Vision Grid Transformer for Document Layout Analysis

Cheng Da*, Chuwei Luo*, Qi Zheng, and Cong Yao†
DAMO Academy, Alibaba Group, Beijing, China
dc.dacheng08,luochuwei,zhengqisjtu,yaocong2010@gmail.com

Abstract

Document pre-trained models and grid-based models have proven to be very effective on various tasks in Document AI. However, for the document layout analysis (DLA) task, existing document pre-trained models, even those pre-trained in a multi-modal fashion, usually rely on either textual features or visual features. Grid-based models for DLA are multi-modality but largely neglect the effect of pre-training. To fully leverage multi-modal information and exploit pre-training techniques to learn better representation for DLA, in this paper, we present VGT, a two-stream Vision Grid Transformer, in which Grid Transformer (GiT) is proposed and pre-trained for 2D token-level and segment-level semantic understanding. Furthermore, a new dataset named D^{4}LA, which is so far the most diverse and detailed manually-annotated benchmark for document layout analysis, is curated and released. Experiment results have illustrated that the proposed VGT model achieves new state-of-the-art results on DLA tasks, e.g. PubLayNet (95.7%→96.2%), DocBank (79.6%→84.1%), and D^{4}LA (67.7%→68.8%). The code and models as well as the D^{4}LA dataset will be made publicly available [1].

1. Introduction

Documents are important carriers of human knowledge. With the advancement of digitization, the techniques for automatically reading [38, 50, 39, 40, 9], parsing [51, 31], and understanding documents [32, 46, 20, 25] have become a crucial part of the success of digital transformation [8]. Document Layout Analysis (DLA) [4], which transforms documents into structured representations, is an essential stage for downstream tasks, such as document retrieval, table extraction, and document information extraction. Technically, the goal of DLA is to detect and identify homogeneous document regions based on visual cues and textual content within the document. However, performing DLA in real-world scenarios is faced with numerous difficulties: variety of document types, complex layouts, low-quality images, semantic understanding, etc. In this sense, DLA is a very challenging task in practical applications.

Table 1. Comparisons of the use of different modalities and pre-training techniques with existing SOTA methods in DLA task.

Models	Vision	Text	Pre-trained
CNN-based [37, 34]	✓	✗	✗
ViT-based [25]	✓	✗	✓
Multi-modal PTM [15, 20]	✓	✗	✓
Grid-based [45, 47]	✓	✓	✗
VGT (Ours)	✓	✓	✓

Basically, DLA can be regarded as an object detection or semantic segmentation task for document images in computer vision. Early works [37, 34] directly use visual features encoded by convolutional neural networks (CNN) [19] for layout units detection [36, 30, 35, 17], and have been proven to be effective. Recent years have witnessed the success of document pre-training. Document Image Transformer (DiT) [25] uses images for pre-training, obtaining good performance on DLA. Due to the multi-modal nature of documents, previous methods [15, 20] propose to pre-train multi-modal Transformers for document understanding. However, these methods still employ only visual information for DLA fine-tuning. This might lead to degraded performances and generalization ability for DLA models.

To better exploit both visual and textual information for the DLA task, grid-based methods [45, 22, 47] cast text with layout information into $2D$ semantic representations (char-grid [23, 22] or sentence-grid [45, 47]) and combine them with visual features, achieving good results in the DLA task. Although grid-based methods equip with an additional textual input of grid, only visual supervision is used for the model training in the DLA task. Since there are no explicit textual objectives to guide the linguistic modeling, we consider that the capability of semantic understanding is limited in grid-based models, compared with the existing document pre-trained models [20]. Therefore, how to effectively model semantic features based on grid representations is a

*Equal contribution. † Corresponding author.
[1] https://github.com/AlibabaResearch/AdvancedLiterateMachinery

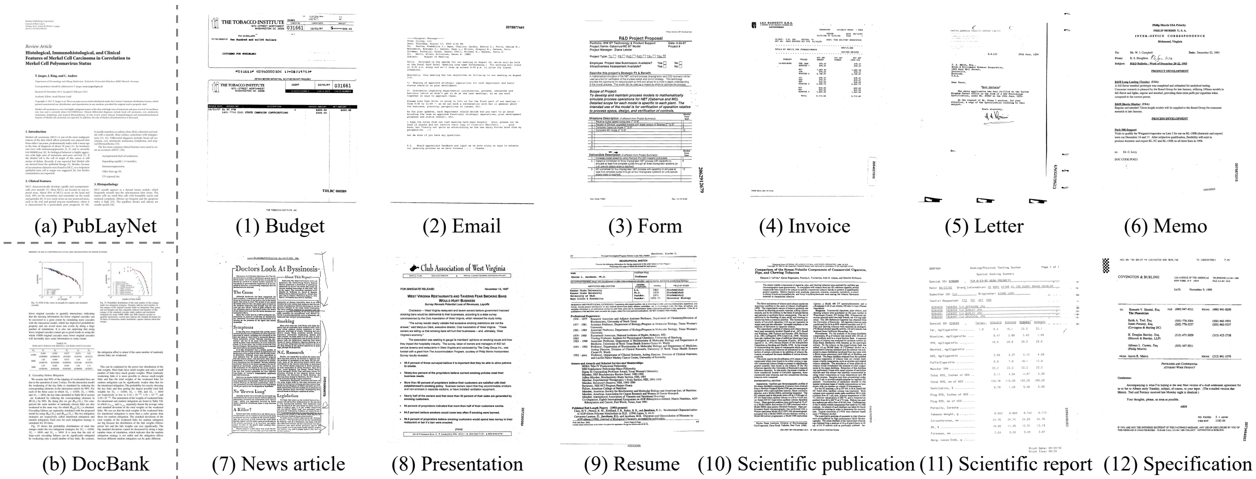


Figure 1. Document examples in the public dataset PubLayNet (a) and DocBank (b) and document examples in real-world applications.

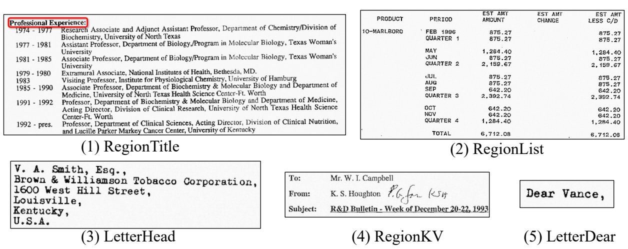


Figure 2. Some special layout categories of D^{4}LA dataset.

vital step to improve the performance of DLA. The differences between existing DLA methods are shown in Table 1.

As a classic Document AI task, there are many datasets for document layout analysis. However, the variety of existing DLA datasets is very limited. The majority of documents are scientific papers, even in the two large-scale DLA datasets PubLayNet [48] and DocBank [26] (in Figure 1 (a) & (b)), which have significantly accelerated the development of document layout analysis recently. As shown in Figure 1, in real-world scenarios, there are diverse types of documents, not limited to scientific papers, but also including letters, forms, invoices, and so on. Furthermore, the document layout categories in existing DLA datasets are tailored to scientific-paper-type documents such as titles, paragraphs, and abstracts. These document layout categories are not diverse enough and thus are not suitable for all types of documents, such as the commonly encountered Key-Value areas in invoices and the line-less list areas in budget sheets. It is evident that a significant gap exists between the existing DLA datasets and the actual document data in the real world, which hinders the further development of document layout analysis and real-world applications.

