Table Detection in Document Images using Foreground and Background Features

1st Saman Arif
Department of Computing
NUST-SEECS
Islamabad, Pakistan
sarif.mscs16seecs@seecs.edu.pk

2nd Faisal Shafait
Department of Computing
NUST-SEECS
Islamabad, Pakistan
faisal.shafait@seecs.edu.pk

***Abstract*—Table detection is an important step in many document analysis systems. It is a difficult problem due to variety of table layouts, encoding techniques and the similarity of tabular regions with non-tabular document elements. Earlier approaches of table detection are based on heuristic rules or require additional PDF metadata. Recently proposed methods based on machine learning have shown good results. This paper, based on foreground and background features, describes performance improvement to these table detection techniques. Proposed solution is based on the observation that tables tend to contain more numeric data and hence it applies color coding/coloration as a signal for telling apart numeric and textual data. Deep learning based Faster R-CNN is used for detection of tabular regions from document images. To gauge the performance of our proposed solution, publically available UNLV dataset is used. Performance measures indicate improvement when compared with best in-class strategies.**

***Index Terms*—table detection, heuristic rules, document image analysis, metadata**

I. Introduction

Tables are a very common way for presenting and organizing important information in structured documents. Tables are present in vast class of documents like books, scientific papers, journals and even in more complex documents like newspapers, magazines, technical and commercial notes, internal reports, financial statements, business reports and invoices. Table detection aims at localizing/spotting the tables in the document images by marking their boundaries. Table structure recognition and understanding has been an active research area because of its significance in document analysis. Table detection is also important for being very first step in table recognition and analysis.

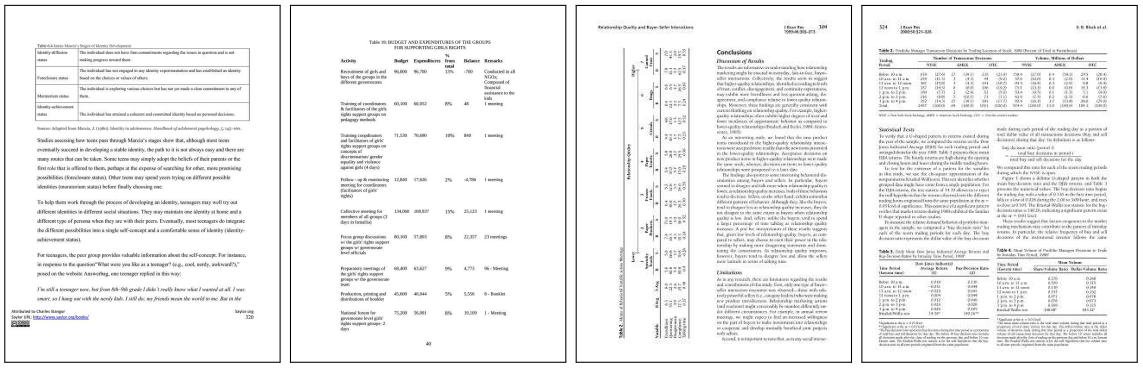
Table detection is a hard problem due to huge variation in table structure and encoding techniques. It is hard to formally define a table because of diverse table contents, erratic use of ruling lines and varying layouts. Moreover, significant similarity of tables with other elements of documents i.e. bar charts, flow charts or graphics etc makes table detection even more challenging. Keeping in view all these factors, a generalized solution is difficult to design.

Regardless of several years of research, existing methods could not provide a generalized method for table detection. Many existing solutions are based on hand-engineered features. They are designed for a specific class of documents. In order to accomplish the proposed task they use heuristic rules and work well under these rules/heuristics but fail where these conditions do not meet. Therefore, more efficient, robust and genearlized solutions for table detection are needed.

Spotting objects in natural images comes under the object detection research domain where Convolutional Neural Networks (CNNs) are playing the leading role. The problem of detecting tabular regions in document images is much similar to the object detection. Therefore, using state-of-the-art object detection approaches for solving table spotting problem can outperform conventional table detection approaches. The grand success of deep models for various detection and recognition tasks is highly dependent on the availability of huge training datasets. However, the scarcity of labeled training data is a major problem in the domain of document analysis. This problem can be addressed by utilizing the approach of domain adaptation and transfer learning.

