task back in 2002. Their proposed method called *Tabfinder* [9] first transforms a document into an MXY tree representation and then searches for blocks surrounded by horizontal or vertical lines. A subsequent depth-first search starting at such nodes yields potential table candidates.

Another early data-driven approach by Silva [10] develops more and more complex Hidden-Markov-Models (HMMs) which model the joint probability distribution over sequential observations of visual page elements and the hidden state of a line belonging to a table or not. In her Ph.D. thesis [11] Silva builds on her earlier findings and emphasizes the importance of probabilistic models and the combination of multiple approaches over brittle heuristics.

Kasar et al. derive a set of hand-crafted features which they subsequently use to train a classifier based on an SVM [12]. Although no heuristic rules or user-defined parameters are needed, the method's area of application stays limited because it relies heavily on the presence of visible ruling lines.

With the help of unsupervised learning of weak labels for every line in a document as well as linguistic information extracted from a region, Fan and Kim [13] successfully trained an ensemble of generative and discriminative classifiers to detect tables.

Recently, the first method we know about applying deep learning techniques to table detection in PDF documents was published by Hao et al. [14]. In addition to the learned features the authors also make use of loose heuristic rules as well as meta information from the underlying PDF documents.

Not based on machine learning but reporting competitive results on the well-known ICDAR 2013 dataset [15], Tran et al. propose a method based on regions of interest and the spatial arrangement of extracted text blocks [16]. Different from most other approaches, their method works directly on document images. *Since the authors do not disclose which parts of the ICDAR 2013 table competition dataset were used for design and analysis of their algorithm, their results can not be directly compared to ours. The same is true for the follow-up works published by this group.*

B. Table Structure Recognition

Directly compared to table detection, research in table structure identification is rather scarce. One of the earliest successful systems described in literature is the T-RECS approach by Kieninger and Dengel [17] where words are first grouped into columns by evaluating their horizontal overlaps and subsequently further divided into cells based on the columns' margin structure.

Wang et al. [18] developed a seven-step process based on probability optimization to solve the table structure understanding problem similar to the X-Y cut algorithm. The probabilities used by their system are derived from measurements taken from a training corpus. Hence their approach is also data-driven.

Bearing the adaptability to different input sources in mind, the system proposed by Shigarov et al. [19] offers thorough configuration of the algorithms, thresholds, and rule sets used for decomposing tables. Their approach therefore relies heavily on PDF metadata like font and character bounding boxes as well as ad-hoc heuristics.

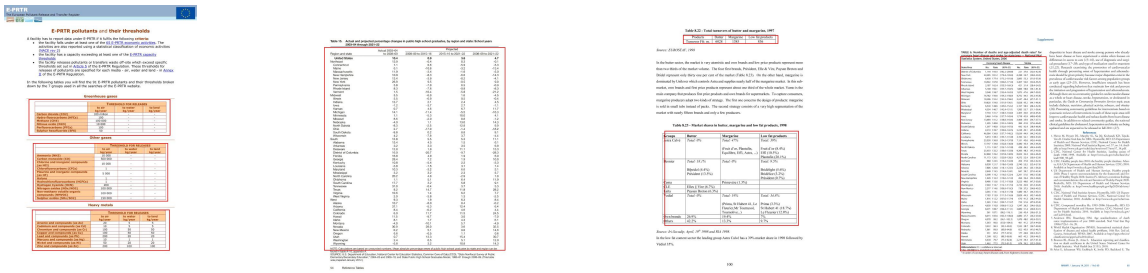
III. DeepDeSRT: The Presented Approach

This section provides details about the proposed DeepDeSRT system, which consists of two separate parts for table detection and structure recognition. Since the two tasks are inherently different, each is tackled by a unique solution strategy utilizing deep learning methods.

A. Deep Learning for Table Detection

The first step in table understanding is detecting the locations of tables within a document. Conceptually, the problem is similar to the detection of objects in natural scene images. Therefore, in the presented approach we used domain adaptation and transfer learning by utilizing deep learning-based object detection frameworks originally created for natural scene images and tested their ability to cope with tabular structures in scanned document images. Due to the compelling performance and publicly available code base, we choose *Faster R-CNN* [6], subsequently called FRCNN, as the basic framework used in our detection system. The FRCNN approach, disregarding its age, does still yield state-of-the-art performance and is an inherent part of many modern architectures [20]–[22].

