Table Structure Recognition Using CoDec Encoder-Decoder

Bhanupriya Pegu, Maneet Singh(✉), Aakash Agarwal, Aniruddha Mitra, and Karamjit Singh

AI Garage, Mastercard, Gurgaon, India
{bhanupriya.pegu,maneet.singh,aakash.agarwal,aniruddha.mitra, karamjit.singh}@mastercard.com

Abstract. Automated document analysis and parsing has been the focus of research since a long time. An important component of document parsing revolves around understanding tabular regions with respect to their structure identification, followed by precise information extraction. While substantial effort has gone into table detection and information extraction from documents, table structure recognition remains to be a long-standing task demanding dedicated attention. The identification of the table structure enables extraction of structured information from tabular regions which can then be utilized for further applications. To this effect, this research proposes a novel table structure recognition pipeline consisting of row identification and column identification modules. The column identification module utilizes a novel Column Detector Encoder-Decoder model (termed as *CoDec* Encoder Decoder) which is trained via a novel loss function for predicting the column mask for a given input image. Experiments have been performed to analyze the different components of the proposed pipeline, thus supporting their inclusion for enhanced performance. The proposed pipeline has been evaluated on the challenging ICDAR 2013 table structure recognition dataset, where it demonstrates state-of-the-art performance.

Keywords: Table structure recognition · Encoder-Decoder · Document analysis

1 Introduction

The volume of digital information getting generated is growing at an astonishing rate, where text documents correspond to a major portion of it. Parsing such documents and extracting the required information is a challenging task since many such documents contain tables with varying layouts and colour schemes. For example, Fig. 1 show sample tabular regions of various layouts in different document types, such as invoices, research papers, and reports. To enable automated processing of these documents, accurate tabular parsing methodology is required. Significant efforts have been made in the past to extract this tabular information from documents using automated processes [6,7,12,14,20,23].

E. H. Barney Smith and U. Pal (Eds.): ICDAR 2021 Workshops, LNCS 12917, pp. 66–80, 2021.
https://doi.org/10.1007/978-3-030-86159-9_5

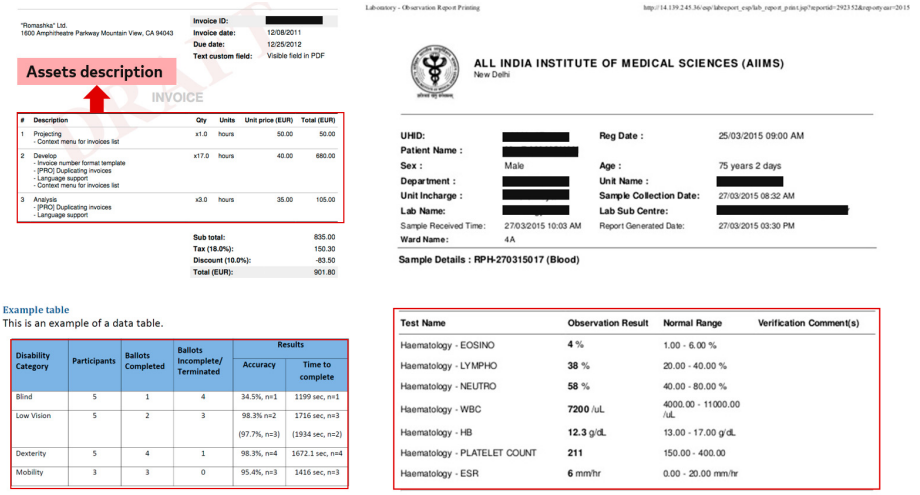


Fig. 1. Table structure recognition has applications involving automated extraction of tabular content for further analysis. For example, extraction of related fields from invoices, research publications, or reports.

The problem of successful table parsing can be decomposed into two sub-problems [22]: (i) table detection and (ii) structure recognition. The first sub-problem of table detection can be solved by detecting the pixels representing the tabular region in a document. Several methods have been proposed in the past to solve this problem [6,20,23] which have shown high detection results on publicly available datasets. Once a tabular region is successfully detected, the next sub-problem is to identify the structure of a table by understanding its layout and detecting the cell region in it [7]. Detection of cell regions can further be broken down into row and column identification which can ultimately be combined to discover the corresponding cells in a table [20]. The problem of structure recognition is extremely challenging due to significant intra-class variability, e.g., tables can have different layouts, several colour schemes, the erratic use of ruling lines for tables, structure delineation, or simply due of diverse table contents [2]. While recent techniques such as the CascadeTabNet [18] have shown almost near perfect results for table detection, the task of table structure identification still requires dedicated attention. To this end, this paper focuses on the table structure recognition sub-problem.

In this paper, an end-to-end pipeline is proposed for table structure recognition containing two components: (i) column identification module, and (ii) row identification module. The column identification module utilizes a novel Column Detector Encoder-Decoder model (termed as *CoDec* Encoder-Decoder) which is trained via a novel loss function containing *Structure loss* and *Symmetric loss*. The intuition of the proposed method is to develop a small and compact deep learning architecture which can be used to train models with limited training

data and have split second inference time to enable real-time applications. In particular, the contributions of this research are as follows:

- This research proposes an end-to-end pipeline for table structure recognition using a small and compact deep learning architecture. The relatively lower trainable parameters enables model training with limited data and split second inference time for applicability in real-world scenarios.
- The proposed pipeline utilizes a novel column identification module, termed as the *CoDec Encoder-Decoder* model. The CoDec model is trained with a novel loss function consisting of a Structure and Symmetric loss for faster and accurate learning.
- The performance of the proposed pipeline has been evaluated on the challenging ICDAR 2013 dataset [7]. The proposed pipeline demonstrates improvement from the state-of-the-art networks even without explicitly training or fine-tuning on the ICDAR 2013 dataset, thus supporting the generalizability of the proposed technique. Further, analysis has been performed on the proposed pipeline via an ablation study which supports the inclusion of different components.