In proposed technique, document image is first preprocessed. Preprocessing consists of two phases; coloration and transformation. Coloration or color coding relies upon the observation that numeric data contributes more towards a table and it helps to distinguish numeric and textual data. In second phase image transformation is applied to the document image for separating text regions from nontext regions. Following preprocessing, preprocessed image is passed on to the detection module. Proposed system has employed Faster Region-based Convolutional Neural Network (Faster R-CNN) [1] as detection module. The significant contribution of our proposed system lies in the fact that we account both foreground and background features for table detection. In addition to this, it is an efficient, robust and completely data driven approach. It is invariant to changes in layout and structure of the table because it can be fine-tuned to work on any dataset. UNLV dataset [2] is used for evaluation of our approach and it gave better results as compared to previous best in-class techniques.

The remaining part of the paper is structured as: in the next section a review of the literature on table detection/classification is presented. Section III provides details about training data collection. Section IV explains our proposed methodology. Section V and VI describes performance measures and experimental results respectively. Section VII concludes the paper and provides future directions.



(a) Table with ruling lines (b) Table without ruling lines (c) Vertical Table (d) Inside column and span column table

Fig. 1. Few images from training dataset.

II. Related Work

The problem of table detection has been addressed by many researchers. Table detection techniques reported in literature can be divided into two classes; traditional ruled-based approaches and recently proposed machine learning based techniques. First we will discuss few traditional approaches and then we will focus on machine learning based solutions.

Kieninger et al. [3]–[5] is among the pioneers who laid the groundwork for table spotting and structure recognition. They proposed a system for table detection and structure recognition called T-Recs. Bottom-up approach is utilized to form clusters of word bounding boxes by building segmentation graph. The main limitation of this approach is its dependency on word bounding boxes and also it fails in the presence of multicolumn layout.

Hu et al. [6] developed a table detection system based on dynamic programming. In this approach, to find out which input line(s) can be taken as part of table, certain characteristics are measured like merit, line correlation and scores etc. The group of lines that maximizes the characteristics to a threshold is considered as part of table. Single column document is required as input by this approach and fails in the presence of multi-column documents.

Wang et al. [7] proposed a system for table identification and decomposition based on statistical learning. For the identification of table linesthis approach utilizes the space among the consecutive words. It determines the table entity candidates by grouping horizontally adjacent words and vertically adjacent lines with large gaps. As last step, it applies statistical table refinement algorithm to reduce false alarms and refine table candidates. This approach is designed for specific set of document layouts and applicable to those layouts only.

Gatos et al. [8] presented approach, after preprocessing, makes some estimation about presence of vertical and horizontal lines and improves these estimations by removing text area. In last step, it finds out the line intersections and group these line intersections horizontally and vertically to reconstruct the table. The major shortcoming of this technique lies in its applicability to tables with ruling lines only.

Table detection approach presented by Mandel et al. [9] is based on assumption that table columns usually have large inter-word gap as compared to rest of the text lines. As a preprocessing step, all lines are removed this results into inaccurate detections for partially filled tables.

An early data driven approach developed by A. C. e Silva [10] back in 2009 focuses on table detection using hidden markov models (HMM). In first step text from the PDF files is extracted by the system and then uses it for computing feature vector. The key shortcoming of this technique lies in its applicability to noise free PDF files only.

The first method using deep learning based technique for table detection was presented by Hao et al. [11] back in 2016. Along with learned features, some loose heuristic rules and meta-information from PDF documents is also used for table detection purpose. The major limitation is that it fails to spot span-column tables i.e. tables that are spanned across multiple columns and also it only works for PDF documents.

