Multi-scale Cell-based Layout Representation for Document Understanding

Yuzhi Shi[1]*, Mijung Kim[2], and Yeongnam Chae[2]
[1]Chubu University
[2]Rakuten Institute of Technology
shi@mprg.cs.chubu.ac.jp, {mijung.a.kim, yeongnam.chae}@rakuten.com

Abstract

Deep learning techniques have achieved remarkable progress in document understanding. Most models use coordinates to represent absolute or relative spatial information of components, but they are difficult to represent latent rules in the document layout. This makes learning layout representation to be more difficult. Unlike the previous researches which have employed the coordinate system, graph or grid to represent the document layout, we propose a novel layout representation, the cell-based layout, to provide easy-to-understand spatial information for backbone models. In line with human reading habits, it uses cell information, i.e. row and column index, to represent the position of components in a document, and makes the document layout easier to understand. Furthermore, we proposed the multi-scale layout to represent the hierarchical structure of layout, and developed a data augmentation method to improve the performance. Experiment results show that our method achieves the state-of-the-art performance in text-based tasks, including form understanding and receipt understanding, and improves the performance in image-based task such as document image classification. We released the code in the repo [a].

1. Introduction

Document understanding can parse layout and extract key information from various documents such as scanned forms and receipts, which are widely used in many industries. However, it is a challenging task due to its cross-modality nature including textual, visual, and layout characteristics. With the development on natural language processing (NLP) and computer vision (CV) techniques, extracting textual and visual information has been easier, however, the utilization of layout information has received relatively less attention. Thus, we try to develop a layout representation that complies with human reading habits.

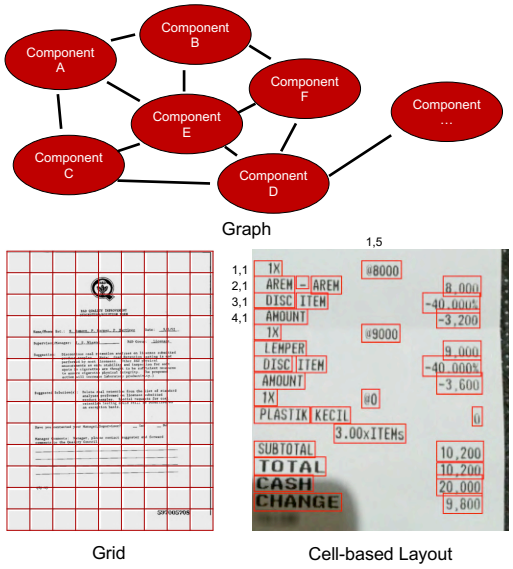

Figure 1. **Three layout representations: graph, grid, and cell-based layout.** The number in the cell-based layout is the [row, column] index of the cells. The images are sampled from FUNSD and CORD.

A document consists of many independent components, such as text blocks, figures, and tables. The positions of those components usually follow specific patterns. For example, the answer is usually written on the right or right below to the corresponding question as a pair. Also, contextually similar components are written on the same row or column. Such arrangement of text components enables us to read more easily and fast, which is one of the crucial features for the document understanding.

Previous works [24, 25, 7, 13, 3] have used absolute and relative coordinates extracted from optical character recognition (OCR) models to represent the position of components. Since those approaches are tied to their OCR algorithms, the coordinates are influenced by the limitation of the algorithms. In addition, it is hard to learn relational information among components. Therefore, there is a possibility to confuse the model, which results in training the

*work conducted during an internship at Rakuten Group, Inc.

[a]https://github.com/mijungkim-rakuten/multi-scale-cell-based

model even more difficult. To address this problem, we introduce the cell-based layout to improve the representation of spatial positions. A cell is a unit of components in an image used to represent layout, which corresponds to a bounding box. Specifically, we calculate the position of rows and columns in a document based on coordinates by giving the same row/column index to the cells that have similar y/x coordinates, and then give a row index and column index to every cell based on coordinates. By learning from row/column indexes, the model can learn if two components are in the same row or column and how many components are in each row and column. Therefore, the model can understand relatively higher level of the spatial relationship of components in documents than previous approaches.

Regarding the layout representation, there are two popular ways: graph [26] and grid [10] as shown in Figure 1. Graph model can learn from the relationship among all components, however, suffering from heavy calculation. Grid divided document image into several patches with the same height and width. It could capture distance information between patches and has less calculation, however, the size of component is not always equal to the patches. Sometimes many components are put into a patch and sometimes a component is divided into several patches. To resolve these problems, we propose the cell-based layout, which makes every component in a cell.

Generally, document layout is a hierarchical structure. For example, a text block could be divided into a few sentences, and a sentence could be divided into several words. To represent such complex layout, we propose the multi-scale layout via using word-level cells and token-level cells as input data. One or several words make up a token, a named entity. Furthermore we propose a data augmentation to simulate hand-writing words and camera motion, which randomly zoom in or out named entities in the documents.

Our contributions are as follows: 1. We propose a novel layout representation for document understanding, the cell-based layout, which is more in line with natural human reading habits. 2. We propose the multi-scale layout to learn the hierarchical structure in documents and propose a new data augmentation to improve the results. 3. Our method achieved the SoTA performance of named entity recognition on the FUNSD [8] and CORD [19] datasets and improve the performance of document classification on the RVL-CDIP [5] dataset comparing with the baseline models. Furthermore, we conducted extensive ablation studies to analyze the effect of the multi-scale cell-based layout.