The rest of the paper is organized as follows: Sect. 2 outlines an overview of the related work on tabular structure recognition. Section 3 presents the detailed description of the proposed pipeline. Section 4 elaborates upon the details of the experiments and datasets. Section 5 presents the results and analysis of the proposed pipeline, and Sect. 6 presents the concluding remarks.

2 Related Work

The concept of recognising table structure has evolved gradually from pre-Machine-Learning (ML) era, when it used to be completely heuristic based, to the recent age of deep learning. Comprehensive summary of the algorithmic evolution can be traced in the surveys available describing and summarizing the state-of-the-art in the field [2,3,11,24,29]. One of the earliest successful developments in table understanding could be found in T-RECS by Kieninger and Dengel [12], where they built a framework to group words into columns on basis of its horizontal overlaps, followed by dividing those word-groups into cells with respect to the column's margin structure. In the same period many handcrafted features based algorithms were introduced [5,9,28], which were task specific and demonstrated heavy utilization of the domain knowledge. Another early data driven approach by Wang et al. [27] proposes a seven step formulation based on probability optimization. Considering the high intra-class variability, Shigarov et al. [21] proposed a table decomposing algorithm with sets of domain-specific rules, where they also rely on PDF metadata like font and character bounding boxes as well as ad-hoc heuristics.

The proposed table structure recognition pipeline is conceptually connected with recent Deep Learning (DL) based developments on this subject. The remainder of this Section thus focuses on recent works which set benchmarks

utilizing deep-learning techniques. Though research related to table detection in PDF documents can be traced back to the technique published by Hao et al. [8], research on table structure recognition still remains limited owing to the challenging and complex nature of the problem. Schreiber et al. [20] tackle the problem of scarce labelled data, which hinders high parameterized DL training, by leveraging Domain Adaptation and Transfer Learning. The authors used Fully-Convolution Networks (FCN) based general object detection models to adapt to the domain of documents using Transfer Learning. However, their performance metric restricts itself to identifying rows/columns instead of using the cell-level information. Siddiqui et al. [22] propose to constrain the problem space for obtaining improved performance. Qasim et al. [19] modelled the problem of table recognition with Graph Neural Networks where Convolution Neural Network (CNN) and Optical Character Recognition (OCR) engine are employed to extract the feature maps and word positions, respectively. The representation of the features are learned through an interaction network. The representations are then concatenated and fed into a dense neural network to classify the vertex pairs. Recently, TableNet [16] architecture was proposed which is a multi-task network built on a VGG based encoder followed by task specific decoders, to model the inter-dependency between the twin tasks of table detection and table structure identification. Further, recently, CascadeTabNet [18] was proposed, where the tasks of table detection and structure recognition are accomplished by a single CNN model utilizing cascade mask region-based CNN and a high-resolution network.

In literature, the closest technique to the current manuscript is the TableNet architecture [16]. The TableNet model utilizes a large-scale pre-trained VGG network as the back-bone architecture, and performs semantic segmentation on the given input image for generating a column mask, along with the utilization of domain knowledge for generating the row co-ordinates. In comparison, the proposed table structure recognition pipeline utilizes a novel light-weight CoDec Encoder-Decoder model which is trained using a novel loss function for column detection. Further, as opposed to semantic segmentation which involves generating a mask for each class, the proposed CoDec Encoder-Decoder constructs a single image containing the column mask, resulting in further reduction in the number of trainable parameters. Detailed description of the proposed pipeline is provided in the next Section.

3 Table Structure Recognition

In order to perform table structure recognition, this research follows a top-down approach. A two-step approach is followed: (i) identification of columns, followed by (ii) identification of rows. Figure 2 presents a broad over-view of the proposed table structure recognition pipeline. A given input table image is processed to identify the column details via the proposed Column Detector Encoder-Decoder model (termed as the *CoDec* Encoder-Decoder), followed by the identification of different rows in the table using the domain knowledge and different image

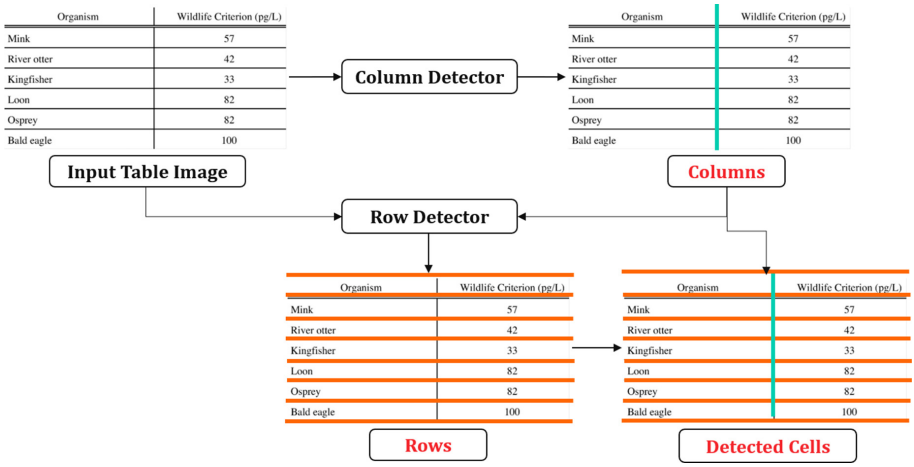


Fig. 2. Proposed pipeline for table structure recognition. The input tabular image is provided to the row detection and column detection modules, which return the row and column co-ordinates. The information from the two modules is then fused together to generate the cell co-ordinates.