In this paper, we present **VGT**, a two-stream multi-modal **V**ision **G**rid **T**ransformer for document layout analysis, of which **Gr**id **T**ransformer (GiT) is proposed to directly

Table 2. Comparisons with previous DLA datasets.

Dataset	Type	Category	Labeling	Training	Validation
PubLayNet	1	5	XML	335,703	11,245
DocBank	1	13	LaTeX	400,000	50,000
D^{4}LA	12	27	Manual	8,868	2,224

model $2D$ language information. Specifically, we represent a document as a $2D$ token-level grid as in the grid-based methods [45, 22, 47] and feed the grid into GiT. For better token-level and segment-level semantic awareness, we propose two new pre-training objectives for GiT. First, inspired by BERT [11], the **M**asked **G**rid **L**anguage **M**odeling (MGLM) task is proposed to learn better token-level semantics for grid features, which randomly masks some tokens in the $2D$ grid input, and recovers the original text tokens on the document through its $2D$ spacial context. Second, the **S**egment **L**anguage **M**odeling (SLM) task is proposed to enforce the understanding of segment-level semantics in the grid features, which aims to align the segment-level semantic representations from GiT with pseudo-features generated by existing language model (*e.g.* , BERT [3] or LayoutLM [43]) via contrastive learning. Both token-level and segment-level features are obtained from the $2D$ grid features encoded by GiT via RoIAlign [17], according to the coordinates. Combining image features from **Vi**sion **T**ransformer (ViT) further, VGT can make full use of textual and visual features from GiT and ViT respectively and leverage multi-modal information for better document layout analysis, especially in text-related classes.

In addition, to facilitate the further advancement of DLA research for real-world applications, we propose the **D^{4}LA** dataset, which is the most **D**iverse and **D**etailed **D**ataset ever for **D**ocument **L**ayout **A**nalysis. The differences from the existing datasets for DLA are listed in Table 2. Specifically,

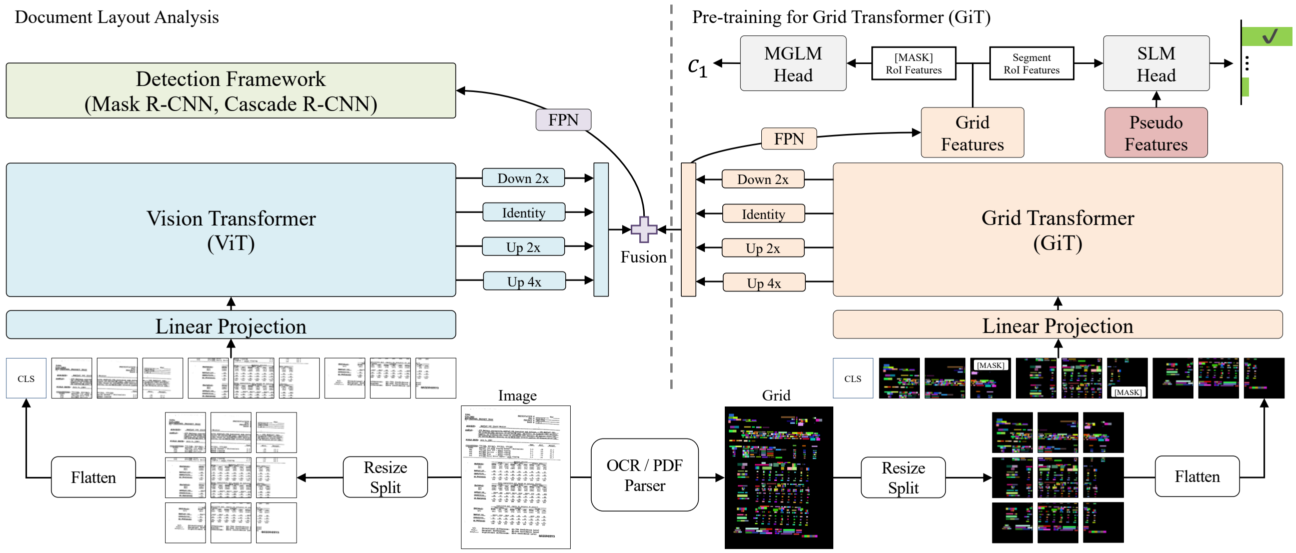


Figure 3. The model architecture of Vision Grid Transformer (VGT) with pre-training objectives for the GiT branch.

D^{4}LA dataset contains 12 types of documents as shown in Figure 1. We define 27 document layout categories and **manually** annotate them. Some special layout classes are illustrated in Figure 2, which are more challenging and text-related. Experiment results on DocBank, PubLayNet and D^{4}LA show the SOTA performance of VGT.

The contributions of this work are as follows:

1) We introduce VGT, a two-stream Vision Grid Transformer for document layout analysis, which can leverage token-level and segment-level semantics in the text grid by two new proposed pre-training tasks: MGLM and SLM. To the best of our knowledge, VGT is the first to explore grid pre-training for $2D$ semantic understanding in documents.
2) A new benchmark named D^{4}LA, which is the most diverse and detailed manually-labeled dataset for document layout analysis, is released. It contains 12 types of documents and 27 document layout categories.
3) Experimental results on the existing DLA benchmarks (PubLayNet and DocBank) and the proposed D^{4}LA dataset demonstrate that the proposed VGT model outperforms previous state-of-the-arts.

2. Related Works

Document Layout Analysis (DLA) is a long-term research topic in computer vision [4]. Previous methods are rule-based approaches [5, 21] that directly use the image pixels or texture for layout analysis. Moreover, some machine learning methods [12, 14] employ low-level visual features for document parsing. Recent deep learning works [37, 34] consider DLA as a classic visual object detection or segmentation problem and utilize convolutional neural networks (CNN) [19] to solve this task [36, 30, 35, 17].

Self-supervised pre-training techniques have given rise to blossom in Document AI [43, 44, 27, 2, 33, 20, 25, 15, 32]. Some document pre-trained models [20, 25, 15] have been applied to the DLA task and achieved good performances. Inspired by BEiT [3], DiT [25] trains a document image transformer for DLA and obtains promising performance, while neglecting the textual information in documents. Unidoc [15] and LayoutLMv3 [20] model the documents in a unified architecture with the text, vision and layout modalities, but they only use the vision backbone without text embeddings for object detection during fine-tuning the downstream DLA task. Different from the methods that regard DLA as a vision task, LayoutLM [43] regards the DLA task as a sequence labeling task to explore DLA only in text modality. The experimental results of LayoutLM show the possibility to use NLP-based methods to process DLA tasks. However, these methods solely use a single modality for the DLA task, and most of them focus on visual information and ignore textual information.

