Table Detection in Invoice Documents by Graph Neural Networks

Pau Riba*, Anjan Dutta*, Lutz Goldmann[†], Alicia Fornés*, Oriol Ramos*, Josep Lladós*

*Computer Vision Center, Universitat Autònoma de Barcelona, Barcelona, Spain

[†]omni:us GmbH, Berlin, Germany

Email: {priba, adutta, afornes, oriolrt, josep}@cvc.uab.cat, lutz@omnius.com

Abstract—Tabular structures in documents offer a complementary dimension to the raw textual data, representing logical or quantitative relationships among pieces of information. In digital mail room applications, where a large amount of administrative documents must be processed with reasonable accuracy, the detection and interpretation of tables is crucial. Table recognition has gained interest in document image analysis, in particular in unconstrained formats (absence of rule lines, unknown information of rows and columns). In this work, we propose a graph-based approach for detecting tables in document images. Instead of using the raw content (recognized text), we make use of the location, context and content type, thus it is purely a structure perception approach, not dependent on the language and the quality of the text reading. Our framework makes use of Graph Neural Networks (GNNs) in order to describe the local repetitive structural information of tables in invoice documents. Our proposed model has been experimentally validated in two invoice datasets and achieved encouraging results. Additionally, due to the scarcity of benchmark datasets for this task, we have contributed to the community a novel dataset derived from the RVL-CDIP invoice data. It will be publicly released to facilitate future research.

Keywords-Table Detection, Administrative Documents, Graph Representations, Geometric Deep Learning, Graph Neural Network

I. INTRODUCTION

Extracting information from administrative documents in digital mailroom processes is a common task in various domains including finance, insurance, manufacturing, and trading. The manual extraction of the relevant information is often a tedious and time consuming process. The ultimate goal of the automatic extraction is to reduce the manual effort and speed up the overall process. For forms and structured documents with simple named entities (e.g. names, dates, prices), the existing extraction methods can already achieve a high accuracy. For unstructured documents and challenging content (e.g. addresses, tables, details), the automatic extraction is not yet good enough and still requires human assistance and validation.

Administrative documents are very often semi-structured, without a fixed layout, but sharing a common set of components (e.g. header, footer, sender, recipient). This spatial arrangement can be roughly perceived as a tabular layout. On the other hand, one part of the documents is often a table. We refer to tabular layouts in the broadest sense, i.e. infor-

mation terms organized in a two-dimensional arrangement where some perceptual organization rules (horizontal and vertical alignment, repetitivity) dominate. Such organization of information offers a complementary dimension to the plain document contents, showing logical or quantitative relationships among pieces of information.

Tabular layouts have been over centuries one of the main instruments to communicate ideas through documents. Tables are graphical structures that visually show relationships among named entities, giving rich semantic messages beyond the basic literality of the constituent terms. Structured layouts are present in documents of different types and time periods. Parishes have been registering over centuries birth, marriage and death events in manuscripts structured in tabular arrangements. In 1869 the Russian chemist Dmitri Mendeleev presented in a manuscript a system to arrange the chemical elements. The periodic table has become an icon of science and culture (the United Nations has declared 2019 as the International Year of the Periodic Table). The public administration, due to socio-political models based on the rule of law introduced in the 19th century, had to develop mechanisms for collecting and quantifying the composition of the population and sources of wealth. Civil and notarial documents like census, tax, election records among many others, all of them mainly in tabular form, proliferated as information collection documents. As a consequence, nowadays administrative documents are rich in tabular layouts.

A key observation is that humans, when reading, perceive tables because of the observation of repetitive patterns. The Gestalt principles of visual perception [20] can be applied: table items have a regular arrangement, with a continuity in horizontal and vertical directions. In this work we propose a Graph Neural Network (GNN) approach for the detection and interpretation of tabular structures. Due to their representational power, graphs are suitable models for structured layouts. Graph nodes represent segmented text regions (isolated words or text lines). Graph edges are inferred in terms of visibility relations. Figure 1 illustrates the representation of a document in terms of an attributed visibility graph. When a table is present in the document, due to the repetitivity principle, its corresponding graph can be decomposed into a set of repeated and connected graphlets (small induced subgraphs). Therefore, the detection of tables is formulated in terms of a frequent graphlet

discovery algorithm. Since these local structures have a high variability, a supervised learning method is required, training the system with a priori knowledge about the graphlets that correspond to table parts. GNNs offer a solid foundation to achieve this objective by extending the formalism of Convolutional Neural Networks (CNNs) to graph domains, utilizing the principle of graph diffusion [28]. Therefore, the graph convolutions applied to the nodes can be seen as a way of constructing a graph node embedding that encodes the context of the node. Figure 1 shows this (middle figure). A graph convolution embeds into a node a combination of the information of the neighbouring nodes and edges. Then, after each convolution, nodes structurally similar tend to receive a similar encoding (similar colors in the nodes of the Figure 1 right). Our method classifies nodes as output of the GNN.

