Table Detection Using Deep Learning

Conference Paper · September 2017

DOI: 10.1109/ICDAR.2017.131

4 authors, including:

Azka Gilani
National University of Sciences and Technology
1 PUBLICATION **0** CITATIONS
SEE PROFILE

Faisal Shafait
University of Western Australia
162 PUBLICATIONS **2,688** CITATIONS
SEE PROFILE

Table Detection using Deep Learning

Azka Gilani*, Shah Rukh Qasim*, Imran Malik† and Faisal Shafait‡
*National University of Sciences and Technology (NUST), Islamabad, Pakistan
Email: agilani(dot)mscs15seecs,14beesqasim,malik(dot)imran,faisal(dot)shafait(at)seecs.edu.pk

***Abstract*— Table detection is a crucial step in many document analysis applications as tables are used for presenting essential information to the reader in a structured manner. It is a hard problem due to varying layouts and encodings of the tables. Researchers have proposed numerous techniques for table detection based on layout analysis of documents. Most of these techniques fail to generalize because they rely on hand engineered features which are not robust to layout variations. In this paper, we have presented a deep learning based method for table detection. In the proposed method, document images are first pre-processed. These images are then fed to a Region Proposal Network followed by a fully connected neural network for table detection. The proposed method works with high precision on document images with varying layouts that include documents, research papers, and magazines. We have done our evaluations on publicly available UNLV dataset where it beats Tesseract's state of the art table detection system by a significant margin.**

I. Introduction

Tables are widely used for presenting structural and functional information. They are present in diverse classes of documents including newspapers, research articles and scientific documents, etc. Tables enable readers to rapidly compare, analyse and understand facts present in documents. Table detection in documents is significant in the field of documents analysis and recognition; hence it has attracted a number of researchers to make their contributions in this domain.

Table detection is carried out by layout and content analysis of documents. Tables have varying layouts and variety of encodings. Because of this reason, writing a general algorithm for table detection is very hard. Hence, table detection is considered a hard problem in scientific society. Large number of researches have been carried out in this field but most of them have their limitations. Existing commercial and open source techniques for document analysis including Tesseract lack the capability to completely detect table regions from document images [1].

In the recent years, deep learning techniques have greatly improved the results on various computer vision problems. Recently, Hao et al. [2] presented an approach for table detection in documents using deep learning. Their proposed method employs combination of custom algorithms and machine learning in order to generate region proposals and to detect whether a table exists in the proposed region or not. The major limitation of this method is that it is limited to PDF (Portable Document Format) documents only which are non-raster. Another limitation is that it works well on the tables that have ruling lines but fails to detect those without ruling lines and those which are spanned across multiple columns. Hence in order to improve the performance of table detection and to make up for the limitations of prior techniques, this paper proposes a methodology for table detection based on purely based on deep learning without using extensive pre or post processing. To explain it further, the document image is transformed into a new image. Then, this paper uses Faster Recurrent Convolutional Neural Network (Faster R-CNN) as the deep learning module. In contrary to Hao et al. [2] technique, the Faster-RCNN computes region proposals itself and then helps in determining whether the selected area is a table or not. Our approach has a major advantage of being invariant to changes in table structure and layout as it can be fined-tuned to work on any dataset very easily. This capability is not present in any of the existing approaches. Hence, we make a significant contribution to table detection problem by making it data-driven. Additionally, we have used publicly available UNLV dataset for evaluation of our proposed methodology [3] where it gives better results than Tesseract's table detection system. We have also compared our results with the commercial market leading OCR Engine, Abbyy Cloud OCR SDK [4].

The rest of the paper is organized as follows: Section II describes researches related to table detection. Section 3 describes our proposed methodology that consists of pre-processing and detection module. Section IV explains describes performance measures that have been used to evaluate our system and explains experimental results. Section V concludes the paper and provides some directions for future research.

II. Literature Review

Several researchers have reported their work regarding table detection in document images. Kieninger et al. [5]–[7] proposed an algorithm for table spotting and structure extraction from documents called T-Recs. This system takes word bounding boxes as input. They are clustered to form segmentation graph using bottom-up approach. The key problem with this technique is that it depends entirely on word bounding boxes and is unable to perform well in presence of multi-column layouts.

