Efficient Document Image Classification Using Region-Based Graph Neural Network

Jaya Krishna Mandivarapu
jmandivarapu1@student.gsu.edu
American Family Insurance
Madison, Wisconsin, USA

Eric Bunch
ebunch@amfam.com
American Family Insurance
Madison, Wisconsin, USA

Qian You
qyou@amfam.com
American Family Insurance
Madison, Wisconsin, USA

Glenn Fung
gfung@amfam.com
American Family Insurance
Madison, Wisconsin, USA

ABSTRACT

Document image classification remains a popular research area because it can be commercialized in many enterprise applications across different industries. Recent advancements in large pre-trained computer vision and language models and graph neural networks has lent document image classification many tools. However using large pre-trained models usually requires substantial computing resources which could defeat the cost-saving advantages of automatic document image classification. In the paper we propose an efficient document image classification framework that uses graph convolution neural networks and incorporates textual, visual and layout information of the document.

CCS CONCEPTS

• **Applied computing → Document scanning**.

KEYWORDS

document intelligence, graph neural networks, document classification

1 INTRODUCTION

Gartner has estimated 80% of enterprises data is unstructured (emails, PDF and other documents). These documents contain rich information and knowledge about internal and external business communication and transactions. They also have ubiquitous applications in numerous industrial sectors such as finance, health care, and law etc. Therefore, being able to automatically and efficiently sort, analyze, and extract structure and content from document images can improve efficiency and reduce cost for many business workflows. Document image classification is an import task in these automation solutions, and has been a popular research area for decades. Early works usually build classifiers that rely on Optical Character Recognition (OCR) to extract text information, and employ heuristics to model layout structural features. In light of the advancement of computer vision and deep learning, VGG-16 [20] pre-trained on ImageNet [5] reported good classification performance on data sets mixed of business letters, print advertisement, emails and magazine articles [13]. Both [7] and [21] created document representations by encoding layout coordinates into positional embeddings as inputs to pre-trained BERT [8] or transformer architectures. The latest PubLayNet [24] addresses the limited public available document image data sets by training a Mask R-CNN [10] model on 360k images of scientific articles, and enables transfer learning to other document domains. Motivated by the development of graph neural network algorithms researchers [15] attempted to use graph convolutions to model the interactions among structural components of a document and between the visual and textual features, as an alternative to pixel level or token level document modeling. In contrast to fast moving research progress in document analysis and classification, few have systematically studied the time and hardware resources when using different methods and the financial implications of the model design. However, as document image classifications have been primarily motivated by its potential in commercialization, it is imperative to study its model performance with computing resources requirements and financial implications. In this paper we propose an efficient document image classification framework as shown in Fig. 1. Semantic regions of a document is extracted by pre-trained PubLayNet, textual features are extracted by text embedding models and the image features are extracted by a pre-trained VGG-16 model. Graphs formed for the document, with the document class labels are used to train a sort pooling graph convolution network [23] which normalizes and classify arbitrary graphs therefore documents. The major contributions of our papers are as follows:

- We propose a novel document image classification framework which applies a graph convolution neural network to a document image graph formed by semantic regions extracted from a pre-trained document segmentation model. Moreover both image and text features of the regions are extracted and assigned to the nodes so that information from both modalities are captured and propagated in the graph convolutions. To our best knowledge, our framework is the the first in effectively and economically integrating image, text, and layout information for document image classification using a graph convolution neural network.
- We have rigorously bench marked our proposed method against state-of-the-art pre-trained vision models and transformer language models on document image data sets. These include an insurance related document image data set consisted of 11 classes and an open source data set of 10 classes. The results showed the classification results of our method are comparable to those of baseline models, if not better.

Document Intelligence Workshop at KDD, 2021, Virtual Event
.