To model the documents in multi-modality for the DLA task, like the grid-based models [23, 10, 29] in vision information extraction, some works [45, 47] use text and layout information to construct the text grid (text embedding map) and combine it with the visual features for DLA. Yang *et al.* [45] build a sentence-level grid that is concatenated with visual features in the model for the DLA task. VSR [47] uses two-stream CNNs where the visual stream and semantic stream take images and text grids (char-level and sentence-level) as inputs, respectively for the DLA task. However, the text grids in these methods are only as model inputs or extra features, there are no semantic supervisions during DLA task training. Therefore, it is difficult to achieve remarkable semantic understandings.

Previous DLA datasets [1, 7, 45] often focus on newspapers, magazines, or technical articles, the size of which is relatively small. Recently, the introduction of large-scale DLA datasets such as DocBank [26] and PubLayNet [48] has promoted significant progress in DLA research. DocBank [26] has 500K documents of scientific papers with 12 types of layout units. PubLayNet [48] includes 360k scientific papers with 5 layout types such as text, title, list, figure, and table. Since the majority of documents of them are scientific papers, the variety of document types and layout categories are very limited. Furthermore, the document layout categories designed for scientific papers in the existing DLA datasets are difficult to transfer to other types of documents in real-world applications.

3. Vision Grid Transformer

The overview of Vision Grid Transformer (VGT) is depicted in Figure 3. VGT employs a Vision Transformer (ViT) [13] and a **Gri**d **T**ransformer (GiT) to extract visual and textual features respectively, resulting in a two-stream framework. Additionally, GiT is pre-trained with the MGLM and SLM objectives to fully explore multi-granularity textual features in a self-supervised and supervised learning manner, respectively. Finally, the fused multi-modal features generated by the multi-scale fusion module are used for layout detection.

3.1. Vision Transformer

Inspired by ViT [13] and DiT [25] , we directly encode image patches as a sequence of patch embeddings for image representation by linear projection. In Figure 3, a resized document image is denoted as $\mathbf{I} \in \mathbb{R}^{H\times W\times C_I}$, where H, W and C_I are the height, width and channel size, respectively. Then, $\mathbf{I}$ is split into non-overlapping $P \times P$ patches, and reshaped into a sequence of flattened $2D$ patches $\mathbf{X}_I \in \mathbb{R}^{N\times(P^2 C_I)}$. We linearly project $\mathbf{X}_I$ into D dimensions, resulting in a sequence of $N = HW/P^2$ tokens as $\mathbf{F}_I \in \mathbb{R}^{N\times D}$. Following [13], standard learnable $1D$ position embeddings and a $[CLS]$ token are injected. The resultant image embeddings serve as the input of ViT.

3.2. Grid Transformer

The architecture of GiT is similar to ViT in Section 3.1, while the input patch embeddings are grid [23, 10]. Concretely, given a document PDF, PDFPlumber [2] is used to extract words with their bounding boxes. While for images, we adopt an open-sourced OCR engine. Then, a tokenizer is used to tokenize the word into sub-word tokens, and the width of the word box is equally split for each sub-word.

[2]https://github.com/jsvine/pdfplumber

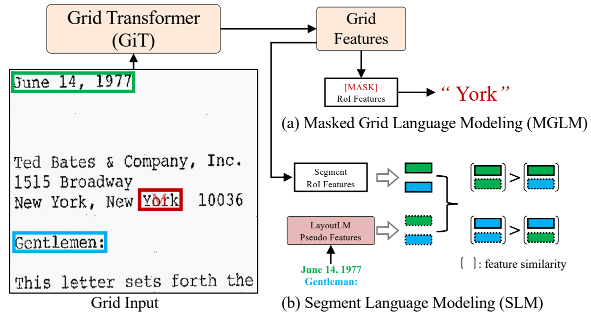


Figure 4. Schematic overview of the pre-training for GiT.

The complete texts are represented as $\mathcal{D} = \{(c_k, \mathbf{b}_k)|k = 0, \dots, n\}$, where c_k denotes the k-th sub-word token in the page and $\mathbf{b}_k$ is the associated box of c_k. Finally, the grid input $\mathbf{G} \in \mathbb{R}^{H\times W\times C_G}$ is constructed as follows:

$$\mathbf{G}_{i,j} = \begin{cases} \mathbf{E}(c_k), & if\ (i,j) \prec \mathbf{b}_k, \\ \mathbf{E}([\text{PAD}]), & otherwise, \end{cases} \tag{1}$$

where $\prec$ means point (i,j) is located in the box $\mathbf{b}_k$ and thus all pixels in $\mathbf{b}_k$ share the same text embedding $\mathbf{E}(c_k)$ of c_k. $\mathbf{E}(\cdot)$ represents an embedding layer, which maps tokens into feature space. The background pixels with non-text are set as the embedding of a special token [PAD].

As in ViT, $\mathbf{G}$ is split into $P \times P$ patches and flattened into a sequence of patches $\mathbf{X}_G \in \mathbb{R}^{N\times(P^2 C_G)}$. We also utilize linear projection to transcribe $\mathbf{X}_G$ into patch embeddings $\mathbf{F}_G \in \mathbb{R}^{N\times D}$. Similarly, $\mathbf{F}_G$ embeddings added with learnable $1D$ position embeddings and a learnable $[CLS]$ token are transferred into GiT, generating $2D$ grid features.

3.3. Pre-Training for Grid Transformer

To facilitate the $2D$ understanding of token-level and segment-level semantics in GiT, we propose Masked Grid Language Modeling (MGLM) and Segment Language Modeling (SLM) objectives for GiT pre-training. Notably, we decouple visual and language pre-training and **ONLY** pre-train GiT within grid inputs in VGT. The reasons are as follows: (1) Flexibility: different pre-training strategies can be used for ViT pre-training, such as ViT [13], BEiT [3] and DiT[25]. (2) Alignment: language information rendered as $2D$ grid features by GiT are naturally well-aligned with image ones in spatial position, and so that the alignment learning in [44, 20] is not that necessary. (3) Efficiency: it can speed up the pre-training process. The schematic overview of the pre-training for GiT is shown in Figure 4.

Masked Grid Language Modeling (MGLM). MLM objective [11] predicts the masked tokens based on the contextual representations, of which the output features are readily accessible from a $1D$ sequence via index. The input and output of GiT, however, are $2D$ feature maps. We extract the region textual feature with RoIAlign [18] as in Region-

CLIP [49]. Specifically, we randomly mask some tokens in $\mathbf{G}$ with [MASK] as the input of GiT, and utilize FPN [28] to generate refined features of GiT as in Figure 3. Then, the region feature $\mathbf{e}_{c_k}$ of masked token c_k is cropped on the largest feature map (*i.e.* P_2 of FPN) by RoIAlign with the box $\mathbf{b}_k$. The pre-training objective is to maximize the log-likelihood of the correct masked tokens c_k based on $\mathbf{e}_{c_k}$ as

$$L_{MGLM}(\theta) = -\sum_{k=1}^{N_M} \log\ p_\theta(c_k \mid \mathbf{e}_{c_k}), \tag{2}$$

where θ is the parameters of GiT and FPN, N_M is the number of masked tokens.