processing rules. The row and column information is then combined to generate the cell co-ordinates. Detailed explanation of each component is provided in the following subsections.

3.1 Column Identification via Proposed CoDec Encoder-Decoder

Column identification involves identifying the columns in a given tabular image. As shown in Fig. 3, the task suffers from several challenges such as varying tabular formats, presence/absence of columns lines, differing space between different columns, etc. Existing techniques in literature have either utilized hand-crafted techniques or focused on specific table designs only. In order to develop a more generalized solution, this research proposes a novel deep learning based CoDec Encoder-Decoder formulation for identifying columns in the given input table.

Figure 4 presents a diagrammatic overview of the proposed CoDec Encoder-Decoder model. Given an input image, the model outputs a mask with the column identifiers. The loss function of the proposed CoDec Encoder-Decoder model utilizes a (i) Structure loss and a (ii) Symmetric loss for identifying the columns from the given tabular image. For n training samples, the loss function of the proposed CoDec Encoder-Decoder model is given as follows:

$$\mathcal{L}_{CoDec} = \frac{1}{n}\sum_{i=1}^{n}\left(\underbrace{\|f(g(x^i)) - x^i_{mask}\|_2^2}_{Structure\ Loss} + \underbrace{\lambda\|f(g(x^i)) - \mathcal{P}(f(g(x^i)))\|_2^2}_{Symmetric\ Loss}\right) \quad (1)$$

where, x^i and x^i_{mask} refer to the i^{th} training image and the corresponding column mask. $g(.)$ and $f(.)$ refer to the Encoder and Decoder modules, respectively, while

Instance	Size	Algorithm	r_{best}	$\overline{r}$	r_σ	$\overline{t}$ (s)
1	5	SA	12.776	13.693	0.62	0.82
		CA-PSLS	10.942	11.704	0.49	4.31
		PSO	11.046	11.716	0.49	4.89
2	6	SA	16.004	17.377	0.69	1.66
		CA-PSLS	14.686	15.590	0.67	7.76
		PSO	14.320	15.349	0.56	8.28
3	9	SA	20.849	22.328	0.87	6.84
		CA-PSLS	18.157	19.797	1.03	21.07
		PSO	18.579	19.205	0.49	22.84
4	20	SA	29.969	31.680	0.98	92.01
		CA-PSLS	27.927	33.129	5.48	125.52
		PSO	32.596	34.426	2.23	138.00

(a)

Language	Names	Avg. Len.
German	3153	15.1
English	1660	13.6
Serbocroatian	1474	14.3
Italian	1151	16.2
French	1141	15.8
Polish	1057	16.0
Spanish	1031	14.0
Danish	817	15.7
Dutch	809	15.1
Swedish	746	15.7
Czechoslovak	653	13.6
Norwegian	622	16.2
Portuguese	600	11.1
Total	14914	14.8

(b)

File	Words	Dice	Jaccard	Overlap	WN1
Bra14	931	0.699	0.701	0.711	0.742
Bra02	959	0.637	0.685	0.697	0.753
Brb20	930	0.672	0.674	0.693	0.731
Bra15	1071	0.653	0.651	0.684	0.732
Bra13	924	0.667	0.673	0.682	0.735
Bra01	1033	0.650	0.648	0.674	0.714
Brb13	947	0.649	0.650	0.674	0.722
Bra12	1163	0.626	0.622	0.649	0.717
Bra11	1043	0.634	0.639	0.648	0.708
Brc01	1100	0.625	0.627	0.638	0.688

(c)

	Census-Income				DBGEN			
	cardinalités	sans tri	d1 ... d10	d10 ... d1	cardinalités	sans tri	d1 ... d10	d10 ... d1
d1	7	42 427	32	42 309	2	$0{,}75\times10^6$	24	$0{,}75\times10^6$
d2	8	36 980	200	36 521	3	$1{,}11\times10^6$	38	$1{,}11\times10^6$
d3	10	34 257	1 215	28 975	7	$2{,}58\times10^6$	150	$2{,}78\times10^6$
d4	47	$0{,}13\times10^6$	12 118	$0{,}13\times10^6$	9	$0{,}37\times10^6$	100 6	$3{,}37\times10^6$
d5	51	35 203	17 789	28 803	11	$4{,}11\times10^6$	10 824	$4{,}11\times10^6$
d6	91	$0{,}27\times10^6$	75 065	$0{,}25\times10^6$	50	$13{,}60\times10^6$	$0{,}44\times10^6$	$1{,}42\times10^6$
d7	113	12 199	9 217	12 178	2 526	$23{,}69\times10^6$	$22{,}41\times10^6$	$23{,}69\times10^6$
d8	132	20 028	14 062	19 917	20 000	24.00×10^6	$24{,}00\times10^6$	$22{,}12\times10^6$
d9	1 240	29 223	24 313	28 673	400 000	$24{,}84\times10^6$	$24{,}84\times10^6$	$19{,}14\times10^6$
d10	99 800	$0{,}50\times10^6$	$0{,}48\times10^6$	$0{,}30\times10^6$	984 298	$27{,}36\times10^6$	$27{,}31\times10^6$	$0{,}88\times10^6$
total	-	$1{,}11\times10^6$	$0{,}64\times10^6$	$0{,}87\times10^6$	-	$0{,}122\times10^9$	$0{,}099\times10^9$	$0{,}079\times10^9$