Another approach was proposed by Wang et al. [8]. It detects table lines depending on distance between consecutive words. After that, horizontal consecutive words are grouped together with vertical adjacent lines in order to propose table entity candidates. This statistical approach assumes that maximum number of columns in the document is two and designs the algorithm according to three layout templates (single column, double column, mixed column). Then, column

classification algorithm is applied to find out column layout of the page and use this information as prior knowledge for table spotting. Major limitation of this technique is that it can only work on those templates for which it has been designed.

Hu et al. [9] presented an approach for table detection while assuming that input images are single columned. Like previous methods, this technique can not be applied on multi-column layouts. Shafait et al. [10] presented another approach for table detection in heterogeneous documents. This system is integrated into open source Tesseract OCR engine. It works well on large variety of documents but major limitation is that it is a traditional technique and not data-driven.

Tupaj et al. [11] proposed an OCR based table detection technique. The system searches for sequences of table-like lines based on the keywords that might be present in the table headers. The line that contains keyword is regarded as the starting line while subsequent lines are then analyzed to match with predefined set of tokens which are then categorized as table structure. The limitation of this technique is that it depends highly on the keywords that might appear in table headers.

Harit et al. [12] proposed a technique for table detection based on the identification of unique table start and trailer pattern. The major limitation of this method is that it will not work properly whenever the table start patterns are not unique in document images.

Gatos et al. [13] proposed an approach for table detection by finding area of intersection between the horizontal and vertical lines. Table are then reconstructed by drawing corresponding horizontal and vertical lines that are connected to intersection pairs. The limitation of this system is that it works only for the documents in which the table rows and columns are separated by ruling lines. Costa e Silva [14] presented a technique for table detection using Hidden Markov Models (HMMs). The system extracts text from PDF files using pdftotext Linux utility. Feature vectors are then computed on basis of spaces present between the text. The major limitation of this technique is that it works only on non-raster PDF files that do not have any noise.

Kasar [15] presented a method to locate tables by identifying column and row line separators. This system then employs run-length approach in order to detect horizontal and vertical lines from input image. From each group of horizontal and vertical lines, a set of 26 low level features are extracted and passed to Support Vector Machine (SVM) which then detects the table. The major limitation of this approach is that it will fail on tables without ruling lines.

Jahan et al. [16] presented a method that uses local thresholds for word spacing and line height for localization and extraction of table regions from document images. The major limitation of this method is that it detects table regions along with surrounding text regions. Hence it cannot be used for localization of table regions only.

Anh et al. [17] presented a hybrid approach for table detection in document images. This system first classifies document in text and non-text regions. On the basis of that, it uses a hybrid method to find candidate table regions. These regions are then examined to get table regions. This approach will fail if table is spanned across multiple columns in the document. Moreover, it will not work for scanned images as it does not use any heuristic filter to cater for noisy images.

Hao et al. [2] presented deep learning based approach for table detection. This system computes region proposals from document images through some predefined set of rules. These region proposals are then passed to the CNN that detects whether a certain region proposal belongs to table region or not. The major limitation is that it works well for tables with ruling lines but fails to localize table regions if the table is spanned across multiple columns. Another limitation is that it works only on non-raster PDF documents.

In order to make up for the limitations of prior methodologies, this paper attempts to adapt Faster R-CNN, a deep learning technique used for object detection in natural images, to solve table detection problem.

III. Proposed Methodology

The proposed method consists of two major modules: Image transformation and table detection. Documents consist of content region and blank spaces. Image transformation is applied in order to separate these regions while the table detection module uses Faster R-CNN as a basic element of deep network. Faster R-CNN is highly dependant on combined network that is composed of Region Proposal Networks (RPN) and Fast R-CNN. In this section we will describe each module in detail.

A. Image Transformation

Image Transformation is the initial step of our proposed methodology. Faster R-CNN [18] was initially proposed for natural images. Hence image transformation plays a preliminary role in conversion of document images to natural images as close as possible so that we can easily fine-tune on existing Faster R-CNN models. Distance transform [19]–[21] is a derived representation of digital image. It calculates the precise distance between text regions and white spaces present in the document image which can give a good estimate about presence of a table region. In our proposed methodology, we have used different types of distance transforms so that different features can be stored in all three channels. Image transformation is done using the following procedure:

procedure IMAGE TRANSFORMATION(I)
  b $\leftarrow$ EuclideanDistanceTransform(I)
  g $\leftarrow$ LinearDistanceTransform(I)
  r $\leftarrow$ MaxDistanceTransform(I)
  P $\leftarrow$ ChannelMerge(b,g,r)
  return P

