Document Image Classification Method Based on Graph Convolutional Network

Yangyang Xiong[1], Zhongjian Dai[1], Yan Liu[2(✉)], and Xiaotian Ding[2]

[1] Beijing Institute of Technology, Beijing 100081, China
[2] Taikang Insurance Group, Beijing 100031, China
liuyan146@taikanglife.com

Abstract. Automatic and reliable document image classification is an essential part of high-level business intelligence. Previous studies mainly focus on applying Convolutional Neural Network (CNN)-based methods like GoogLeNet, VGG, ResNet, etc. These methods only rely on visual information of images but textual and layout features are ignored, thereby their performances in document image classification tasks are limited. Using multi-modal content can improve classification performances since most document images found in business systems carry explicit semantic and layout information. This paper presents an innovative method based on the Graph Convolutional Network (GCN) to learn multiple input image features, including visual, textual, and positional features. Compared with the CNN-based methods, the proposed approach can make full use of the multi-modal features of the document image to lead the model competitive with other state-of-the-art methods with much fewer parameters. In addition, the proposed model does not require large-scale pre-training. Experiments show that the proposed method achieves an accuracy of 93.45% on the popular RVL-CDIP document image dataset.

Keywords: Graph convolutional network · Document classification · Image processing

1 Introduction

Document digitization plays a critical role in the automatic retrieve and management of document information. Most of these documents are still processed manually, with billions of labor costs each year in industry. Thus, researches on automatic document image classification have great practical value. The document image classification task attempts to predict the type of a document image by analyzing the document's appearance, layout, and content representation. Traditional solutions to this challenge mainly include the image-based classification method and the text-based classification method. The former tries to extract patterns in the pixels of the image to match elements with a specific category, such as shapes or textures. The latter tries to understand the text printed in the document and associate it with its corresponding class.

T. Mantoro et al. (Eds.): ICONIP 2021, LNCS 13108, pp. 317–329, 2021.
https://doi.org/10.1007/978-3-030-92185-9_26

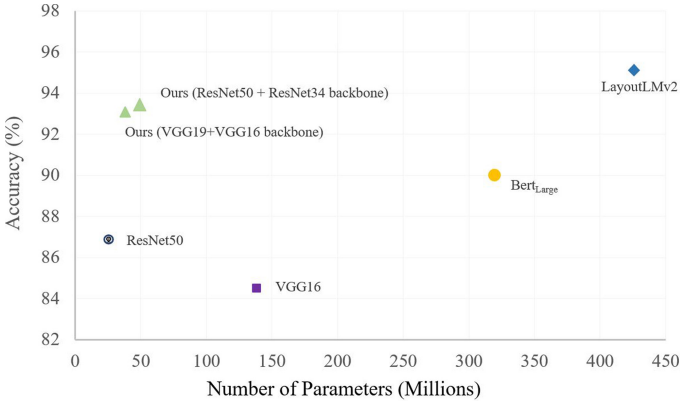

Fig. 1. Model size vs. classification accuracy on the RVL-CDIP dataset. The LayoutLMv2 [1] is currently the state-of-the-art method. However, this model has much more parameters (426M) and requires tens of millions of data for model pre-training to achieve the best accuracy.

However, in real-business applications, the same kind of document often contains different layouts. This intra-class difference makes visual-based classification difficult, and it is impossible to perform rigid feature detection and feature matching, like SURF [2], SIFT [3], and ORB [4]. In addition, different kinds of documents sometimes show high visual similarity, which increases the difficulty of classification. For example, some news articles contain tables and figures, make them look like scientific publications. Therefore, it is difficult for pure visual methods, including CNN, to classify document images with ideal classification accuracy.

If judging from the content of the text, these documents have a similar structure: the address and date usually appear at the top, and the signature usually appears at the bottom. Making full use of the information in the document images, including visual, positional, and textual features, can improve document classification accuracy. In recent years, researchers have started to use the graph concept, including the GCN [5], to do some graph node classification and link prediction tasks with the feature aggregation capabilities of GCN. Therefore, we propose a framework based on the GCN architecture, which can make full use of the multimodal characteristics of the document image. The model incorporates three types of input features: (1) Compact image feature representations for the slice of each text block and the whole document image; (2) Textual features from the text content of each text block; and (3) Positional features denoting the positions of texts within a document image. By doing so, the model can aggregate the visual features and textual features in the document image, and the accuracy of document image classification can be effectively improved.

