2017 14th IAPR International Conference on Document Analysis and Recognition

# Multi-scale Multi-task FCN for Semantic Page Segmentation and Table Detection

Dafang He*, Scott Cohen‡, Brian Price‡, Daniel Kifer† and C. Lee Giles*
*School of Information Science and Technology, Penn State University
†Computer Science and Engineering, Penn State University
‡Adobe Research

***Abstract*—Page segmentation and table detection play an important role in understanding the structure of documents. We present a page segmentation algorithm that incorporates state-of-the-art deep learning methods for segmenting three types of document elements: text blocks, tables, and figures. We propose a multi-scale, multi-task fully convolutional neural network (FCN) for the tasks of semantic page segmentation and element contour detection. The semantic segmentation network accurately predicts the probability at each pixel of the three element classes. The contour detection network accurately predicts instance level "edges" around each element occurrence. We propose a conditional random field (CRF) that uses features output from the semantic segmentation and contour networks to improve upon the semantic segmentation network output. Given the semantic segmentation output, we also extract individual table instances from the page using some heuristic rules and a verification network to remove false positives. We show that although we only consider a page image as input, we produce comparable results with other methods that relies on PDF file information and heuristics and hand crafted features tailored to specific types of documents. Our approach learns the representative features for page segmentation from real and synthetic training data. The learning-based property makes it a more general method than existing methods in terms of document types and element appearances. For example, our method reliably detects sparsely lined tables which are hard for rule-based or heuristic methods.**

## I. Introduction

Page segmentation and table detection are important topics in document understanding. They could be used in many applications such as document indexing, accessibility, re-flow, and document classification.

The goal of page segmentation is to segment a document page into homogeneous parts, and we consider text regions, figure regions, and table regions in this paper.

In addition to page segmentation, we also consider the problem of obtaining an accurate location for each table in the document. For most previous works, page segmentation and table detection are typically considered separately using completely different algorithms. In this work, we present a unified framework of combining deeply-learned model and heuristic rules to tackle both tasks. The ability of this unified framework removes some assumptions made by previous researchers. For example, previous researchers in page segmentation typically assume that no table exists in the document, which is not the case in many scenarios. In this work, we first train a multi-scale, multi-task deep fully convolutional neural network (FCN) [1] for two tasks: (1) predict the class label for each pixel, and (2) predict the instance level edges for document elements, i.e. the boundaries of individual occurrences such as a single table or figure. The two tasks are naturally trained together with two branches.

After obtaining the two prediction maps for each document, we jointly train a conditional random field (CRF) on top of the semantic segmentation and contour detection to produce a refined semantic segmentation output. The unary term of the CRF is defined by the deep features from the semantic segmentation network. The pairwise term is defined by the color difference and deep contour features. The CRF can produce better semantic segmentation results than the raw network output.

After obtaining the pixel level segmentation prediction, we build a heuristic but effective approach on top of the prediction map to get each table instance with a verification network that can further removes false positive. Fig. 5 shows several segmentation results and Fig. 8 shows some identified table regions by our algorithm.

Our contributions are in the following aspects:

1. A hybrid network having two complementary branches is presented to solve the problem of both page segmentation and table detection. To our knowledge, this is the first work that segments page content and detects tables in arbitrary types of documents in a unified framework with a deeply-learned model.

2. Contour detection, which serves as a way of generating 'instance edges', presents a unique way of tackling the problem. This is the first work to predict such boundaries of document elements using a deep neural network. Combined with the semantic segmentation and contour detection, a conditional random field is trained to generate better segmentation results.

3. A novel way of synthesizing documents is presented and our model, trained in part on such synthetic documents, achieves good performance on real document for both page segmentation and table detection. Our network only takes input of a document image, so the framework could also handle many scanned pages as well.

## II. Related Work

### A. Page Segmentation

Early works on document physical layout analysis can be classified into two categories: (1) bottom up methods

2379-2140/17 $31.00 © 2017 IEEE
DOI 10.1109/ICDAR.2017.50

[2], [3] and (2) top-down methods [4]. Bottom up methods typically attempt to do connected component analysis on characters and group them into words or lines. On the other hand, top-down methods start from the document level, and recursively segment the document based on some predefined assumptions. A systematic survey can be found in [5]. All these methods used heuristic rules to segment documents, and hence do not generalize well to a wide range of document types and their element appearances. They typically focused on several specific types of documents. Recently, new methods have been proposed [6], [7], [8], [9], [10] that start to use machine learning based methods with texture features for page segmentation. These methods had different focus, but they typically couldn't handle documents with table.

