M2Doc: A Multi-Modal Fusion Approach for Document Layout Analysis

Ning Zhang[1,2], **Hiuyi Cheng**[1], **Jiayu Chen**[2], **Zongyuan Jiang**[1],
Jun Huang[2], **Yang Xue**[1], **Lianwen Jin**[1, 3*]

[1] South China University of Technology
[2] Platform of AI(PAI), Alibaba Group
[3] SCUT-Zhuhai Institute of Modern Industrial Innovation, Zhuhai, China
johnning2333@gmail.com, {eechenghiuyi, eejiangzongyuan}@mail.scut.edu.cn,
{yunji.cjy, huangjun.hj}@alibaba-inc.com, {yxue, eelwjin}@scut.edu.cn

Abstract

Document layout analysis is a crucial step for intelligent document understanding. However, many existing methods primarily focus on the visual aspects and overlook the textual features of documents. Although document pre-trained models utilize multi-modal features during the pre-training phase, they tend to operate as a unimodal pipeline when it comes to layout analysis tasks. Furthermore, current multi-modal methods perform worse than unimodal detectors on complex layout analysis datasets. To address these limitations, we propose an effective and pluggable multi-modal fusion approach named M2Doc, which fuses visual and textual features for better layout detection. M2Doc contains two pluggable multi-modal fusion modules, early-fusion and late-fusion, which align and fuse visual and textual features at the pixel level and block level. Benefitting from the concision and effectiveness of M2Doc, it can be easily applied to various detectors for better layout detection, including two-stage and end-to-end object detectors. Our experimental results demonstrate significant performance improvements in detectors equipped with M2Doc on datasets such as DocLayNet (+11.3 mAP) and M6Doc (+1.9 mAP). Furthermore, through the integration of the DINO detector with M2Doc, we achieve state-of-the-art results on DocLayNet (89.0 mAP), M6Doc (69.9 mAP), and PubLayNet (95.5 mAP). The code will be publicly released at https://github.com/johnning2333/M2Doc.

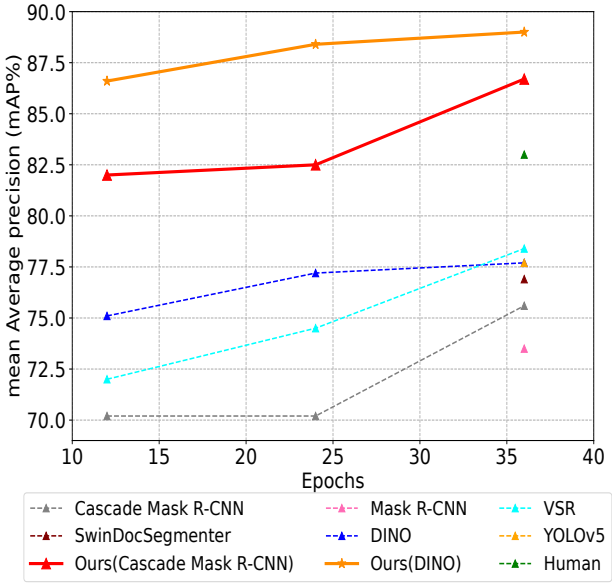

Figure 1: The mAP curves on DocLayNet test set for our method and previous methods.

Introduction

Document layout analysis (DLA) is a fundamental task in document understanding (Namboodiri and Jain 2007), which aims to detect and segment different types of regions and analyze their relationships within document image. DLA can be divided into two categories, physical and logical layout analysis (Lee et al. 2019). Physical layout analysis focuses on detecting the fundamental blocks within a document, such as text, figure, and table. Representative datasets include PubLayNet (Zhong, Tang, and Jimeno Yepes 2019) and PubMed (Li et al. 2020a). Logical layout analysis requires a finer-grained layout detection based on the document's structures and content semantics, where representative datasets include PRImA (Antonacopoulos et al. 2009), DocBank (Li et al. 2020b), DocLayNet (Pfitzmann et al. 2022), and M^{6}Doc (Hiuyi et al. 2023).