In summary, the main contributions of this work are: (1) A **GNN model** for tabular layout detection in administrative documents based on the classification of graph node embeddings. It is not constrained to rigid tabular layouts in terms of single rows, columns or presence of rule lines. (2) The **graph edge information** is considered in the node convolutions (related works do only consider node attributes). It boosts the structural knowledge in the computation of the feature embeddings. (3) The proposed model is **language independent**, i.e. although an OCR is used as preprocessing, only character type attributes are considered, but not the transcription of the OCR. This has the side advantage that privacy can can be preserved if required. Administrative documents contain sensitive information (e.g. names, addresses, bank details, health information...) A privacy policy may impose limitations with regards to the document handling that may prevent the document to be sent to a cloud service. Since we do not rely on the document contents, text recognition could be run locally, and only the graph should be used in a cloud service. (4) A **dataset**[1] consisting of 518 invoice pages from RVL-CDIP [9] dataset augmented with ground truth for table detection and layout analysis has been created and publicly released.

The rest of the paper is organized as follows. Section II reviews the state of the art in this field, whereas section III describes the system architecture. section IV evaluates our approach on several datasets. Finally, section V draws conclusions and outlines future research directions.

II. RELATED WORK

A. Graph Neural Networks

Geometric deep learning[2] [3], [15] refers to deep learning algorithms that apply neural networks to non-Euclidean domains, such as *graphs* and *manifolds*. In particular, Graph Neural Networks (GNN) have been coined to refer to neural networks applied to graph-structures. These algorithms

[1] Available at: https://zenodo.org/record/3257319

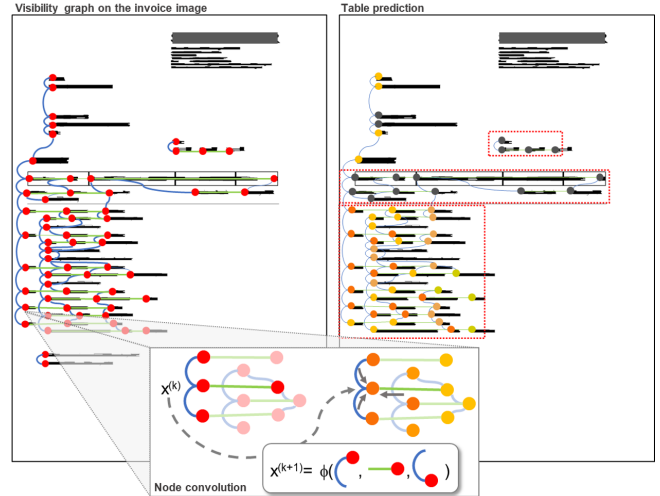[2] www.geometricdeeplearning.com

Figure 1. Graph representation of an invoice (left). Graph convolution idea (middle). Similar embeddings in table nodes after convolutions (right).

allow to learn representations at node, edge and graph level considering the underlying topological information. Depending on the elementary architecture, GNN methods are roughly divided into two groups: *spatial* and *spectral* methods. Spatial methods extend the idea of Convolutional Neural Networks (CNNs) for images and define a set of operations involving the local neighbourhood to compute a new representation [4], [18]. On other hand, spectral methods take the advantage of spectral graph theory [26] and consider graph Laplacians for defining convolution operations in graph domain [3], [15]. Gilmer *et al.* [8] generalized the above two domains of GNN, and defined most of the existing methods in terms of a message passing pipeline. These basic architectures are further extended to new tasks involving graphs, such as variational graph autoencoder [14], learning graph edit distance between a pair of graphs [23], graph matching [29] etc.

B. Table Detection

As mentioned before table detection is the first step for table recognition and the approaches can be grouped according to the type of the input documents (spreadsheets, textual documents). Here we focus on the latter which is more challenging due to the lack of explicit row and column information.