(d)

Dimensions				Measures	
location	time	salesman	product	cost	profit
Montreal	March	John	shoe	100$	10 $
Montreal	December	Smith	shoe	150$	30 $
Quebec	December	Smith	dress	175$	45 $
Ontario	April	Kate	dress	90$	10 $
Paris	March	John	shoe	100$	20 $
Paris	March	Marc	table	120$	10 $
Paris	June	Martin	shoe	120$	5 $
Lyon	April	Claude	dress	90$	10 $
New York	October	Joe	chair	100$	10 $
New York	May	Joe	chair	90$	10 $
Detroit	April	Jim	dress	90$	10 $

(e)

Fig. 3. Sample tabular regions having different formats, varying column identifiers (lines/no lines), and varying spacing. The presence of different formats makes the problem of structure identification a challenging task.

λ is the weight for controlling the contribution of the Symmetric Loss. $\mathcal{P}(.)$ refers to the flip operator such that the input image is mirrored across the x-axis.

As mentioned above, the proposed CoDec Encoder-Decoder model is trained via a combination of the (i) Structure loss and the (ii) Symmetric loss. As shown in Fig. 4, the *Structure loss* minimizes the distance between the decoded sample and the column mask for the input tabular image. That is, unlike traditional Encoder-Decoder models, the input is not reconstructed at the output, instead the Decoder is trained to generate the column mask for the input by recognizing the columns in the given input. Thus, the Encoder learns a latent representation for the given image, which is then up-sampled by the Decoder for generating the corresponding column mask. It is our belief that the different convolutional filters are able to encode the variations observed in the varying tabular formats, making it possible to recognize the column structure from the given image. The second component of the CoDec Encoder-Decoder model corresponds to the *Symmetric loss* which attempts to benefit from the symmetric structure observed in tabular regions. As observed from Fig. 3, the column information is mostly symmetric across the x-axis, and thus the Symmetric loss attempts to ensure that the decoded column mask also maintains the same property by being symmetric across the x-axis. Therefore, as part of the CoDec loss function, the distance between the generated column mask and the flipped column mask is also minimized, which ensures stricter boundaries across the length of the column. In the CoDec loss function, $\mathcal{P}(.)$ (from Eq. 1) refers to the flip operator, which flips the provided image across the x-axis. Mathematically, for an input image x, the $\mathcal{P}(.)$ operator is represented as:

$$\mathcal{P}(x) = (x^R)^R \tag{2}$$

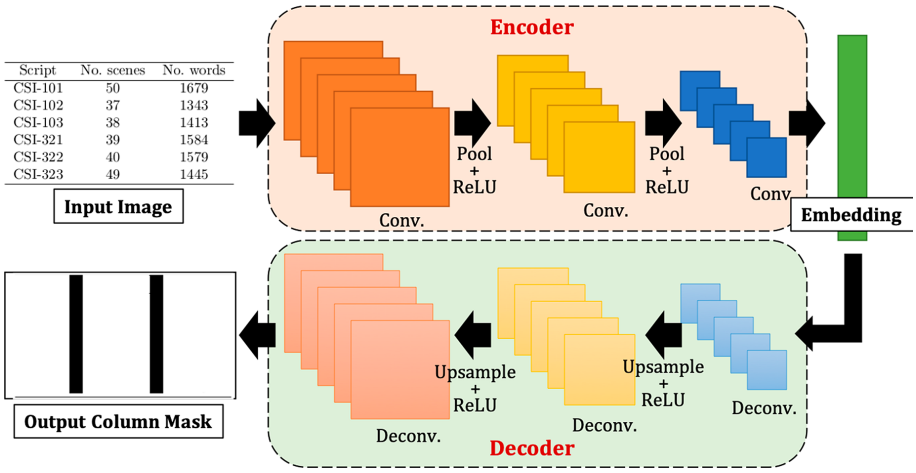


Fig. 4. Diagrammatic representation of the CoDec Encoder-Decoder model for extracting the column information from a given tabular image. The model is trained via a combination of Structure loss and Symmetric loss for identifying the columns.

where, x^R refers to rotating the image by 90°. The Symmetric Loss is thus developed using the available domain knowledge for tabular regions for extracting an accurate column mask.

During training, the CoDec Encoder-Decoder is optimized using the Structure loss and the Symmetric loss (Eq. 1). At inference, the trained model is used to output the column mask for the given tabular image. The generated column mask is then post-processed via binary thresholding to identify the columns. Given the extracted column information, the input image is processed via the row identification module for row recognition. Details regarding the row identification module are provided in the next subsection.