Many current DLA models, such as TransDLANet (Hiuyi et al. 2023), SelfDocSeg (Subhajit et al. 2023), and SwinDocSegmenter (Ayan et al. 2023) focus on enhancing generic object detectors to more suitably match layout analysis tasks. However, these models tend to rely solely on visual features while overlooking textual features of documents. In recent years, self-supervised models such as LayoutLM (Yupan et al. 2022) and StructText (Yu et al. 2023) have demonstrated remarkable progress in a variety of Document AI tasks. These models primarily focus on developing better pre-training tasks to align cross-modal features and enhance models' ability to represent multiple modalities. Despite incorporating various modality inputs and applying multiple pretext tasks during the pre-training phase, these models are only used to initialize the backbone of generic object detectors when transferred to layout analysis tasks. Essentially, these pipelines treat DLA as an image-centric

*Corresponding author.

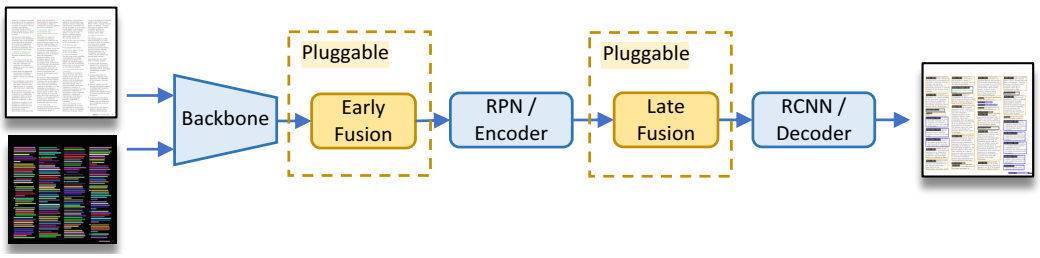

Figure 2: The overall framework of M2Doc plugs into detectors.

object detection problem rather than a multi-modal problem.

Currently, numerous multi-modal DLA models are in the process of being developed, with VSR (Peng et al. 2021) being a representative example. VSR employs a complex network, incorporating multiple granularity textual modality inputs, a two-stream backbone, and Transformer layers for relation modeling. However, VSR exhibits limited effectiveness and occasionally performs worse than unimodal object detectors when applied to complex logical layout analysis.

Considering the aforementioned limitations and issues, we have rethought the distinctions between generic object detection and DLA, and have identified two main distinctions: (1) DLA scenarios mostly involve rich text documents, which makes it more appropriate and intuitive to use multi-modal methods; (2) The textual instances in documents contain connectivity and logical relationships. For instance, text positioned beneath an image is likely to be a caption, while instances that are contextually connected are likely to belong to the same category. Considering the abundant textual content and logical relationships in the majority of DLA application scenarios, a multi-modal model of combining visual and textual features is a promising solution.

To this end, we propose an effective and pluggable multi-modal fusion approach named M2Doc, which aims to convert unimodal detectors into multi-modal detectors for DLA tasks. As illustrated in Fig 2, it can be easily implemented on both two-stage and end-to-end detectors. Firstly, we obtain textual grid representations using a pre-trained language model BERT (Jacob et al. 2019). As textual representation is aligned to visual representation at pixel level, we use a single backbone to extract both textual and visual features. Specifically, we densely fuse each scale visual and textual feature using early-fusion module. Additionally, we use late-fusion module to explicitly align block-level visual and textual features. By combining these early-fusion and late-fusion modules within the M2Doc approach, we effectively align and fuse visual and textual features at both pixel and block levels, enabling improved performance in DLA tasks.

To validate the effectiveness of our proposed approach, we conducted extensive experiments on physical and logical layout analysis datasets. Our results demonstrate that Cascade Mask R-CNN (Cai and Vasconcelos 2018) and DINO (Hao et al. 2023) show promising improvements with the use of M2Doc on the DocLayNet dataset, as shown in Fig. 1. Moreover, ablation study results show that various detectors can benefit from M2Doc.

The contributions of this paper are summarized as follows:

- To endow existing unimodal detectors with multi-modal capabilities when handling DLA tasks, we propose a pluggable multi-modal fusion approach M2Doc.
- M2Doc can be easily integrated into existing two-stage and end-to-end detectors. Our experimental results indicate that many detectors benefit from M2Doc, showcasing its versatility and wide applicability.
- Our experimental results demonstrate that DINO with M2Doc outperforms previous models by large margin and achieves state-of-the-art performance on complex logical layout analysis datasets.