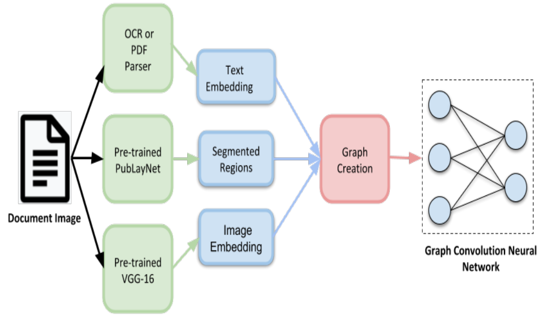


Figure 1: Eff-GNN Framework overview: textual embedding, segmented regions and image embeddings of an image are integrated when the graph of document is formed. Created graph is fed into the Graph Convolution Neural Network for graph classification as document image classification.

- We extensively bench marked the computing resources required by all methods. The results showed our framework needs substantially less computing resources and less time, further indicating the cost advantages of training, deployment and hosting at scale. Efficient model also helps accelerate model iterations and update.

We also discussed a few potential document image classification applications and the infrastructure to deploy our framework. The potentially large scale adoption of document image classification further reinforced the need for an efficient document image classification method.

2 RELATED WORK

Early document image classification algorithms relied on OCR to extract content information and exploited the visual structure and layout of a document image, e.g. using tree-related data structures to model a document [6, 9, 19]. Advancement in Deep Convolutional Neural Networks (DCNN) lend new tools for document image classification [11], because DCNN could extract salient and hierarchical visual feature representations which can somewhat reflect hierarchical nature of document layout. Quite a few DCNN training strategies for document image classification are proposed and extensively reviewed [2, 11]. Variants of VGG-16 [4] achieved the state-of-the art on publicly available Tobacco data sets [14]. Document understanding and analysis community have also been leveraging word embedding techniques [16] in NLP and large language models [8] to create contextualized embedding for textual content in an document image. BERTGrid [7] uses both the contextualized word embeddings and its 2D layout coordinates to extract information by predicting segmentation masks and bounding boxes. Assuming syntactic features matter less than content categories in document classification. DocBERT achieved an economical solution for document classification task by distilling BERT. DocBert [1] assumes syntax features matter less if only the categories of the document need to be decided. And DocBert successfully distills trained BERT into a much smaller LSTM model. This gives us some insight that token level modeling for document classification may not be necessary. LayoutLM [21] jointly models interactions between text and layout information by inputting both text embeddings along with its 2D layout positional embeddings extracted using OCR and Region of Interest (ROI) Regressions. Considering both the image and textual modalities in the document images, multi-modalities methods [3, 22] are adapted to document classification tasks as well.

3 METHODOLOGY

In this section, we briefly describe our proposed efficient graph neural network, **Eff-GNN**

3.1 Document Segmentation

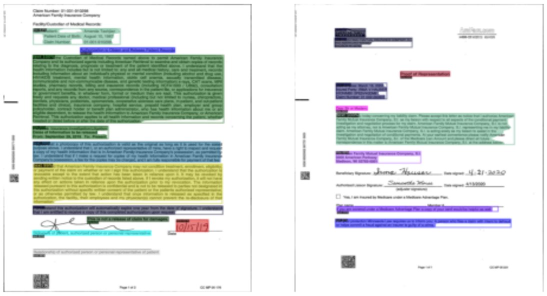
A business document contains visually salient structural components such as header, footer, paragraph, table etc. Intuitively one can classify a document image by its layout and structural components without accessing much of its content. The regions of structural components are usually pixels or tokens with similar appearances or groupings. Hence regions are a higher level abstraction and representation which we can leverage to classify this document. Research in the computer vision community has provided plenty of tools of segmenting document images. The latest advancement is PubLayNet which trained a Mask R-CNN model for 360 thousand document images from scientific articles. The segmented region results from PubLayNet can be found at Figure 2 (a) & (b).

