Graph Attention Networks for Efficient Text Line Detection on Receipt-Layout Documents*

ABSTRACT

Text line detection from OCR detections is an essential step in many information-extraction processes, particularly when working with unstructured documents such as purchase receipts, where utilizing this information is crucial for matching key-value pairs that are on the same line. Existing models, however, are limited to structured documents and do not generalize well to unstructured ones. To address this issue, we have created a GNN-based line detection model that is optimized for receipt-layout documents. Experiments show that the proposed method outperforms other approaches in accuracy, processing time and resource consumption.

1 INTRODUCTION

In recent years, information extraction from documents has gained prominence. Advances in Natural Language Processing and Computer Vision facilitate the creation of Document Intelligence systems that can outperform human workers in this task [4, 9, 20].

Most of these systems use an OCR engine to extract individual text segments (typically at the word level) and then a layout extraction stage to detect lines, columns, or paragraphs in the document. In unstructured documents, such as purchase receipts, where key-value pairs are typically separated by tabs or broad spaces and printed on the same line, line detection is crucial for comprehension.

Multiple line detection models have been created in recent years in an attempt to handle this problem accurately and effectively. [6, 12, 14, 19]. In fact, the majority of OCR engines provide line predictions alongside text segment information, although their precision is restricted. Other models work directly over the images and provide line-level polygon detections [6, 12], obviating the requirement for an OCR engine, but are inapplicable to systems that require word-level information.

Recently, new approaches have been developed to group by lines the OCR engine's recognized segments. In addition, several Graph Neural Networks (GNNs) are built on this method, which has been proved to be effective for link prediction tasks. [6, 14]. However, these models are only applicable to structured documents, where the layouts are constant and some assumptions such as the inter-word distance can be done (for instance scientific papers or tabular documents). Thus, they do not generalize well to complex layouts. To the best of our knowledge, there is no published study focused on unstructured documents.

To meet this demand, we built a fast and accurate GNN-based line detection model for receipt-layout documents. Our work mainly contributes:

- Custom nodes features extracted from the rotated bounding boxes of the segments.
- Edge sampling strategy optimized for improving the connectivity on unstructured documents.
- Graph Attention Layers (GAT) based GNN architecture with residual connections and a global document node for enhanced performance.
- Optimized connected components algorithm for clustering the GNN predictions.

In the experiments, we show that our model outperforms other approaches in accuracy, processing time, and resources.

2 RELATED WORK

Multiple competitions and datasets have been published reflecting the increasing interest in text line detection. [5, 13, 22, 23]. We can distinguish two research lines: single-stage models based on image detection and/or segmentation, and models that operate over OCR detections, mostly based on GNNs.

2.1 Line Detection based on image segmentation

In [12], the authors offer a semantic segmentation-based approach to assess document layout, where a pixel-wise classification map is constructed for the layout and a task-dependent post-processing step generates a polygon bounding box from the binary map. Another image-segmentation-based approach for historical line recognition is [6]. A modified version of UNET [15], ARU-Net, classifies pixels as baseline, separator, etc. As a second stage, postprocessing steps are introduced to convert the labeled pixel maps into a polygon for the required baseline. In recent years, there are other relevant approaches [1–3, 7, 11]. Besides, in this category, we can also find models based purely on image detection, but they are more oriented towards tabulated formats, such as tables. This is the case of [17], authors offer a two-model technique. First predicts table grid pattern and splits cells. The second predicts which cells to merge for more complex layouts. Also, in [16], they use deformable convolutions to recognize specific regions in a table, similar to FasterRCNN.

2.2 Line detection based on GNNs

In [14], the authors present a GNN-based approach for recognizing cells, rows, and columns in tabular documents based on OCR detections. As input features, they employ image features derived by a CNN, positional features from the OCR bounding box, and segment length.After processing the nodes' features with the GNN, they sample pairs of nodes that could be connected for each task and apply edge prediction to determine the links between the segments. Another GNN-based model is presented in [19], but for line and paragraph detection in multi-column structured documents. From the OCR bounding box and line predictions, they first broke multi-paragraph lines and then clustered them into paragraphs. A different approach is followed in [10], where lines and paragraphs are detected at once. Starting with OCR bounding boxes, they use a GNN to enrich node features and then evaluate three types of connections: if the first box is above, below, or on the left of the second. All of these models are focused on locating relations over

*A patent has been applied for that covers the subject matter described in this article.

structured documents, where there are limits for word spacing, and are therefore unsuitable for unstructured text.