MGLM also differs from variants of MLM used in most of the previous works on layout-aware language modeling. The key difference between them lies in the way they utilize the $2D$ layout information. In MLM variants (*e.g.* , MVLM in LayoutLM [35]), the $2D$ position embeddings of text boxes ($[T, 4, C]$) are explicitly computed and added to the embeddings of text sequences ($[T, C]$), whereas in MGLM the $2D$ spatial arrangement is naturally preserved in the $2D$ grid $\mathbf{G}$ ($[H, W, C]$) and explicit $2D$ position embeddings of text are unnecessary.

Segment Language Modeling (SLM). Token-level representation can be efficiently explored vis MGLM task. However, extremely precise token-level features may not be that crucial in DLA. Segment-level representation is also essential for object detection. Thus, the SLM task is proposed to explore the segment-level feature learning of text. Concretely, we use PDFMiner [3] to extract text lines with bounding boxes as segments. Then, an existing language model (*e.g.* , BERT or LayoutLM) is used to generate the feature $\mathbf{e}^*_{l_i}$ of segment l_i as pseudo-target. The segment feature $\mathbf{e}_{l_i}$ of l_i is produced by RoIAlign with its line box. Given the aligned segment-target feature pairs $\{(\mathbf{e}_{l_i}, \mathbf{e}^*_{l_i})\}$, contrastive loss [49] is used for SLM task, which is computed as

$$p_\theta(\mathbf{e}_{l_i}, \mathbf{e}^*_{l_i}) = \frac{\exp\left(\mathbf{e}_{l_i} \cdot \mathbf{e}^*_{l_i}/\tau\right)}{\exp\left(\mathbf{e}_{l_i} \cdot \mathbf{e}^*_{l_i}/\tau\right) + \sum_{k \in \mathcal{N}_{l_i}} \exp\left(\mathbf{e}_{l_i} \cdot \mathbf{e}^*_{l_k}/\tau\right)}$$

$$L_{SLM}(\theta) = -\frac{1}{N_S}\sum_{i=1} \log\ p_\theta(\mathbf{e}_{l_i}, \mathbf{e}^*_{l_i}).$$

Here, $\cdot$ represents the cosine similarity between segment feature $\mathbf{e}_{l_i}$ from FPN and the pseudo-target feature $\mathbf{e}^*_{l_i}$ generated by language model. $\mathcal{N}_{l_i}$ represents a set of negative samples for segment l_i, and τ is a predefined temperature. We sample N_S segments on one page. Finally, L_{MGLM} and L_{SLM} are equally used for GiT pre-training.

3.4. Multi-Scale Multi-Modal Feature Fusion

FPN [28] framework is widely used to extract multi-scale features in object detection [18]. To adapt the single-scale ViT to the multi-scale framework, we use 4 resolution-modifying modules at different transformer blocks, following [25]. In this way, we obtain multi-scale features of ViT and GiT, denoted as $\{\mathbf{V}_i \in \mathbb{R}^{H/2^i \times W/2^i \times D} | i = 2 \ldots, 5\}$ and $\{\mathbf{S}_i \in \mathbb{R}^{H/2^i \times W/2^i \times D} | i = 2 \ldots, 5\}$, respectively. Then, we fuse the features at each scale i respectively as

$$\mathbf{Z}_i = \mathbf{V}_i \oplus \mathbf{S}_i, \quad i = 2 \ldots, 5, \tag{3}$$

where $\oplus$ represents an element-wise sum function in our implements. Then, we employ FPN to refine pyramid features $\{\mathbf{Z}_i \in \mathbb{R}^{H/2^i \times W/2^i \times D} | i = 2 \ldots, 5\}$ further. Finally, we can extract RoI features from different levels of the feature pyramid according to their scales for later object detection.

4. D^{4}LA Dataset

In this section, we introduce the proposed D^{4}LA dataset.

Document Description. The images of D^{4}LA are from RVL-CDIP [16], which is a large-scale document classification dataset in 16 classes. We choose 12 types documents with rich layouts from it, and sample about $1,000$ images of each type for **manual** annotation. The noisy, handwritten, artistic or less text images are filtered. The OCR results are from IIT-CDIP [24]. The statistics on different document types of D^{4}LA dataset are listed in Table 3.

Category Description. We define detailed 27 layout categories for real-world applications, *i.e.* , DocTitle, ListText, LetterHead, Question, RegionList, TableName, FigureName, Footer, Number, ParaTitle, RegionTitle, LetterDear, OtherText, Abstract, Table, Equation, PageHeader, Catalog, ParaText, Date, LetterSign, RegionKV, Author, Figure, Reference, PageFooter, and PageNumber. For example, we define 2 region categories for information extraction task, *i.e.* , RegionKV is a region that contains Key-Value pairs and RegionList for wireless form as in Figure 2. The statistics of each category of D^{4}LA are listed in Table 4. More detailed descriptions and examples can be found in supplementary materials.

Characteristics. Documents in existing large-scale DLA datasets [48, 26] are mainly scientific papers, where documents in real-world scenarios are not well represented. In contrast, D^{4}LA includes 12 diverse document types and 27 detailed categories. The variety of types and categories is substantially enhanced, closer to the use of real-world applications. Moreover, the image quality of D^{4}LA may be poor, *i.e.* , scanning copies are noisy, skew or blurry. The increased diversities and low-quality scanned document images constitute a more challenging benchmark for DLA.

[3] https://github.com/euske/pdfminer

Table 3. Statistics of different document types of training and validation sets in the D^{4}LA dataset.

Category	**Budget**	**Email**	**Form**	**Invoice**	**Letter**	**Memo**
	845 / 212	780 / 195	650 / 163	574 / 144	793 / 199	817 / 205
Training / Validation	**News article**	**Presentation**	**Resume**	**Scientific publication**	**Scientific report**	**Specification**
	682 / 171	721 / 181	854 / 214	760 / 190	616 / 155	776 / 195

Table 4. Statistics of different layout categories of training and validation sets in the D^{4}LA dataset (#instances / percentage %).