To sum up, the contribution of this work lies in three folds:

(1) A one-step, end-to-end approach is developed to handle document image classification tasks by a single GCN-based model. The model possesses great scalability to take such a task across various document images with complex layouts.
(2) The model uses the concept of graphs to classify documents and innovatively proposes a method for constructing node features that combine visual, positional, and textual features, which can greatly improve the model performances with fewer parameter sizes, and the best accuracy-speed trade-off is achieved. As shown in Fig. 1.
(3) In practical applications, the model can be trained from scratch and does not require large-scale pre-training.

2 Related Work

Document image classification tasks were generally solved using semantic-based methods in the past. And Bag of Words (BOW)-based methods have shown great success in document image classification [6,7]. However, the primary mechanism of the BOW-based process is to calculate the frequency information of the corresponding word dictionary and ignore the unique layout position information between the document image components, which limits the ability to describe document images.

With the development of deep learning methods in various fields of computer vision, such as target recognition, scene analysis, and natural language processing, deep learning methods show better performance than traditional methods. Some scholars use deep CNN in the field of document image classification and achieve satisfying performances. For the first time, Le Kang et al. use CNN to classify document images [8]. Their results prove that the performance of the CNN is better than the traditional methods. Later, Afzal et al. propose to design a deeper neural network [9], pre-train the network on the ImageNet dataset [10], and then perform transfer learning on the document image dataset. They get better results on the same document image classification dataset with a 12.25% improvement of accuracy. Their experiments show that training a CNN requires many data, and the transfer learning techniques are practical and feasible. However, the CNN-based model can only handle visually different documents, and the performance is deficient on visually similar documents.

To classify document images from the content, some researchers combine the Optical Character Recognition (OCR) [11–14] with Natural Language Processing (NLP) [15]. These methods can deal with visually similar documents well, but do not make full use of the visual information of the document images. Moreover, the document images usually contain defects, including rotation, skew, distortion, scanning noise, etc. All of these bring significant challenges to the OCR system and directly affect subsequent NLP modules. Although enormous efforts have been paid, the OCR + NLP approaches are still short of satisfying performance for the above reasons.

Recently, some researchers notice that the classification of complex document images requires multi-modal feature fusion. For example, LayoutLMv2 [1] realizes to combine textual, visual, and positional information for the document classification task, achieving state-of-the-art performance. Still, it has many parameters (426M) to achieve the optimal result, and requires tens of millions of pre-training data.

3 Proposed Approach

We propose a document image classification framework, which constructs a graph representation for each document image, and the overall architecture is shown in Fig. 2. The first CNN sub-module (CNN1) is responsible for extracting the whole image's visual features. For each OCR text block, the second CNN sub-module (CNN2) is used for extracting local-aware visual features for the text image slice of the block. Textual features are extracted by a Tokenize-Embedding-GRU (Gated Recurrent Unit) pipeline from text contents. Positional features are extracted by a Fully Connected layer (FC1) from text block coordinates. The GCN sub-module is designed to fuse and update the above visual, textual, and positional features and extract graph representations for the document image. At last, the graph representations are passed to a Fully Connected layer (the classification layer, FC3 in Fig. 2) to get the specific category of the document image.

The input of the model includes four parts from the document image, which are: (1) the full image of the document; (2) the image slices of each text block; (3) the text contents of each text block; and (4) the coordinates of each text block. In practice, the text block information is generated by an off-the-shelf OCR system, from which we can get the text content and the coordinates of the four vertices for each text block. One text block from the OCR results is taken as one graph node. Based on this information, an innovative graph node feature construction method is proposed, which combines the full image feature and the feature of each text block.

3.1 Graph Node Feature Extraction

Node features of the graph are constructed from two parts. They are full input image features and text block features, where the text block features include text image features, text content features, and text position features.

The whole image features are extracted by a CNN sub-module (CNN1 in Fig. 2). In our experiments, we attempt to use different CNN backbones, including ResNet50 and VGG19. For these backbones, the final Fully Connected layer is removed, and the size of the Adaptive Average Pooling layer is changed to 7×7. The full document image is resized to a fixed size and then passed to this module to get a $7\times7\times C$ feature map, where C is the image feature channel. Then, this feature map is split into 7×7 parts along the x-direction and y-direction, so 49 parts of features are obtained along the channel-direction ($1\times1\times C$). Finally, each part of the features is squeezed and taken as one node feature of the graph.