2. Related work

2.1. Document understanding

The approaches of document understanding could be divided into three categories: heuristic rule-based approaches, conventional machine learning approaches, and deep learning approaches. To develop rule-based approaches, researchers summarized some heuristic rules via manually observing the layout information of documents and processed documents with fixed layout information. The rule-based approaches [4, 11, 17, 21] contains three types of analysis methods: bottom-up [11, 21], top-down [4] and hybrid strategy [18].

With the development of conventional machine learning, statistical machine learning approaches [16, 20] have become the standard for document segmentation tasks in the last decade. [20] models document layout as a grammar and performs a global search for the optimal parse based on a grammatical cost function.

Recently, deep learning methods have become the mainstream of many machine learning problems. Doc-former [1] proposed a novel multi-modal attention layer capable of fusing text, vision, and spatial features. SelfDoc [13] proposed a modality-adaptive attention mechanism to fuse language and vision features. Many novel unsupervised pre-training tasks are proposed to encourage multi-modal feature collaboration, such as the text-image alignment task in LayoutLMv2 [25], which aligns the text lines and the corresponding image regions. LayoutLMv3 [7] introduces a word patch alignment objective to learn cross-modal alignment.

There are research effort on improving document representation ability of the model by making full use of cross-modal information. ViLBERT [15] proposed a model for learning task-agnostic joint representations of image content and natural language. VL-BERT [22] adopts the Transformer model as the backbone, and extends it to take both visual and linguistic embedded features as input.

2.2. Layout representation

For document understanding, the main research direction is the introduction of new pre-training objectives[24, 25, 7, 26] and the attention mechanism [13]. PICK [26] introduces a novel method for the KIE task and uses the improved graph learning module to learn the layout representation. Chargrid [10] introduced a novel type of text representation, which is achieved by encoding each document page as a two-dimensional grid of characters. [9] learns the boundary points and the pixels in the text lines and then follows the most simple observation that the boundaries and text lines in both horizontal and vertical directions should be kept after dewarping to introduce a novel grid regularization scheme. Distinct from them, we propose a novel layout representation, the cell-based layout. It does not need additional tool, data, or modules. It just analyzes the OCR results to generate a row index and column index for each bounding box, and uses them to improve the existing methods.

3. Method

In this section, we first introduce the cell-based layout, as shown in Figure 3. We utilize the row and column indexes as spatial position representations and sort out the input data by row/column index to improve the layout representation. Then we introduced the multi-scale layout, which uses word- and token-level cells to learn the multi-scale document layout. Finally, we propose a data augmentation method to simulate handwritten words and camera motion effects.

3.1. Cell-based layout

One reason for the difficulty of document understanding is that the models cannot understand the spatial relationship of the components because of the difficulty in learning the habit of human reading. Generally, existing methods learn from the coordinates of the bounding boxes. However, it is inefficient to train the document understanding model for three reasons. The first reason is that the detected coordinates of the bounding box will be biased due to the limitation of OCR technology and real word environment, especially for handwriting words. For example, the detected y-coordinates of two words on the same line are often a few pixels apart. Besides, unnoticed handwriting positional deviations will also be magnified, resulting in an inaccurate representation of the document layout.

The second reason is that the coordinate of a bounding box could not provide any relational information with other bounding boxes, so that the model has to learn the latent information in all coordinates information. Although [25, 7] puts the index of a word in the corresponding named entity and the distance between the former boundary box in the model to solve this problem, it is still difficult to explore the latent rules of the document. We use row index and column index to present spatial relationship of cells, it could emphasize the relationship of the cells on the same line or on the same column. Besides, the current cell number indicates that the max number of previous cells in the same row/column indirectly by simply subtracting 1 from the row/column index. In addition, the coordinates have a wide range that makes it difficult to converge, for example, [24, 25, 7] normalized the range of the coordinates in the range of $[1 \sim 1000]$. Contrarily, the maximum number of rows/columns is usually only a few dozen. Therefore, the cell-based layout is a more efficient representation of the document layout.

The third reason is that the coordinates do not fit the way humans remember. The cell-based layout is more in line with human understanding of documents. We think about information of the document layout, for example, the specific meaning of words on the same column. There are certain rules in documents, such as item names are usually in the same column on a invoice. The row/column index could emphasize the relationship among cells in the same line/column, it could help the model to find the latent rule in the documents.

Generation cell-based layout. First, we sort the x- and y-coordinates of top left corners of the bounding boxes to get a sequence of the x-coordinates X_{tl} and the y-coordinates Y_{tl}. The coordinates of i^{th} row r_i and column c_i are calculated as follows;

$$r_1 = min(Y_{tl});\ c_1 = min(X_{tl}).$$

Then we define two sets of coordinates Y^i_{tl} and X^i_{tl}, $i > 1$ and $i \in \mathbb{N}^+$.

$$Y^i_{tl} = \{y_{tl} | y_{tl} > r_{i-1} + \theta * H\}$$

$$X^i_{tl} = \{x_{tl} | x_{tl} > c_{i-1} + \theta * W\}$$

$$r_i = min(Y^i_{tl});\ if\ len(Y^i_{tl}) > 0$$

$$c_i = min(X^i_{tl});\ if\ len(X^i_{tl}) > 0,$$

where H is the height of the document and W is the width of the document. θ is a threshold to control the number of cells in the document, and make the cell-based layout clearer. We use 0.005 as θ unless otherwise indicated. Then we give every bounding box a row index r_{index} and a column index c_{index} based on the y-coordinate y_{tl} and x-coordinate x_{tl} of its top left corner.

$$r_{index} = o;\ if\ c_o \leq y_{tl} < c_{o+1};\ for\ o \in \mathbb{N}^+$$

$$c_{index} = p;\ if\ c_p \leq x_{tl} < c_{p+1};\ for\ p \in \mathbb{N}^+$$

In the cell-based layout, we update the coordinate of the top-left corner of the bounding box using the coordinates of the corresponding row and column.