Table detection and recognition in unconstrained documents is considered a challenging task and recently has received significant attention within the community [2], [7], [11], [16], [19], [22], [24]. Available OCR systems only provide textual information without considering the actual tabular structures that exist in a document. However, recognizing tabular structures is crucial for getting a contextual meaning of the recognized textual information, which acted as the main motivation behind this research line. Early works

on this topic were mostly bottom-up in nature [1], [12], [13], [27], and they often start by detecting words or parallel lines following some heuristic to group homogeneous elements to detect tabular components (*i.e.* rows and columns), and hence tables in a document. As a consequence, most of the methods do not work well on multi-column document images for making some simplifying assumptions [25]. Later, Ghanmi and Belaïd [6] proposed Conditional Random Field (CRF) to localize tabular components in unconstrained handwritten documents. However, the recent developments on table detection are focused on the current progresses of deep learning techniques. Among them, Gilani *et al.* [7] proposed a variant of region proposal network where they feed pre-processed document images for detecting tables. A similar approach based on a region proposal network is also proposed in DeepDeSRT [24] for detecting tables, they further extended it to rows and column detection. In [22], Rashid *et al.* used a pre-trained neural network model to distinguish whether a word belongs to a table or not, and depending on the outcome applied some post-processing techniques for detecting tables. A saliency based fully connected neural network performing multi-scale reasoning on visual cues followed by a fully connected CRF for localizing tables and charts is proposed by Kavasdis *et al.* [11]. In [2], Clinchant *et al.* proposed two graph-based methods and compared them for the table detection task, where the first method relies on graph Conditional Random Fields (gCRF) [17], while the second method is based on Graph Convolutional Networks (GCN) [15]. At the same time, Koci *et al.* [16] proposed another graph-based method, where instead of a GCN, they proposed the remove and conquer algorithm for detecting tables. Very recently, Lohani *et al.* [19] proposed the same GCN-based technique for recognizing different fields in an invoice.
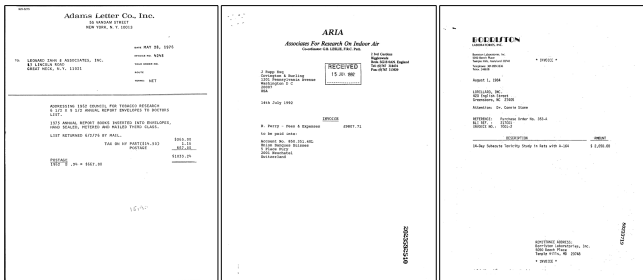

Figure 2. Example of invoices taken from the RVL-CDIP dataset showing the appearance variabilities in tables.

III. TABLE DETECTION FRAMEWORK

Tables are complex document entities composed of different elements (headers, rows, columns, etc.). These elements are distributed on document pages following repetitive structures. When dealing with structured data, we propose to use the high representation power of graphs to discover these repetitive patterns characterizing the tabular structure. However, this repetitive structures are not consistent among different documents and require a priori knowledge to be able to identify them as tables. Figure 2 shows some examples of these documents. With the aim of obtaining a priori knowledge from examples, data driven techniques such as deep learning provide an efficient tool to learn the key properties to deal with these complex documents.

In this work, we propose a GNN architecture, which is trained in a supervised manner knowing which is the corresponding class of each node, as well as if an edge is part of the same region. Thus, we formulate the problem of table detection as a classification problem by learning how entities are related. In this setting, Cross entropy is the proposed objective function which is optimized by Stochastic Gradient Descent (SGD) with momentum and weight decay for parameter estimation.

A. Graph Representation

Given a document image, we apply physical layout techniques to detect graphical and textual regions. Additionally, we apply a commercial OCR tool to encode textual attributes (numeric, alphabet or symbol), but not the recognized text. Given an invoice document, we represent each detected entity with the available information that can be used keeping the anonymity of the document. In this case, each entity corresponds to a 7-dimensional vector containing the bounding box position and a histogram of its content (numeric, alphabet or symbol). This encoded information is the one that will be used for table detection. From this set of entities, we generate a visibility graph in order to represent the structural information of the document.