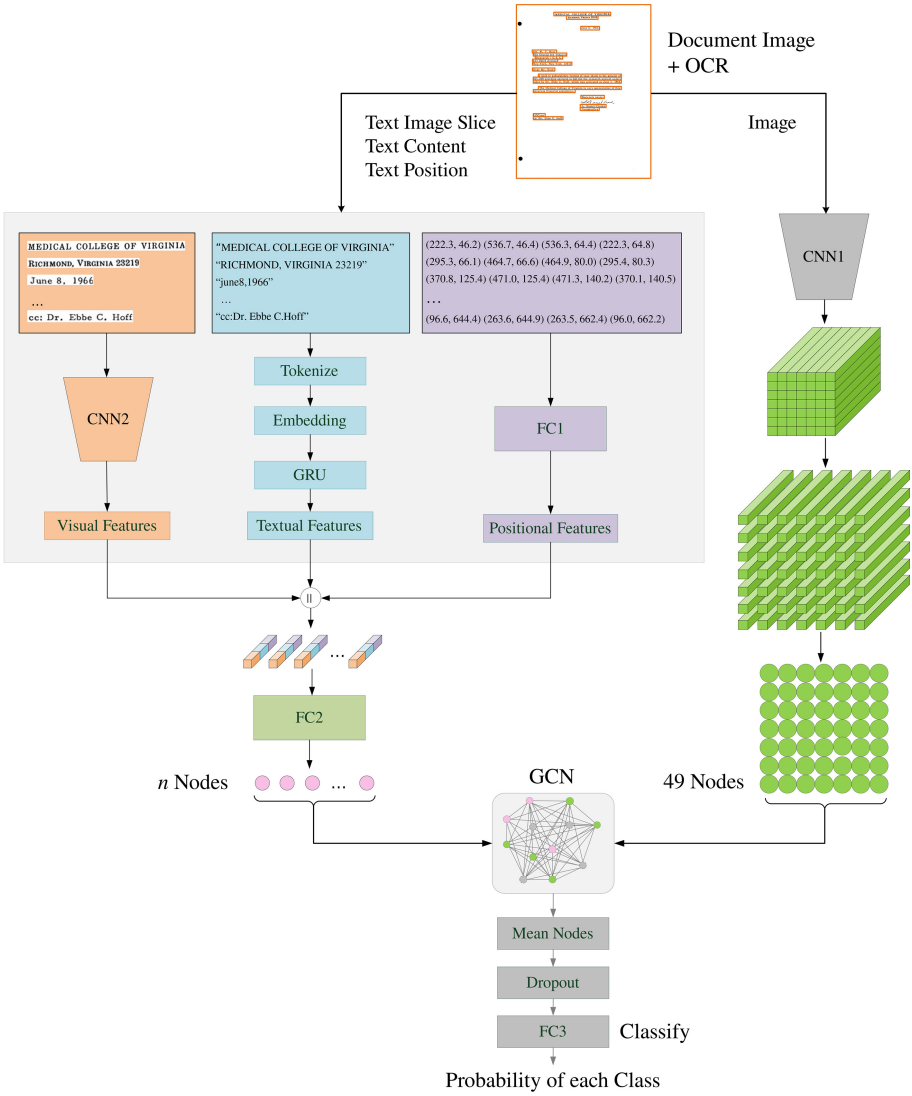

Fig. 2. Overall design of the proposed model. The model employs CNN1 and CNN2 as the backbone network for extracting the full-image visual features and the local-aware visual features. The embedding layer is responsible for converting text information into textual features. The FC1 converts the position vector into the positional feature. The GCN sub-module is designed to fuse and update node features and extract graph representations for the document image. FC1, FC2, and FC3 is Fully Connected layer.

From a computer vision point of view, this is similar to dividing the original image into 7×7 sections and then extracting a node feature by the CNN for each section.

The first 49 nodes' features are prepared from the full input image's CNN feature as described above, and the next is to prepare node features from each OCR text block. The image slice features of each text block are extracted by another CNN sub-module (CNN2). Similar to CNN1, we choose ResNet34 and VGG16 as CNN2 backbones in different experimental setups, respectively. The difference between CNN2 and CNN1 is that, after removing the last fully connected layer, the size of the Adaptive Average Pooling layer of the CNN2 is 1×1. Thus the size of the visual features generated by the CNN2 for each text block is $1\times1\times C$.