Recently in 2017, Rashid et al. [12] presented an approach for table spotting and recognition based on machine learning. This technique employs bottom-up approach and utilizing geometric position of a word and its relation with the neighboring words, it derive a set of hand-crafted features like white-space distance and font variation etc. Auto MLP is trained on this feature set to classify words as table/non-table category and classification results are improved using post-processing. The limitation of this technique lies in its dependency on hand-crafted features and also it is unable to make column boundaries during table recognition.

A deep learning based approach is presented by He et al. [13] for semantic page segmentation and table detection. On the top of semantic segmentation and contour detection network, CRF is used to improve results of semantic segmentation. Individual table instances are detected using semantic segmentation output along with some heuristic rules. However, this approach fails to detect the tables without any ruling lines covering the whole page.

Another recent data-driven technique presented by Schreiber et al. [14] focuses on table detection and structure analysis

as well. This technique first detects the table regions from document images using Faster R-CNN and then in next step performs structure recognition. But this technique confuses the tables with other graphical elements that look similar to the tables.

Gilani et al. [15] also used Faster R-CNN for table detection. In first step they perform some distance transformations to the raw images and in next step apply Faster R-CNN for table detection. This technique out performs previous table detection techniques but it does not consider fore-ground features of the table.

In this paper, we propose a solution for table detection able to overcome the shortcomings of the previous approaches. Proposed solution for table spotting is based on foreground and background features. And it then adopts deep learning based Faster R-CNN to detect tabular regions from document images.

III. Training Data Collection

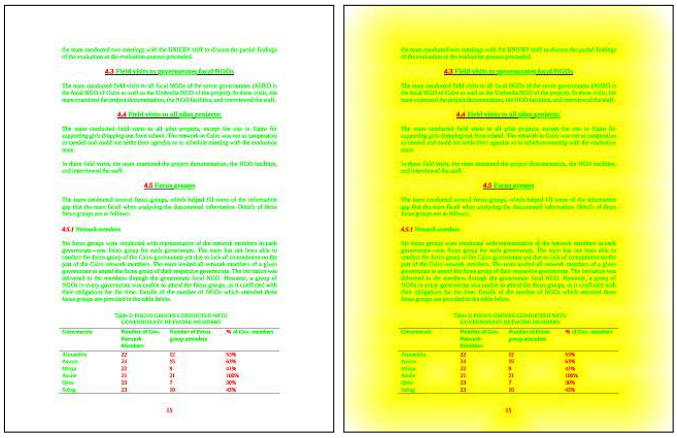
Document image analysis has always been an active research area. Most of the work done in this field depends on metadata and heuristic rules which does not require large training sets. This results into scarcity of sufficient labeled data for training of deep learning based models.

In order to meet the requirement of training data for proposed deep learning method, a huge number of document images annotated for table regions is required. For stated purpose, a new dataset is generated by collecting a large number of documents containing table regions. These documents cover a wide variety of sources like newspapers, magazines, research articles, business letters, annual reports and books etc. The dataset shows a great variety in table styles from inside-column tables to span-column tables, from horizontal tables to vertical tables and from horizontally and vertically ruled tables to tables having no ruling lines. Dataset comprises of tables with varying sizes from large tables covering the whole document page to small tables occupying a minor portion of the document image. Nonruled tables, small tables with fewer than five rows, sparse tables and tables with complex header structures are very common in this dataset.

We used labelImg tool [16] to label our training data for table regions. It generates annotations in PASCAL VOC format. One important point should be kept in mind while annotating the table regions that table caption and table footnote is not the tabular structure [17]. So while preparing the ground-truth these regions should be excluded. Our training set consists of 1274 document images having 1795 tables. The proposed dataset is used for training of the model, however the generalization and performance of our model is evaluated on publically available UNLV dataset. Few sample images from our training data are shown in Fig. 1.

IV. Methodology

The proposed approach consists of two important steps: Preprocessing and table detection. Detail about each module is given in following subsections.



(a) Color Coded Image (b) Color Coded and Transformed Image

Fig. 2. Preprocessed Images

A. Preprocessing

Preprocessing is the initial and the most crucial step of proposed approach and comprises of the two steps; coloration and transformation.