Category	**DocTitle**	**ListText**	**LetterHead**	**Question**	**RegionList**	**TableName**	**FigureName**	**Footer**	**Number**
Training	7391 / 6.30	4581 / 3.90	570 / 0.49	113 / 0.10	3741 / 3.19	640 / 0.55	295 / 0.25	642 / 0.55	7289 / 6.21
Validation	1893 / 6.41	1137 / 3.85	127 / 0.43	56 / 0.19	891 / 3.02	178 / 0.60	85 / 0.29	170 / 0.58	1833 / 6.21
Total	**ParaTitle**	**RegionTitle**	**LetterDear**	**OtherText**	**Abstract**	**Table**	**Equation**	**PageHeader**	**Catalog**
Training	4962 / 4.23	5469 / 4.66	871 / 0.74	15229 / 12.98	807 / 0.69	2733 / 2.33	54 / 0.05	3941 / 3.36	21 / 0.02
117322	1333 / 4.51	1352 / 4.58	228 / 0.77	3703 / 12.54	200 / 0.68	656 / 2.22	20 / 0.07	933 / 3.16	14 / 0.05
Total	**ParaText**	**Date**	**LetterSign**	**RegionKV**	**Author**	**Figure**	**Reference**	**PageFooter**	**PageNumber**
Validation	32328 / 27.55	3148 / 2.68	738 / 0.63	12322 / 10.50	1384 / 1.18	2201 / 1.88	574 / 0.49	3164 / 2.70	2114 / 1.80
29524	8400 / 28.45	786 / 2.66	175 / 0.59	2947 / 9.98	371 / 1.26	592 /2.01	148 / 0.50	797 / 2.70	499 / 1.69

5. Experiments

5.1. Implementation Details

Model Configuration. VGT is built upon two ViT-Base models, which adopt a 12-layer Transformer encoder with 12-head self-attention, $D = 768$ hidden size and $3,072$ intermediate size of MLP [13]. The patch size P is 16 as in [13] for both ViT and GiT. For grid construction, WordPiece tokenizer [11] is used to tokenize words. We initialize the text embeddings of GiT with those of LayoutLM [43] and reduce the embedding size 768 to $C_G = 64$ for memory constraints. **G** has the same shape as the original image.

Model Pre-Training. For ViT pre-training, we directly use the weights of the DiT-base model [25]. For GiT, we also initial it with the weights of the DiT-base model and perform pre-training for GiT on a subset of IIT-CDIP [24] dataset with about 4 million images via MGLM and SLM tasks. Specifically, we randomly set some tokens as [MASK] tokens, and recover the masked tokens on MGLM task as in BERT [11]. For SLM task, we randomly select $N_S = 64$ segments of one page and employ LayoutLM [43] to generate $\mathbf{e}^*$ as pseudo-targets. We use Adam optimizer with 96 batch size for 150,000 steps. We use a learning rate $5e{-}4$ and linearly warm up 2% first steps. The image shape is 768×768 and τ is 0.01.

Model Fine-Tuning. We treat layout analysis as object detection and employ VGT as the feature backbone in the Cascade R-CNN [6] detector with FPN [28], which is implemented based on the Detectron2 [41]. AdamW optimizer with 1,000 warm-up steps is used, and the learning rate is $2e - 4$. We train VGT for 200,000 steps on DocBank and 120,000 steps on PubLayNet with 24 batch size. Since D^{4}LA is relatively small, we train VGT for 10,000 steps on it with 12 batch size. The other settings of Cascade R-CNN are the same with DiT [25].

5.2. Datasets

Besides the proposed D^{4}LA dataset, two benchmarks for document layout analysis are used for evaluation. For the visual object detection task, we use the category-wise and overall mean average precision (mAP) @IOU[0.50:0.95] of bounding boxes as the evaluation metric.

PubLayNet [48] contains 360K research PDFs for DLA released by IBM. The annotation is in object detection format with bounding boxes and polygonal segmentation in 5 layout categories (Text, Title, List, Figure, and Table). Following [15, 25, 48], we train models on the training split (335,703) and evaluate on the validation split (11,245).

DocBank [26] includes 500K document pages with fine-grained token-level annotations released by Microsoft. Moreover, region-level annotations in 13 layout categories (Abstract, Author, Caption, Equation, Figure, Footer, List, Paragraph, Reference, Section, Table and Title) are proposed for object detection. We train models on the training split (400K), and evaluate on the validation split (5K) [26].

Since both PubLayNet and DocBank datasets are relatively large, we construct two sub-datasets PubLayNet2K and DocBank2K that sample $2,000$ images for training and $2,000$ images for validation respectively, to quickly verify the effects of different modules of VGT in the early experiments. We train VGT for $10,000$ steps on them.

5.3. Discussions on GiT

We deeply study the effectivenesses of GiT in Table 5, where (a) is a single-stream baseline with only ViT.

Effectiveness of Only GiT. We can directly employ GiT for layout detection since the grid input of GiT naturally contains the fine-grained layout and textual information.

Table 5. Effect of different modules of GiT on PubLayNet2K and DocBank2K.

Tag	Image Backbone	Grid Backbone	Grid Embedding	PTM	PubLayNet 2K	DocBank 2K
(a)	ViT	-	-	-	86.92	59.61
(b)	ViT	ViT	Image	✗	86.98	58.57
(c)	-	GiT	[UNK]	✗	64.12	40.56
(d)	-	GiT	LayoutLM	✗	65.88	49.15
(e)	-	GiT	LayoutLM	✓	74.96	55.46
(f)	ResNeXt-101	-	-	-	83.54	57.04
(g)	ResNeXt-101	GiT	LayoutLM	✓	85.58	63.05
(h)	ViT	GiT	BERT	✗	87.97	63.97
(i)	ViT	GiT	LayoutLM	✗	87.76	64.01
(j)	ViT	GiT	LayoutLM	✓	**88.44**	**65.94**

To disentangle the effect of layout and text, we use GiT as the feature backbone and set all the sub-word tokens as the [UNK] token in (c), where no textual messages are used. We then adopt the original tokens as input in (d) to verify the effectiveness of the text. Both (c) and (d) are not pre-trained, and we pre-train GiT with MGLM and SLM objectives in (e). We observe that only layout information can produce rough detection results, adding text brings improvements, and pre-training objectives can further exploit the ability of GiT. Notably, DocBank2K contains more linguistic categories than PubLayNet2K, such as “Date”, “Author” and so on. Therefore, the improvement on DocBank2K is more remarkable than that on PubLayNet2K in (d) and (e). These results demonstrate that sub-word layout information can be directly used for layout analysis, textual cues of the grid can indeed facilitate layout analysis, and a well pre-trained Grid Transformer for grid inputs is indispensable.

Effectiveness of VGT. We compare the performance between different word embeddings, *i.e.* BERT [11] in (h) and LayoutLM [43] in (i). The results show that the VGT with both word embeddings can lead to better performance than (a). Since LayoutLM is pre-trained on documents and possesses the capability of layout modeling, we use the embeddings of LayoutLM in the following experiments. Moreover, we introduce a pre-training mechanism in (j), resulting in significant improvements over (i). These results verify the effectiveness of VGT and the pre-training for GiT.

Compatibility of GiT. Due to the decoupling framework of VGT, we can perform solely pre-training for GiT and further integrate GiT with not only ViT but also CNNs. We train a Cascade R-CNN with ResNeXt-101 [42] backbone and FPN in (f) as a baseline. Typically, we construct a hybrid model (g) with ResNeXt-101 and the pre-trained GiT. Clearly, the results of (g) can surpass that of (f), demonstrating the good compatibility of the pre-trained GiT.

Effect of Parameters. Using the two-stream framework inevitably leads to an increase in the number of model parameters. We conduct experiments to analyze the effect of model parameters. In (b), we replace GiT with one ViT, and thus simply construct a two-stream ViT framework with two image inputs. Comparing (b) with (a), introducing more parameters with double image inputs can not enhance the capability of the model. Referring to (g), (h), (i) and (j), GiT models the layout and textual information as a supplement to image information, resulting in obvious improvements.