In preparing text features for each text block, we pad or cut the text content to a fixed length of 16 words. Then, the Bert Word Piece Tokenizer is used to convert the text into id indexes. Different from BERT [16] training, the [CLS] and [SEP] tokens are removed. An embedding layer is employed to convert these id indexes into 64-d features. Finally, each line of text is transformed into a 128-d textual feature by a 128-unit GRU layer.

The positional information for each text block is obtained from the coordinates of the four vertices of the text block. Each coordinate is composed of two values in x-direction and y-direction. Therefore, the position vector for each text block is constructed and then transformed into a 128-d feature vector by a Fully Connected layer (FC1).

For each OCR text block, the visual, textual, and positional features are prepared by the above steps. Next, they are concatenated together and passed to a Fully Connected layer (FC2) to get the final node feature vector. According to this setting, we can get n nodes' feature if there are n OCR text blocks. As previously introduced, 49 node features have been prepared from CNN1. Thus the graph representation of the input image has 49+n nodes.

3.2 Graph Convolutional Network Module

Unlike CNN, which performs convolution operations in a regular Euclidean space such as a two-dimensional matrix, GCN extends the convolution operation to non-Euclidean data with a graph structure. GCN takes the graph structure and node features as input and obtains a new node representation by performing graph convolution operations on the neighboring nodes of each node in the graph and then pooling all nodes to represent the entire graph.

A multi-layer GCN is defined by the following layer-wise propagation rule [5]:

$$H^{(l+1)} = \sigma(\widetilde{D}^{-1/2}\widetilde{A}\widetilde{D}^{-1/2}H^{(l)}W^{(l)}) \tag{1}$$

Therefore, as long as the input feature X and the adjacency matrix A are known, the updated node feature can be calculated. In our model, the input feature X is the n+49 nodes' features. Since the graph in our model is a Fully Connected graph, every two nodes have a connection, so the adjacency matrix A is $N \times N$ full-one matrix. We build a GCN module with two graph convolutional layers, as shown in Fig. 3. Each layer of graph convolution is followed by a SiLU

activation function. The graph is defined by the fully connected N nodes and initialized from the node features prepared by the above steps. States and features are propagated across the entire graph by the two graph convolutional layers. The final node states vector of the graph is the $N \times 512$ vector. Then, the final node states are averaged to a 1×512 vector, which is the graph representation of the input data. Finally, the 512-d enriched graph representation is then passed to a $512 \times k$ FC layer (FC3 in Fig. 2), where k is the number of the classes of document images.

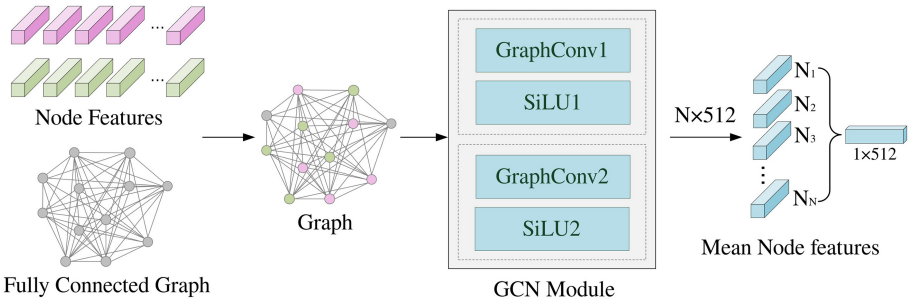

Fig. 3. Schematic depiction of multi-layer Graph Convolutional of aggregating node characteristics. The model's input includes a graph definition with a total of N nodes and the node features.

4 Experiments

4.1 Datasets Description

The model is applied to the document image classification task on the Medical Insurance Document Image (MIDI) dataset and the Ryerson Vision Lab Complex Document Information Processing (RVL-CDIP) dataset [17].

The MIDI Dataset. This dataset contains scanned and photo images collected from the real business system. It has a total of 160,000 images in 20 categories, and sample images are shown in Fig. 4. We split these images into 120,000 training images, 20,000 validation images, and 20,000 testing images. These images are collected from various provinces and cities in China. This dataset has the characteristics of significant intra-class differences and slight inter-class differences.

Fig. 4. OverallSample Images from the MIDI dataset. From left to right: Claim form, Personal information form, Medical invoice, Medical imaging report, Claim notice.