The transformation algorithm takes binary image as an input. It then computes Euclidean distance transform, linear distance transform and max distance transform [19]–[21] on blue, green and red channels of the image respectively. Result

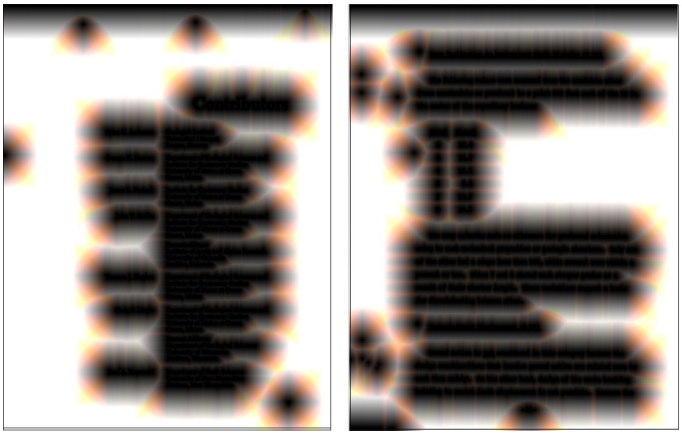


Fig. 1: Transformed Images

of the image transformation algorithm on document images is shown in Figure 1.

B. Table detection

For detection, our approach employed Faster R-CNN [18]. Faster R-CNN was originally proposed for object detection and classification in natural images. It is composed of two modules. The first module is a RPN that propose regions. The region proposals are fed to the second module which is the detector module that was originally proposed in Fast R-CNN [22]. The entire system is a unified network for object detection. Figure 2 shows the architectural diagram for our system.

1) Region Proposal Network: As described in [18], Region Proposal Network (RPN) takes the transformed image as an input and returns an output of a set of rectangular object proposals, each with an objectness score. RPN shares common set of convolutional layers with detector module of Faster-RCNN. Ren et al. has used Zeiler and Fergus model (ZF) [23] and Simonyan and Zisserman model (VGG-16) in their experiments.

In order to generate region proposals, a small network is slided over the convolutional feature map output by the last shared feature map. RPN takes an $n \times n$ spatial window of the input convolutional feature map as an input. It maps each sliding window to a lower dimensional (256-d for ZF model) feature. The complete architecture of network is shown in Figure 2. The feature map is then passed to two fully connected layers that include a regression layer and a classification layer. For this paper we have used default implementation of Faster R-CNN that takes n=3. The fully connected layers of the network are shared across all spatial locations. This architecture [18] is naturally implemented with an $n \times n$ convolutional layer followed by two 1×1 convolutional layers for regression and classification. At each sliding window, Faster R-CNN simultaneously predicts multiple region proposal for each location that can be denoted by *k*. So classification layer has *2k* output scores while regression layer has *4k* outputs that

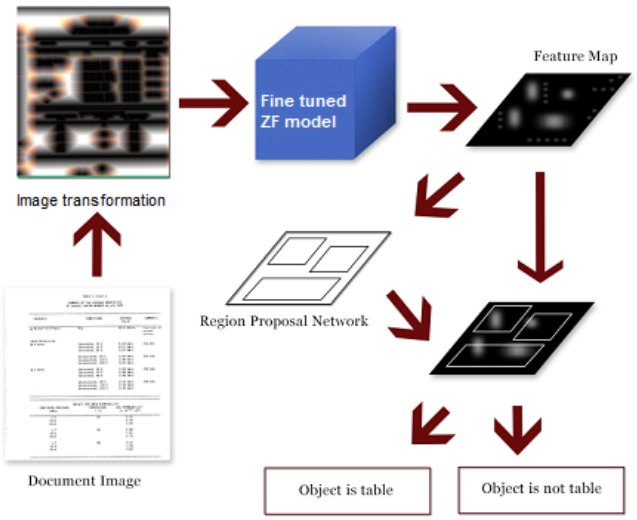


Fig. 2: Our approach: The document image is first transformed and then fed into a fine-tuned CNN model. It outputs a feature map which are fed into region proposal network for proposing candidate table regions. These regions are finally given as input to fully connected detection network along with the convolutional feature map to classify them into tables or non-tables.