Figure 2 gives an overview of the application of the cell-based layout. We propose two ways to take advantage of cell information.

Spatial position representation. The first way is taking the cell information as a part of the spatial position representation, and making the backbone model learn latent information in the cell-based layout from the cell information.

$$E_{x_{tl}} = Emb_x(x_{tl});\ E_{y_{tl}} = Emb_y(y_{tl})$$

$$E_w = Emb_w(w);\ E_h = Emb_h(h)$$

$$E_r = Emb_r(r_{index});\ E_c = Emb_c(c_{index})$$

$$SPR = Concat(E_{x_{tl}}, E_{y_{tl}}, E_w, E_h, E_r, E_c),$$

where x_{tl} and y_{tl} are the x/y coordinates of the top left corner, w and h are the width and height of the bounding box. Finally, we combine these embeddings to represent the spatial representation. Embs are the embedding layers.

Order of the input data. The other way is sorting the sequence of cells by row or column index. It does not need

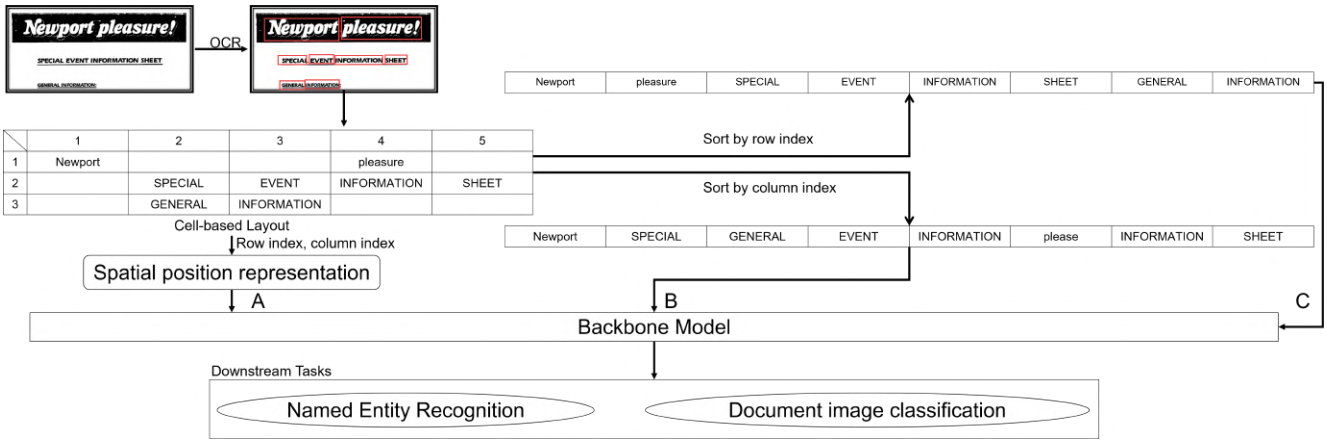

Figure 2. **Overview of the application of the cell based layout.** We define a row and column index for each bounding box to generate the cell-based layout. We use three ways to take advantage of cell information: A. using row and column index as spatial position representation. B. sorting input data by column index. C. sorting input data by row index. The images are sampled from FUNSD.

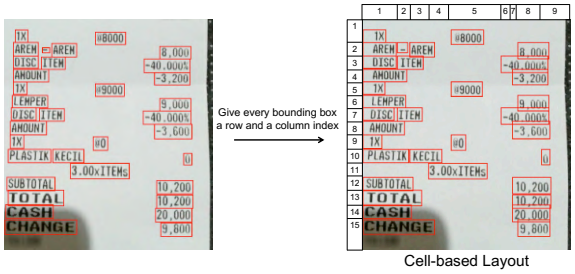

Figure 3. **A cell-based layout sample of a receipt image.** The numbers in the cell-based layout are row and column indexes. The images are sampled from CORD.

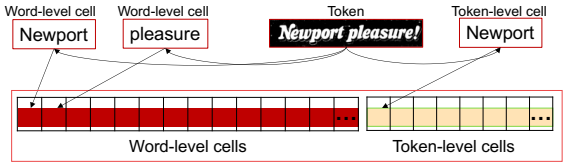

Figure 4. **The multi-scale layout.** The feature of the first word in a token is used as the feature of the token. The image is sampled from the FUNSD dataset.

to change model structure, the existing models could be reused directly. By using the specific order of the input data, the latent rules among the cells which on the same line or in the same column could be understood more easily. Such rules can be found by humans easily, and they appear in the most documents due to the human writing customs.