Tables in document images share some important characteristics with objects in natural scene images, e.g. they can be visually distinguished from background rather easily and there are other elements on a page that look similar but actually belong to different classes. These analogies lead to the assumption that existing object detection systems should be able to cope with table detection rather well but will also



(a) Multiple tables (b) Large table (c) Small table (d) Page column alike

Fig. 1. DeepDeSRT table detection results on the ICDAR 2013 table competition dataset.

suffer from the same limitations. This hypothesis is verified by our results.

FRCN models consist of two distinct parts: First they generate region proposals based on the input image by a so-called region proposal network (RPN). Afterwards, these proposals are classified using a Fast-RCNN [23] network. Both modules share parameters and can be trained end-to-end [6]. As the backbone of these two modules, we use ZFNet proposed by Zeiler and Fergus [24] and the much deeper VGG-16 network by Simonyan and Zisserman [25]. Ren et al. provide readily trained FRCNN models for both these base networks which can thus be used for fine-tuning in our experiments. Using two different base architectures allows for evaluating the impact of network depth on the final results.

B. Deep Learning for Structure Recognition

After a table has successfully been detected and its location is known to the system, the next challenge in understanding its contents is to recognize and locate the rows and columns which make up the physical structure of the table. This step is inherently different from the preceding table detection. The key difference is not only that there are significantly more rows and columns present in a table image than there are tables in a document but these tabular structures are generally located in very close proximity. These two factors make this task so difficult for FRCNN and ask for a different approach.

When thinking about fine-grained segmentation of images what comes to mind are the recent successes of deep learning-based semantic segmentation tools. The FCN-Xs architectures by Shelhamer et al. [7] combine fully convolutional networks for arbitrary input sizes with skip connections, a technique also known as skip pooling [26] or Hyper Features [27] used to integrate semantically coarse but naturally high resolution features from lower layers, and fractionally strided convolutions which increase the resolution of the final segmentation masks.

While their highest resolution architecture FCN-8s does include features from the pool4 and pool3 layers of the underlying VGG-16 [25] base network and the authors report only minuscule improvements when fusing in additional pooling layers [7], we strongly believe that extra details extracted by shallower layers can help with obtaining cleaner delineation results for rows and columns. The reason behind this assumption is that the basic features detected by early network layers, e.g. edges and changes in color, can facilitate boundary detection. Therefore, we added two extra skip connections incorporating features from the pool2 and pool1 layers resulting in an FCN-2s architecture, which is also briefly mentioned in [28] where it is used for edge detection.

In a first implementation of FCN-2s, we adhered to the approach of Shelhamer et al. where skip-pooled features are scaled by a fixed factor before getting used for scoring and fusion. This factor decreases by two orders of magnitude with every pooling level resulting in a scaling factor of 10^{-8} for pool1 features. While this turned out to work comparatively well for column segmentation and detection, the corresponding row models were lagging behind in performance. To alleviate this issue, we introduced the network to the possibility of learning the scaling factors itself during training. For this purpose, we exchanged the scale layers for normalization layers which also provide learnable scaling capabilities. The implementation of this layer type was first introduced by Liu et al. in [29] and later improved for application in their *Single Shot Multibox Detector* [30]. For our purposes, we chose the latter variant.

IV. Experiments and Results

This section provides details on the different experiments performed to evaluate DeepDeSRT on the tasks of table detection and structure recognition. DeepDeSRT is evaluated on a publicly available dataset (the well-known ICDAR 2013 table competition dataset [15]) as well as a closed dataset containing documents from a major European aviation company.