The RVL-CDIP Dataset. This dataset consists of 400,000 grayscale images in 16 classes, with 25,000 images per class. There are 320,000 training images, 40,000 validation images, and 40,000 testing images. The images are resized to a maximum length of 1000 pixels. Some sample images of this dataset can be seen in Fig. 5.

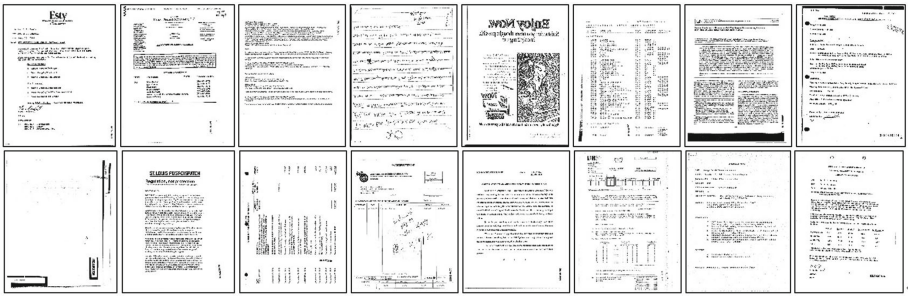

Fig. 5. Sample images from the RVL-CDIP dataset. From left to right: Letter, Form, Email, Handwritten, Advertisement, Scientific report, Scientific publication, Specification, File folder, News article, Budget, Invoice, Presentation, Questionnaire, Resume, Memo.

4.2 Model Training and Evaluation

For each experiment, a trainable end-to-end pipeline was built according to Fig. 2, and the output of the pipeline was the enriched feature of the original image. After the final classifier (FC3), the category to which the image belongs was predicted. To test the impact of different visual backbones (CNN1 and CNN2 in Fig. 2) on the model performances, we tested ResNet50 and VGG19 for CNN1 and tested ResNet34 and VGG16 for CNN2. In order to compare our model with the CNN-based visual model, we also tested the performance of the VGG16 and ResNet50 on the MIDI dataset. For the RVL-CDIP dataset, we followed the same model and hyper-parameter setups with the MIDI experiments.

The training epochs were set to 20 for all experiments with gradient accumulation technology to ensure stable convergence of the model. All models were trained on an NVIDIA Tesla V100 machine, using the Cross-Entropy Loss function and AdamW optimizer. The max learning rate was set to 8e-5, and the cosine learning rate scheduler was set. In addition, the learning rate warm-up steps were set to 50000 for the RVL-CDIP dataset, 10000 for the MIDI dataset, respectively. During training, all input data were shuffled at each epoch begin.

5 Results and Discussion

On the MIDI dataset, the classification accuracies of the proposed models and the CNN-based models are shown in Table 1. The results suggest that our models with different backbone setups significantly surpass CNN-based methods. The experiments reach the best classification accuracy of 99.10%, with 6.58% and 5.71% accuracy improvement than the CNN-based VGG16 and ResNet50. The outstanding performance means that the proposed models can be directly used in industrial applications since this dataset is the actual business dataset. The proposed models have much fewer parameters than the VGG16 because we removed the large-parameter FC layer.

Table 1. Classification accuracy of different models on the MIDI dataset.

Models	Accuracy(%)	Parameter size (millions)
VGG16	92.62	130
ResNet50	93.39	26
Ours (VGG 19 + VGG 16 backbone)	**99.04**	**38**
Ours (ResNet50 + ResNet34 backbone)	**99.10**	**49**

Table 2 shows the result of our model compared with VGG16, ResNet50, and other models, including text-only models and image-only models on the RVL-CDIP dataset. The table shows that the proposed model outperforms those text-only or image-only models as it leverages the multi-modal information within the documents. The proposed model uses the fewest parameters but shows the best classification accuracy.

It is worth noting that although the RVL-CDIP dataset is larger than the MIDI dataset. Due to the higher image resolution, higher OCR character recognition accuracy, and color images, the classification accuracy on the MIDI dataset is higher than on the RVL-CDIP dataset when using the same model setup and training setups. The OCR engine in our experiments is a general multi-language engine and not specially optimized for English data. Thus, the OCR character recognition accuracy is unsatisfactory due to the OCR engine optimization and the low pixel resolution of texts in several images.

Table 2. Comparison of accuracies on RVL-CDIP of best models from other papers.