1) Coloration: Tables, an important part of documents, usually contain more numeric data as compared to textual data like mathematical and statistical tables and tables in research articles etc. On the basis of this observation, coloration aims at exploiting fore-ground features of the document images. For this purpose all the numeric information in the table is color coded as red and all the textual information is color coded as green. This color coding helps in differentiating numeric data from the textual data which can provide a fair estimate about presence of table regions.

2) Transformation: Following coloration is the image transformation phase, in which distance transformation is applied to the color coded images to capture the background features. Image transformation segregates text regions from nontext regions and gives detection module additional clues about presence of the table regions. Contrary to [15], which applies distance trasformation on all three red, blue and green channels, we apply distance transform to blue channel only. Euclidean distance transform is employed in our proposed system. We performed experiments with transformation on red and green channels as well but it turns out that transformation on blue channel is more robust.

As a result of coloration and transformation, red and green channel contain foreground features while blue channel contains background features. Result of preprocessing on document image is shown in Fig. 2.

B. Table Detection

The problem of detecting tabular regions in document images is same as object detection in natural scene images. Therefore, in the proposed system Faster R-CNN, deep learning based object detection framework, is used that was originally created for natural scene images. Later on due

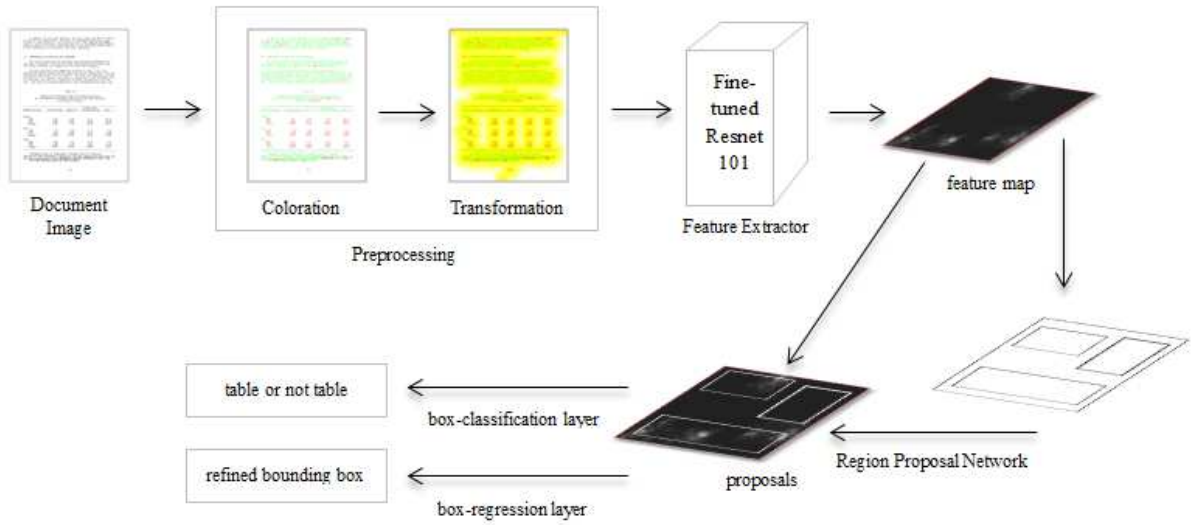


Fig. 3. The proposed approach: In preprocessing, coloration and transformation is applied to the document image and then it is fed to feature extractor, fine-tuned Convolutional Neural Network. It outputs a feature map which is then passed to the RPN for generation of table region proposals. Detection network take these proposals as input and classify them into table and nontable regions along with refined bounding boxes.

to compelling performance of Faster R-CNN it is used in different domains for detection purpose. In this work, the ability of Faster R-CNN is evaluated for detection of tabular structures in document images.

Faster R-CNN is composed of two modules: Region Proposal Network (RPN) and Detection Network. RPN generates the candidate region proposals and Detection Network uses these proposals to detect the tabular regions. Fast R-CNN [18], predecessor of Faster R-CNN, uses selective search for region proposals generation. Key contribution of Faster R-CNN is the introduction of RPN for region proposals generation. Time cost of region proposals generation is much less when using RPN as compare to selective search. Fig. 3 illustrates the work flow of our proposed approach.