### B. Table Detection

Tables are important document elements, and previous works typically regard their detection as separate problems. Line information is the most useful feature that helps in identifying a table region. Chen et al. [11] studied the problem of table detection in noisy handwritten images with a correlation-based approach. Kasar et al. [12] proposed to identify table regions from a set of low-level features extracted from horizontal or vertical lines. Most of these methods are still using hand-crafted features with predefined rules for extracting table regions. A recent work using a convolutional neural network (CNN) [13] has achieved state-of-the-art performance in table detection. However, the ability of the CNN is not fully explored since they used a lot heuristic rules on finding possible table regions. The CNN is only applied as a classifier to determine whether a given image patch is table or not. In contrast, our work uses a CNN on a full-page level and table regions are obtained directly from the CNN output. Hao[13] used pdf information while we only needs images. Xiao [14] used fcn for semantic page segmentation, while our work focus on segmentation and table detection. This allows our system to continuously adapt with training data rather than requiring humans to develop more and more heuristic rules.

### C. Semantic Segmentation

The performance of semantic segmentation in natural images increased significantly after the introduction of fully convolutional neural networks [1]. The ability of CNNs to extract high level feature representations makes the model effective for pixel-wise prediction. Several further works have also addressed the problem of multi-scale feature extraction [15], [16], [17]. This is the key to the success of FCN model. In our work, we used a multi-scale FCN model for page segmentation for the purpose of capturing both large context information as well as detailed features. This is crucial for predicting table and figure regions accurately.

### D. Contour Detection

Contour detection was originally proposed to extract semantic contours in natural images. Several recent works [18], [19] using deep convolutional neural networks have achieved great success in this task. Unlike low-level edge detection, the deeply learned contour preserves the semantic meaning of an image as specified in the training data. In this work, we bring the idea of contour detection into document processing and show that a network can be trained to learn such contours in documents as well. Besides, document contours have unique features that distinguish themselves from those in natural images.

## III. Training Data Collection

Document analysis has always been an active research field. Most of the works published in this area rely heavily on heuristic rules [11], [2], [3], [4]. Such approaches do not require large training datasets, leading to a lack of sufficient training data for deep learning models.

Fully-annotated document images are needed for training FCNs. To do that, we designed a novel method of generating partially synthetic document pages. We first manually labeled over 2000 documents pages. We replaced elements in these labeled pages randomly with other elements. We follow the simple rule that any type of elements could be replaced with either a table or figure. Note that we do not replace elements with text blocks because, typically, text styles and fonts within a document are similar or follow similar rules. Replacing elements with random text block might make the document unrealistic. We synthesize 20000 more full page training annotations with this method. Our experiments show that the segmentation model trained on these real and synthetic data produces good segmentations on real pages. In Fig. 1 we show examples of real annotated pages and synthetic pages by element replacement.

To create our synthetic document pages, we needed labeled instances of tables and figures to use to replace real page elements. We gathered these new elements as part of a separate data collection effort. For figures, we gathered both natural images and graphic figures. We used ImageNet and MSCOCO as sources of natural images. We used graphics figures such as those published in [20].

Table sources are harder to find. We considered two ways of gathering table images from pre-existing content. First, we gathered 2000 tables from real documents by first generating candidate table regions from pdfs with existing tools in Adobe Acrobat and then asking Mechanical Turkers to label the tables as correct or not. Second, we gathered a large number of different kinds of public table images from the internet to cover more table styles. Additionally, we synthesized many table elements from a large corpus of text. We randomized the characteristics of the table such as the text and the number of rows and columns. We wrote code to synthesize latex files and render the table images.

## IV. Multi-scale, Multi-task FCN

In this section, we describe the design of our architecture and rationale behind it. The full architecture is in Fig. 2.

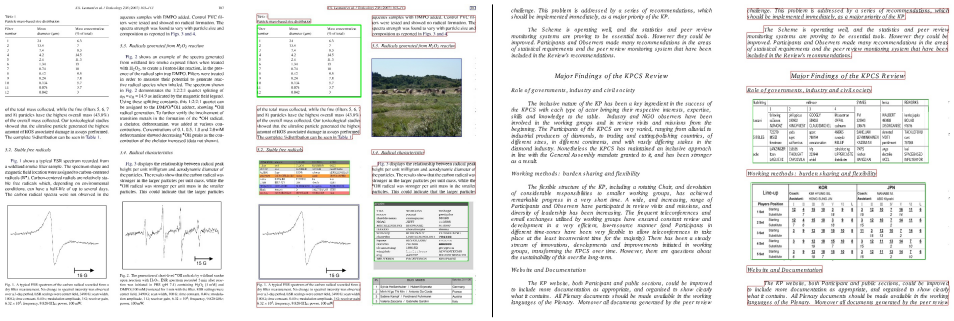

Fig. 1: Examples of two pairs of partially synthetic document images. For each pair of image, the left image is the original labeled document. The right image is synthesized by replacing elements from the left. We show the label in the synthetic document. Red: text block. Blue: figure. Green: Table. Better when zoomed in.