3 METHODOLOGY

3.1 Problem Definition

Given a list of text segments generated by an OCR engine from a receipt-layout document image (such as a purchase receipt), is to group the segments that belong to the same text line. The only available information for each segment is the text string and the rotating bounding box. The document depicted within the image may be slightly rotated and may contain flaws such as creases or ripples.The considered use case is illustrated in Fig. 1.

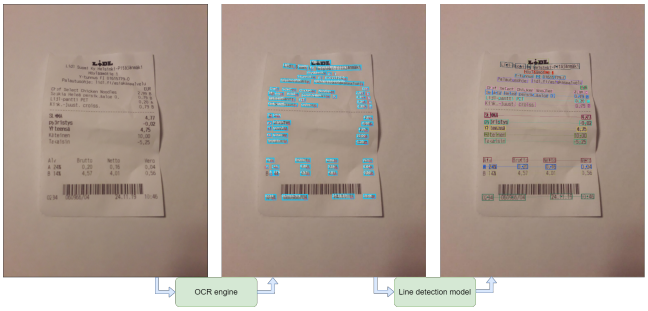

Figure 1: The left image shows an incoming purchase receipt. First, the OCR engine detects the text segments (blue boxes in the middle image). Next, the line detection model groups the segments by line (linked boxes in the right image).

We describe some of the inherent difficulties of the task. First, since we are dealing with unstructured texts, we cannot assume any distance limits between neighboring segments that belong to the same or adjacent lines. Additionally, the documents may exhibit some 3D rotation relative to the camera. This adds complexity not only to the edge sample phase, but also to the node feature selection, as more variables must be considered. In addition, the OCR module has flaws and inconsistencies, such as typographical errors in the detected text segment, noisy bounding boxes, repeated detections and missing detections.

3.2 Proposed Method

Our proposed method is based on Graph Neural Networks (GNNs), as shown in Fig. 2. Reasons include:

- The problem can be treated as a link prediction task in which the objective is to anticipate a relation between two (spatial) segments that should be linked together. GNNs work well for this task.
- Documents may have dozens or hundreds of segments. Methods based on fixed input sizes, like Fully Connected Neural Networks (FCNN), are not suited.
- The number of connections to evaluate can be limited based on the box coordinates, accelerating the inference.

3.2.1 Feature Extraction. From the two sources of information available for each segment (the bounding box and the text string), we discard the text information, as we believe there is no generalizable relationship between the text of two segments that can help to automatically determine if they are on the same line and it can produce overfitting. This way we also mitigate the impact of the text errors coming from the OCR.

Regarding the bounding box, we select the following features: left and right center coordinates, and the angle of the bounding box in radians, between -PI/2 and PI/2. Notice that using the left and right center we are losing the information related to the height of the bounding box. We do this on purpose, as we observed that the model tended to overfit using this feature. We normalize both centers using the width of the document, as it is the most stable dimension. Finally, the features are concatenated to generate the embedding (with 5 float values).

3.2.2 Edge Sampling. The GNN uses the edges to accomplish message passing, while the edge prediction head uses them to create the final predictions. Consequently, it is essential to choose a sampling function that includes all possible true positives. Due to the fact that we are working with extremely unstructured texts, we cannot rely on the typical sampling functions, such as k-nearest neighbor or beta-skeleton, ([10, 14, 19]), as they are prone to miss connections between distant segments.

Thus, we developed a custom sampling function to ensure that all the segments in the same line are connected: an edge from segment A to segment B is created if the vertical distance between their centers (C) is less than the height (H) of segment A by a constant (K) (Eq. 1). We set this K constant to two.

$$edge_{A-B} = |C_A^y - C_B^y| < H_A * K \tag{1}$$

3.2.3 GNN Architecture. Selecting the most appropriate type of layer is another important step in the model design. Most of the GNN layer implementations require an additional scores vector for performing a weighted message passing. It decides the contribution of each neighbor node. This adds complexity to network, as we need to develop a module for computing the weights In our case, the information needed for computing these weights is related to the box coordinates, already embedded in the node features. Thus, we select Graph Attention Layers (GAT) [18], as the weights for the message passing are computed directly inside the layer using the input node features. The proposed architecture is shown in (Fig. 3).

Another enhancement is the use of a global node, inspired by [21]. This node is connected bidirectionally to the rest and its feature embedding is computed by averaging all. It has a double function in the network: it provides some context information to the nodes, and it acts as a regularization term for the GAT layer weights.

3.2.4 Edge Prediction Head. After the node features have been passed through the network and enriched with the information from the neighbor nodes, for each pair of segments that are connected by an edge, we extract the confidence that they belong to the same line. For this task, we first concatenate the output features of both nodes and then process them with a MLP composed of two linear layers with an output size of 32 and 1, respectively. After the first layer, we apply another SiLU activation. Finally, we apply a sigmoid function after the last MLP layer to obtain the confidences.