1) Feature Extractor: In order to obtain the feature map, the preprocessed document image is passed through a convolutional network. The original Faster R-CNN used Simonyan and Zisserman model (VGG-16) [19] and Zeiler and Fergus model (ZF) [20] pretrained on ImageNet dataset [21] as feature extractor. But since then there have been lot of different models with varying number of parameters that can be used as feature extractor in Faster R-CNN. Feature extractor selection is important as type of layers and number of parameters has great impact on speed and performance of detector.

2) Region Proposal Network: Region Proposal Network (RPN), a deep fully convolutional neural network, takes convolutional feature map produced by base network as input and outputs the bounding boxes along with the objectness score. In order to make region proposal generation task almost cost free RPN shares convolutional layers with detection network. Utilizing sliding window approach, RPN generates k candidate anchors of different scales and ratios for each position in the last convolutional feature map. Default configuration of Faster R-CNN uses three scales and three aspect ratios yielding k=9 anchors. For generation of region proposals, RPN slides a small network of nxn over the last convolutional feature map and generates a lower dimensional feature map. This feature map is then passed to two 1x1 convolutional layers; box-classification layer and box-regression layer. We have used default implementation of Faster R-CNN that takes n=3.

Box-regression layer have 4k outputs. Each set of 4 outputs parameterize a bounding box. Box classification layer generates 2k outputs, where each pair of 2 outputs is the probability that the corresponding bounding box contains table or it is just background. Nonmaximum suppression is used to minimize the number of bounding box proposals.

3) Detector Network: After training of network for region proposal generation, generated region proposals are then passed to the detection module, Fast R-CNN. Detection module utilizes these proposals to detect tables and returns the bounding box coordinates of detected tables along with confidence scores.

C. Training

A large amount of labeled training data is required to train a deep neural network from scratch and this is a serious constraint of deep learning research domain. However, fine tuning of existing pre-trained deep models that have been trained on millions of images is a good practice to extract the useful information from small datasets. Due to limited number of images in our training set, instead of training from

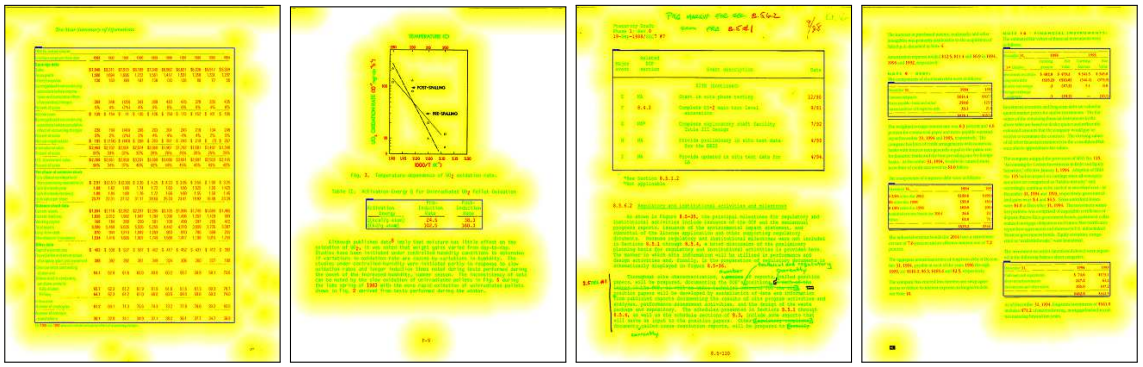
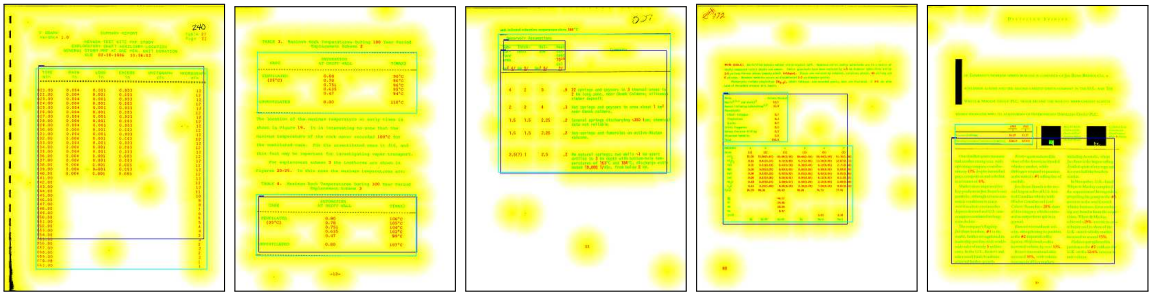