A. Table Detection

As DeepDeSRT is based on a data-driven approach, there was the need for a sufficiently large dataset. The largest publicly available dataset is the Marmot dataset for table recognition[1] published by the Institute of Computer Science and Technology of Peking University and further described in [31]. Since there is no default split for the dataset available, we set up a random 80-20 split into training and validation data, respectively. This split resulted in $1,600$ training images and left another 399 images for validation. The ratio of positive

[1] http://www.icst.pku.edu.cn/cpdp/data/marmot_data.htm

to negative images is approximately 1:1 for both sets. In order to achieve the best results possible, we cleaned out errors in the ground-truth annotations of the dataset resulting in our version called *Marmot clean*, subsequently referred to as *MarmotC*. Because the number of images in MarmotC is not sufficient for training deep neural networks from scratch, we rely on the powerful techniques of transfer learning and domain adaptation to get our models to converge to good weight configurations. *We want to emphasize at this point that we did not use any part of the ICDAR 2013 table competition dataset [15] during training or validation of our models.*

We trained a group of FRCNN models based on the different backbone CNN architectures described in Section III-A. For fine-tuning we used the models provided by [6] which are pre-trained on one of three different datasets: ImageNet [32], Pascal VOC [8], or Microsoft COCO [33]. The remaining training parameters were taken from [6]. Since our training set consists of roughly $1,600$ images and the original training schedule of Ren et al. accounted for about 28 epochs, we trained all our models for $30,000$ iterations with a batch size of two to ensure convergence. To detect possible over-fitting, we monitored performance on the validation set during training.

We evaluated all our models on the MarmotC validation split and chose the best performing network to be trained again on complete MarmotC. This training process yields the model we apply in our DeepDeSRT system and for which we also report performance on ICDAR 2013. For reporting model performance, we chose the metrics prevalent in the document processing community, i.e. recall, precision and F1-measure. We computed these measures the way it is described in [15] by first computing the scores for each document individually and subsequently taking their average across all documents. We also added average precision (AP) and average recall (AR) to have aggregated metrics as well.

The results reported in this paper are achieved when limiting the detections to those with prediction confidence scores greater than 99%. Based on this criteria, DeepDeSRT achieves state-of-the-art performance across all metrics on the well-known ICDAR 2013 table competition dataset [15] with only one confusion with a non-table element. Table I compares our proposed system with results reported by other authors on ICDAR 2013. It is important to mention that the systems which are processing PDF documents are not directly comparable with DeepDeSRT, as they have access to lots of metadata included in the PDF files, while DeepDeSRT only uses the raw images with no additional metadata. This makes the problem more challenging than using PDF files. The results of the systems operating on PDFs are listed in Table I only for completeness.

Figure 1 shows some sample detections directly taken from this evaluation. They illustrate DeepDeSRT's ability to accurately locate multiple medium-sized tables within a page as well as large page-filling tables, very small tables only a few inches in size, and even tables which could be mistaken for columns of the page layout. Examples for existing issues of the system, like false negatives when using high confidence scores or bar charts mistaken for a table are given in Figure 3.

TABLE I
TABLE DETECTION PERFORMANCE OF DEEPDESRT AND STATE-OF-THE-ART METHODS. *Existing PDF-based approaches are not directly comparable as they operate on a different input format with access to metadata.*

Input	Method	Recall	Precision	F1-measure
Image	**DeepDeSRT**	**0.9615**	**0.9740**	**0.9677**
	Tran et al. [16]	0.9636	0.9521	0.9578
PDF	Hao et al. [14]	0.9215	0.9724	0.9463
	Silva [11]	0.9831	0.9292	0.9554
	Nurminen [15]	0.9077	0.9210	0.9143
	Yildiz [34]	0.8530	0.6399	0.7313

TABLE II
TABLE STRUCTURE RECOGNITION PERFORMANCE OF DEEPDESRT AND STATE-OF-THE-ART METHODS. *Existing PDF-based approaches are not directly comparable as they operate on a different input format with access to metadata.*

Input	Method	Recall	Precision	F1-measure
Images	**DeepDeSRT**	**0.8736**	**0.9593**	**0.9144**
PDFs	Shigarov et al. C_1 [19]	0.9121	0.9180	0.9150
	Shigarov et al. C_2 [19]	0.9233	0.9499	0.9364
	Nurminen [15]	0.9409	0.9512	0.9460
	Silva [11]	0.6401	0.6144	0.6270
	Hsu et al. [15]	0.4811	0.5704	0.5220

In addition to the evaluation on ICDAR 2013, DeepDesRT is also tested on a randomly selected sample from the above mentioned closed dataset of an aviation company. There it achieves an F1-measure of 91.37%. It is important to mention that the documents in the closed dataset are more complex and show a broader variability of table styles than the tables contained in the ICDAR 2013 dataset.