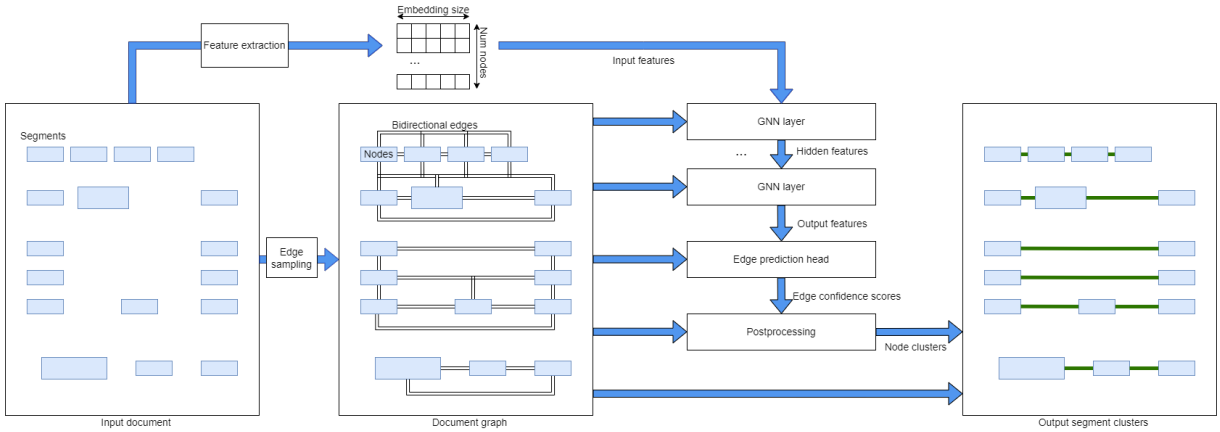

Figure 2: Overview of the proposed line detection model.

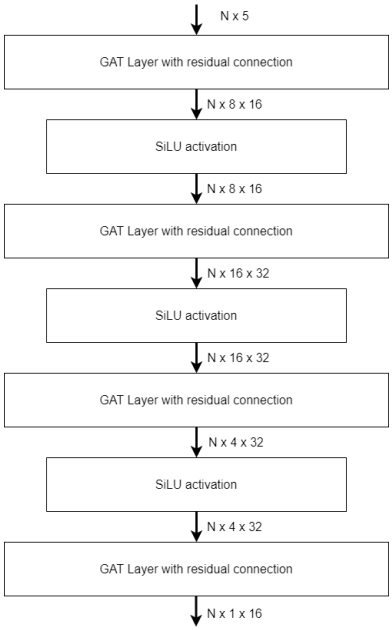

Figure 3: Proposed GNN architecture.

3.2.5 Postprocessing. The last stage consists of grouping the segments by line using the confidences matrix. First we binarize the matrix using a threshold computed with a grid search over the validation set. We considered applying the connected components algorithm but we noticed that in most of the cases where the model was erroneously connecting two segments, they were far away from each other. Thus, to reduce these errors, we present a modified version, called Limited Connected Components (LCC), so each segment can only be connected to the closest segment to his left and one to his right. Its contribution is reported in the experiments.

4 EXPERIMENTS

4.1 Datasets

4.1.1 Private Dataset. It consists of 2733 purchase receipt images from different countries. Receipts vary widely in height, density, and image quality. They may contain rotation and all kinds of wrinkles. Some examples are shown in Figure 4. The dataset also contains the receipt region annotation, so we preprocess the dataset for all the models by cropping the images and filtering the segments outside the receipt. We split the dataset into training, validation and test sets using a ratio of 70/20/10.

4.1.2 Public dataset: CORD. Consolidated Receipt Dataset (CORD) [13] is composed of 1000 Indonesian receipts which contains images and box/text annotations for OCR, and multi-level semantic labels. Although it is mainly used for semantic parsing tasks such as entity tagging or entity linking, it provides also line annotations. The dataset is split in train (800), validation (100), and test (100).