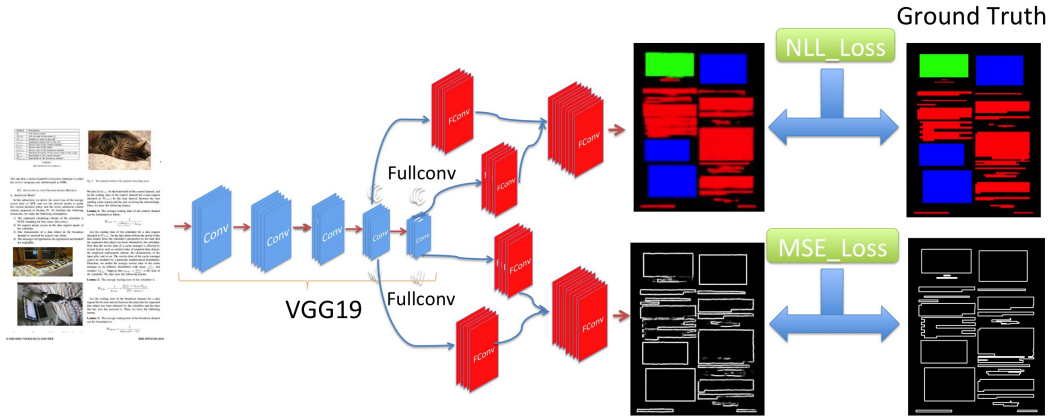


Fig. 2: Our architecture of multi-scale, multi-task fully-convolutional neural network for page segmentation. This CNN is used to extract high-level feature representations. Two branches of the FCNs are used to produce the semantic segmentation and contour detection. We do not show the output for each scale in this image. However, there are actually two outputs associated with each scale for each task that is trained in the same criterion. We only show the joint prediction in the image. For the semantic segmentation, blue represents figure, green represents table and red represents text block.

### *A. Multi-scale*

Elements in documents are highly context dependent. Fig. 3 shows several examples demonstrating why we need a large context for accurate semantic segmentation. A text region could actually be part of a table, since text is an important component in a table. For example, in the left image of Fig. 3, the information within the red bounding box indicates that it is a text block. But it is actually part of a table. The right image shows examples of parts of graphics figure that are hard to be distinguished from a small part of table without larger context.

Based on the above observations, we propose to use a multi-scale model which can provide a larger context as well as details for prediction at each pixel. We initialized our CNN model with VGG19 [21]. Then unpooling[22] and full-convolution are added to upscale the feature map. For each task of semantic segmentation and contour detection,

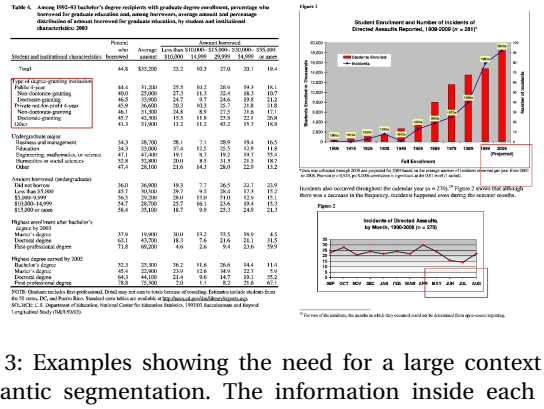

Fig. 3: Examples showing the need for a large context for semantic segmentation. The information inside each red box is not enough for accurate region identification.

we have features from two scales and there is an output associated with each scale too. This will be discussed in the next section. The final output for each task is trained with features jointly from two scales by concatenation and further convolution.

### B. Optimization for Semantic Segmentation

A semantic segmentation network is trained to predict the label for each pixel. However, training a multi-scale prediction is generally harder than with only one scale. Here we use a per-scale strong supervision on each scale to enforce training of each scale. This training scheme is used in several previous works [15], [23] and is claimed to generate more discriminative features for each scale. Equation 1 shows the loss function.

$$S_L(\theta_s,\theta_{\text{sh}},D) = -\sum_k \log P(Y=y^k|x^k,\theta_s,\theta_{\text{sh}}) + \sum_{i=1}^{N} -\alpha_i \times (\sum_j \log P(Y=y_i^j|x_i^j,\theta_s,\theta_{\text{sh}})) \quad (1)$$

In the equation, $N$ represents the total number of scales that we use. $S_L$ represents semantic loss. $\theta_s$ represents the parameters for the decoder part of semantic network. $\theta_{\text{sh}}$ represents the shared convolutional part of the network. In our experiments, $N=2$. $x^j$ and $y^j$ represent the ground truth and prediction, respectively. $\alpha_i$ controls the weight for scale $i$. In training, we start with larger $\alpha_i$ for each separate scale and then decrease $\alpha_i$ to focus on the final output in the first term.