	Models	Accuracy (%)	Parameter size (millions)
Text-only models	BERT-Base [16]	89.81	110
	UniLMv2-Base [18]	90.06	125
	BERT-Large [16]	89.92	340
	UniLMv2-Large [18]	90.20	355
Image-only models	VGG16	84.52	138
	ResNet50	86.83	26
	Document section-based models + AlexNet transfer learning [17]	89.80	–
	AlexNet + spatial pyramidal pooling + image resizing [19]	90.94	–
	Transfer Learning from AlexNet, VGG16,GoogLeNet and ResNet50 [20]	90.97	–
	Transfer Learning from VGG16 trained on Imagenet[21]	92.21	–
Proposed models	Ours (VGG19 + VGG16 backbone)	**93.06**	**38**
	Ours (ResNet50 + ResNet34 backbone)	**93.45**	**49**

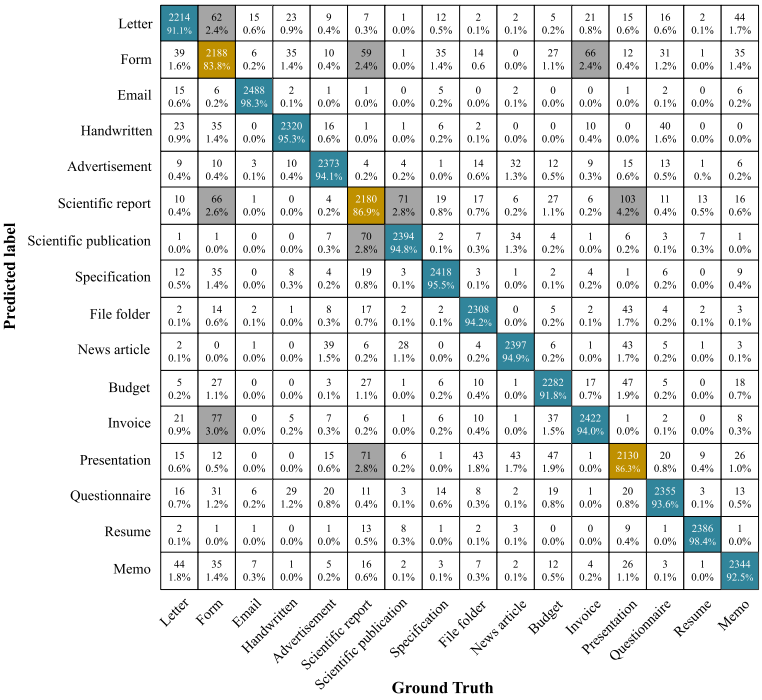

Fig. 6. Confusion matrix of the proposed model on the RVL-CDIP dataset.

Figure 6 reports the confusion matrix of the proposed model on the RVL-CDIP dataset. It shows that the proposed model performs very well on most categories of images. However, the classification accuracy for the three categories is less than 90%, which is form, scientific report, and presentation. This is because there are overlaps of definitions among the three categories. For example, some pages of scientific reports usually contain data forms, which make them be defined as the "form" category.

6 Conclusion

This paper presents a document image classification framework based on GCN. We propose a novel multi-modal graph node feature construction method to combine the visual, textual and positional features of each text block in the image and the visual feature of the full document image. All of these make the feature expression more abundant. By transmitting information to the GCN network, the meaningful features are enriched for classification. Experiments were carried out on the MIDI dataset and the RVL-CDIP dataset. The proposed model obtained classification accuracies of 99.10% and 93.45% on the two datasets, respectively, which are superior to CNN algorithms. Experimental data have shown that our model is effective and efficient. Moreover, our end-to-end pipeline does not require handcrafted features or largescale pre-training as other works.

In our experiments, the OCR engine we can obtain is not optimized for the English data. The lower gain on the RVL-CDIP dataset is directly affected by the high error rate of OCR recognition and the low image resolution of several images. Therefore, we will further find commercial OCR systems suitable for English text recognition to tackle this problem. We also consider adding more features to the GCN model, such as learning the relationship between text blocks, to make full use of the GCN capabilities and various information of the document image.