3.2 Row Identification Module

In literature, row identification of tabular regions has mostly been performed via the use of domain knowledge and business rules. Similarly, in this research, a combination of different rules is used for the identification of rows in tabular regions. As observed in Fig. 3, a new row is often signified by the presence of an entry in the first column of the table. In order to utilize this domain knowledge, once the column mask has been extracted, the input image and the mask are provided to the row extraction module for the identification of rows. Further, the co-ordinates of each word are also extracted using the Tesseract OCR [26] which are also utilized for identifying the row details in the given tabular image. The following process is followed by the row identification module:

1. Image processing based line detection is applied on the input image. The given image is converted into grayscale, followed by Canny edge detection [1].

Hough transform [10] based line detection is then applied on the processed image for detecting the horizontal lines (lines with a large gap between their y_1 and y_2 co-ordinates are eliminated as vertical lines). Post-processing is performed wherein detected lines with a gap of less than a chosen threshold of pixels along the y-axis are removed. This is done to eliminate duplicate row lines and double boundaries.

2. Since all tabular regions do not contain row boundaries (lines), parallely, the row co-ordinates are also estimated using the y-coordinates obtained with the extracted words. Initially, each y-coordinate beyond a chosen threshold is identified as a new row line. The column information is then utilized for modeling multi-line cells in un-bordered tables. If a given row contains text in very few columns (less than 80% of the total columns), it is deemed as the continuation of the previous row, and the information of that y-coordinate is updated.
3. As a final step, the rows identified by the above two techniques are fused together to generate the final row coordinates.

3.3 Table Structure Recognition and Data Extraction

The row and column coordinates obtained by the two modules are fused together to generate the overall structure of the table. Parallely, as mentioned above, data extraction is performed from the tabular region using the Tesseract OCR [26]. The OCR returns the content in the given image along with the coordinates of each word containing the x co-ordinate, y co-ordinate, and width of the word. These coordinates are then used to divide the content into the corresponding cells created using the row and column coordinates obtained via the proposed table structure recognition pipeline.

Once the content has been split into the different cells of the table, the information is then post-processed for comparison with the ground-truth. Similar to the existing techniques [16], 1-D tuples are generated for each cell containing the content of the neighbouring cells (upper, lower, immediate left, and immediate right cells). These tuples are then compared with the ground-truth information provided with the datasets. The datasets available for table structure recognition often contain an XML file for each table containing the details regarding the structure and the content (coordinates of each cell along with the content). The XML files are thus used for generating the ground-truth 1-D tuples for each cell, following which matching is performed with the tuples generated using the proposed table structure recognition pipeline.

4 Datasets and Protocols

Two datasets have been used for experiments: (i) Marmot dataset [4] and the (ii) ICDAR 2013 dataset [7]. Details regarding each are as follows:

– **Marmot Dataset**[1] [4]**:** The Marmot dataset contains over 1000 PDF documents in English and Chinese languages with tabular regions. The ground-truth annotations of the table structure (row and column co-ordinates) have also been provided as an XML file for each document. As part of this research, the Marmot dataset has been used for training the novel CoDec Encoder-Decoder model. Specifically, the 509 English documents are pre-processed for the extraction of the tabular region (input of the Encoder) along with the creation of the column mask (output of the Decoder) based on the ground-truth provided with the dataset. Owing to the limited training data, data augmentation has been performed on the tabular regions, specifically, mirroring along the $y-axis$ and incorporation of minor Gaussian noise.

– **ICDAR 2013 Dataset**[2] [7]**:** The ICDAR 2013 dataset is one of the most popular and commonly used dataset for table structure recognition. In this research as well, the ICDAR 2013 dataset has been used for evaluating the proposed pipeline for table structure recognition. The dataset contains a total of 67 PDF documents with tabular regions. The ICDAR dataset contains both vertical and horizontal tables - over 30% of the total tables are vertical in nature. The dataset provides XML files for each document containing ground-truth annotations with respect to the table position and its structure. Information such as the cell co-ordinates and the content has also been provided. Consistent with existing techniques and in order to compare with recent state-of-the-art algorithms [16,20], the standard protocol is followed on this dataset, wherein 34 images are used for evaluating the model. Existing techniques often utilize the remaining samples for fine-tuning the pre-trained architecture, however, we do not utilize the ICDAR 2013 dataset for training or fine-tuning.

4.1 Implementation Details

As elaborated in the previous Section, column identification has been performed using the proposed CoDec Encoder-Decoder model which consists of an Encoder module and a Decoder module. The Encoder is composed of four convolutional layers with 3×3 kernels and filter sizes of $[32, 16, 8, 4]$. We use $ReLU$ [15] as the activation function after each convolution layer. Max-pooling is also applied post each convolution layer for reducing the dimension of the feature. The Decoder model is the mirror of the encoder architecture with four transposed convolutional layers having 3×3 kernels and filter sizes of $[4, 8, 16, 32]$. After each layer, $ReLU$ activation function has been used. An image of dimension 224×224 is provided as input to the CoDec Encoder-Decoder model. The model is implemented in PyTorch [17], and the Adam optimizer [13] has been used to train the model with an initial learning rate of 0.01. The weight for the Symmetric loss (λ in Eq. 1) is set to 0.01. In the row identification module, a minimum gap of 20

[1] https://www.icst.pku.edu.cn/cpdp/sjzy/index.htm.
[2] http://www.tamirhassan.com/html/competition.html.