B. Table Structure Recognition

For recognizing the structure of tables we first simply applied the same FRCNN-based technique as before. While the results achieved when detecting columns were at least mediocre, this approach yielded only very bad performance when rows were considered. Further investigation on this issue brought to light that the biggest problems with table structure recognition, especially with rows, are the vast number of objects in a very confined space as well as the extreme aspect ratios of the structure elements. The large effective strides of 16 pixels at layer conv5_3 of FRCNN and similar models probably induce the network to overlook important visual features that could help detect and differentiate between row instances.

A different approach for dividing images into their constituent parts is semantic segmentation. Using the architecture described in Section III-B significantly improves performance when compared to the FRCNN approach. However, the results were still not satisfactory: Although semantic segmentation metrics looked promising at first, only very few rows were detected by the model. Further analysis of the input segmentation masks suggested that the gaps of background pixels between

the individual rows are just not big enough to sufficiently penalize the model during training. Therefore, the model simply learned that everything inside a table is accumulated row pixels. Hence, increasing the importance of background pixels was the area we focused on next.

To increase the amount of background separating each row from the remaining structure components, we introduce an additional pre-processing step to the model: before being processed by the network, all tables are stretched vertically to facilitate the separation of rows and in a second, independent run horizontally to make the delimitations between columns easier to spot. This pre-processing is only minimally invasive and feels very natural.

Exchanging scaling for normalization layers as described in Section III-B yielded in conflicting results: Without the aforementioned input pre-processing, the learned scaling is superior to fixed scaling parameters while when class-specific pre-processing is included, this advantage diminishes or gets even reversed. Also, row and column models behave the exact opposite way.

We also add some lightweight post-processing to the system which fixes three problems we have encountered: spurious detection fragments as well as severed and conjoined structures. The first one is fixed by simply removing all bounding boxes which cover less than 0.5% of the pixels of the input image. Severed structures are brought together by horizontally (vertically) merging detected row (column) structures with a significant vertical (horizontal) overlap. Finally, conjoined structures are separated by a morphological opening. All thresholds were identified experimentally by visual inspection of results on images not related to the training or validation set.

The FCN-based segmentation models of DeepDeSRT were trained for $60,000$ iterations employing a standard SGD optimizer with a fixed learning rate of 10^{-10} and classical momentum of 0.99. The batch size was set to one as suggested by the original paper [7]. Table II shows the results of the system for table structure recognition. We want to emphasize, that the scores obtained by our system and the other listed approaches can not be compared directly: We were only able to test DeepDeSRT on a randomly chosen test split of the ICDAR 2013 table competition dataset [15] which contains just 34 images since we used the remaining images for training. Furthermore, while all other methods operate on PDF files, we process raw images instead. We are going to alleviate the first issue in the future by using a dedicated training set for our models.

Figure 2 shows the qualitative results of DeepDeSRT. These examples clearly show that DeepDeSRT successfully learned to cope with missing ruling lines even when rows and columns are in close vicinity. On the other hand, although it achieves state-of-the-art results for structure recognition, it is still not perfect. Figure 3 shows the cases where DeepDeSRT has problem with nested row hierarchies or extremely close adjacent rows.

(a) Row detection, no ruling lines present

(b) Column detection, no ruling lines present

Fig. 2. DeepDeSRT table structure recognition samples from the ICDAR 2013 table competition dataset.