Fig. 4. Few images from UNLV dataset showing table detection results of proposed approach.



(a) Partial Detection (b) Missed Tables (c) Over-Segmented (d) Under-Segmented (e) False Positives

Fig. 5. An illustration of different segmentaion errors. Ground truth is aqua while detected regions are blue in color.

scratch we used transfer learning and domain adaptation to converge our model to good weight configurations. We have used Tensorflow Object Detection API [22] as it supports many different base models trained on various datasets for Faster R-CNN. For fine tuning we used Resnet101 [23] trained on KITTI dataset [24]. Momentum optimizer with momentum of 0.9 and learning rate of 0.001 was used. We have trained our network for 70,000 iterations. We trained our system for 2 classes i.e. background and table regions. To detect overfitting, during training we monitored performance on validation set. It is important to mention that we haven't used any part of UNLV dataset for training or validation of our model. We made a random split of our training data into 80:20 for training and validation respectively. The split resulted in 1019 images for training and left 255 validation images. We have evaluated performance of our model on publically available UNLV dataset.

V. Performance Measures

In literature various performance measures have been used for evaluation of table detection algorithms. These measures range from simple to more complex and sophisticated measures. Since we are focusing only on table detection, the evaluation measures described in [17] have been employed as these performance measures provide more detailed and elaborative measure of how good is our detection algorithm.

As described in [17], consider G_i represents the ground truth box and D_j represents the bounding box detected by our proposed system. The amount of overlap between two bounding boxes G_i and D_j is given by formula:

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

$|G_i \cap D_j|$ represents the intersecting area of the two bounding boxes i.e, ground truth box and detected box. $|G_i|$ shows the area of ground truth box and $|D_j|$ represents the area of detected box. The value of A ranges from 0 to1 depending upon the amount of overlap between G_i and D_j.

A. Correct Detections:

It represents the number of ground truth tables that have one to one correspondence with detected tables and have significant amount of overlap $(A \geq 0.9)$.

B. Partial Detections:

Partial detections depict the number of ground truth tables that have major overlap with one of the detected tables but overlap $(0.1 < A < 0.9)$ is not large enough to classify it as correct detection.

C. Over-Segmented Tables:

Over-segmentation represents the splitting errors. It means that larger tables are broken down into number of smaller ones and it is number of ground truth tables that have major overlap $(0.1 < A < 0.9)$ with more than one predicted tables.

TABLE I
PERFORMANCE COMPARISON OF DIFFERENT APPROACHES

Performance Measures	Accuracy(%)		
	Schreiber et al. [11]	***Gilani et al. [15]***	***Our Approach***
Precision	96.15	82.3	86.33
Recall	97.40	90.67	93.21
F1 Score	96.77	86.29	89.64

D. Under-Segmented Tables:

Under-segmentation represents the merging errors. It means that different tables are combined by the detection algorithm to form single one. It represents the number of ground truth tables that have major overlap $(0.1 < A < 0.9)$ with one predicted table but detected table has overlap with other ground truth tables as well.

E. Missed Detections:

Missed detections show number of ground truth tables that do not have major overlap with any of the detected tables $(A \leq 0.1)$. It represents the number of ground truth tables that were missed by the detection algorithm.