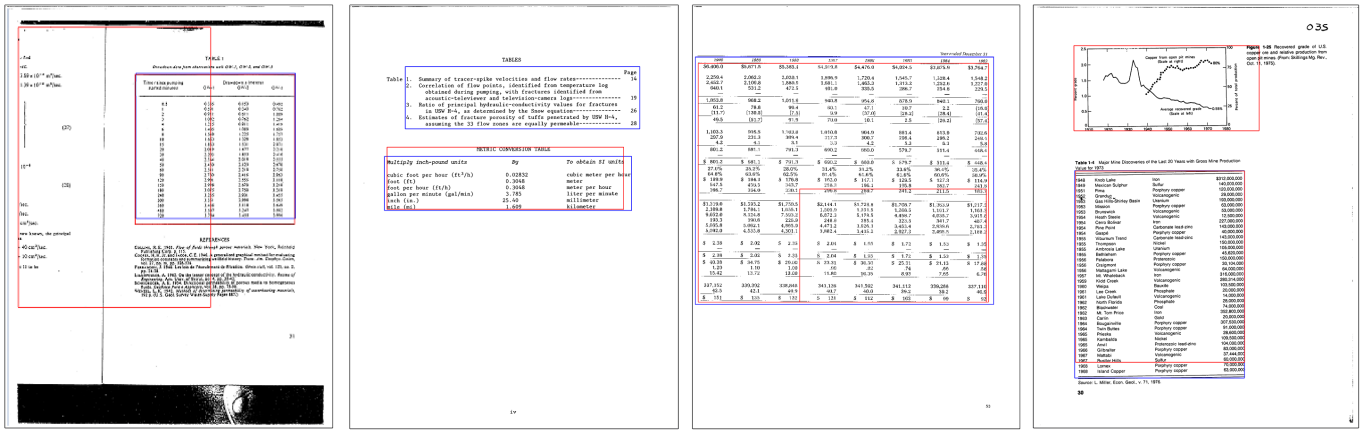
are encoding coordinates of *k* boxes. The *k* regional proposals are then parametrized in relevance to *k* reference boxes which are known as anchors. Faster R-CNN yields *k=9* anchors at each sliding position.

The important fact is that Faster R-CNN generates region proposals that are scale and translational invariant. The RPN is then trained end-to-end by Stochastic Gradient Descent (SGD) and back propagation.In this paper, all the layers are fine tuned by ZF network.

2) Detection network: After the training of network for region proposal generation, these proposals are then passed to the region based object detection CNN's module that will utilize these proposals. Detection module is highly based on the unified network that is composed of RPN and Fast R-CNN with shared convolutional layers. Resultantly, it detect tables from test set and returns the coordinates of bounding boxes of predicted tables.

C. Training

We have used Caffe based implementation of Faster R-CNN [18] to fine tune on our images. Momentum Optimizer with learning rate of 0.001 and a momentum of 0.9 was used. Number of training iterations was 10,000. We trained our system on 2 classes i.e. background and table region. Background class has been used as the negative example (table region is missing) while table class has been used as the positive example (containing table region). Due to this reason, our proposed system doesn't search aggressively for table regions on negative samples.



(a) (b) (c) (d)

Fig. 3: Results showing: (a) Partial detection, (b) Missed, (c) Over-Segmented, and (c) False Positives