3.2. Multi-scale layout

A document layout can be considered as a hierarchical structure largely consisted of header, content, and footer. The content can be divided into multiple sentences, a sentence can be divided into multiple phrases, and a phrase can be divided into several words. It is important to understand the hierarchical structure of document for document understanding, therefore we propose the multi-scale layout to express the multi-level structure accurately. Since there are not enough annotations to learn the full hierarchical structure, we feed word-level and token-level cells into the model as shown in Figure 4.

To apply multi-scale layout on existing methods conveniently, we only modify the input data without changing the architecture of the existing models or increasing calculation. Specifically, we use the cell feature of the first word in the token as the token cell feature. Then we feed N_w word cells and the N_t token cells into the model, where N_w means the number of word cells and N_t means the number of token cells. Token cells are less than word cells, because token cells could be divided into several word cells. For the named entity recognition task, we take the classification results of the token cells as the results. As a result, the calculation of the token feature is not needed. The multi-scale layout could make use of backbone model to learn the multi-scale layout without additional calculation and modification of the model. Therefore it could be used in any models which use a sequence of components as input data, as long as corresponding annotations exist.

The multi-scale cell-based layout. We implement the multi-scale cell-based layout by using the cell information and the multi-scale layout at the same time. The multi-scale cell-based layout could learn the hierarchical structure based on the multi-scale layout, explore latent layout information based on cells. It is a natural layout representation and more easily to understand.

3.3. Data augmentation

The handwritten words are difficult to localized by OCR tools. Additionally, camera motion and shooting settings would change the document images which often happen in

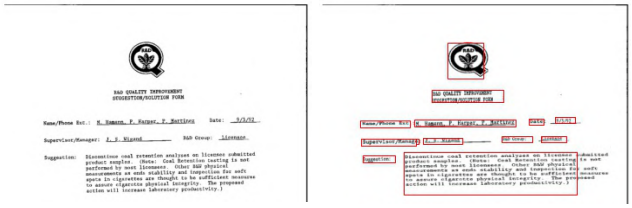

Figure 5. **A sample of proposed data augmentation.** The red boxes represent the original bounding boxes from the dataset. The images are sampled from FUNSD.

the real-world. To simulate these problems, we propose a data augmentation method, shown in Figure 5, which randomly enlarges or shrinks the bounding boxes to generate new examples. We crop the image patch in the every bounding boxes and set the color in all bounding boxes white, firstly. Then we enlarge or shrink these image patches and put them in the original position based on the top left corner of the corresponding bounding box. In this manner, the cell-based layout would not be influenced, therefore it could increase the diversity of document layout without changing the latent rules in the cell information. We use the scale factor Θ to control the resizing range of the bounding boxes.

4. Experiments

In this section, we introduce the implementation details of our proposed method and the datasets that we used in experiments at first. Then we compare with existing methods and analyze the spatial position representation. In addition to it, we used the experimental results on the FUNSD [8] and CORD [19] datasets to test the performance on the named entity recognition task. Furthermore, we use the RVL-CDIP [5] dataset to evaluate the proposed method for document classification. Finally, we conducted several ablation studies to analyze the proposed methods.

4.1. Implementation details

We evaluate our method on three datasets, the CORD dataset, the FUNSD dataset, and the RVL-CDIP dataset. The FUNSD dataset is a document noisy scanned form understanding dataset sampled from the RVL-CDIP dataset. It contains 199 documents with comprehensive annotations for 9,707 semantic entities. We focus on semantic entity a label among "question", "answer", "header", or "other". The dataset is divided into training dataset with 149 samples and test dataset with 50 samples. The CORD dataset is a receipt key information extraction dataset with 30 semantic labels defined under 4 categories. It consists of 800 training samples, 100 validation samples, and 100 test samples. The RVL-CDIP dataset is a subset of the IIT-CDIP collection [12] labeled with 16 categories. It contains 400,000 document images and is split into train/val/test (320,000/40,000/40,000 document images) dataset.

We apply the cell-based layout to the baseline models, LayoutLMv2/v3 [25, 7], to compare it with SoTA performance and evaluate our methods. Baseline models have two different versions with different parameter number, model$_{BASE}$ and model$_{LARGE}$. We use the LayoutLMv3$_{BASE}$ for the comparison purpose unless stated otherwise. We use both the baseline model and the pre-trained baseline model. The pre-trained LayoutLMv3 is pre-trained on a large IIT-CDIP dataset, which contains about 11 million document images and can split into 42 million pages. Due to the limitation of GPU, we use 1 GPU[b] to fine-tune on the FUNSD dataset and the CORD dataset, and use 4 GPUs to fine-tune on the RVL-CDIP dataset. The batch size per GPU is 8 for model$_{BASE}$, and the batch size per GPU of model$_{LARGE}$ is 4.

4.2. Fine-tuning on Multi-modal Tasks

We compare our method with existing methods and categorize them by the position representation as follows.

P(A) is the absolute 1D-position, used to preserve the positional relationship of the components within the document. It represents the difference of the position for each component simply, however could not represent spatial position of components.

Co(A) means the absolute position based on coordinates of bounding boxes, such as coordinates of top-left (x_{tl}, y_{tl}) and bottom right (x_{br}, y_{br}) corners, width w and height h of the bounding box. It provides detailed spatial position of bounding boxes and is used by many methods [23, 24, 13].