Table 1. Table structure recognition performance on the ICDAR 2013 dataset, along with comparison with recent state-of-the-art algorithms. Owing to the same protocol, results have directly been taken from the published manuscript [16].

Algorithm	Recall	Precision	F1-Score
TableNet + Semantic features (fine-tuned) [16]	0.9001	0.9307	0.9151
TableNet + Semantic features [16]	0.8994	0.9255	0.9122
TableNet [16]	0.8987	0.9215	0.9098
DeepDeSRT (fine-tuned) [20]	0.8736	**0.9593**	0.9144
Proposed	**0.9289**	0.9337	**0.9304**

pixels is maintained between each detected row in order to eliminate duplicate line boundaries and incorrect row co-ordinates. Default parameters have been used for the Canny detector which remain consistent across all grayscale input images. For the Hough transform, the 'minLineLength' is a function of the size of the image, and the 'maxLineGap' is set to 10. Other parameters are kept as default. The parameters of the row-identification module are configured once and then used consistently across all the images. That is, once trained/configured, the entire pipeline is automated in nature without any manual intervention. Given a tabular image, the pipeline outputs the table structure using the (i) column detector followed by the (ii) row detector, resulting in an end-to-end framework. During the generation of the 1-D tuples, the extracted words (obtained via the ground-truth XML file or the Tesseract OCR), are converted to lower case after removing any trailing or preceding white spaces. All special characters are replaced with a '_', followed by matching between the words.

5 Results

Table 1 presents the results obtained on the ICDAR 2013 dataset for table structure recognition. The top result is presented in bold, while the second best result is underlined. As per the existing research, performance has been reported in terms of the recall, precision, and F-1 score:

$$Recall = \frac{TP}{TP \ + \ FP}; \ Precision = \frac{TP}{TP \ + \ FN}; \tag{3}$$

$$F1 \ Score = \frac{2 \times Precision \times Recall}{Precision \ + \ Recall} \tag{4}$$

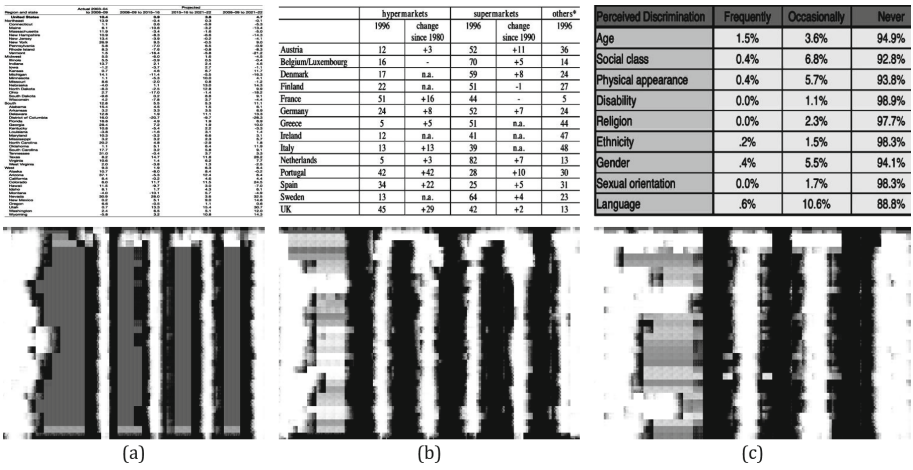
It can be observed that the proposed technique achieves a recall of 0.9289, thus demonstrating an increase of over 2% as compared to the state-of-the-art technique (0.9001 obtained by TableNet [16]). The proposed technique also obtains the second best performance for precision by reporting 0.9337, while DeepDeSRT [20] obtains 0.9593. It is important to note that the best results of DeepDeSRT and the TableNet architecture are obtained after fine-tuning on the ICDAR 2013

Table 2. Ablation study on the proposed table structure recognition pipeline. Analysis has been performed by modifications of the row detection module (removal of the image processing (IP) based row detection and co-ordinate based row detection) and the column detection module (removal of the Symmetric loss).

Algorithm	Recall	Precision	F1-Score
Proposed - IP based row detection	0.7832	0.7984	0.7882
Proposed - Coordinate based row detection	0.8545	0.8910	0.8636
Proposed - $\mathcal{L}_{Symmetric}$	0.8335	0.8918	0.8556
Proposed	**0.9289**	**0.9337**	**0.9304**

dataset, while the proposed technique is only trained on the Marmot dataset and does not utilize the ICDAR 2013 dataset during training or fine-tuning. The improved performance obtained without explicit training/fine-tuning on the ICDAR 2013 dataset supports the generalizable behavior of the proposed pipeline. Finally, an overall F1 score of 0.9304 is obtained via the proposed technique demonstrating an improvement from the state-of-the-art TableNet (0.9151). The improved results obtained on the standard benchmark dataset demonstrates the efficacy of the proposed algorithm. Further, Fig. 5 presents sample images from the ICDAR 2013 dataset, along with the raw column mask generated by the proposed CoDec Encoder-Decoder model. The model is able to identify the column demarcations in the absence of column lines (Fig. 5(a)), as well as in the presence of line demarcations (Fig. 5(b–c)). The high performance obtained on the ICDAR 2013 dataset, without explicitly training or fine-tuning on it further promotes the utility of the column detection model on unseen datasets not used during the training of the model.