### C. Optimization for Contour Detection

Before describing the optimization for contour detection, we first define the contours we use in the document domain. Object contours in natural images are typically a subset of color edges that preserve the semantic meaning. Instead, here we define the contour as a set of pixels that surround each instance of a document element (text block, table, or figure). The contour separates the element from other elements or background. For example, a contour for a text block is a polygon around its boundary.

In order to train the network to predict such element contours, we use mean squared error as our loss function. However, simply labeling all pixels on the contour as 1 and others as 0 is not effective for two reasons: (1) The ground truth labeling of contour might have a small shift from where it is supposed to be. This is due to slight imperfections in our training data. (2) Predicting a contour a few pixels away from the ground truth is acceptable for our task. Based on the two observations, Equation 2 shows the definition of the ground truth, and Fig. 6 shows several ground truth illustration document images.

$$x_i = \begin{cases} 1 & \text{if } i \in S_{\text{contour}} \\ 0.9 & \text{if dist}(i,j) == 1 \text{ and } \exists j \in S_{\text{contour}} \\ 0.6 & \text{if dist}(i,j) <= 3 \text{ and } \exists j \in S_{\text{contour}} \end{cases} \quad (2)$$

$S_{\text{contour}}$ represents the pixels in the contour of each instance. The training of the contour network follows the same pattern as in semantic segmentation - jointly optimize the network with two different scales. Equation 3 shows the criterion we use to optimize. $C_L$ represents contour loss.

$$C_L(\theta_c,\theta_{\text{sh}},D) = -\sum_k (y^k - x^k)^2 - \sum_{i=1}^{N} \alpha_i \times (\sum_j (y_i^j - x_i^j)^2) \quad (3)$$

$\theta_c$ represents the parameters for the decoder part of contour network. Note the convolutional part of it is shared with the semantic network. N represents the number of scales. $\alpha_i$ represents the weight for scale $i$. $k$ and $j$ represent output pixel indices. Fig 4 shows several example output images for the contour detection network.

### D. Joint Training

By combining the semantic loss and contour loss, the total loss function $J_L$ could be represented as Eq. 4, where $\beta$ is the training weight for contour loss, it is set to 15 so that the losses from two branches are comparable. Several training examples and their ground truth annotation of both semantic segmentation and contour detection are in Fig. 6.

$$J_L(\theta_{\text{sh}},\theta_c,\theta_s,D) = S_L + \beta \times C_L \quad (4)$$

## V. CRF for Joint Prediction

After obtaining the semantic segmentation image and contour detection image, we propose to use a CRF to optimize the final result of the semantic segmentation.

An FCN-based semantic segmentation suffers from the fact that each prediction(pixel) in the output layer is independent, and thus the output is not smooth. A traditional way to overcome this in natural image is by applying a CRF on top of the deep features generated by the network as the unary term and using color differences in the pairwise term. In documents, however, there may not be a color difference at some points along an element boundary – e..g. a table boundary without an explicit line around the table. Instead, we use both the color as a low level feature edge term and the contour as a deep edge term in our CRF formula. The energy $E$ of a document image $x$ with label $y$ is in equation 5. $\theta_{\text{crf}}$ represents the parameters of the CRF. $\phi^{(1)}$ is the function that produces the unary potential. In our setting, we used the 32-dimensional feature vector generated from the semantic network for each pixel location. $\phi^{(2)}$ is the function that produces the pair-wise potential for each edge. It is modeled by the color difference and the deep contour features. $P$ is the set of pixels in the output layer. $N$ is the set of pixel neighbors. We only consider 4 directions for edge connection for each pixel in output layer.

$$E(y,x;\theta_{\text{crf}}) = \sum_{p\in P} \phi^{(1)}(y^p,x;\theta_{\text{crf}}) + \sum_{(p,q)\in N} \phi^{(2)}(y^p,y^q,x;\theta_{\text{crf}}) \quad (5)$$

Fig. 4: Contour detection results. For each pair of images, left is the input image and right is the output contour map.