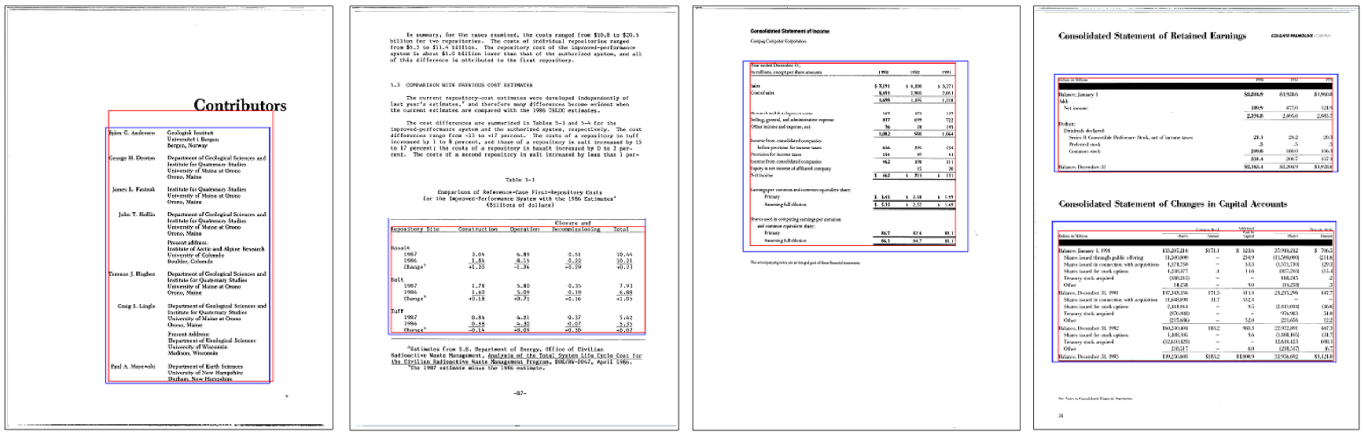


Fig. 4: Some sample images from the UNLV dataset showing detection results of proposed Table Detection approach. Ground truth is blue while the detected regions are red.

IV. Performance Measures

Different performance measures have been mentioned in the literature for evaluation of table detection algorithms. These measures include precision and recall [9], [24] that have been used for evaluating various table detection algorithms [1], [8], [24]–[26]. We have compared our proposed methodology with Shafait et al. [10] and a commercial engine, Abbyy Cloud OCR SDK [4]. The evaluation measures described in [10] have been employed.

We have used open sourced UNLV dataset [3] as used in [10] to make a fair comparison of both methodologies. Due to this dataset, we didn't compare our methodology with systems that were proposed in ICDAR 2013 Table Detection competition. So, it wouldn't be a fair comparison to their proposed approach. Most of the techniques that were proposed in ICDAR 2013 [15], [27] are not data driven and are highly dependent on table layout and extraction of custom features from the images. This makes those techniques non robust to varying layout.

Considering Ground truth bounding box is represent by G_i while the bounding box detected by our system is represented by D_j. The formula for finding the overlapped region between two bounding boxes is given by [10].

$$A(G_i, D_j) = \frac{2 \times |G_i \cap D_j|}{|G_i| + |D_j|}, A \in [0, 1] \quad (1)$$

$A(G_i, D_j)$ represents the overlapped region between ground truth and detected bounding boxes. Depending on the area of intersecting region, its value will lie between zero and one. Note that we are using the same threshold values as described by Shafait et al. [10] to make a fair comparison with their technique.

Figure 3 shows some of the errors (partial detection, over segmentation, and false positive detection) that occurred during table detection. Here the blue region represents the ground truth bounding boxes while red region represents bounding boxes of detected regions.

A. Correct Detections

These are the number of ground truth tables that have a major overlap (A $\geq$ 0.9) with one of the detected tables. The area has been calculated using eq.1

B. Partial Detections

These are the number of ground truth tables that have a partial overlap (0.1 < A < 0.9) with one of the detected tables.

C. Over-Segmented Tables

These are the number of ground truth tables that have overlap (0.1 < A < 0.9) with more than one detected tables. It means that different parts of the ground truth table have been detected as separate tables.

D. Under-Segmented Tables

These are the number of ground truth tables that have major overlap (0.1 < A <0.9) with detected table but that detected table also overlaps with several other ground truth tables. It means that more than one tables were merged during detection and were reported as single table.

E. False Positive Tables

This indicates the number of detected tables that do not have an overlap (A $\leq$ 0.1) with any of the ground truth tables. Such tables are missed during detection.

F. Missed Tables

This indicates the number of ground truth tables that do not have an overlap (A $\leq$ 0.1) with any of the detected tables. It means that these tables are missed by the detecting algorithm.

G. Precision

Precision measure has been used for evaluating the overall performance of table detection method. It finds the percentage of detected tables that actually belong to table regions of ground truth document image. Formula for calculating precision is as follows:

$$\frac{\text{Area of Ground truth regions in Detected regions}}{\text{Area of all Detected table regions}} \tag{2}$$

H. Recall

It is evaluated by finding the percentage of ground truth table regions that were marked as detected table regions. Formula for calculating recall is as follows:

$$\frac{\text{Area of Ground truth regions in Detected regions}}{\text{Area of all Ground truth table regions}} \tag{3}$$

I. F1 Score

It considers both precision and recall to compute the accuracy of methodology.

$$\frac{2 \times \text{Precision} \times \textit{Recall}}{\text{Precision} \times \text{Recall}} \tag{4}$$