3.2 Efficient GNN for Document Image Classification

Ideally an effective document classification method need to leverage both textual, image and layout information. However, training or fine tuning CNNs or large language models do not only run into resource constraints (e.g. GPUs, memory), but also prevent fast model iterations. We attempted to solve this problem by using graph representations to represent document. We then assign image and text features to the nodes of the graph and apply a graph convolution neural network. Finally we classify the document as classifying a graph.

Details of training our proposed **Eff-GNN** can be found in Algorithm 1. For each image with class label, we extract its text using OCR PyTesseract and we extract its semantic regions using PubLayNet.The segmented region results from PubLayNet can be found at Figure 2 (a) & (b). Each image is converted into a graph where each node is a region. To generate the text feature of the nodes, we use Word2Vec to create the embeddings for words in the region; to generate image feature of the nodes, we extract visual features from that region using VGG-16 pre-trained on ImageNet. Text features and image features of the node can be concatenated and assigned to the nodes. A graph convolution neural network classifier with a SortPooling [23] layer is then trained on this data. We adapted to this specific graph convolution neural network because it preserve features of individual nodes and also enforces learning from graph global topology.

Till this end we have integrated textual, image and layout information into a document classification task using a graph convolution neural network.



(a) Medical Authorization (b) Attorney Correspondence

Figure 2: Examples of documents from the Insurance data set. The document classes (a) Medical Authorization, and (b) Attorney Correspondence including the visualization of the output of the PubLayNet model.

class label	0	1	2	3	4	5	6	7	8	9	10
Insurance	13	5	6	18	5	12	3	11	8	13	11
Tobacco	5	9	3	11	10	5	2	9	10	2	-

Table 1: Median number of nodes per class for the Insurance and Tobacco data sets.

4 EXPERIMENTAL SETUP

4.1 Datasets

We use the following two datasets to evaluate our proposed model: Insurance dataset (Fig.2), Tobacco-3482 dataset[13]. Insurance dataset contains 5772 document images which spans across 11 categories and Tobacco-3482 dataset consists of 3482 images which spans across 10 categories. See Fig. 2 for examples of documents from the Insurance dataset.

For the Insurance dataset, we use the splits of 4544 images for train set and 1280 images for test set; for Tobacco-3482 dataset, we use the standard splits of 2482 images for training, 800 images for testing, and rest 200 images for validation set.

We also summarized the statistics of graphs that those documents formed in Table 1. As evidenced in the tables, the graph size of each document is significantly smaller relative to the number of pixels or the number of words in a document, which significantly simplifies the subsequent modeling and computation.

4.2 Document Preprocessing

To construct a graph for each document image we need to utilize the different layout regions' information such as paragraphs, title, list, table present in each document. We process the scanned document images using pre-trained PubLayNet to obtain the necessary bounding boxes for each region. To extract the textual information present in the bounding box of each region in the document images we apply PyTesseract, an open-source python OCR package. All the results of the PubLayNet and PyTesseract were then serialized and stored in tensor format using PyTorch[17].

4.3 Hyper parameters and infrastructure

We use Hedwig[1], an open-source deep learning toolkit which contains implementation of several document classification models. We use a Tesla K80 GPU for all models requiring GPU for train, and use amazon EC2-t2.micro and EC2-c type machine when only CPU is needed . We use PyTorch 1.5 as the backend framework, and gensim [18] package for computing the node feature vectors using word2vec. As evidenced in the table 1, the graph size of each document is significantly smaller relative to the number of pixels or the number of words in a document, which significantly simplifies the subsequent modeling and computation.

5 RESULTS AND DISCUSSIONS

5.1 Comparing classification accuracy

We compare our proposed approach with the state-of-the-art deep learning models, VGG-16 pre-trained on ImageNet and BERT$_{base}$ pre-trained on Wikipedia. When using BERT$_{base}$ for classification, we extracted tokens from document images as input and fine tune BERT$_{base}$ with class labels. When using pre-trained VGG-16, we used it directly to extract document image features for classification. No further fine-tuning is used. We also compared our model with DocBERT which is specially designed for document level classification by simplifying BERT models with knowledge distillation.