Co(R) means the relative position and distance based on coordinates of neighboring bounding box, for example, the Euclidean distance from each corner of a bounding box to the corresponding corner in the adjacent bounding box. Since Co(A) does not include the relationship among the bounding boxes, the model has to learn the latent spatial relationship from dataset. Co(R) could represent the spatial information between adjacent components.

T(R) is semantic relative position, such as the index of a word in the corresponding token [25, 7]. It presents the position of a small component in a large component, which make use of the order of small components in a large component.

Cell represents the row index and column index based on the cell-based layout. It provides rich spatial information by emphasizing the spatial relationship of the cells that are in the same row or in the same column. On the other hand, the number of rows and columns are less than the range of coordinates, which make the layout easier to understand.

We follow the existing methods to fine-tune the baseline models for three multi-modal tasks on public available

[b]NVIDIA Tesla V100-SXM2-32GB

Table 1. **Comparison with existing methods on the FUNSD, CORD, and RVL-CDIP datasets.** "T/L/I" denotes "text/layout/image" modality. "R/G/P" denotes "region/grid/patch" image embedding. The score † is reached when only using the 10% RVL-CDIP dataset and the score ‡ is not reached due to resource limitation. Thus the scores are not directly comparable to other scores.

Model	Parameters	Modality	Image Embedding	Position Representaion	FUNSD (F1)	CORD (F1)	RVL-CDIP (Acc.)
BERT$_{BASE}$ [2]	110M	T	None	P(A)	60.26	89.68	89.81
RoBERTa$_{BASE}$ [14]	125M	T	None	P(A)	66.48	93.54	90.06
BROS$_{BASE}$ [6]	110M	T+L	None	Co(R)+P(A)	83.05	95.73	-
LiLT$_{BASE}$ [23]	-	T+L	None	Co(A)+T(R)	88.41	96.07	95.68*
LayoutLM$_{BASE}$ [24]	160M	T+L+I(R)	ResNet-101(fine-tune)	Co(A)+P(A)	79.27	-	94.42
SelfDoc [13]	-	T+L+I(R)	ResNeXt-101	Co(A)	83.36	-	92.81
Udoc [3]	272M	T+L+I(R)	ResNet-50	Co(A)	87.93	96.86	95.05
LayoutLMv2$_{BASE}$ [25]	200M	T+L+I(G)	ResNeXt101-FPN	Co(A+R)+T(R)+P(A)	82.76	94.95	95.25
DocFormer$_{BASE}$ [1]	183M	T+L+I(G)	ResNet-50	Co(A+R)+P(A)	83.34	96.33	**96.17**
LayoutLMv3$_{BASE}$ [7]	133M	T+L+I(P)	Linear	Co(A+R)+T(R)+P(A)	90.29	96.56	95.44
Ours$_{BASE}$	133M	T+L+I(P)	Linear	Cell+Co(A+R)+T(R)+P(A)	**93.76**	**97.23**	90.7†
BERT$_{LARGE}$ [2]	340M	T	None	P(A)	65.63	90.25	89.92
RoBERTa$_{LARGE}$ [14]	355M	T	None	P(A)	70.72	93.80	90.11
BROS$_{LARGE}$ [6]	340M	T+L	None	Co(R)+P(A)	84.52	97.40	-
LayoutLM$_{LARGE}$ [24]	343M	T+L	None	Co(A)+P(A)	77.89	-	91.90
LayoutLMv2$_{LARGE}$ [25]	426M	T+L+I(G)	ResNeXt101-FPN	Co(A+R)+T(R)+P(A)	84.20	96.01	95.64
DocFormer$_{LARGE}$ [1]	536M	T+L+I(G)	ResNet-50	Co(A+R)+P(A)	84.55	96.99	95.50
LayoutLMv3$_{LARGE}$ [7]	368M	T+L+I(P)	Linear	Co(A+R)+T(R)+P(A)	92.08	97.46	**95.93**
Ours$_{LARGE}$	368M	T+L+I(P)	Linear	Cell+Co(A+R)+T(R)+P(A)	**93.52**	**97.49**	-‡

* LiLT uses image features with ResNeXt101-FPN backbone in fine-tuning RVL-CDIP.

Table 2. **Comparison on the FUNSD dataset.** Ours(P) means the proposed position representation, and Ours(R) and Ours(C) mean that the input data are sorted by row/column index.

Method	Pre-trained	F1(%)
LayoutLMv3	No	21.84
Ours(P)	No	**26.92**
LayoutLMv3	Yes	90.29
LayoutLMv3$_{LARGE}$	Yes	92.08
Ours(P)	Yes	92.39
Ours(R)	Yes	92.50
Ours(C)	Yes	**93.76**

benchmarks, including form understanding on FUNSD, receipt understanding on CORD, and document image classification on RVL-CDIP. Results are shown in Table 1.

4.3. Named entity recognition

Named entity recognition (NER) is a subtask of information extraction that aims to locate and classify named entities mentioned in documents into pre-defined categories such as organizations and locations. The model is trained to learn from the input data, a document image, words, and bounding box information, in order to predict a classification result of each named entity. It could be used for form understanding, receipt understanding, and key information extraction. We apply the cell-based layout into pre-trained and original LayoutLMv3 models to improve the results on NER task. We report F1 scores for this task.

FUNSD dataset. We use LayoutLMv3$_{BASE}$ as the baseline model, and it reaches the 90.29% F1 score after pre-training the model using the CDIP dataset. However, it only gets 21.84 % of F1 score without pre-training, because the FUNSD dataset only has 149 document images for training. Pre-training could greatly improve the performance of the baseline model.