References

1. Xu, Y., Xu, Y., Lv, T., Cui, L., Wei, F., Wang, G., et al.: LayoutLMv2: multi-modal pre-training for visually-rich document understanding, pp. 1–16 (2020)
2. Bay, H., Ess, A., Tuytelaars, T., Van Gool, L.: Speeded-up robust features (SURF) original publication. Comput. Vis. Image Underst. **110**, 346–359 (2008)
3. Low, D.G.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vis. **60**, 91–110 (2004)
4. Rublee, E., Rabaud, V., Konolige, K., Bradski, G.: ORB: an efficient alternative to SIFT or SURF. In: Proceedings of IEEE International Confernce Computing Vision, pp. 2564–2571 (2011)
5. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: 5th International Conference Learning Representation ICLR 2017 - Conference Track Proceedings, pp. 1–14 (2017)

6. Barbu, E., Héroux, P., Adam, S., Trupin, É.: Using bags of symbols for automatic indexing of graphical document image databases. In: Liu, W., Lladós, J. (eds.) GREC 2005. LNCS, vol. 3926, pp. 195–205. Springer, Heidelberg (2006). https://doi.org/10.1007/11767978_18
7. Kumar, J., Prasad, R., Cao, H., Abd-Almageed, W., Doermann, D., Natarajan, P.: Shape codebook based handwritten and machine printed text zone extraction. In: ProcSPIE (2011). https://doi.org/10.1117/12.876725
8. Kang, L., Kumar, J., Ye, P., Li, Y., Doermann, D.: Convolutional neural networks for document image classification. In: 2014 22nd International Conference on Pattern Recognition, p. 3168–3172 (2014)
9. Afzal, M.Z., Capobianco, S., Malik, M.I., Marinai, S., Breuel, T.M., Dengel, A., et al.: Deepdocclassifier: document classification with deep convolutional neural network. In: Proceedings of International Conference Document Analysis and Recognition, ICDAR, pp. 1111–1115. IEEE (2015)
10. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. Commun. ACM **60**, 84–90 (2017)
11. Zhou, X., Yao, C., Wen, H., Wang, Y., Zhou, S., He, W., et al.: EAST: an efficient and accurate scene text detector. In: Proceedings - 30th IEEE Conference oComputer Vision Pattern Recognition, CVPR 2017, 2017-January, pp. 2642–2651 (2017)
12. Shi, B., Bai, X., Yao, C.: An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. IEEE Trans. Pattern Anal. Mach. Intell. **39**, 2298–304 (2017)
13. Tian, Z., Huang, W., He, T., He, P., Qiao, Yu.: Detecting text in natural image with connectionist text proposal network. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9912, pp. 56–72. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46484-8_4
14. Zhang, C., Liang, B., Huang, Z., En, M., Han, J., Ding, E., et al.: Look more than once: an accurate detector for text of arbitrary shapes. In: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2019-June, pp. 10544–10553 (2019)
15. Noce, L., Gallo, I., Zamberletti, A., Calefati, A.: Embedded textual content for document image classification with convolutional neural networks. In: DocEng 2016 - Proceedings 2016 ACM Symposium Document Engineering, pp. 165–73 (2016)
16. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: NAACL HLT 2019–2019 Conference North America Chapter Association Computer Linguistius Human Language Technology - Proceedings Conference, pp. 1:4171–1:4186 (2019)
17. Harley, A.W., Ufkes, A., Derpanis, K.G.: Evaluation of deep convolutional nets for document image classification and retrieval. In: Proceedings of International Conference on Document Analysis Recognition, ICDAR, 2015-November, pp. 991–995 (2015)
18. Bao, H., Dong, L., Wei, F., Wang, W., Yang, N., Liu, X., et al.: Unilmv2: pseudo-masked language models for unified language model pre-Training. In: 37th International Conference on Machine Learning, ICML 2020, Part F16814, pp. 619–629 (2020)
19. Tensmeyer, C., Martinez, T.: Analysis of convolutional neural networks for document image classification. In: Proceedings of International Conference on Document Analysis Recognition, ICDAR, vol. 1, pp. 388–393 (2017)

20. Afzal, M.Z., Kolsch, A., Ahmed, S., Liwicki, M.: Cutting the error by half: investigation of very deep cnn and advanced training strategies for document image classification. In: Proceedings of International Conference on Document Analysis Recognition, ICDAR, vol. 1, pp. 883–888 (2017)
21. Das, A., Roy, S., Bhattacharya, U., Parui, S.K.: Document image classification with intra-domain transfer learning and stacked generalization of deep convolutional neural networks. In: Proceedings - International Conference on Pattern Recognition, 2018-August, pp. 3180–3185 (2018)