Let $G = (V, E)$ be such a *visibility graph*. The set of nodes V corresponds to the detected entities of the document. The set of edges E represents visibility relations between nodes. Two entities are connected with an edge if and only if the bounding boxes are vertically or horizontally visible, *i.e.* a straight horizontal or vertical line can be traced between the bounding box of two entities without crossing any other. It is enough to take these two directions to check the visibility since it follows the way which tables are usually organized in documents. Finally, long edges covering more than a quarter of the page height are discarded.

B. GNN Architecture

Figure 3 provides an overview of the whole architecture, once the document has been processed and the graph has been generated. Firstly, a learned embedding layer makes use of a linear mapping (*i.e.* fully-connected layer) to project our 7-dimensional node input space to a higher order space encoding individual node features. This new embedding will be used in the following layers to share information between neighbour nodes. Message passing algorithms, in this case graph residual blocks, are used to propagate information.
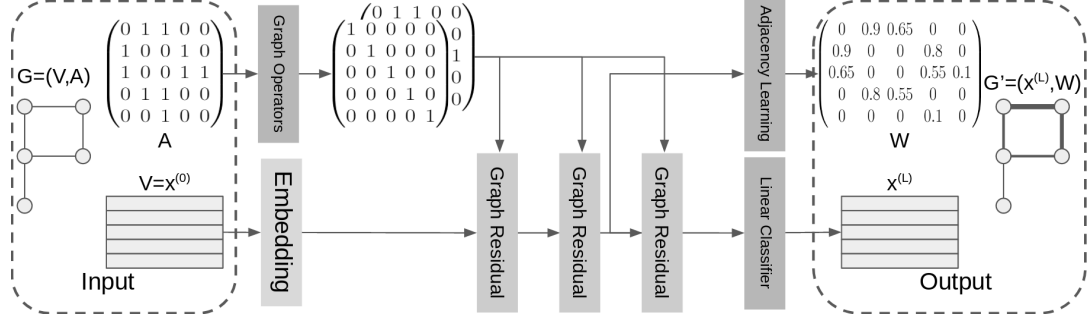
Figure 3. Overview of the proposed table detection framework. **Input**: visibility graph of document entities. **Output**: document entity labels and weighted adjacency matrix. The weights of the output adjacency matrix W are the probability of each pair of adjacent nodes to belong to the same document entity. Our framework is composed of an Embedding layer, 3 Graph Residual Blocks and 2 classifiers for nodes and edges respectively.
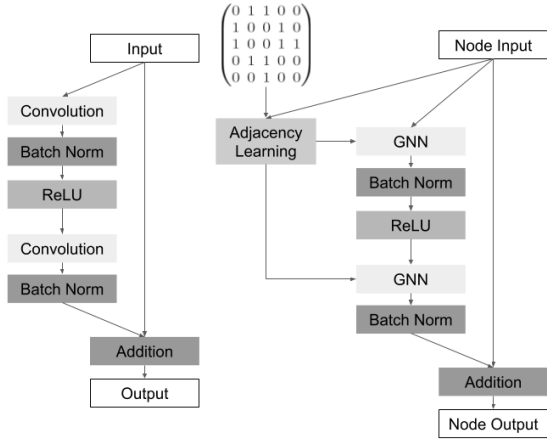

Figure 4. Residual Block layer: **Left**: Residual Block used in image recognition. **Right**: Graph Residual Block. The CNN layer is replaced by a GNN layer. The graph adjacency layer learns the weights of the adjacency matrices $W^{(k)}$ for the residual block. In case of size mismatch between input and output, a linear layer is used in the skip connection.

Graph Residual Block: A graph residual block follows the usual architecture of a residual block introduced by He *et al.* [10] but replacing the CNN layers by GNN layers. Figure 4 shows the proposed Graph Residual Block and a comparison with the corresponding block in a CNN. The proposed block, consists of two GNN layers with a non-linearity in between as well as a residual connection. However, in the specific case of graphs, edge weights are learned according to the node features at the beginning of the block and are used in both GNN layers. Moreover, the residual connection contains an optional linear mapping in case of size changes at the output of the block.

GNN Layer: Following the notation introduced in [5], let us consider an input signal $x^{(k)} \in \mathbb{R}^{|V| \times d_k}$ on the vertices of the given graph G and a set of graph intrinsic linear operators $\mathcal{A}^{(k)}$ both at the k layer of our network. In our approach we consider the powers of *adjacency operator* which encodes the j-hop neighborhoods into the graph.