Related Work

This paper analyzes the document layout analysis task from the perspective of the modalities used, including unimodal and multi-modal models.

Unimodal Document Layout Analysis

Unimodal layout analysis utilizes visual features to analyze document layout. Several approaches attempt to utilize object detection and instance segmentation methods to detect and segment document regions.

PubLayNet (Zhong, Tang, and Jimeno Yepes 2019) directly use Faster R-CNN (Ren et al. 2015) and Mask-RCNN (He et al. 2017) for paper layout analysis. Lee et al. (Lee et al. 2019) propose a trainable multiplication layers combined with U-Net (Ronneberger, Fischer, and Brox 2015). Li et al. (Li, Yin, and Liu 2020) add a domain adaptation module based on Faster R-CNN. TransDLANet (Hiuyi et al. 2023) uses three parameter-shared multi-layer-perception (MLP) on top of ISTR. SwinDocSegmenter (Ayan et al. 2023) utilizes both high-level and low-level features of document images to initialize the query of DINO. SelfDocSeg (Subhajit et al. 2023) generates pseudo-layouts to pretrain the image encoder before fine-tuning on layout analysis datasets following BYOL (Grill et al. 2020).

Although researchers have attempted to improve the performance of original models, these approaches unable to utilize the semantic information of document.

In recent years, many researchers have attempted to introduce multi-modal information from a self-supervised perspective. Such as LayoutLM (Yupan et al. 2022), BEiT (Hangbo et al. 2022), DiT (Li et al. 2022), UDoc (Jiuxiang

et al. 2021), and StrucText (Yu et al. 2023). The training of these models comprises two phases: (1) the pre-training phase, where models are self-supervised training on massive unlabeled data utilizing multi-modal inputs; (2) the fine-tuning phase, in which models are supervised training on labeled data to complete downstream tasks using pre-trained weights. With sufficient pre-training on enormous document data, these models perform well when transferred to various downstream document AI tasks, including image classification, layout analysis, and information extraction. For instance, DiT is self-supervised trained on 42 million document data using Masked Image Modeling (MIM) as its pre-training objective. However, during the fine-tuning phase, it merely integrates the well-pre-trained model as a feature backbone of Cascade R-CNN (Cai and Vasconcelos 2018). Despite detector benefiting from DiT, it remains essentially an unimodal pipeline, without utilizing semantic features associated with the downstream DLA datasets.

Multi-Modal Document Layout Analysis

Multi-modal DLA models focus on utilizing the multi-modal features of text blocks. For instance, MFCN (Yang et al. 2017) uses skip-gram to obtain sentence-level textual features and combines textual and visual features in the decoder. VSR (Peng et al. 2021) combines three granularity: Chargrid (Katti et al. 2018), Wordgrid, and Sentencegrid (Denk and Reisswig 2019), into the full text embedding maps as textual input. It then uses two backbones to extract visual and textual features, which are fused in the multi-scale-adaptive-aggregation (MSAA) module. Furthermore, VSR emphasizes the importance of relation modeling for layout analysis, which uses Transformer layers (Vaswani et al. 2017) to model the relation of text blocks for further feature enhancement.

These methods have the problem of shallow modality fusion or complex network structure, which makes their robustness and effectiveness compromised on complex datasets.

Method

M2Doc is a pluggable approach that can be directly applied to enhance the existing document layout analysis detectors, as shown in Fig 2. The detail architecture of M2Doc is depicted in Fig. 3 (a). The main pipeline of our method consists of four phases: (1) Textual Grid Representation, where a pre-trained BERT is utilized to convert images to textual grid representations; (2) Feature Extraction, employing a single backbone to extract both visual features and textual features; (3) Early Fusion, where textual and visual features are fused at corresponding scales; (4) Late Fusion, fusing the visual features and textual features of text blocks, which is determined based on the Intersection over Union (IoU) of the candidate bounding boxes and optical character recognition (OCR) bounding boxes. Finally, the layout bounding boxes and categories are predicted based on the fused features.