Via using our proposed position embedding, LayoutLMv3 without pre-training could reach the F1 score of 26.92%. The F1 score could be improved by 5.08%. Furthermore, we tested our method using pre-trained LayoutLMv3$_{BASE}$. We put the cell information into spatial position embedding layers in the fine-tuning phase. As a result, the proposed position embedding reached 92.39% of the f1 score, improving LayoutLMv3$_{BASE}$ by 2.1%. It is better than the performance of LayoutLMv3$_{LARGE}$.

Furthermore, by sorting the input data by row index or column index, we reach 92.5% and 93.76%, respectively. Therefore the three ways of using cell information could improve the baseline model. The cell-based layout could improve the performance of LayoutLM-v3 on FUNSD dataset with pre-training and without pre-training. It is noted that we did not pre-train LayoutLMv3 using our method, we only use the cell information in the fine-tuning phase.

CORD dataset. In order to prove the generic of our method, we evaluated our method using LayoutLM$_{BASE}$ on the CORD dataset. LayoutLMv3 reached a F1 score of 53.13% without pre-training. Although the performance is lower than the pre-trained LayoutLMv3, the performance is much better than the results on the FUNSD dataset, due to 800 document images for training. As presented in Table 3, the F1 score is pushed to 85.25% using the multi-scale cell-based layout. In addition, the proposed spatial position representation improved the F1 score by 16.41%. The F1 score could be improved by 13.76% and 12.04% via sorting the input data by row and column index, respectively. Because

Table 3. **Comparison on the CORD dataset**. Ours(P) means the proposed position representation, and Ours(R) and Ours(C) mean that the input data are sorted by row/column index. Ours(M) means the multi-scale layout.

Method	Pre-trained	F1(%)
LayoutLMv3	No	62.56
Ours(P)	No	69.54
Ours(R)	No	66.89
Ours(C)	No	65.17
Ours(M+P)	No	**85.25**
LayoutLMv3	Yes	96.56
Ours(P)	Yes	96.97
Ours(M)	Yes	97.01
Ours(M+P)	Yes	**97.23**

Table 4. **Results on the RVL-CDIP dataset for Document Classification task.** We choose a 0.1%, 10% samples randomly to evaluate the methods. Ours(P) means the proposed position representation, and Ours(R) and Ours(C) mean that the input data are sorted by row/column index.

Method	Pre-trained	Num. of samples	Acc.(%)
LayoutLMv3	No	400 (0.1%)	27.50
Ours(P)	No	400 (0.1%)	40.00
LayoutLMv3	No	40,000 (10%)	77.32
Ours(P)	No	40,000 (10%)	77.89
Ours(R)	No	40,000 (10%)	77.67
Ours(C)	No	40,000 (10%)	77.82
Ours(P+R)	No	40,000 (10%)	**78.69**
Ours(P+C)	No	40,000 (10%)	78.24
LayoutLMv3	Yes	400 (0.1%)	62.50
Ours(P)	Yes	400 (0.1%)	70.00
Ours(R)	Yes	400 (0.1%)	70.00
Ours(C)	Yes	400 (0.1%)	70.00
LayoutLMv3	Yes	40,000 (10%)	90.22
Ours(P)	Yes	40,000 (10%)	**90.70**
Ours(R)	Yes	40,000 (10%)	90.62
Ours(C)	Yes	40,000 (10%)	90.34

of its layout containing latent rules and sufficient number of samples, the cell-based layout could analyze the layout more correctly on receipt dataset.

Furthermore, we evaluate the multi-scale layout and the proposed position representation using the pre-trained LayoutLMv3. As a result, the F1 score increases by 0.67% using the multi-scale and the proposed spatial position representation as shown in Table 3. It shows the multi-scale cell-based layout can benefit LayoutLMv3 and pre-trained LayoutLMv3 for the NER task.

4.4. Document classification

To demonstrate the generalizability of the cell-based layout from the multi-modal domain to the visual domain, we evaluated our method on the task of document classification. The document classification task aims to predict the category of visually rich document images. We conduct experiments on the RVL-CDIP dataset, and extract textual and layout information using tesseract 4.1.1[c]. The evaluation metric is the overall classification accuracy on the test dataset. Note that we use the different OCR tool with LayoutLMv3, and due to resource limitation, we use 4 GPU and up to 10% examples of the dataset. To perform a detailed analysis, we evaluate models using different numbers of samples, as presented in Table 4 You can observe that our method improves accuracy by 7.5% using 0.1% of the RVL-CDIP dataset that contains 400 document samples. When using 10% samples, the proposed spatial position representation could improve the performance by 0.48%. In the condition of using 10% samples and LayoutLMv3, the proposed method could reach an accuracy of 78.69%, improving the baseline model by 1.37%.

We observe two trends from the experiment results. The first is that when more samples are used, the improvement will be less. We consider document classification is a relatively simple task. Compared with latent layout information, intuitive text information and image information are more helpful for document understanding. By using more document samples, the model have more text and image information to learn and reach remarkable performance, even though the cell-based layout give a better layout representation . The second trend is that the pre-trained model is harder to improve, which could also be observed in other experiment results. Considering that the pre-trained model has prior knowledge from the IIT-CDIP dataset, and the cell-based layout that is not used in the pre-training phase, we consider it is a reasonable phenomenon.