V. Conclusion and Future Work

This paper presents a novel end-to-end system for table detection and structure recognition. In this paper, it is shown that existing object detectors based on CNN architectures which were originally developed for objects in natural scene images are also very effective for detecting tables in documents thanks to the powerful approaches of transfer learning and domain adaptation. Subsequently, we went one step further and utilized recently published insights from deep learning-based semantic segmentation research for recognizing structures within tables. Performance of our proposed system DeepDeSRT is evaluated on the publicly available ICDAR 2013 table competition dataset for both tasks and on a closed dataset containing documents from a big European aviation company for table detection only. Evaluation results of DeepDeSRT outperform all of the existing methods, even though they are not comparable due to extensive use of PDF metadata, which is not available when processing raw images. Qualitative detection samples are given for both table understanding sub-disciplines pointing out the high quality of our method.

In the future, we are going to enhance DeepDeSRT by resolving its persisting issues with recognizing structures which are in very close proximity to other elements of interest in an image. Also, we are going to train the structure recognition network on a dedicated dataset so we can report performance on the full ICDAR 2013 table competition dataset.

References

[1] B. Coüasnon and A. Lemaitre, "Recognition of Tables and Forms," in *Handbook of Document Image Processing and Recognition*, D. Doermann and K. Tombre, Eds. London: Springer London, 2014, pp. 647–677.

[2] R. Zanibbi, D. Blostein, and J. R. Cordy, "A survey of table recognition: Models, observations, transformations, and inferences," *International Journal on Document Analysis and Recognition (IJDAR)*, vol. 7, no. 1, pp. 1–16, 2004.

[3] D. W. Embley, M. Hurst, D. Lopresti, and G. Nagy, "Table-processing paradigms: A research survey," *International Journal of Document Analysis and Recognition (IJDAR)*, vol. 8, no. 2-3, pp. 66–86, 2006.

[4] A. C. e Silva, A. M. Jorge, and L. Torgo, "Design of an end-to-end method to extract information from tables," *International Journal of Document Analysis and Recognition (IJDAR)*, vol. 8, no. 2-3, pp. 144–171, 2006.

(a) Close columns (b) Bar chart confusion (c) Nested row hierarchy (d) Very close-by rows

Fig. 3. Some failure cases of DeepDeSRT.

[5] S. Khusro, A. Latif, and I. Ullah, "On methods and tools of table detection, extraction and annotation in PDF documents," *Journal of Information Science*, vol. 41, no. 1, pp. 41–57, 2015.

[6] S. Ren, K. He, R. B. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," Microsoft Research, Tech. Rep., 2015. [Online]. Available: https://arxiv.org/abs/1506.01497

[7] E. Shelhamer, J. Long, and T. Darrell, "Fully Convolutional Networks for Semantic Segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 4, pp. 640–651, 2017.

[8] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal Visual Object Classes (VOC) Challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, 2010.

[9] F. Cesarini, S. Marinai, L. Sarti, and G. Soda, "Trainable Table Location in Document Images," in *16th International Conference on Pattern Recognition, ICPR 2002*. IEEE Computer Society, 2002, pp. 236–240.

[10] A. C. e Silva, "Learning Rich Hidden Markov Models in Document Analysis: Table Location," in *ICDAR 2009: 10th International Conference on Document Analysis and Recognition*. IEEE Computer Society, 2009, pp. 843–847.

[11] ——, "Parts that add up to a whole: a framework for the analysis of tables," Ph.D. dissertation, University of Edinburgh, Edinburgh, Scotland, UK, 2010.

[12] T. Kasar, P. Barlas, S. Adam, C. Chatelain, and T. Paquet, "Learning to Detect Tables in Scanned Document Images Using Line Information." IEEE Computer Society, 2013, pp. 1185–1189.

[13] M. Fan and D. S. Kim, "Detecting Table Region in PDF Documents Using Distant Supervision," Tech. Rep., 2015. [Online]. Available: http://arxiv.org/abs/1506.08891

[14] L. Hao, L. Gao, X. Yi, and Z. Tang, "A Table Detection Method for PDF Documents Based on Convolutional Neural Networks," in *12th IAPR Workshop on Document Analysis Systems (DAS) 2016*. IEEE Computer Society, 2016, pp. 287–292.