Table 2 shows the model comparison results of classification AUC on the two data sets. In the insurance data set, our proposed model achieves 90.7% to 91.0 %, very competitive as compared to models in BERT families (91.95 %) and VGG-16 (90.6 %). In the Tobacco-3482 data set, our models achieved 73.5 % to 77.5 % , comparable to models in BERT families (82.3 %) and VGG-16 (81.5 %). The fact our proposed model shows more advantages when classifying the insurance data set could be due to the high intra-class variance and low inter-class variance in the Tobacco-3482 data set [12].

We also experimented with combining text and image embedding features as node features in the graph neural network. The combination does provides ample AUC improvements in our proposed method on Tobacco-3482 dataset i.e. 73.5 % for Eff-GNN + Word2Vec and for 77.5 % for Eff-GNN + Word2Vec + Image Embedding. In the insurance data set, Eff-GNN + Word2Vec + Image Embedding shows little improvement over using Word2Vec text features alone. That could be due to the fact that both the textual and image content in the insurance data set provides enough information for classification. This assumption can be further justified by the similar classification results achieved by models in BERT family and VGG-16.

In addition to classification performance, we compare the number of trainable parameters of each model. In both the insurance data set and Tobacco-3482, our model size is drastically smaller than models in BERT families and VGG-16. We calculate the parameters of our model as the sum of parameters in the graph neural network and the parameters in trained word2vec. The small sizes of graph neural network model (Table 1) results in only 160,000 parameters. The Word2Vec model is also relatively light weight because each of the data set contains a very limited vocabulary. Note we did not include

[1] https://github.com/castorini/hedwig

Dataset Results					
Insurance			**Tobacco-3482**		
Model	**AUC**	**# Parameters**	**Model**	**AUC**	**# Parameters**
DocBert [1]	91.95 %	110M	DocBERT [1]	82.3 %	110M
BERT [8]	91.95 %	110M	BERT [8]	79.0 %	110M
VGG-16 [20]	90.6 %	130M	VGG-16 [20]	81.5%	130M
Eff-GNN + Word2Vec [16]	91.0 %	124k + 610k	Eff-GNN+ Word2Vec [16]	73.5 %	124k + 610k
Eff-GNN + Word2Vec [16] + Image Embedding	91.0 %	126k + 610k	Eff-GNN + Word2Vec [16] + Image Embedding	77.5 %	126k + 610k

Table 2: Classification accuracy on the Insurance, Tobacco-3482 dataset.

Insurance	**Batch Size**	**Epochs**	**Training Time**	**GPU Memory (Training)**	**Inference Time**
VGG	32	28	2.30 hours	7.08GB	103 seconds
Eff-GNN (GPU)	32	50	3.5 mins.	470MB + 3.5 GB	0.79 seconds
Eff-GNN (CPU)	32	50	4.1 mins.	NA	0.79 seconds
BERT	16	15	6.2 hours	10.5GB	40 seconds
DocBERT	16	15	6.3 hours	8.1GB	"40 times faster than BERT" [1]

Table 3: Memory, hardware and time required by different models on the Insurance dataset.

the 44.2 million parameters of PubLayNet, because in our framework we do not train or fine-tune any parameters of the PubLayNet.