Hence, $\mathcal{A}^{(k)} = \{A^j\}_{j=0...n}$, where A is the adjacency matrix. In this setting, the GNN layer, $G_C(\cdot)$ that produces a signal $x^{(k+1)} \in \mathbb{R}^{V \times d_{k+1}}$ is defined as:

$$x^{(k+1)} = G_C(x^{(k)}) = \rho \left(\sum_{B \in \mathcal{A}^{(k)}} B x^{(k)} \theta_B^{(k)} \right) \quad (1)$$

where $\theta_B^{(k)} \in \mathbb{R}^{d_k \times d_{k+1}}$ are trainable parameters and $\rho(\cdot)$ is a Rectified Linear Unit (ReLU) [21].

Graph Adjacency Layer: Note that in the previous definition, the graph operators $\mathcal{A}^{(k)}$ do not depend on the layer we are at, therefore, the importance of the neighbourhood connection would be the same across the whole network. Following the idea introduced in [5], we allowed our network to learn the weights of our operators at layer k. Hence, we define our set of graph intrinsic linear operators $\mathcal{A}^{(k)} = \{\phi_k(A^j)\}_{j=0...n}$ where ϕ_k is defined as,

$$\phi_k(B)_{i,j} = \begin{cases} 0 & \text{if } B_{i,j} = 0 \\ \sigma \left(\mathrm{MLP}_{\tilde{\theta}} \left(\left| x_i^{(k)} - x_j^{(k)} \right| \right) \right) & \text{otherwise} \end{cases} \quad (2)$$

where B is a power of the adjacency matrix, σ is the sigmoid non-linearity and MLP stands for a Multi Layer Perceptron with learnable parameters $\tilde{\theta}$ and ReLU activation function.

This definition allows our framework to learn the importance of the neighbourhood connections up to j-hops making use of the initial adjacency matrix and the hidden states of the nodes at layer k. The absolute difference between adjacent nodes vectors provides our system with the symmetry property, therefore, we are always in the setting of an undirected graph.

Final classifier: The information of the output signal $x^{(k)}$ returned by the last graph residual block, is used to feed two classifiers for nodes and edges respectively. On one hand, node classification makes use of a linear classifier *i.e.* fully connected layer with a Softmax operation. Note that during training additional classes such as *header* or *address* are considered, even though the initial focus is on table detection. On the other hand, a Graph Adjacency Layer

is used as an edge classifier. The learned adjacency matrix is updated one last time expecting to distinguish edges connecting nodes inside the same region, labelled with a 1 in the ground-truth against these edges that connect two different regions, labelled as 0.

C. Table Detection

Once the nodes have been classified a grouping algorithm is applied. The edge classifier allows our system to discard connections between nodes that are not likely to belong to the same region. As a result, connected subgraphs where the nodes have been classified with the same label can be extracted. Those connected components where the nodes are classified as a table with a good average confidence are then grouped and proposed as a predicted table. Two thresholds are used, the first one to decide whether or not to discard edges and the second one to decide whether or not to detect a table. Small connected components are also discarded.

IV. EXPERIMENTAL VALIDATION

A. Datasets

CON-ANONYM This is a particular dataset of 960 documents which has been used as part of an industrial collaboration. The documents are annotated with 8 regions including the following *address*, *table*, *header*.

RVL-CDIP The original RVL-CDIP dataset [9] contains $400,000$ grayscale images, which are divided in 16 classes with $25,000$ images per class. For the evaluation of our framework, we selected 518 images from the *invoice* class, which have been annotated with 5 regions including the same as above *table*, *address*, *header*.

For the sake of keeping anonymity, we apply ABBYY OCR on both datasets for extracting the text and encode it into a sequence of attribute types. For example, 'NNS' would encode '24$' and 'AAAAAA' would encode 'Google', where 'A', 'N', 'S' respectively denote the type of alphabetical, numeric and special character. In addition to the attribute type, we also keep the bounding boxes of each word that appears in these documents. Table I shows a comparison of both datasets. Note that in both datasets there are pages without tables.

Table I
SUMMARY OF THE DATASETS STATISTICS AS WELL AS THE PROPOSED DIVISION IN TRAIN, VALIDATION AND TEST SETS.