[15] M. C. Göbel, T. Hassan, E. Oro, and G. Orsi, "ICDAR 2013 Table Competition." IEEE Computer Society, 2013, pp. 1449–1453.

[16] D. N. Tran, T. A. Tran, A.-R. Oh, S.-H. Kim, and I.-S. Na, "Table Detection from Document Image using Vertical Arrangement of Text Blocks," *International Journal of Contents*, vol. 11, no. 4, pp. 77–85, 2015.

[17] 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.

[18] 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.

[19] A. Shigarov, A. Mikhailov, and A. Altaev, "Configurable Table Structure Recognition in Untagged PDF documents," in *Proceedings of the 2016 ACM Symposium on Document Engineering, DocEng 2016*, R. Sablatnig and T. Hassan, Eds. ACM, 2016, pp. 119–122.

[20] S. Hong, B. Roh, K.-H. Kim, Y. Cheon, and M. Park, "PVANet: Lightweight Deep Neural Networks for Real-time Object Detection," 1st International Workshop on Efficient Methods for Deep Neural Networks (EMDNN 2016), Conference in Neural Information Processing Systems NIPS 2016, Barcelona, Spain.

[21] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, and K. Murphy, "Speed/accuracy trade-offs for modern convolutional object detectors," Tech. Rep., 2016. [Online]. Available: http://arxiv.org/abs/1611.10012

[22] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," Tech. Rep. [Online]. Available: https://arxiv.org/abs/1703.06870

[23] R. Girshick, "Fast R-CNN." Piscataway, NJ and Piscataway, NJ: IEEE Computer Society, 2015, pp. 1440–1448.

[24] M. D. Zeiler and R. Fergus, "Visualizing and Understanding Convolutional Networks," in *Computer Vision – ECCV 2014*, ser. Lecture Notes in Computer Science, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham, Switzerland: Springer International Publishing, 2014, pp. 818–833.

[25] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," in *Proceedings of the 3rd International Conference on Learning Representations (ICLR 2015)*, 2015.

[26] S. Bell, C. L. Zitnick, K. Bala, and R. Girshick, "Inside-Outside Net: Detecting Objects in Context with Skip Pooling and Recurrent Neural Networks." IEEE Computer Society, 2016, pp. 2874–2883.

[27] T. Kong, A. Yao, Y. Chen, and F. Sun, "HyperNet: Towards Accurate Region Proposal Generation and Joint Object Detection." IEEE Computer Society, 2016, pp. 845–853.

[28] S. Xie and Z. Tu, "Holistically-Nested Edge Detection." Piscataway, NJ and Piscataway, NJ: IEEE Computer Society, 2015, pp. 1395–1403.

[29] W. Liu, A. Rabinovich, and A. C. Berg, "ParseNet: Looking Wider to See Better," 2016.

[30] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. E. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single Shot MultiBox Detector," ser. Lecture Notes in Computer Science, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds., vol. 9905. Cham, Switzerland: Springer International, 2016, pp. 21–37.

[31] J. Fang, X. Tao, Z. Tang, R. Qiu, and Y. Liu, "Dataset, Ground-Truth and Performance Metrics for Table Detection Evaluation," in *10th IAPR International Workshop on Document Analysis Systems (DAS) 2012*, M. Blumstein, U. Pal, and S. Uchida, Eds. IEEE Computer Society, 2012, pp. 445–449.

[32] 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," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[33] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common Objects in Context," ser. Lecture Notes in Computer Science, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds., vol. 8693. Cham, Switzerland: Springer International Publishing, 2014, pp. 740–755.

[34] B. Yildiz, K. Kaiser, and S. Miksch, "pdf2table: A Method to Extract Table Information from PDF Files," in *Proceedings of the 2nd Indian International Conference on Artificial Intelligence (IICAI) 2005*, B. Prasad, Ed., 2005, pp. 1773–1785.

[35] *ICDAR 2013: 12th International Conference on Document Analysis and Recognition*. IEEE Computer Society, 2013.

[36] *2015 IEEE International Conference on Computer Vision (ICCV)*. Piscataway, NJ and Piscataway, NJ: IEEE Computer Society, 2015.

[37] *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, 2016.