Table 6. Ablation study of pre-training objectives.

Tag	MGLM	SLM	DocBank2K
(a)	-	-	64.010
(b)	✓	✗	64.539
(c)	✗	✓	65.112
(d)	✓	✓	65.167

Table 7. Document Layout Detection mAP @ IOU [0.50:0.95] on PubLayNet validation set.

Models	Text	Title	List	Table	Figure	mAP
ResNeXt-101	93.0	86.2	94.0	97.6	96.8	93.5
DiT-Base	94.4	88.9	94.8	97.6	96.9	94.5
LayoutLMv3-Base	94.5	90.6	95.5	97.9	97.0	95.1
VSR	**96.7**	93.1	94.7	97.4	96.4	95.7
VGT (ours)	95.0	**93.9**	**96.8**	**98.1**	**97.1**	**96.2**

5.4. Ablation Study of Pre-Training Objectives

We investigate the effect of the proposed pre-training objectives on more text-related DocBank2K in Table 6. Model (a) is the baseline of VGT without pre-training. We pre-train (b) with only MGLM and (c) with only SLM. Then, both MGLM and SLM objectives are utilized for (d). All models are trained on 0.5 million images with $525,000$ steps for experimental efficiency. Model (b) obtains better performance than (a) indicating that predicting masked tokens in MLM objective [11] makes sense in $2D$ textual space. Notably, model (c) attains a small improvement over (b). We speculate that the textual features of segments are more suitable for the layout detection task, and thus segment-level SLM works better than token-level MGLM. Model (d) with MGLM and SLM can achieve the best results.

5.5. Comparison with State-of-the-arts

We evaluate the performance of VGT on three datasets, namely PubLayNet, DocBank, and our proposed D^4LA.

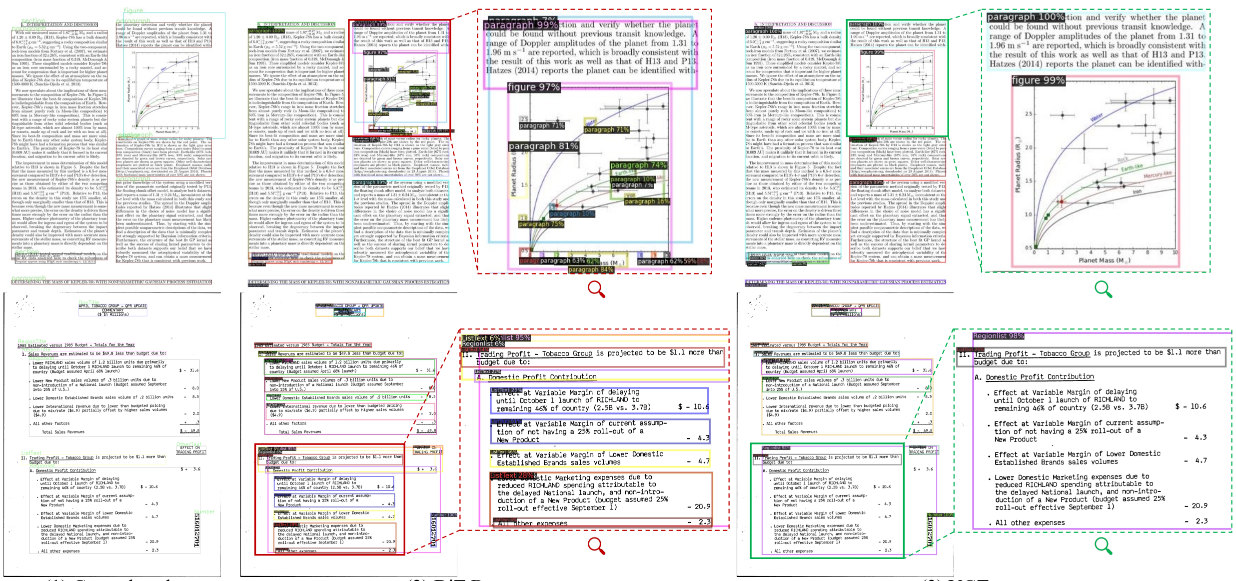
PubLayNet. The results of document layout detection on PubLayNet are reported in Table 7. Generally, PubLayNet contains 5 relatively simple layout categories, of which the visual information may be sufficient for layout detection. Thus, all of the methods can obtain promising mAPs ($> 90\%$). The results of DiT-Base are better than that of ResNeXt-101, showing the powerful capabil-

Table 8. Document Layout Detection mAP @ IOU [0.50:0.95] on DocBank validation set.

Models	Abstract	Author	Caption	Date	Equation	Figure	Footer	List	Paragraph	Reference	Section	Table	Title	mAP
ResNeXt-101	89.7	72.6	82.3	69.7	76.4	73.6	78.2	78.3	66.2	81.7	75.9	77.3	84.1	77.4
DiT-Base	91.1	75.4	83.1	73.4	77.8	75.7	80.2	82.7	67.3	83.8	77.0	80.8	86.8	79.6
LayoutLMv3-Base	90.5	73.6	81.2	73.5	76.0	74.4	78.1	80.7	65.8	82.8	76.6	78.6	86.3	78.3
VGT (ours)	**92.4**	**79.9**	**88.8**	**79.1**	**86.7**	**76.6**	**84.8**	**88.6**	**75.8**	**85.6**	**81.5**	**83.9**	**89.8**	**84.1**

Table 9. Document Layout Detection mAP @ IOU [0.50:0.95] on D^{4}LA validation set.

Models	**DocTitle**	**ListText**	**LetterHead**	**Question**	**RegionList**	**TableName**	**FigureName**	**Footer**	**Number**	
ResNeXt-101	70.6	71.0	**82.8**	48.4	76.1	66.0	45.9	76.2	83.0	
DiT-Base	**73.1**	70.6	82.2	55.0	80.1	68.4	**51.8**	**81.2**	83.2	
LayoutLMv3-Base	66.8	56.5	78.5	39.3	72.1	64.3	32.1	72.2	82.1	
VGT (ours)	72.6	**71.3**	82.3	**63.9**	**80.2**	**68.4**	46.6	79.7	**83.2**	
Models	**ParaTitle**	**RegionTitle**	**LetterDear**	**OtherText**	**Abstract**	**Table**	**Equation**	**PageHeader**	**Catalog**	
ResNeXt-101	60.3	63.8	73.4	56.4	65.7	**86.3**	11.5	53.7	32.0	
DiT-Base	**63.2**	**67.5**	74.5	59.2	73.8	86.2	9.2	56.5	44.8	
LayoutLMv3-Base	55.6	59.5	70.8	50.8	68.2	80.6	7.3	53.1	37.3	
VGT (ours)	63.0	67.2	**76.7**	**60.0**	**80.4**	86.0	**19.9**	**56.9**	**40.9**	
Models	**ParaText**	**Date**	**LetterSign**	**RegionKV**	**Author**	**Figure**	**Reference**	**PageFooter**	**PageNumber**	**mAP**
ResNeXt-101	85.2	68.4	69.3	68.2	62.6	76.7	83.4	62.2	57.9	65.1
DiT-Base	**86.4**	69.7	71.6	68.8	66.0	**77.2**	83.4	65.5	58.3	67.7
LayoutLMv3-Base	81.6	62.5	60.4	59.4	59.3	72.2	74.9	62.1	52.8	60.5
VGT (ours)	86.2	**71.3**	**75.5**	**70.1**	**67.6**	76.7	**85.6**	**66.5**	**58.7**	**68.8**