Ablation Study and Effect of λ: In order to analyze the proposed pipeline for table structure recognition, an ablation study has been performed on the same. Table 2 presents the results obtained by removing different components from the structure recognition pipeline. Experiments have been performed by removing: (i) image processing based row detection from the row detection module, (ii) co-ordinate based row detection from the row detection module, and (iii) symmetric loss from the column detection module. As demonstrated from Table 2, removal of any component from the proposed pipeline results in a reduction in the precision, recall, and F1-score performance. Specifically, maximum drop in performance is observed by removing the image processing based row detection component (almost 15% drop in F1 score). Drop in performance is also observed upon removing the Symmetric Loss from the proposed CoDec Encoder-Decoder model for column detection (Eq. 1). The drop in performance further reinstates the benefit on incorporating the Symmetric loss across the x-axis. Further, experiments were also performed to analyze the impact of λ (Eq. 1) which controls the contribution of the Symmetric Loss in the CoDec model. With a much smaller value ($\lambda = 0.001$), the F-1 score reduces to 84.78%, while a larger λ ($\lambda = 0.1$)



	hypermarkets		supermarkets		others*
	1996	change since 1980	1996	change since 1990	1996
Austria	12	+3	52	+11	36
Belgium/Luxembourg	16	-	70	+5	14
Denmark	17	n.a.	59	+8	24
Finland	22	n.a.	51	-1	27
France	51	+16	44	-	5
Germany	24	+8	52	+7	24
Greece	5	+5	51	n.a.	44
Ireland	12	n.a.	41	n.a.	47
Italy	13	+13	39	n.a.	48
Netherlands	5	+3	82	+7	13
Portugal	42	+42	28	+10	30
Spain	34	+22	25	+5	31
Sweden	13	n.a.	64	+4	23
UK	45	+29	42	+2	13

Perceived Discrimination	Frequently	Occasionally	Never
Age	1.5%	3.6%	94.9%
Social class	0.4%	6.8%	92.8%
Physical appearance	0.4%	5.7%	93.8%
Disability	0.0%	1.1%	98.9%
Religion	0.0%	2.3%	97.7%
Ethnicity	.2%	1.5%	98.3%
Gender	.4%	5.5%	94.1%
Sexual orientation	0.0%	1.7%	98.3%
Language	.6%	10.6%	88.8%

(a) (b) (c)

Fig. 5. Sample images from the ICDAR 2013 dataset and the corresponding column mask generated by the proposed CoDec Encoder-Decoder. The model is able to process images with/without column lines and is able to identify column details well.

results in a F-1 score of 86.16%. The reduction in performance upon the removal of different components demonstrates the benefit of each component in the final table structure recognition pipeline, along with the choice of appropriate hyper-parameters.

Comparison on Number of Trainable Parameters: The proposed table structure recognition pipeline utilizes a light-weight Encoder Decoder architecture for column extraction. The proposed column detection model contains only 9,269 trainable parameters, whereas existing state-of-the-art column detection models such as the TableNet [16] and the DeepDeSRT [20] contain at least 1.38M (VGG-19 and VGG-16 architectures [25], respectively). The light-weight nature of the proposed framework results in lesser number of trainable parameters and also reduced size of the model. This enables the proposed pipeline to be trained with lesser number of images, and also makes it deployable in real world scenarios with less resource requirement. The lesser number of parameters prevents the model from over-fitting on the training dataset (Marmot dataset), thus resulting in a generalized behavior on a new dataset as well (ICDAR 2013 dataset). Further, the entire pipeline takes less than 1 s. for inference on an input image.

6 Conclusion

The requirement of automated detection and identification of tables from document images has been increasing over the last two decades. Information extraction from tabular regions is useful for automated content generation and summarization. Extraction of relevant content from tabular regions also requires table structure recognition, which corresponds to identifying the exact structure

of the table along with the cell information. Despite the wide-spread applicability, table structure recognition has received limited attention and continues to be a challenging task due to the complexity and diversity in the structure and style of different tables. To this effect, this paper presents an end to end pipeline for table structure recognition containing two components: (i) column identification module, and (ii) row identification module. The column identification module utilizes a novel Column Detector Encoder-Decoder model (termed as *CoDec* Encoder-Decoder) which is trained via a novel loss function containing Structure loss and Symmetric loss. The detection of the columns is followed by the identification of different rows in the table using domain information and different image processing rules. The performance of the proposed pipeline is evaluated on the ICDAR 2013 dataset, where it demonstrates improvement from the state-of-the-art networks even without explicitly training or fine tuning on the ICDAR 2013 dataset, thereby suggesting a generalizable behavior of the proposed pipeline. Further, as part of this research, ablation study has also been performed by removing different components from the structure recognition pipeline, and results of each experiment have been discussed in this paper. Another key contribution of this research revolves around the limited number of trainable parameters of the proposed CoDec Encoder-Decoder model as compared to existing techniques. There are only 9,269 trainable parameters (in comparison to over 1.39M of existing techniques) in the proposed column detection model which makes the proposed framework trainable with limited number of images, and also makes it deployable with less resource requirement in real world scenarios. As part of future work, the proposed technique can be improved to better model multi-column variations for table structure recognition.