	CON-ANONYM	RVL-CDIP
Total # documents (tr, va, te)	950 (665, 95, 195)	518 (362, 52, 104)
Total # pages	1252	518
Total # tables	1202	485
Total # classes	8	6
Avg. # nodes/page	245.50	124.03
Avg. # edges/page	1354.81	619.55

B. Node and edge classification

Node and edge classification are the first step in the pipeline. In particular, node labels define our candidate regions whereas edges help to refine the final detection.
Node classification: Given an invoice, the nodes are classified into several classes such as *header*, *supplier*. Considering all these classes provides information about the global layout of the page that is helpful for table classification. For instance, a *table* will be always be below a *header*. We have experimentally observed that training a binary classifier does not lead to better results. Table II shows a comparison in terms of node classification accuracy. We also provide the accuracy taking into account only *table* nodes during test. In these experiments, we make use of 2 and 5 adjacency operators (denoted as pow 2 and 5 respectively) being defined as $\{A^j\}_{j=\{0..N\}}$, where A is the adjacency matrix of the input graph.
Edge classification: Our method may have problems when dealing with border conditions. Nodes that are connected to the *table* region will share some topological features with the *table* nodes. To alleviate this problem, the binary edge classifier enforces this separation by distinguishing the edges that connect two regions. Table II shows that adding the edge information in our model slightly increases the overall performance of our system.

Table II
ACCURACY RESULTS: **ALL:** GENERIC NODE CLASSIFICATION; **TABLE:** BINARY CLASSIFICATION OF NODES (TABLE OR NOT); **EDGE:** BINARY CLASSIFICATION OF EDGES.

Task	CON-ANONYM			RVL-CDIP		
	All	Table	Edge	All	Table	Edge
Pow 2	82.8	96.4	—	57.8	80.9	—
+ Edge	84.2	97.0	93.4	58.2	79.1	84.1
Pow 5	82.7	96.2	—	56.5	82.3	—
+ Edge	84.5	97.2	93.4	62.3	83.9	84.0

C. Table detection

Finally, we validate our framework on the table detection task. Table III shows the F1-score, Precision and Recall of the proposed framework in both benchmarks. The detection task is assessed by *intersection over union* (IoU) of the detected bounding box against the ground truth ones, which can be expressed as $\frac{D \cap G}{D \cup G}$, where D and G are the bounding boxes of the detected and the ground truth regions, respectively. Following the paradigm of object detection, we consider a correctly detected table if the IoU is higher than 0.5. We can observe that by means of the powers of the adjacency matrix, the receptive field increases and the system is able to detect better the table entities. Moreover, the use of edges is more relevant at this stage. Note that in the case of RVL-CDIP, there is a small amount of training data and the variability is much higher, hence, making this the more challenging dataset which is reflected in the results.

Finally, Table IV provides an idea of the edge removal threshold. Note that we make use of connected components to segment the tables, hence we can only remove these edges which are really not important. Otherwise a bad classified edge can introduce a lot of noise. In both datasets it is shown that the best threshold is 0.1 which increases the precision but losing a bit of performance in terms of recall.

Table III
TABLE DETECTION VALIDATION. PREDICTED EDGES WITH CONFIDENCE LESS THAN 0.1 HAVE BEEN REMOVED.

Task	CON-ANONYM			RVL-CDIP		
	F1-Score	Precision	Recall	F1-Score	Precision	Recall
Pow 2	69.4	65.8	73.4	28.6	23.9	35.4
+ Edge	70.8	65.2	77.6	30.8	26.7	36.5
Pow 5	68.4	65.3	71.8	22.6	20.0	26.0
+ Edge	73.7	78.4	69.5	30.8	25.2	39.6

Table IV
STUDY ON THE EDGE CONFIDENCE THRESHOLD.

Thresh.	CON-ANONYM			RVL-CDIP		
	F1-Score	Precision	Recall	F1-Score	Precision	Recall
0.0	68.2	64.7	72.2	27.1	23.3	32.3
0.1	73.7	78.4	69.5	30.8	25.2	39.6
0.2	72.3	68.3	76.8	29.5	23.2	40.6
0.3	72.6	67.0	79.3	27.0	20.5	39.6
0.4	72.3	66.2	79.7	21.3	15.6	33.3
0.5	68.1	59.9	78.8	21.5	15.2	36.5

V. CONCLUSIONS AND FUTURE WORK

This paper has presented the first table detection method based purely on structural information without making use of the raw content of the text. The underlying structure of the document is modeled as a graph, then, table detection is solved as a node classification problem where local node configurations characterize the table structure. Graph Neural Networks provide the tools to discover the local structures from nodes belonging to tables. Additionally, we have contributed to the community a novel dataset derived from the RVL-CDIP invoice data. Moreover, our approach is able to deal with anonymized data, because it does not use the textual contents. While most existing works on table detection do not consider anonymization (a big concern for companies when dealing with sensitive content as in the case of invoices), our method has demonstrated to overcome this limitation. For the future, we will study if this simple but practical architecture is robust enough to be generalized to other unconstrained tabular layouts.