(1) Ground truth (2) DiT-Base (3) VGT

Figure 5. Qualitative comparison between DiT-Base and VGT on DocBank (1st row) and D^{4}LA (2nd row). Best viewed in color.

ity of pre-trained ViT. LayoutLMv3 pre-trains multi-modal Transformer with unified text and image masking objectives. We feed only image tokens into LayoutLMv3 without text embeddings as in [20]. LayoutLMv3 achieves better results than pure visual methods (DiT-Base and ResNeXt-101), especially in "Title" and "List" classes. Grid-based VSR [47] method presents better results in "Title" class, showing the effectiveness of grid for text-related classes. VGT achieves the best average mAPs and presents a remarkable improvement in "Title" and "List" classes over VSR. We attribute this improvement to the textual modeling of GiT and the pre-training objectives.

DocBank. We measure the mAP performance on DocBank and list the results in Table 8. The methods in Table 8 are implemented by the original codes, and the Cascade R-CNN is used for layout detection. Since DocBank provides more detailed layout categories than PubLayNet, the mAPs of the text-related categories of DocBank might reflect the ability of textual modeling. Clearly, the performance of DiT-Base is still better than that of ResNeXt-101, showing the superiority of ViT backbone again. LayoutLMv3 obtains a little worse result than DiT-Base. Since LayoutLMv3 is pre-trained with text but no explicit text embeddings are used in DLA task, we conjecture that the text information may be insufficient for detailed detection on DocBank. VGT obtains the best mAPs of text-related categories and exhibits substantial improvement over other methods, such as "Caption", "Date", "Equation", "List" and "Paragraph", verifying the effectiveness of VGT.

D^{4}LA. Due to the more diverse document types and detailed categories of D^{4}LA, the detection results reported in Table 9 are relatively lower, compared with the mAPs of PubLayNet and DocBank. It reveals that the existing methods do not work well on real-world documents. Similarly, the DiT-Base backbone achieves better performance than ResNeXt-101 in D^{4}LA, due to the well-designed image pre-training objective on documents. LayoutLMv3 obtains worse results than DiT-Base, especially in the text-related classes. VGT achieves the best results in most categories. The significant improvements on text-related categories (*e.g.* , 6.6% on "Abstract" over DiT-Base) verify the superiority of VGT on textual modeling in $2D$ fashion.

5.6. Visualization Cases

We illustrate the detection results of DiT-Base and VGT on samples from DocBank and D^{4}LA in Figure 5. For the sample of DocBank, the text in the chart is misidentified as "Paragraph" in DiT-Base, while VGT removes the predictions of them and produces a precise box of "Figure". This is because there is no text in the chart for grid construction, beneficial to false positive reduction. For the sample of D^{4}LA, the predictions of "ListText" are drastically reduced. Visually, these regions alone are indeed like "ListText" regions. However, they constitute a "RegionList" from the contextual semantics. These qualitative results demonstrate the ability of $2D$ language modeling in VGT.

6. Limitations

Due to the two-stream framework, VGT contains 243M parameters, relatively larger than DiT-Base (138M), and LayoutLMv3 (138M). The inference time of VGT (460ms) is relatively longer than DiT-Base (210 ms) and LayoutLMv3 (270 ms). Thus, a more lightweight and efficient architecture will be our future work. Moreover, since VGT is an image-centric method, we will extend VGT to text-centric tasks, such as information extraction in the future.

7. Conclusion

In this paper, we present VGT, a two-stream Vision Grid Transformer for document layout analysis. VGT is an image-centric method, that is more compatible with object detection. The Grid Transformer of VGT is pre-trained by MGLM and SLM objectives for $2D$ token-level and segment-level semantic understanding. In addition, we propose a new dataset D^{4}LA, which is the most diverse and detailed manually-annotated benchmark ever for document layout analysis. Experimental results show that VGT achieves state-of-the-art results on PubLayNet, DocBank and the proposed D^{4}LA.

References

[1] Apostolos Antonacopoulos, David Bridson, Christos Papadopoulos, and Stefan Pletschacher. A realistic dataset for performance evaluation of document layout analysis. In *ICDAR*, pages 296–300, 2009.

[2] Srikar Appalaraju, Bhavan Jasani, and Bhargava Urala Kota. DocFormer: End-to-end transformer for document understanding. In *ICCV*, pages 4171–4186, 2021.

[3] Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. Beit: Bert pre-training of image transformers. *arXiv preprint arXiv:2106.08254*, 2021.

[4] Galal M Binmakhashen and Sabri A Mahmoud. Document layout analysis: a comprehensive survey. *ACM Computing Surveys (CSUR)*, 52(6):1–36, 2019.

[5] Frank Le Bourgeois, Zbigniew Bublinski, and Hubert Emptoz. A fast and efficient method for extracting text paragraphs and graphics from unconstrained documents. In *ICPR*, pages 272–276, 1992.

[6] Zhaowei Cai and Nuno Vasconcelos. Cascade R-CNN: delving into high quality object detection. In *CVPR*, pages 6154–6162, 2018.

[7] Christian Clausner, Christos Papadopoulos, Stefan Pletschacher, and Apostolos Antonacopoulos. The ENP image and ground truth dataset of historical newspapers. In *ICDAR*, pages 931–935, 2015.

[8] Lei Cui, Yiheng Xu, Tengchao Lv, and Furu Wei. Document AI: Benchmarks, models and applications. *arXiv preprint arXiv:2111.08609*, 2021.

[9] Cheng Da, Peng Wang, and Cong Yao. Levenshtein OCR. In *ECCV*, volume 13688, pages 322–338, 2022.

[10] Timo I Denk and Christian Reisswig. Bertgrid: Contextualized embedding for 2d document representation and understanding. *arXiv preprint arXiv:1909.04948*, 2019.

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

[12] Markus Diem, Florian Kleber, and Robert Sablatnig. Text classification and document layout analysis of paper fragments. In *ICDAR*, pages 854–858, 2011.

[13] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner,

Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021.

[14] Angelika Garz, Markus Diem, and Robert Sablatnig. Detecting text areas and decorative elements in ancient manuscripts. In *ICFHR*, pages 176–181, 2010.

[15] 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. *NIPS*, 34:39–50, 2021.

[16] Adam W. Harley, Alex Ufkes, and Konstantinos G. Derpanis. Evaluation of deep convolutional nets for document image classification and retrieval. In *ICDAR*, pages 991–995, 2015.