4.2 Metrics

4.2.1 Line F1 Score. This metric is very restrictive and aims at evaluating the number of lines that are perfectly formed, highly penalizing the rows that are split or merged with others (see Equation 2). For each predicted text line in a document, we only consider it as a true positive (tp) if it matches exactly the ground truth line.

$$\begin{aligned} precision &= tp/(tp+fp) \\ recall = tp/(tp+fn), tp+fn &= num\_GT\_lines \\ line\_f1\_score &= 2 * \frac{precision * recall}{precision + recall} \end{aligned} \tag{2}$$

4.2.2 ARI. The Adjusted Rand Index (ARI) [8], is more focused on analyzing the quality of the segment clusters rather than checking if they perfectly match the ground truth ones (Equation 3).

$$\begin{aligned} RI &= \frac{num\_agreeing\_pairs}{num\_total\_pairs} \\ ARI &= \frac{RI - expected\_RI}{max(RI) - expected\_RI} \end{aligned} \tag{3}$$

Table 1: Comparison across datasets, baseline models and the proposed method.

Model	Dataset					
	Private		CORD		CORD (priv)	
	F1	ARI	F1	ARI	F1	ARI
TIES [14]	0.934	0.946	0.935	0.959	0.922	0.933
dhSegment [12]	0.946	-	0.920	-	0.896	-
Ours (orig CC)	0.977	0.985	0.972	0.978	0.978	0.987
Ours (LCC)	**0.989**	**0.993**	**0.976**	**0.985**	**0.984**	**0.994**

4.3 Results

We have selected two state-of-the-art and publicly available models for benchmarking: dhSegment [12] based on image segmentation and TIES [14], based on OCR detections and GNN. For each dataset, we select the best weights after each epoch using the validation set. We also test on CORD the models trained on the private dataset to analyze their generalization capabilities.

The results collected in Table 1, show that the proposed method highly outperforms the others in both datasets and for all the metrics, regarding the quality of the clusters and the number of correct lines. These results demonstrate that using only features related to the bounding box is sufficient to solve this task, and that the image features (used in the TIES model) do not provide additional relevant information (at least embedded inside nodes). Furthermore, while the other methods perform worse in the CORD dataset when they are trained using the private one, our model achieves even better results, which demonstrates its capability of generalization on unseen data. Some examples of the performance of the proposed model on challenging samples from the private dataset are presented in Figure 4, with different sizes, lengths, text densities, rotations, and wrinkles, showing that it is able to overcome the challenges discussed in Section 3.1.

Regarding the time consumption, we measure the total time for processing the whole test set of the private dataset and then we calculate the average time per image. For a fair comparison, we also calculate the preprocessing time for all the models (including the graph construction and the feature extraction). All the tests were launched under the same conditions (same hardware in idle state and batch 1). The GPU used is an NVIDIA GTX 1080 TI. The results in Table 2, reveal again the overwhelming superiority of the proposed method, performing 5 times faster than the others. One of the main reasons for this difference is that our model does not use the image, avoiding its loading and preprocessing, and the inference of the image backbone for extracting the feature map.

Finally, we also analyze the GPU memory consumption for the three models. Table 3 shows the proposed model is much more efficient than the other approaches. Again, the main reason for this difference is that our model is not using an image backbone.

Figure 4: Examples of successfully detected lines on challenging cases from the private dataset

Table 2: Comparison of the processing time in seconds

Model	Processing time per sample (seconds)		
	Preprocess	Inference + postprocess	Total
TIES [14]	0.3987	0.2946	0.6933
dhSegment [12]	0.2774	0.3950	0.6724
Ours	**0.1168**	**0.0097**	**0.1265**

Table 3: Comparison of the GPU memory consumption in megabytes (MB)

Model	Memory consumption (MB)
TIES [14]	4911
dhSegment [12]	8075
Ours	**889**

5 CONCLUSIONS AND FUTURE WORK

In this work, we have addressed the problem of detecting text lines on receipt-layout documents using the detected OCR bounding boxes. After reviewing the state of the art, we have concluded that none of the current text line detection models based on OCR detections are suitable for unstructured documents. Thus, we have proposed a GNN-based model optimized for this specific use case. The capabilities and suitability of this model for the considered task have been demonstrated experimentally: it outperforms in accuracy, processing time and resource consumption.

Future work will focus on extending the capabilities of the model for detecting more complex structures, such as paragraphs or semantic entities. To this end new types of features will be considered, such as text or visual features. We believe also that combining the line detection task with other more complex ones can force the model to have a better understanding of the document layout and improve the results for all the tasks.

REFERENCES

[1] Michele Alberti, Lars Vögtlin, Vinaychandran Pondenkandath, Mathias Seuret, Rolf Ingold, and Marcus Liwicki. 2019. Labeling, cutting, grouping: an efficient text line segmentation method for medieval manuscripts. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*. IEEE, 1200–1206.

[2] Adi Azran, Alon Schclar, and Raid Saabni. 2021. Text line extraction using deep learning and minimal sub seams. In *Proceedings of the 21st ACM Symposium on Document Engineering*. 1–4.