ACKNOWLEDGMENTS

REFERENCES

[1] F. Cesarini, S. Marinai, L. Sarti, and G. Soda, "Trainable table location in document images," in *ICPR*, 2002.
[2] S. Clinchant, H. Djean, J. Meunier, E. M. Lang, and F. Kleber, "Comparing machine learning approaches for table recognition in historical register books," in *DAS*, 2018.
[3] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *NeurIPS*, 2016.
[4] D. K. Duvenaud, D. Maclaurin, J. Iparraguirre, R. Bombarell, T. Hirzel, A. Aspuru-Guzik, and R. P. Adams, "Convolutional networks on graphs for learning molecular fingerprints," in *NeurIPS*, 2015.
[5] V. Garcia and J. Bruna, "Few-shot learning with graph neural networks," in *ICLR*, 2018.
[6] N. Ghanmi and A. Belaid, "Table detection in handwritten chemistry documents using conditional random fields," in *ICFHR*, 2014.
[7] A. Gilani, S. R. Qasim, I. Malik, and F. Shafait, "Table detection using deep learning," in *ICDAR*, 2017.
[8] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural message passing for quantum chemistry," in *ICML*, 2017.
[9] A. W. Harley, A. Ufkes, and K. G. Derpanis, "Evaluation of deep convolutional nets for document image classification and retrieval," in *ICDAR*, 2015.
[10] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016.
[11] I. Kavasidis, S. Palazzo, C. Spampinato, C. Pino, D. Giordano, D. Giuffrida, and P. Messina, "A saliency-based convolutional neural network for table and chart detection in digitized documents," *arXiv*, 2018.
[12] T. Kieninger and A. Dengel, "Applying the t-recs table recognition system to the business letter domain," in *ICDAR*, 2001.
[13] T. Kieninger and A. Dengel, "Table recognition and labeling using intrinsic layout features," in *ICAPR*, 1999.
[14] T. N. Kipf and M. Welling, "Variational graph auto-encoders," *NeurIPS BDL*, 2016.
[15] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *ICLR*, 2017.
[16] E. Koci, M. Thiele, W. Lehner, and O. Romero, "Table recognition in spreadsheets via a graph representation," in *DAS*, 2018.
[17] J. D. Lafferty, A. McCallum, and F. C. N. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *ICML*, 2001.
[18] Y. Li, D. Tarlow, M. Brockschmidt, and R. S. Zemel, "Gated graph sequence neural networks," in *ICLR*, 2016.
[19] D. Lohani, A. Belaïd, and Y. Belaïd, "An invoice reading system using a graph convolutional network," in *IWRR*, 2018.
[20] D. G. Lowe, *Perceptual Organization and Visual Recognition*. Norwell, MA, USA: Kluwer Academic Publishers, 1985.
[21] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *ICML WDLASL*, 2013.
[22] S. F. Rashid, A. Akmal, M. Adnan, A. A. Aslam, and A. Dengel, "Table recognition in heterogeneous documents using machine learning," in *ICDAR*, 2017.
[23] P. Riba, A. Fischer, J. Lladós, and A. Fornés, "Learning graph distances with message passing neural networks," in *ICPR*, 2018.
[24] 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 *ICDAR*, 2017.
[25] F. Shafait and R. Smith, "Table detection in heterogeneous documents," in *DAS*, 2010.
[26] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE SPM*, vol. 30, no. 3, 2013.
[27] Y. Wang, I. T. Phillipst, and R. Haralick, "Automatic table ground truth generation and a background-analysis-based table structure extraction method," in *ICDAR*, 2001.
[28] X. Yang, L. Prasad, and L. Latecki, "Affinity learning with diffusion on tensor product graph," *IEEE TPAMI*, vol. 35, no. 1, 2013.
[29] A. Zanfir and C. Sminchisescu, "Deep learning of graph matching," in *CVPR*, 2018.