5.2 Comparing computing resources

We also bench marked the time and memory required for training our proposed Eff-GNN against other models. Table 3 reports the statistics on the insurance data set, 4544 images for training and 1280 images for inference. Eff-GNN models take less than 5 minutes to train 50 epochs whereas VGG-16 or models in BERT Family take hours to train less number of epochs (15 epochs and 28 epochs respectively). In particular Eff-GNN can run on CPU alone and its model training time is comparable to its GPU counterpart. This is consistent with the small size of our model (See Table 2, "Parameters" column). Eff-GNN can achieve these advantages because it models the documents using a graph formed by regions extracted by PubLayNet. The time of using PubLayNet to extract regions for training images are negligible. The size of the resulting graph leads to a small model compared to deep models trained on pixel level information or transformers trained on token level information. Therefore Eff-GNN only uses 470MB in GPU memory with additional 3.5 GB for using PubLayNet. Consequently Eff-GNN requires drastically less time for the inference of 1280 images (0.79 seconds) as compared VGG-16 (103 seconds) and BERT (40 seconds). Note the time of document pre-porcessing steps such as OCR and training Word2Vec model are not included in the table. Although these two steps are extra for our proposed framework, we contend that their addition does not nullify the efficiencies gained through graph neural nets. Even BERT based models require OCR extraction preprocessing step. Just one Word2Vec needs to be trained for the entire data sets and OCR can be optimized by e.g. parallel processing.

Compared with the SOTA pre-trained large models, our proposed Eff-GNN framework achieved competitive classification results on our insurance document image data sets, and achieved comparable results on the the open source Tobacco-3482 data set. We also showed that combining text and image information as the node features in our graph neural network can be advantageous when OCR fails to extract text information or when the two modalities are complimentary. Our proposed method models document representations using extracted semantic regions, instead of using token level or pixel level information. Therefore our model size is dramatically less than other methods, and can be run on CPU machines.

6 CONCLUSION AND FUTURE WORK

In this paper we proposed a novel document image classification that uses graph convolution neural network to integrate text, image, and layout information of a document. We rigorously bench marked our method against the SOTA computer vision and language models on both the insurance dataset and Tobacco dataset. We also compared computing time and hardware resources required for training those models. The results showed our method is not only competitive on classification performance but also is much smaller in size therefore requires much less time and resource. This could translate to big cost advantages of hosting and deployment in real world applications. We are also working on enabling general document classification that can handle hundreds of document classes. A few options include training larger models for domain specific transfer learning, enabling few shot learning and continual learning when dynamically adding new document classes. In addition, we would like to further explore more effective document representations including more sophisticated graph representations or jointly trained layout [21].

REFERENCES

[1] Ashutosh Adhikari, Achyudh Ram, Raphael Tang, and Jimmy Lin. 2019. DocBERT: BERT for Document Classification. *CoRR* abs/1904.08398 (2019). arXiv:1904.08398 http://arxiv.org/abs/1904.08398

[2] Muhammad Zeshan Afzal, Andreas Kölsch, Sheraz Ahmed, and Marcus Liwicki. 2017. Cutting the Error by Half: Investigation of Very Deep CNN and Advanced Training Strategies for Document Image Classification. *CoRR* abs/1704.03557 (2017).

[3] Nicolas Audebert, Catherine Herold, Kuider Slimani, and Cédric Vidal. 2019. Multimodal Deep Networks for Text and Image-Based Document Classification. In *Machine Learning and Knowledge Discovery in Databases,International Workshops of ECML PKDD 2019, Wurzburg, Germany, September 16-20, 2019 (Communications in Computer and Information Science, Vol. 1167)*, Peggy Cellier and Kurt Driessens (Eds.). Springer, 427–443. https://doi.org/10.1007/978-3-030-43823-4_35

[4] Arindam Das, Saikat Roy, and Ujjwal Bhattacharya. 2018. Document Image Classification with Intra-Domain Transfer Learning and Stacked Generalization of Deep Convolutional Neural Networks. *CoRR* abs/1801.09321 (2018). arXiv:1801.09321 http://arxiv.org/abs/1801.09321

[5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li. 2009. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*. IEEE Computer Society, 248–255. https://doi.org/10.1109/CVPR.2009.5206848

[6] Andreas Dengel. 1993. Initial learning of document structure. In *2nd International Conference Document Analysis and Recognition, ICDAR '93, October 20-22, 1993, Tsukuba City, Japan*. IEEE Computer Society, 86–90. https://doi.org/10.1109/ICDAR.1993.395776