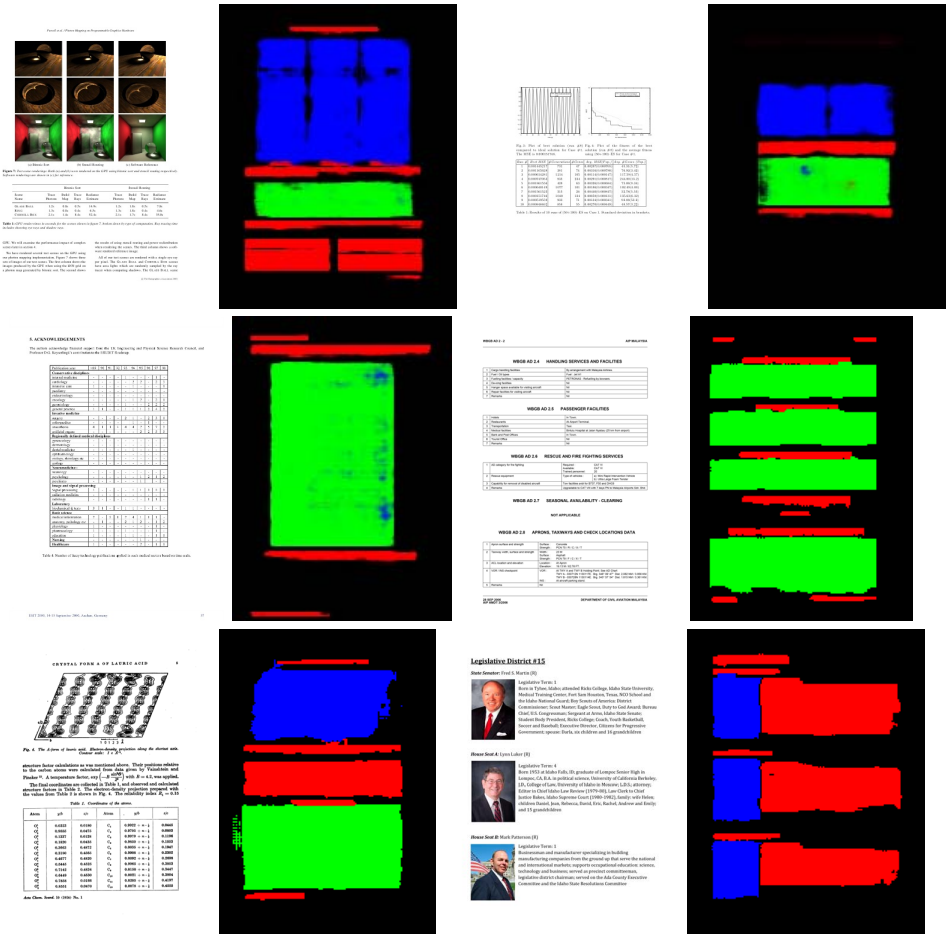

Fig. 5: Semantic segmentation results of our multi-scale, multi-task network with CRF training. Red: text block. Blue: figure. Green: table.

The segmentation then turns into minimizing the energy w.r.t the parameters $\theta_{\text{crf}}$ as in Equation 6.

$$\theta^* = \arg\min_{\theta} E(y, x; \theta_{\text{crf}}) \tag{6}$$

The CRF is learned with structured support vector machines [24]. Fig. 5 shows some example results of our segmentation model.

## VI. Table Detection

Given the multi-scale semantic segmentation output, we obtain the location of each table. We only consider table detection for the reason that table detection is widely

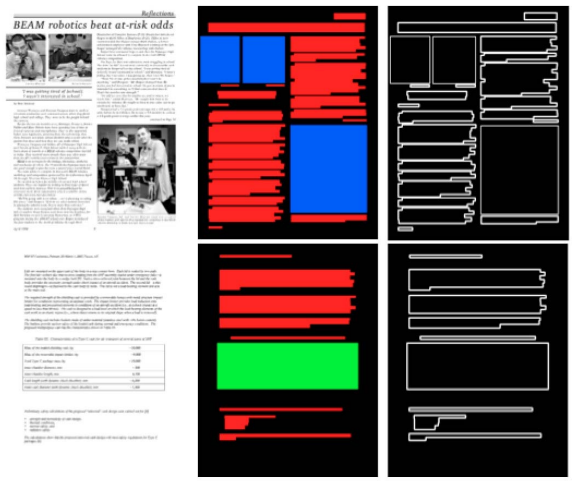


Fig. 6: Two examples illustrate the training input images and corresponding ground truth of our multi-task network. From left to right: input document image, semantic segmentation ground truth, contour detection ground truth. Red, blue, green represent text, figure, and table, respectively.

studied in previous works and most existing algorithms rely on heuristic rules and only focus on specific types of documents such as research articles [25]. By using a deep learning model, we will be able to tackle various kinds of documents without those constraints and learn appropriate features for the task. The pipeline of table detection is in Fig. 7. We start from a probability map generated from the multi-task, multi-scale deep FCN and CRF. We threshold the probability mask for table at 0.5 after applying the CRF and then apply connected components analysis(CCA) on it to get possible locations of table. In order to remove false positives, a deep CNN is then used to classify whether each region is a table or not. We call this network the *verification network*. It is initialized with pretrained VGG16 network [21]. In the training process, we randomly sample negative regions and positive regions from a set of documents. Let $T$ denote the set of ground truth tables. We define a region $x$ as positive if $\exists t \in T$ s.t $IOU(t,r) > 0.8$, and as negative if $\forall t \in T$ s.t $Inter(t,r)/area_r < 0.3$. There is no training data for those cases in between, since this module is mostly used for removing false positive region.