Textual Grid Representations

Considering a document image $\mathbf{I} \in \mathbb{R}^{H\times W\times 3}$ with N words, where H and W represent the height and width of image. And we have follow-up labels (w_i, b_i), for $i \in \{1, \ldots, N\}$, where w_i originating from the OCR result, it signifies the i-th word or sentence. And $b_i = [(x_i^1, y_i^1), (x_i^2, y_i^2)]$ indicates the coordinates of the top-left and bottom-right corners of the bounding boxes corresponding to the i-th word. To obtain the textual representation of the document, we refer to the information extraction method BERTgrid (Denk and Reisswig 2019). We align the OCR results one by one and input them into the pre-trained BERT to generate sequential text embeddings $T_i \in \mathbb{R}^{d\times 1}$, where d represents the feature dimension of the BERT model, with a value of 768, as shown in Eq.(1).

$$(T_1, \ldots, T_N) = \text{BERT}(w_1, \ldots, w_N) \tag{1}$$

Finally, we transform the sequential text embeddings T_i into a 2D grid representation $\mathbf{G} \in \mathbb{R}^{H\times W\times d}$ based on b_i, which is defined as follows,

$$\mathbf{G}_{\mathbf{x},\mathbf{y}} = \begin{cases} T_i, & \text{if}(x, y) \in b_i \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

This grid representation maximizes the preservation of the documents' layout and aligns the textual grid representation $\mathbf{G}$ with the original image $\mathbf{I}$ at pixel level.

Feature Extraction

We employ a single backbone to extract both textual and visual features, as shown in Fig. 3. In contrast to VSR, we contend that the utilization of two distinct backbones is unnecessary due to the precise pixel-level alignment between the visual and textual inputs. Consequently, our proposed method has better generality and demands fewer model parameters in comparison to VSR.

We use convolution layers to align the channel dimensions of visual and textual inputs before feeding them into the first ResNet block (Kaiming et al. 2016). Subsequently, downsampling operations are performed in four ResNet blocks to obtain features at different scales, where each scale becomes $\{1/4, 1/8, 1/16, 1/32\}$ of the original input. And we obtain the corresponding visual features P_θ and textual features S_θ, where $\theta \in \{1, 2, 3, 4\}$.

Early Fusion

Due to end-to-end detectors requires the utilization of features extracted by the backbone for generating anchors, and two-stage detectors necessitates the application of RoIAlign to select appropriate features from backbone. So modality fusion is essential before inputting the features to the Region Proposal Network (RPN) (Ren et al. 2015) or Transformer encoder.

We adopt the gate-like mechanism from the referring image segmentation model LAVT (Zhao et al. 2022) to obtain fusion scores that can adaptively vary with S_θ.

$$\alpha_\theta = \eta(S_\theta) \tag{3}$$

$$F_\theta = \text{LayerNorm}(\alpha_\theta \odot S_\theta + P_\theta) \tag{4}$$

where $\odot$ means element-wise multiplication, and $\eta(\cdot)$ refers to a function consisting of two 1×1 convolutional layers followed by two activation layers. To calculate the score of S_θ,