References

1. Canny, J.: A computational approach to edge detection. IEEE Trans. Pattern Anal. Mach. Intell. **6**, 679–698 (1986)
2. Coüasnon, B., Lemaitre, A.: Recognition of tables and forms. In: Handbook of Document Image Processing and Recognition (2014). https://doi.org/10.1007/978-0-85729-859-1
3. Embley, D.W., Hurst, M., Lopresti, D., Nagy, G.: Table-processing paradigms: a research survey. Int. J. Doc. Anal. Recogn. **8**(2–3), 66–86 (2006)
4. Fang, J., Tao, X., Tang, Z., Qiu, R., Liu, Y.: Dataset, ground-truth and performance metrics for table detection evaluation. In: International Workshop on Document Analysis Systems, pp. 445–449 (2012)
5. Gatos, B., Danatsas, D., Pratikakis, I., Perantonis, S.J.: Automatic table detection in document images. In: Singh, S., Singh, M., Apte, C., Perner, P. (eds.) ICAPR 2005. LNCS, vol. 3686, pp. 609–618. Springer, Heidelberg (2005). https://doi.org/10.1007/11551188_67
6. Gilani, A., Qasim, S.R., Malik, I., Shafait, F.: Table detection using deep learning. In: International Conference on Document Analysis and Recognition, pp. 771–776 (2017)
7. Göbel, M., Hassan, T., Oro, E., Orsi, G.: ICDAR 2013 table competition. In: International Conference on Document Analysis and Recognition, pp. 1449–1453 (2013)

8. Hao, L., Gao, L., Yi, X., Tang, Z.: A table detection method for pdf documents based on convolutional neural networks. In: IAPR Workshop on Document Analysis Systems, pp. 287–292 (2016)
9. Hu, J., Kashi, R.S., Lopresti, D.P., Wilfong, G.: Medium-independent table detection. In: Document Recognition and Retrieval VII, vol. 3967, pp. 291–302 (1999)
10. Illingworth, J., Kittler, J.: A survey of the Hough transform. Comput. Vis. Graph. Image Process. **44**(1), 87–116 (1988)
11. Khusro, S., Latif, A., Ullah, I.: On methods and tools of table detection, extraction and annotation in pdf documents. J. Inf. Sci. **41**(1), 41–57 (2015)
12. Kieninger, T., Dengel, A.: The T-Recs table recognition and analysis system. In: International Workshop on Document Analysis Systems, pp. 255–270 (1998)
13. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
14. Li, M., Cui, L., Huang, S., Wei, F., Zhou, M., Li, Z.: Tablebank: table benchmark for image-based table detection and recognition. In: Language Resources and Evaluation Conference, pp. 1918–1925 (2020)
15. Nair, V., Hinton, G.E.: Rectified linear units improve restricted Boltzmann machines. In: International Conference on International Conference on Machine Learning, pp. 807–814 (2010)
16. Paliwal, S.S., Vishwanath, D., Rahul, R., Sharma, M., Vig, L.: TableNet: deep learning model for end-to-end table detection and tabular data extraction from scanned document images. In: International Conference on Document Analysis and Recognition, pp. 128–133 (2019)
17. Paszke, A., et al.: Automatic differentiation in PyTorch (2017)
18. Prasad, D., Gadpal, A., Kapadni, K., Visave, M., Sultanpure, K.: CascadeTabNet: an approach for end to end table detection and structure recognition from image-based documents. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pp. 572–573 (2020)
19. Qasim, S.R., Mahmood, H., Shafait, F.: Rethinking table recognition using graph neural networks. In: International Conference on Document Analysis and Recognition, pp. 142–147 (2019)
20. Schreiber, S., Agne, S., Wolf, I., Dengel, A., Ahmed, S.: DeepDesrt: deep learning for detection and structure recognition of tables in document images. In: International Conference on Document Analysis and Recognition, vol. 1, pp. 1162–1167 (2017)
21. Shigarov, A., Mikhailov, A., Altaev, A.: Configurable table structure recognition in untagged pdf documents. In: ACM Symposium on Document Engineering, pp. 119–122 (2016)
22. Siddiqui, S.A., Khan, P.I., Dengel, A., Ahmed, S.: Rethinking semantic segmentation for table structure recognition in documents. In: International Conference on Document Analysis and Recognition, pp. 1397–1402 (2019)
23. Siddiqui, S.A., Malik, M.I., Agne, S., Dengel, A., Ahmed, S.: DeCNT: deep deformable CNN for table detection, vol. 6, pp. 74 151–74 161 (2018)
24. e Silva, A.C., Jorge, A.M., Torgo, L.: Design of an end-to-end method to extract information from tables. Int. J. Doc. Anal. Recogn. (IJDAR) **8**(2–3), 144–171 (2006)
25. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
26. Smith, R.: An overview of the Tesseract OCR engine. In: International Conference on Document Analysis and Recognition, vol. 2, pp. 629–633 (2007)

27. Wang, Y., Phillips, I.T., Haralick, R.M.: Table structure understanding and its performance evaluation. Pattern Recogn. **37**(7), 1479–1497 (2004)
28. Wang, Y., Phillips, I., Haralick, R.: Automatic table ground truth generation and a background-analysis-based table structure extraction method. In: International Conference on Document Analysis and Recognition, pp. 528–532 (2001)
29. Zanibbi, R., Blostein, D., Cordy, J.R.: A survey of table recognition. Doc. Anal. Recogn. **7**(1), 1–16 (2004)