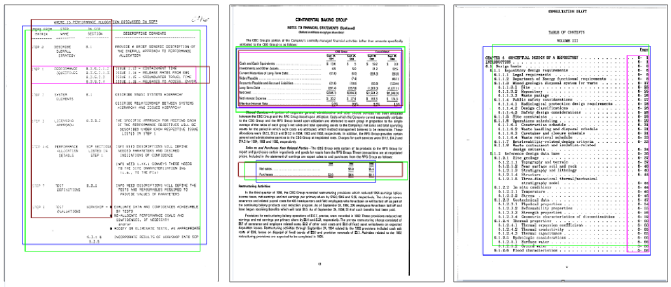


Fig. 5: Visualization of various engines. Ground truth bounding box is represented by blue color while the detected bounding box by our method is represented by green color. Magenta color represents bounding box of Abbyy Cloud OCR SDK while maroon color shows the result of Tesseract.

V. Experiments and Results

In order to evaluate the performance of the proposed methodology, we chose publicly available UNLV dataset [3]. This dataset consists of wide variety of document images ranging from business reports to research papers and magazines that includes varying and very complex table layouts. This dataset contains approximately 10,000 images at different resolutions. For each scanned image, manually keyed ground truth text is provided, along with manually determined zone information. Each zone is further categorized depending on the contents (text, half-tone, table, etc.) of that zone. Amongst 10,000 document images, only 427 contain table regions. We have used all of these 427 images from UNLV dataset for evaluating our proposed technique. As the dataset is small so we have used transfer learning approach [28]. We have used data augmentation approaches including rotation, scaling and flipping to overcome over-fitting.

Performance comparison between open source Shafait et al. [10] technique (Tesseract), a commercial engine (Abbyy Cloud OCR SDK) and our method is shown in Table I. While parsing table, row and column headers are often used as keys. So even if they are missed, it is impossible to extract any information; hence, the whole detected table becomes useless. Thus, number of correct detections is the most expressive performance measure. Tesseract and Abbyy fail to detect tables in presence of complex layouts that consists of wide white spaces. The results exhibit that our approach has better performance as correct detections significantly improve from 44% to 60.5%.

Figure 5 visualizes results of all three engines with respect to ground truth. Overall results of our proposed methodology have been shown in Figure 4.

VI. Conclusion

This paper presented an approach for table detection based on deep learning. The proposed system uses image transformation for separating text regions from non-text regions. It then uses RPN followed by fully connected neural network for detection of table regions in document images. Experimental

	Accuracy (%)			
Performance Measures	**Tesseract**	**Abbyy**	**Without Distance Transform**	**Our Approach**
Correct Detections	44.9	41.28	51.37	60.5
Partial Detections	28.4	32.1	42.2	30.2
Missed Tables	25.68	25.68	6.42	9.17
Over Segmented Tables	3.66	7.33	29.35	24.7
Under Segmented Table	3.66	7.33	42.20	30.27
False Positive Detections	22.72	7.21	5	10.17
Area Precision	93.2	95.0	84.5	82.3
Area Recall	64.29	64.3	89.17	90.67
F1 Score	76.09	76.69	86.77	86.29

TABLE I: Performance comparison of different engines

results show that deep learning based system is robust to layout analysis for table detection as it is not dependent on hand engineered features. The proposed system has been evaluated on publicly available UNLV dataset. It gives better results as compared to the Tesseract's state-of-the-art table detection system. We plan to extend this work in the direction of table structure and content extraction in future.

REFERENCES

[1] J. Hu, R. S. Kashi, D. Lopresti, and G. T. Wilfong, "Evaluating the performance of table processing algorithms," *International Journal on Document Analysis and Recognition*, vol. 4, no. 3, pp. 140–153, 2002.

[2] L. Hao, L. Gao, X. Yi, and Z. Tang, "A table detection method for pdf documents based on convolutional neural networks," in *Document Analysis Systems (DAS), 2016 12th IAPR Workshop on*. IEEE, 2016, pp. 287–292.

[3] A. Shahab, "Table ground truth for the UW3 and UNLV datasets," [Online; accessed 7-April-2017]. [Online]. Available: http://www.iapr-tc11.org/mediawiki/index.php?title=Table_Ground_Truth_for_the_UW3_and_UNLV_datasets

[4] Abbyy. (2017) OCR SDK engine. [Online]. Available: https://www.abbyy.com/en-eu/ocr-sdk/

[5] T. Kieninger and A. Dengel, "A paper-to-html table converting system," in *Proceedings of document analysis systems (DAS)*, vol. 98, 1998.