The $IOU$ function calculates the standard intersection over union between two regions. The $Inter$ function calculates the intersection area between two regions. $area_r$ is the area of region $r$. This formulation is similar to the standard R-CNN [26] framework in object detection. The difference is that in the training of the classifier, we randomly sample positive and negative regions rather than following specific region proposals.

Some detection results are in Fig. 8. Note that with the increasing of IOU requirement, the performance of

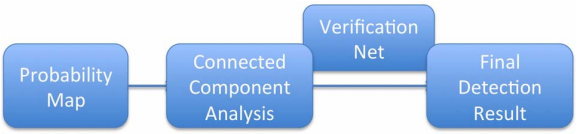


Fig. 7: The pipeline of our table detection model. See the main text for a description.

| method/Element | Text | Table | Figure |
|---|---|---|---|
| FCN | 0.66 | 0.53 | 0.77 |
| MFCN | 0.73 | 0.56 | 0.76 |
| MFCN+CRF | 0.79 | 0.68 | 0.77 |
| MFCN+Contour+CRF | **0.81** | **0.69** | **0.79** |

TABLE I: The pixel IOU performance of page segmentation on the collected dataset. The best result for each element type is shown in bold.

table detection drops significantly as in Fig. 9. This is a common problem in object detection especially for deeply learned models [27]. It could also be seen that even for those successfully detected tables, the bounding box is not perfectly aligned. However, since we already managed to obtain a region of table, simple heuristics could be added to refine the bounding box. For example: we can find text lines or do connected component analysis within the detected region of table, and then refine the table location based on some simple rules. This is much simpler than finding a table in the whole document page since here we can assume that the given region is a table. However, we do not implement such module in this paper, since one of our goals is to see how well a deeply learned vision detection algorithm can do in traditional document processing.

## VII. Experiments

In our experiments, all the document images are resized to a size with the largest dimension equals to 792 without changing the aspect ratio. Then we run our algorithm to obtain the final semantic segmentation output and table regions. We test our algorithm on one public dataset and one collected dataset with various different types of documents. The method MFCN+CRF means multi-scale FCN with CRF trained on color edge term. MFCN+Contour+CRF represents multi-scale FCN with CRF trained on the color and contour edge terms.

### A. Various Doc Dataset

We collected a dataset with 130 pages from various document types, and call it the *Various Doc Dataset*, and it will be made public for research purpose. We evaluate our algorithm on this dataset for page segmentation as in Table I with different models. We can see that by jointly training the semantic features and contour features, a CRF can improve the performance of pixel-level IOU.

### B. Marmot Dataset

The Marmot dataset was originally published by [28]. The dataset contains 2000 Chinese and English document