[3] Berat Kurar Barakat, Ahmad Droby, Reem Alaasam, Boraq Madi, Irina Rabaev, Raed Shammes, and Jihad El-Sana. 2021. Unsupervised deep learning for text line segmentation. In *2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE, 2304–2311.

[4] Manuel Carbonell, Pau Riba, Mauricio Villegas, Alicia Fornés, and Josep Lladós. 2021. Named entity recognition and relation extraction with graph neural networks in semi structured documents. In *2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE, 9622–9627.

[5] Markus Diem, Florian Kleber, Stefan Fiel, Tobias Grüning, and Basilis Gatos. 2017. cbad: Icdar2017 competition on baseline detection. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, Vol. 1. IEEE, 1355–1360.

[6] Tobias Grüning, Gundram Leifert, Tobias Strauß, Johannes Michael, and Roger Labahn. 2019. A two-stage method for text line detection in historical documents. *International Journal on Document Analysis and Recognition (IJDAR)* 22, 3 (2019), 285–302.

[7] Tobias Gruuening, Gundram Leifert, Tobias Strauss, and Roger Labahn. 2017. A robust and binarization-free approach for text line detection in historical documents. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, Vol. 1. IEEE, 236–241.

[8] Maria Halkidi, Yannis Batistakis, and Michalis Vazirgiannis. 2002. Cluster validity methods: part I. *ACM Sigmod Record* 31, 2 (2002), 40–45.

[9] Teakgyu Hong, Donghyun Kim, Mingi Ji, Wonseok Hwang, Daehyun Nam, and Sungrae Park. 2021. BROS: A Pre-trained Language Model Focusing on Text and Layout for Better Key Information Extraction from Documents. *arXiv preprint arXiv:2108.04539* (2021).

[10] Shuang Liu, Renshen Wang, Michalis Raptis, and Yasuhisa Fujii. 2022. Unified Line and Paragraph Detection by Graph Convolutional Networks. In *International Workshop on Document Analysis Systems*. Springer, 33–47.

[11] Olfa Mechi, Maroua Mehri, Rolf Ingold, and Najoua Essoukri Ben Amara. 2019. Text line segmentation in historical document images using an adaptive U-Net architecture. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*. IEEE, 369–374.

[12] Sofia Ares Oliveira, Benoit Seguin, and Frederic Kaplan. 2018. dhSegment: A generic deep-learning approach for document segmentation. In *2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR)*. IEEE, 7–12.

[13] Seunghyun Park, Seung Shin, Bado Lee, Junyeop Lee, Jaeheung Surh, Minjoon Seo, and Hwalsuk Lee. 2019. CORD: a consolidated receipt dataset for post-OCR parsing. In *Workshop on Document Intelligence at NeurIPS 2019*.

[14] Shah Rukh Qasim, Hassan Mahmood, and Faisal Shafait. 2019. Rethinking table recognition using graph neural networks. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*. IEEE, 142–147.

[15] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*. Springer, 234–241.

[16] Shoaib Ahmed Siddiqui, Imran Ali Fateh, Syed Tahseen Raza Rizvi, Andreas Dengel, and Sheraz Ahmed. 2019. DeepTabStR: deep learning based table structure recognition. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*. IEEE, 1403–1409.

[17] Chris Tensmeyer, Vlad I Morariu, Brian Price, Scott Cohen, and Tony Martinez. 2019. Deep splitting and merging for table structure decomposition. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*. IEEE, 114–121.

[18] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903* (2017).

[19] Renshen Wang, Yasuhisa Fujii, and Ashok C Popat. 2022. Post-ocr paragraph recognition by graph convolutional networks. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 493–502.

[20] Zilong Wang, Mingjie Zhan, Xuebo Liu, and Ding Liang. 2020. DocStruct: a multimodal method to extract hierarchy structure in document for general form understanding. *arXiv preprint arXiv:2010.11685* (2020).

[21] Zhenrong Zhang, Jiefeng Ma, Jun Du, Licheng Wang, and Jianshu Zhang. 2022. Multimodal Pre-training Based on Graph Attention Network for Document Understanding. *arXiv preprint arXiv:2203.13530* (2022).

[22] Xu Zhong, Elaheh ShafieiBavani, and Antonio Jimeno Yepes. 2020. Image-based table recognition: data, model, and evaluation. In *European Conference on Computer Vision*. Springer, 564–580.

[23] Xu Zhong, Jianbin Tang, and Antonio Jimeno Yepes. 2019. Publaynet: largest dataset ever for document layout analysis. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*. IEEE, 1015–1022.