F. False Positives:

These are the number of false alarms generated by our detection algorithm as algorithm mistook some nontable regions as tables. And this measure represents the number of predicted tables that have very minor overlap with any of the ground truth tables $(A \leq 0.1)$.

G. Precision:

This measure summarizes the performance of table detection method by calculating the percentage of detected table regions that belong to ground truth table regions in document image. The formula for calculating area precision is:

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

H. Recall:

It is measure of percentage of ground truth table regions that are marked as tables by detection algorithm. The formula for calculating recall is:

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

I. F1 Score:

F1 Score, considers both precision and recall to compute score, calculates as follow:

$$2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{4}$$

Fig. 5 shows different types of segmentation errors (partial detections, over-segmentation, under-segmentation and false positive detections) quantified by the corresponding measure.

VI. EXPERIMENTS AND RESULTS

To assess the performance of our proposed methodology, we chose publically available UNLV dataset. Dataset covers wide variety of document sources like newspapers, magazines, research articles, business letters and technical reports etc. 2889 pages of scanned document images are present in original dataset. Out of 2889 document images only 427 images contain table regions and 570 tables are present in UNLV dataset. Details about this dataset can be found here [17]. We have employed all these 427 images for evaluation of our proposed system. We did not use any part of this data during training and validation of our system. Our approach achieved state-of-the-art performance on metrics described in section V on UNLV dataset. Result of proposed approach on UNLV dataset is shown in Fig. 4.

Proposed approach is benchmarked against, Schreiber et al. [14] and Gilani et al. [15], state-of-the-art in table detection. In both of these techniques Faster R-CNN has been used for detection purpose. Keeping the detection module same, we can have a clear understanding of importance of background and foreground features for table spotting in document images. The comparative analysis of the different techniques is provided in the Table I. Gilani et al. [15] used UNLV dataset for training, validation and testing of the system. Only 20% of the UNLV dataset is used by Gilani et al. for testing the performance of system and Schreiber et al. [14] used ICDAR 2013 table competition dataset for testing. The higher complexity of UNLV datset as compared to ICDAR 2013 table competition dataset makes detection of tabular regions in UNLV dataset difficult [25]. So in order to make a fair comparison with these techniques, we trained these approaches on our proposed dataset and checked the performance of these systems on complete UNLV dataset. Table II provides comparison of proposed system with approaches presented in [15] and [14], trained on our proposed dataset.

Columns and rows header are of great importance while parsing tables. In case the headers are missed out whole table detection becomes useless as no information can be extracted. Hence, number of correct detections becomes most expressive measure. The experimental results exhibit that our approach outperformed the existing approaches as correct detections greatly improve from 58% to 73%. Improvement in other performance measures like partial detections and over-segmentations shows that our approach greatly reduce the segmentaion errors as well. As clear from results foreground features play important role in table spotting and when fore-

TABLE II
PERFORMANCE COMPARISON OF DIFFERENT APPROACHES

Performance Measures	Accuracy(%)			
	Schreiber et al. [11]	*Gilani et al. [15]*	*With Color Coding only*	*Our Approach*
Correct Detections	58.09	62.05	68.70	73.56
Partial Detections	11.51	11.87	10.79	7.01
Over Segmented Tables	22.84	17.62	7.55	6.47
Under Segmented Tables	6.38	5.75	8.63	5.75
Missed Detections	5.75	6.83	8.27	8.99
False Positives	2.31	2.73	1.65	1.48
Precision	72.67	77.28	87.69	86.33
Recall	89.45	90.88	92.13	93.21
F1 Score	80.19	83.53	89.85	89.64

ground features are combined with background features they further improve the performance.

VII. CONCLUSION

This research work presents a novel data driven approach for solving table detection problem. The prime focus of this approach is on the use of foreground and background features of document images for table detection. Proposed system uses color coding or coloration in order to differentiate numeric and textual data and for the separation of text regions from nontext regions image transformation is utilized. It then uses RPN followed by fully connected neural network for detection of tabular regions in color coded and transformed document images. Experimental results show that taking foreground and background features into account make table detection more robust as they provide clear estimate about the presence of table regions. Publicly accessible UNLV dataset has been used for evaluation purpose. Performance measures show great improvement in results as compared to previous state-of-the-art methods. In future, we will focus on table structure recognition.