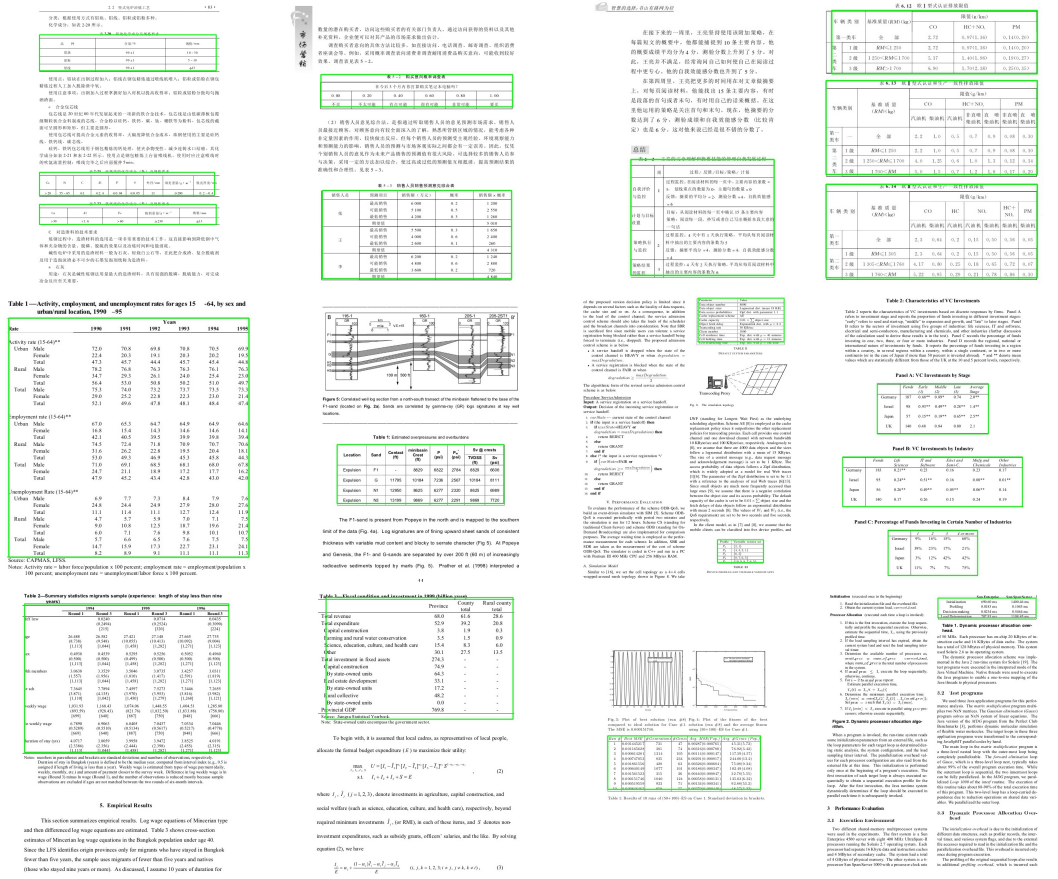

Fig. 8: The table detection results of our algorithm. With training data provided, our system can also remove some graphics data that is relatively similar to table. Better when zoomed in. Note that we are also able to detect tables that have sparse lines as in the result in the first row, fifth column.

images, and it is by far the largest table detection dataset that we know. We used the final CRF output which achieves 5% higher IOU score than MFCN on table segmentation in this dataset. The detection performance of our system w.r.t IOU threshold is shown in Fig. 9.

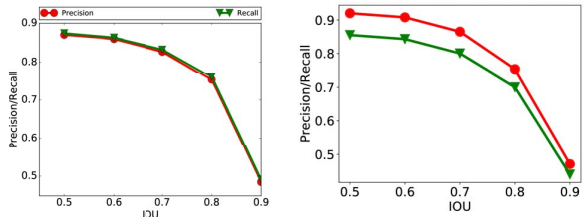


Fig. 9: The precision (red) and recall (green) curves of table detection performance of our algorithm – Left: Chinese Documents. Right: English Documents. Although our training set only has English pages, our model also detects tables on Chinese pages well.

A comparison versus other table detection methods is shown in Table II. All other methods used pdf information rather than image alone.

TABLE II: Table Detection Comparison on Marmot. We gave two results here as IOU requirement to be 0.8 and 0.9 to be considered as a correct detection. This is a really strict requirement in vision-based object detection. Combined with heuristic post-processing within each detected table block, the performance could be significantly improved as could be seen when we lower the IOU threshold.

| Chinese Documents | | | English Documents | | |
|---|---|---|---|---|---|
| method | precision | recall | method | precision | recall |
| Liu [25] | 0.44 | 0.28 | Liu [25] | 0.48 | 0.35 |
| Yildiz [29] | 0.35 | 0.34 | Yildiz [29] | 0.35 | 0.38 |
| Fang [30] | 0.89 | 0.80 | Fang [30] | 0.58 | 0.49 |
| Our(0.9 IOU) | 0.493 | 0.491 | Our(0.9 IOU) | 0.47 | 0.45 |
| Our(0.8 IOU) | 0.77 | 0.761 | Our(0.8 IOU) | 0.753 | 0.70 |

### C. Failure Cases

Fig. 10 shows three examples of failure detection cases of our system. Green boxes represent correct detections. Red

boxes represent false positives and yellow boxes represent tables that we fail to detect.

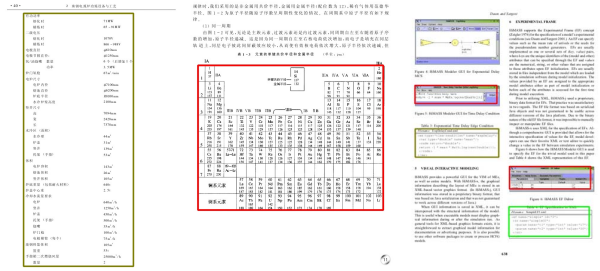

Fig. 10: Three representative failure examples. From left to right: (1) missed table without any lines that covers the whole page. (2) false positive because the periodic table of elements is considered as a table in our definition but not in the Marmot dataset. (3) some graphic elements with table-like structures are incorrectly classified.

## VIII. Conclusion

In this paper, we propose a deep learning based document segmentation and table detection algorithm. It is a purely vision-based algorithm that learns the feature representation for the page segmentation task, and has better generalizability.

## IX. Acknowledgement

Some parts of this work are done while Dafang is in an internship in Adobe Research. The work is supported by NSF grant CCF 1317560 and Adobe System Inc.

## References

[1] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3431–3440.