[6] ——, "Table recognition and labeling using intrinsic layout features," in *International Conference on Advances in Pattern Recognition*. Springer, 1999, pp. 307–316.

[7] ——, "Applying the t-recs table recognition system to the business letter domain," in *Document Analysis and Recognition, 2001. Proceedings. Sixth International Conference on*. IEEE, 2001, pp. 518–522.

[8] Y. Wang, I. Phillip, and R. Haralick, "Automatic table ground truth generation and a background-analysis-based table structure extraction method," in *Document Analysis and Recognition, 2001. Proceedings. Sixth International Conference on*. IEEE, 2001, pp. 528–532.

[9] J. Hu, R. S. Kashi, D. P. Lopresti, and G. Wilfong, "Medium-independent table detection," in *Electronic Imaging*. International Society for Optics and Photonics, 1999, pp. 291–302.

[10] F. Shafait and R. Smith, "Table detection in heterogeneous documents," in *Proceedings of the 9th IAPR International Workshop on Document Analysis Systems*. ACM, 2010, pp. 65–72.

[11] S. Tupaj, Z. Shi, C. H. Chang, and H. Alam, "Extracting tabular information from text files," *EECS Department, Tufts University, Medford, USA*, 1996.

[12] G. Harit and A. Bansal, "Table detection in document images using header and trailer patterns," in *Proceedings of the Eighth Indian Conference on Computer Vision, Graphics and Image Processing*. ACM, 2012, p. 62.

[13] B. Gatos, D. Danatsas, I. Pratikakis, and S. Perantonis, "Automatic table detection in document images," *Pattern recognition and data mining*, pp. 609–618, 2005.

[14] A. C. e Silva, "Learning rich Hidden Markov Models in document analysis: Table location," in *Document Analysis and Recognition, 2009. ICDAR'09. 10th International Conference on*. IEEE, 2009, pp. 843–847.

[15] T. Kasar, P. Barlas, S. Adam, C. Chatelain, and T. Paquet, "Learning to detect tables in scanned document images using line information," in *Document Analysis and Recognition (ICDAR), 2013 12th International Conference on*. IEEE, 2013, pp. 1185–1189.

[16] M. A. Jahan and R. G. Ragel, "Locating tables in scanned documents for reconstructing and republishing," in *Information and Automation for Sustainability (ICIAfS), 2014 7th International Conference on*. IEEE, 2014, pp. 1–6.

[17] T. T. Anh, N. In-Seop, and K. Soo-Hyung, "A hybrid method for table detection from document image," in *Pattern Recognition (ACPR), 2015 3rd IAPR Asian Conference on*. IEEE, 2015, pp. 131–135.

[18] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[19] H. Breu, J. Gil, D. Kirkpatrick, and M. Werman, "Linear time Euclidean distance transform algorithms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 5, pp. 529–533, 1995.

[20] R. Fabbri, L. D. F. Costa, J. C. Torelli, and O. M. Bruno, "2D Euclidean distance transform algorithms: A comparative survey," *ACM Computing Surveys (CSUR)*, vol. 40, no. 1, p. 2, 2008.

[21] I. Ragnemalm, "The Euclidean distance transform in arbitrary dimensions," *Pattern Recognition Letters*, vol. 14, no. 11, pp. 883–888, 1993.

[22] R. Girshick, "Fast R-CNN," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1440–1448.

[23] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *European conference on computer vision*. Springer, 2014, pp. 818–833.

[24] T. Kieninger and A. Dengel, "An approach towards benchmarking of table structure recognition results," in *Document Analysis and Recognition, 2005. Proceedings. Eighth International Conference on*. IEEE, 2005, pp. 1232–1236.

[25] S. Mandal, S. Chowdhury, A. K. Das, and B. Chanda, "A simple and effective table detection system from document images," *International Journal of Document Analysis and Recognition (IJDAR)*, vol. 8, no. 2-3, pp. 172–182, 2006.

[26] F. Shafait, D. Keysers, and T. Breuel, "Performance evaluation and benchmarking of six-page segmentation algorithms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 6, pp. 941–954, 2008.

[27] J. Fang, L. Gao, K. Bai, R. Qiu, X. Tao, and Z. Tang, "A table detection method for multipage pdf documents via visual seperators and tabular structures," in *Document Analysis and Recognition (ICDAR), 2011 International Conference on*. IEEE, 2011, pp. 779–783.

[28] I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal, *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2016.