REFERENCES

[1] 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*, pp. 91–99, 2015.

[2] A. Shahab, "Table Ground Truth for the UW3 and UNLV datasets (DFKI-TGT-2010)." http://tc11.cvc.uab.es/datasets/DFKI-TGT-2010_1, 2010. Accessed: 2017-04-07.

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

[4] T. Kieninger and A. Dengel, "Table recognition and labeling using intrinsic layout features," in *International Conference on Advances in Pattern Recognition*, pp. 307–316, Springer, 1999.

[5] T. Kieninger and A. Dengel, "Applying the t-recs table recognition system to the business letter domain," in *icdar*, p. 0518, IEEE, 2001.

[6] J. Hu, R. S. Kashi, D. P. Lopresti, and G. Wilfong, "Medium-independent table detection," in *Document Recognition and Retrieval VII*, vol. 3967, pp. 291–303, International Society for Optics and Photonics, 1999.

[7] Y. Wangt, I. Phillipst, 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*, pp. 528–532, IEEE, 2001.

[8] B. Gatos, D. Danatsas, I. Pratikakis, and S. J. Perantonis, "Automatic table detection in document images," in *International Conference on Pattern Recognition and Image Analysis*, pp. 609–618, Springer, 2005.

[9] 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.

[10] 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*, pp. 843–847, IEEE, 2009.

[11] 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*, pp. 287–292, IEEE, 2016.

[12] S. F. Rashid, A. Akmal, M. Adnan, A. A. Aslam, and A. Dengel, "Table recognition in heterogeneous documents using machine learning," in *Document Analysis and Recognition (ICDAR), 2017 14th IAPR International Conference on*, vol. 1, pp. 777–782, IEEE, 2017.

[13] D. He, S. Cohen, B. Price, D. Kifer, and C. L. Giles, "Multi-scale multi-task fcn for semantic page segmentation and table detection," in *Document Analysis and Recognition (ICDAR), 2017 14th IAPR International Conference on*, vol. 1, pp. 254–261, IEEE, 2017.

[14] S. Schreiber, S. Agne, I. Wolf, A. Dengel, and S. Ahmed, "Deepdesrt: Deep learning for detection and structure recognition of tables in document images," in *Document Analysis and Recognition (ICDAR), 2017 14th IAPR International Conference on*, vol. 1, pp. 1162–1167, IEEE, 2017.

[15] A. Gilani, S. R. Qasim, I. Malik, and F. Shafait, "Table detection using deep learning," in *Document Analysis and Recognition (ICDAR), 2017 14th IAPR International Conference on*, vol. 1, pp. 771–776, IEEE, 2017.

[16] Tzutalin, "LabelImg. Git code." https://github.com/tzutalin/labelImg, 2015. Accessed: 2018-03-30.

[17] F. Shafait and R. Smith, "Table detection in heterogeneous documents," in *Proceedings of the 9th IAPR International Workshop on Document Analysis Systems*, DAS '10, (New York, NY, USA), pp. 65–72, ACM, 2010.

[18] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, pp. 1440–1448, 2015.

[19] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

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

[21] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pp. 248–255, Ieee, 2009.

[22] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, *et al.*, "Speed/accuracy trade-offs for modern convolutional object detectors," in *IEEE CVPR*, vol. 4, 2017.

[23] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[24] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets Robotics: The KITTI Dataset," *International Journal of Robotics Research (IJRR)*, 2013.

[25] T. A. Tran, H. T. Tran, I. S. Na, G. S. Lee, H. J. Yang, and S. H. Kim, "A mixture model using random rotation bounding box to detect table region in document image," *Journal of Visual Communication and Image Representation*, vol. 39, pp. 196 – 208, 2016.