[17] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, pages 2961–2969, 2017.

[18] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask R-CNN. In *ICCV*, pages 2980–2988, 2017.

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[20] Yupan Huang, Tengchao Lv, Lei Cui, Yutong Lu, and Furu Wei. Layoutlmv3: Pre-training for document ai with unified text and image masking. In *ACM Multimedia*, 2022.

[21] Nicholas Journet, Véronique Eglin, Jean-Yves Ramel, and Rémy Mullot. Text/graphic labelling of ancient printed documents. In *ICDAR*, pages 1010–1014, 2005.

[22] Frédéric Kaplan, Sofia Ares Oliveira, Simon Clematide, Maud Ehrmann, and Raphaël Barman. Combining visual and textual features for semantic segmentation of historical newspapers. *Journal of Data Mining & Digital Humanities*, 2021.

[23] Anoop Raveendra Katti, Christian Reisswig, Cordula Guder, Sebastian Brarda, Steffen Bickel, Johannes Höhne, and Jean Baptiste Faddoul. Chargrid: Towards understanding 2d documents. In *EMNLP*, 2018.

[24] David Lewis, Gady Agam, and Shlomo Argamon. Building a test collection for complex document information processing. In *SIGIR*, pages 665–666, 2006.

[25] Junlong Li, Yiheng Xu, Tengchao Lv, Lei Cui, Cha Zhang, and Furu Wei. DiT: Self-supervised pre-training for document image Transformer. In *ACM Multimedia*, pages 3530–3539, 2022.

[26] Minghao Li, Yiheng Xu, Lei Cui, Shaohan Huang, Furu Wei, Zhoujun Li, and Ming Zhou. Docbank: A benchmark dataset for document layout analysis. *COLING*, 2020.

[27] Yulin Li, Yuxi Qian, Yuechen Yu, Xiameng Qin, Chengquan Zhang, Yan Liu, Kun Yao, Junyu Han, Jingtuo Liu, and Errui Ding. Structext: Structured text understanding with multi-modal transformers. In *ACM Multimedia*, pages 1912–1920, 2021.

[28] Tsung-Yi Lin, Piotr Dollár, Ross B. Girshick, Kaiming He, Bharath Hariharan, and Serge J. Belongie. Feature pyramid networks for object detection. In *CVPR*, pages 936–944, 2017.

[29] Weihong Lin, Qifang Gao, Lei Sun, Zhuoyao Zhong, Kai Hu, Qin Ren, and Qiang Huo. Vibertgrid: a jointly trained multi-modal 2d document representation for key information extraction from documents. In *ICDAR*, pages 548–563, 2021.

[30] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *ECCV*, pages 21–37, 2016.

[31] Shangbang Long, Xin He, and Cong Yao. Scene text detection and recognition: The deep learning era. *International Journal of Computer Vision*, 129:161–184, 2018.

[32] Chuwei Luo, Changxu Cheng, Qi Zheng, and Cong Yao. Geolayoutlm: Geometric pre-training for visual information extraction. In *CVPR*, pages 7092–7101, 2023.

[33] Chuwei Luo, Guozhi Tang, Qi Zheng, Cong Yao, Lianwen Jin, Chenliang Li, Yang Xue, and Luo Si. Bi-vldoc: Bidirectional vision-language modeling for visually-rich document understanding. *arXiv preprint arXiv:2206.13155*, 2022.

[34] Devashish Prasad, Ayan Gadpal, Kshitij Kapadni, Manish Visave, and Kavita Sultanpure. Cascadetabnet: An approach for end to end table detection and structure recognition from image-based documents. In *CVPR Workshops*, pages 2439–2447, 2020.

[35] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

[36] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *NIPS*, 28, 2015.

[37] Sebastian Schreiber, Stefan Agne, Ivo Wolf, Andreas Dengel, and Sheraz Ahmed. Deepdesrt: Deep learning for detection and structure recognition of tables in document images. In *ICDAR*, pages 1162–1167, 2017.

[38] Baoguang Shi, Xiang Bai, and Cong Yao. An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39:2298–2304, 2015.

[39] Baoguang Shi, Mingkun Yang, Xinggang Wang, Pengyuan Lyu, Cong Yao, and Xiang Bai. Aster: An attentional scene text recognizer with flexible rectification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41:2035–2048, 2019.

[40] Peng Wang, Cheng Da, and Cong Yao. Multi-granularity prediction for scene text recognition. In *ECCV*, volume 13688, pages 339–355, 2022.

[41] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. `https://github.com/facebookresearch/detectron2`, 2019.

[42] Saining Xie, Ross B. Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *CVPR*, pages 5987–5995, 2017.

[43] Yiheng Xu, Minghao Li, Lei Cui, and Shaohan Huang. LayoutLM: Pre-training of text and layout for document image understanding. In *KDD*, pages 1192–1200, 2020.

[44] Yang Xu, Yiheng Xu, Tengchao Lv, Lei Cui, Furu Wei, Guoxin Wang, Yijuan Lu, Dinei Florencio, Cha Zhang, Wanxiang Che, Min Zhang, and Lidong Zhou. Layoutlmv2: Multi-modal pre-training for visually-rich document understanding. In *ACL*, 2021.

[45] Xiao Yang, Ersin Yumer, Paul Asente, Mike Kraley, Daniel Kifer, and C Lee Giles. Learning to extract semantic structure from documents using multimodal fully convolutional neural networks. In *CVPR*, pages 5315–5324, 2017.

[46] Zhibo Yang, Rujiao Long, Pengfei Wang, Sibo Song, Humen Zhong, Wenqing Cheng, Xiang Bai, and Cong Yao. Modeling entities as semantic points for visual information extraction in the wild. In *CVPR*, pages 15358–15367, 2023.

[47] Peng Zhang, Can Li, Liang Qiao, Zhanzhan Cheng, Shiliang Pu, Yi Niu, and Fei Wu. Vsr: a unified framework for document layout analysis combining vision, semantics and relations. In *ICDAR*, pages 115–130, 2021.

[48] Xu Zhong, Jianbin Tang, and Antonio Jimeno Yepes. Publaynet: largest dataset ever for document layout analysis. In *ICDAR*, pages 1015–1022, 2019.

[49] Yiwu Zhong, Jianwei Yang, Pengchuan Zhang, Chunyuan Li, Noel Codella, Liunian Harold Li, Luowei Zhou, Xiyang Dai, Lu Yuan, Yin Li, and Jianfeng Gao. Regionclip: Region-based language-image pretraining. In *CVPR*, pages 16772–16782, 2022.

[50] Xinyu Zhou, Cong Yao, He Wen, Yuzhi Wang, Shuchang Zhou, Weiran He, and Jiajun Liang. East: An efficient and accurate scene text detector. *CVPR*, pages 2642–2651, 2017.

[51] Yingying Zhu, Cong Yao, and Xiang Bai. Scene text detection and recognition: recent advances and future trends. *Frontiers of Computer Science*, 10:19–36, 2015.