[2] L. O'Gorman, "The document spectrum for page layout analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 11, pp. 1162–1173, 1993.

[3] K. Kise, A. Sato, and M. Iwata, "Segmentation of page images using the area voronoi diagram," *Computer Vision and Image Understanding*, vol. 70, no. 3, pp. 370–382, 1998.

[4] G. Nagy, S. Seth, and M. Viswanathan, "A prototype document image analysis system for technical journals," *Computer*, vol. 25, no. 7, pp. 10–22, 1992.

[5] S. Mao, A. Rosenfeld, and T. Kanungo, "Document structure analysis algorithms: a literature survey," in *Electronic Imaging 2003*. International Society for Optics and Photonics, 2003, pp. 197–207.

[6] A. VilâĂŹkin, I. Safonov, and M. Egorova, "Algorithm for segmentation of documents based on texture features," *Pattern recognition and image analysis*, vol. 23, no. 1, pp. 153–159, 2013.

[7] O. K. Oyedotun and A. Khashman, "Document segmentation using textural features summarization and feedforward neural network," *Applied Intelligence*, vol. 45, no. 1, pp. 198–212, 2016.

[8] M. Mehri, "Historical document image analysis: a structural approach based on texture," Ph.D. dissertation, Université de La Rochelle, 2015.

[9] P. Barlas, S. Adam, C. Chatelain, and T. Paquet, "A typed and handwritten text block segmentation system for heterogeneous and complex documents," in *Document Analysis Systems (DAS), 2014 11th IAPR International Workshop on*. IEEE, 2014, pp. 46–50.

[10] K. Chen, H. Wei, J. Hennebert, R. Ingold, and M. Liwicki, "Page segmentation for historical handwritten document images using color and texture features," in *Frontiers in Handwriting Recognition (ICFHR), 2014 14th International Conference on*. IEEE, 2014, pp. 488–493.

[11] J. Chen and D. Lopresti, "Table detection in noisy off-line handwritten documents," in *Document Analysis and Recognition (ICDAR), 2011 International Conference on*. IEEE, 2011, pp. 399–403.

[12] 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.

[13] 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.

[14] X. Yang, E. Yumer, P. Asente, M. Kraley, D. Kifer, and C. Lee Giles, "Learning to extract semantic structure from documents using multimodal fully convolutional neural networks," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[15] L.-C. Chen, Y. Yang, J. Wang, W. Xu, and A. L. Yuille, "Attention to scale: Scale-aware semantic image segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3640–3649.

[16] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," *arXiv preprint arXiv:1511.07122*, 2015.

[17] X. Bian, S. N. Lim, and N. Zhou, "Multiscale fully convolutional network with application to industrial inspection," in *Applications of Computer Vision (WACV), 2016 IEEE Winter Conference on*. IEEE, 2016, pp. 1–8.

[18] J. Yang, B. Price, S. Cohen, H. Lee, and M.-H. Yang, "Object contour detection with a fully convolutional encoder-decoder network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 193–202.

[19] Y. Liu, M.-M. Cheng, X. Hu, K. Wang, and X. Bai, "Richer convolutional features for edge detection," *arXiv preprint arXiv:1612.02103*, 2016.

[20] R. Al-Zaidy, S. Choudhury, and C. Giles, "Automatic summary generation for scientific data charts," 2016. [Online]. Available: http://www.aaai.org/ocs/index.php/WS/AAAIW16/paper/view/12661

[21] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, "Return of the devil in the details: Delving deep into convolutional nets," *arXiv preprint arXiv:1405.3531*, 2014.

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

[23] D. He, X. Yang, C. Liang, Z. Zhou, A. G. Ororbi, II, D. Kifer, and C. Lee Giles, "Multi-scale fcn with cascaded instance aware segmentation for arbitrary oriented word spotting in the wild," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[24] I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altun, "Support vector machine learning for interdependent and structured output spaces," in *Proceedings of the twenty-first international conference on Machine learning*. ACM, 2004, p. 104.

[25] Y. Liu, K. Bai, P. Mitra, and C. L. Giles, "Tableseer: automatic table metadata extraction and searching in digital libraries," in *Proceedings of the 7th ACM/IEEE-CS joint conference on Digital libraries*. ACM, 2007, pp. 91–100.

[26] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.

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

[28] J. Fang, X. Tao, Z. Tang, R. Qiu, and Y. Liu, "Dataset, ground-truth and performance metrics for table detection evaluation," in *Document Analysis Systems (DAS), 2012 10th IAPR International Workshop on*. IEEE, 2012, pp. 445–449.

[29] B. Yildiz, K. Kaiser, and S. Miksch, "pdf2table: A method to extract table information from pdf files," in *IICAI*, 2005, pp. 1773–1785.

[30] 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.