4.5. Ablation study

We used ablation studies to analyze the efficiency and generalizability of our methods. More extensive ablation studies are provided in the supplementary.

Comparison of spatial position representation. To analyze the efficiency of the spatial position representation, we compare different spatial position representations as shown in Table 5. We use LayoutLMv3$_{BASE}$ as the baseline model and product experiments on the CORD dataset. PR$_1$ reached the best performance by inserting a row / column index in the spatial position representation. The F1 score is improved by 16.01%, however, the embedding size and the number of parameters are also increased. To evaluate the influence of the parameter number, we develop PR$_2$

[c]https://github.com/tesseract-ocr/tesseract

Table 5. **Comparison results using different spatial position representation (SPR)**. x_{tl} and y_{tl} means the coordinate of top left corner of bounding box, and x_{br} and y_{br} presents the bottom right corner of the bounding box. **1** means the fixed value 1. w and h means the height and width of the bounding box. r and c means the row index and column index. The coordinates† means that coordinates are updated according to the row/column index as mentioned in Section 3.1.

Method	SPR	E. size	F1(%)
BASE	$x_{tl}, y_{tl}, x_{br}, y_{br}, h, w$	786	53.70
PR_1	$x_{tl}\dagger, y_{tl}\dagger, x_{br}, y_{br}, h, w, r, c$	1024	**69.71**
PR_2	$x_{tl}\dagger, y_{tl}\dagger, x_{br}, y_{br}, h, w, \mathbf{1}, \mathbf{1}$	1024	52.20
PR_3	$x_{tl}\dagger, y_{tl}\dagger, x_{br}, y_{br}, h, w, h, w$	1024	51.74
PR_4	$x_{tl}\dagger, y_{tl}\dagger, x_{br}, y_{br}, r, c$	786	58.87
PR_5	$x_{tl}\dagger, y_{tl}\dagger, h, w, r, c$	786	69.54

and PR_3 to compare. PR_2 uses the fixed value 1 to simulate the row index and column index. Considering that the fixed value may bring bad effect to training, we use height and width information again to increase the diversity of input, PR_3. As a result, PR_2 and PR_3 reached a worse result than the baseline model. Additionally, we develop PR_4 by replacing the height and width of the bounding box with the row / column index information, which would not increase the embedding size and the parameter number. PR_4 improved the performance of based model by 5.74%. However without the height and width information, the performance is reduced by 10.84% comparing with PR_1. Consequently, increasing parameters cannot improve performance without appropriate and useful information. The row / column index could provide more important information for spatial position representation.

The x_{tl} and x_{br} share the same embedding layer, and y_{tl} and y_{br} share the same embedding layer in LayoutLMv3. Considering that x_{tl} and y_{tl} are the coordinates of the top left corner, they should provide different information with x_{br} and y_{br}. Sharing the same embedding layer would confuse the information of the two corners. Therefore, we remove x_{tl} and y_{tl} and add row/column index to the position representation to develop PR_5. PR_5 archived a F1 score of 69.54%, which is close to PR_1. By removing x_{tl} and y_{tl} and increasing row/column index, the total embedding size would not be changed. Therefore, PR_5 could easily be employed by existing pre-trained models. By inserting cell information into the position representation, the model would learn the latent layout information from the cell-based layout and reach a better result.

Data augmentation. As presented in Table 6, the data augmentation could improve the results. We use LayoutLMv3_{BASE} as the baseline model and evaluate the method on CORD dataset. Basically, when the bounding box is not updated with the resize of component, the performance is better. The accurate bounding box could make the model overfit on the layout information provided by the training test. Using the bounding box information, which is different from the real size of bounding boxes in document images, would not change the cell-based layout but only change the size of the bounding box. It is helpful to inhibit overfitting in training and improve performance.

Table 6. **Comparison results use data augmentation or not**. Scale factor Θ is used to control the level of the data augmentation. "Update bbox" controls if use the resized bounding box information.

Pattern	Scale factor Θ	Update bbox	F1 score
LayoutLMv3	-	-	53.70
DA	0.2	Yes	53.67
DA	0.3	Yes	53.41
DA	0.2	No	53.79
DA	0.3	No	53.95
DA	0.4	No	**54.07**

Table 7. **Comparison results using LayoutLMv2 with/without the cell information.**

Method	F1 score
LayoutLMv2_{BASE}	82.76
Ours_{BASE}	**83.09**

Evaluation on other model. To evaluate the generic of the cell-based layout, we test it using another baseline model, LayoutLMv2 [25]. We applied the proposed spatial position representation on the baseline model and used the FUNSD dataset to evaluate. As shown in Table 7, the F1 score is improved by 0.33% when using the proposed method. Therefore, we consider that the cell-based layout could improve the performance of other methods.

5. Conclusion

In this paper, we present a novel and natural layout representation for document understanding tasks, i.e. the multi-scale cell-based layout. Unlike graph and grid approaches, it provides natural and efficient spatial representation for document understanding. We believe that it could be easily adapted to other tasks that need spatial information like image detection. Furthermore, we propose a data augmentation method to improve the results. We evaluate the method using 2 baseline model, 3 public datasets, and 2 tasks, named entity recognition and document classification. The multi-scale cell-based layout improve the performance and reached the SoTA on FUNSD and CORD dataset and improve the baseline model on RVL-CDIP dataset in the environment where GPU resource is limited.