[7] Timo I. Denk and Christian Reisswig. 2019. BERTgrid: Contextualized Embedding for 2D Document Representation and Understanding. *CoRR* abs/1909.04948 (2019). arXiv:1909.04948 http://arxiv.org/abs/1909.04948

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[9] Michelangelo Diligenti, Paolo Frasconi, and Marco Gori. 2003. Hidden Tree Markov Models for Document Image Classification. *IEEE Trans. Pattern Anal. Mach. Intell.* 25, 4 (2003), 519–523. https://doi.org/10.1109/TPAMI.2003.1190578

[10] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. 2017. Mask R-CNN. *CoRR* abs/1703.06870 (2017). arXiv:1703.06870 http://arxiv.org/abs/1703.06870

[11] Le Kang, Jayant Kumar, Peng Ye, Yi Li, and David S. Doermann. [n.d.]. Convolutional Neural Networks for Document Image Classification. In *22nd International Conference on Pattern Recognition, ICPR 2014, Stockholm, Sweden, August 24-28, 2014*.

[12] Andreas Kölsch, Muhammad Zeshan Afzal, Markus Ebbecke, and Marcus Liwicki. 2017. Real-Time Document Image Classification Using Deep CNN and Extreme Learning Machines. In *14th IAPR International Conference on Document Analysis and Recognition, ICDAR 2017, Kyoto, Japan, November 9-15, 2017*. IEEE, 1318–1323. https://doi.org/10.1109/ICDAR.2017.217

[13] Jayant Kumar, Peng Ye, and David Doermann. 2014. Structural similarity for document image classification and retrieval. *Pattern Recognition Letters* 43 (2014), 119–126.

[14] David D. Lewis, Gady Agam, Shlomo Argamon, Ophir Frieder, David A. Grossman, and Jefferson Heard. 2006. Building a test collection for complex document information processing. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Seattle, Washington, USA, August 6-11, 2006*, Efthimis N. Efthimiadis, Susan T. Dumais, David Hawking, and Kalervo Järvelin (Eds.). ACM, 665–666. https://doi.org/10.1145/1148170.1148307

[15] Xiaojing Liu, Feiyu Gao, Qiong Zhang, and Huasha Zhao. 2019. Graph Convolution for Multimodal Information Extraction from Visually Rich Documents. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019*, Anastassia Loukina, Michelle Morales, and Rohit Kumar (Eds.). Association for Computational Linguistics, 32–39. https://doi.org/10.18653/v1/n19-2005

[16] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* (2013).

[17] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, and Lin. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Eds.). Curran Associates, Inc., 8024–8035. http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

[18] Radim Řehůřek and Petr Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. ELRA, Valletta, Malta, 45–50. http://is.muni.cz/publication/884893/en.

[19] Christian K. Shin, David S. Doermann, and Azriel Rosenfeld. 2001. Classification of document pages using structure-based features. *Int. J. Document Anal. Recognit.* 3, 4 (2001), 232–247. https://doi.org/10.1007/PL00013566

[20] Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).

[21] Yiheng Xu, Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, and Ming Zhou. 2019. LayoutLM: Pre-training of Text and Layout for Document Image Understanding. *CoRR* abs/1912.13318 (2019). arXiv:1912.13318 http://arxiv.org/abs/1912.13318

[22] Xiao Yang, Mehmet Ersin Yümer, Paul Asente, Mike Kraley, Daniel Kifer, and C. Lee Giles. 2017. Learning to Extract Semantic Structure from Documents Using Multimodal Fully Convolutional Neural Network. *CoRR* abs/1706.02337 (2017).

[23] Muhan Zhang, Zhicheng Cui, Marion Neumann, and Yixin Chen. 2018. An End-to-End Deep Learning Architecture for Graph Classification. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, Sheila A. McIlraith and Kilian Q. Weinberger (Eds.). AAAI Press, 4438–4445.

[24] 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 2019, Sydney, Australia, September 20-25, 2019*. IEEE, 1015–1022. https://doi.org/10.1109/ICDAR.2019.00166