References

[1] Srikar Appalaraju, Bhavan Jasani, Bhargava Urala Kota, Yusheng Xie, and R Manmatha. Docformer: End-to-end transformer for document understanding. In *International Conference on Computer Vision*, pages 993–1003, 2021.

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. 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, Volume 1 (Long and Short Papers)*, pages 4171–4186, 2019.

[3] Jiuxiang Gu, Jason Kuen, Vlad I Morariu, Handong Zhao, Rajiv Jain, Nikolaos Barmpalios, Ani Nenkova, and Tong Sun. Unidoc: Unified pretraining framework for document understanding. In *Advances in Neural Information Processing Systems*, pages 39–50, 2021.

[4] Jaekyu Ha, Robert M Haralick, and Ihsin T Phillips. Recursive xy cut using bounding boxes of connected components. In *Proceedings of 3rd International Conference on Document Analysis and Recognition*, pages 952–955, 1995.

[5] Adam W Harley, Alex Ufkes, and Konstantinos G Derpanis. Evaluation of deep convolutional nets for document image classification and retrieval. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 991–995, 2015.

[6] Teakgyu Hong, Donghyun Kim, Mingi Ji, Wonseok Hwang, Daehyun Nam, and Sungrae Park. Bros: A pre-trained language model focusing on text and layout for better key information extraction from documents. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 10767–10775, 2022.

[7] Yupan Huang, Tengchao Lv, Lei Cui, Yutong Lu, and Furu Wei. Layoutlmv3: Pre-training for document ai with unified text and image masking. *arXiv preprint arXiv:2204.08387*, 2022.

[8] Guillaume Jaume, Hazim Kemal Ekenel, and Jean-Philippe Thiran. Funsd: A dataset for form understanding in noisy scanned documents. In *International Conference on Document Analysis and Recognition Workshops (ICDARW)*, pages 1–6, 2019.

[9] Xiangwei Jiang, Rujiao Long, Nan Xue, Zhibo Yang, Cong Yao, and Gui-Song Xia. Revisiting document image dewarping by grid regularization. In *Computer Vision and Pattern Recognition*, pages 4543–4552, 2022.

[10] Anoop R Katti, Christian Reisswig, Cordula Guder, Sebastian Brarda, Steffen Bickel, Johannes Höhne, and Jean Baptiste Faddoul. Chargrid: Towards understanding 2d documents. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 4459–4469, 2018.

[11] Frank Lebourgeois, Zbigniew Bublinski, and Hubert Emptoz. A fast and efficient method for extracting text paragraphs and graphics from unconstrained documents. In *11th IAPR International Conference on Pattern Recognition. Vol. II. Conference B: Pattern Recognition Methodology and Systems*, pages 272–273, 1992.

[12] David Lewis, Gady Agam, Shlomo Argamon, Ophir Frieder, David Grossman, and Jefferson Heard. 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*, pages 665–666, 2006.

[13] Peizhao Li, Jiuxiang Gu, Jason Kuen, Vlad I Morariu, Handong Zhao, Rajiv Jain, Varun Manjunatha, and Hongfu Liu. Selfdoc: Self-supervised document representation learning. In *Computer Vision and Pattern Recognition*, pages 5652–5660, 2021.

[14] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[15] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *Advances in Neural Information Processing Systems*, pages 13–23, 2019.

[16] Simone Marinai, Marco Gori, and Giovanni Soda. Artificial neural networks for document analysis and recognition. *IEEE Transactions on pattern analysis and machine intelligence*, pages 23–35, 2005.

[17] Lawrence O'Gorman. The document spectrum for page layout analysis. *IEEE Transactions on pattern analysis and machine intelligence*, pages 1162–1173, 1993.

[18] Masayuki Okamoto and Makoto Takahashi. A hybrid page segmentation method. In *Proceedings of 2nd International Conference on Document Analysis and Recognition (ICDAR'93)*, pages 743–746, 1993.

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

[20] Michael Shilman, Percy Liang, and Paul Viola. Learning nongenerative grammatical models for document analysis. In *International Conference on Computer Vision*, pages 962–969, 2005.

[21] Anikó Simon, J-C Pret, and A Peter Johnson. A fast algorithm for bottom-up document layout analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 273–277, 1997.

[22] Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. Vl-bert: Pre-training of generic visual-linguistic representations. In *International Conference on Learning Representations*, 2020.

[23] Jiapeng Wang, Lianwen Jin, and Kai Ding. LiLT: A simple yet effective language-independent layout transformer for structured document understanding. In *ACL*, pages 7747–7757, 2022.

[24] Yiheng Xu, Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, and Ming Zhou. Layoutlm: Pre-training of text and layout for document image understanding. pages 1192–1200, 2020.

[25] Yang Xu, Yiheng Xu, Tengchao Lv, Lei Cui, Furu Wei, Guoxin Wang, Yijuan Lu, Dinei Florencio, Cha Zhang,

Wanxiang Che, et al. Layoutlmv2: Multi-modal pre-training for visually-rich document understanding. In *ACL*, pages 2579–2591, 2021.

[26] Wenwen Yu, Ning Lu, Xianbiao Qi, Ping Gong, and Rong Xiao. PICK: Processing key information extraction from documents using improved graph learning-convolutional networks. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 4363–4370, 2021.