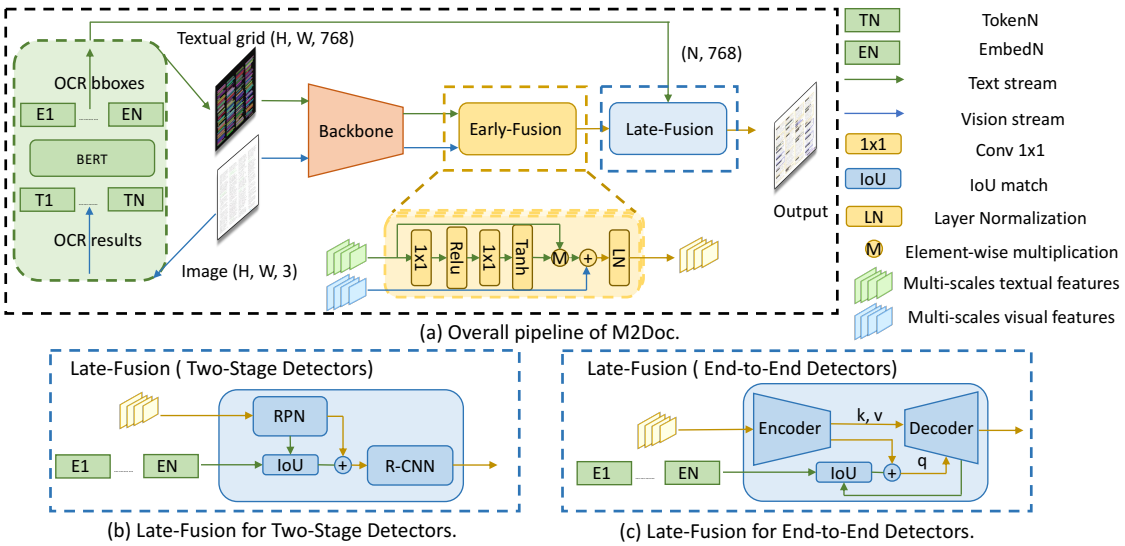

Figure 3: The pipeline of our proposed method. The modules with green, red, yellow, and blue backgrounds represent the Textual Grid Representation, Feature Extraction, Early Fusion, and Late Fusion, respectively.

we use a 1×1 convolutional layer followed by a ReLU activation layer (Nair and Hinton 2010). We then apply another 1×1 convolutional layer and a Tanh activation layer to restrict the score to the range of $(0, 1)$. We obtain the weighted textual feature by multiplying the score α_θ with S_θ. Finally, we add the weighted textual feature to the visual features P_θ to obtain the fused feature F_θ.

Since the textual grid representation $\mathbf{G}$ equals 0 in pixels without text as given in Eq.(2), we normalize the fused features using a LayerNorm normalization layer. This normalization ensures the distribution of the fused features is more consistent. After the early fusion phase, well-fused features F_θ are generated.

Late Fusion

After feeding F_θ into either the RPN or Transformer encoder, we generate numerous candidate bounding boxes. We then fuse features based on the candidate bounding boxes. Specifically, we fuse the visual features P_θ with the assigned block-level textual features for each candidate bounding box. This fusion allows us to obtain more accurate predictions of the bounding box locations and classifications.

Since we have candidate bounding boxes and sequential text embeddings T_i based on Eq.(1), we can assign each candidate bounding box its own block-level textual features through an IoU matching operation.

Accounting for the distinct networks employed in generating candidate bounding boxes, we will discuss end-to-end detectors and two-stage detectors separately.

End-to-End Detectors We use DINO (Hao et al. 2023) as the representative end-to-end detector for illustration. As shown in Fig. 3 (c), DINO initially flattens the fused features with corresponding positional encodings and feeds them into the Transformer encoder layers to enhance the feature representation. The output of the Transformer encoder serves as the keys and values of each layer of the Transformer decoder. With regards to the queries of the decoder layers, DINO splits them into two parts, positonal queries and content queries. The positional queries explicitly indicate the position of candidate bounding boxes, while the content queries are obtained through learnable embeddings that represent the features of candidate bounding boxes. In each decoder layer, DINO refines the positions and categories of candidate bounding boxes gradually.

To enhance the multi-modal feature representations of the content queries, we calculate the $\text{IoU}_{i,j}$ between the predicted candidate boxes r_j where $j = \{1, \ldots, K\}$ and the OCR bounding boxes b_i where $i = \{1, \ldots, N\}$.

$$\text{IoU}_{i,j} = \frac{|r_j \bigcap b_i|}{|r_j \bigcup b_i|} \tag{5}$$

When the $\text{IoU}_{i,j}$ is greater than the threshold, it means that the word bounding box b_i is inside the candidate box r_j. We use $J_j = \{i = \{1, \ldots, N\} | (\text{IoU}_{i,j} > \text{IoU}_{\text{threshold}})\} \in \mathbb{R}^{N \times 1}$ to represent the inclusion of the candidate bounding boxes r_j for all N words. Block level textual feature $E_j \in \mathbb{R}^{c \times 1}$ can then be constructed as follows,

$$E_j = \Gamma(\mathbf{T} \cdot J_j) \tag{6}$$

where $\mathbf{T} = (T_1, \ldots, T_N) \in \mathbb{R}^{d \times N}$ is the textual feature matrix obtained in Eq.(1), and Γ represents the MLP layers that map the textual embedding channel dimensions d to Transformer decoder channel dimensions c. We add the block level textual features E_j to the content queries Query_j to obtain multi-modal content queries.

$$\text{Query}_j = \text{Query}_j + \lambda_1 E_j \tag{7}$$

Where λ_1 is an adjustable hyper-parameter. We then use the new content queries as input to the decoder layer to obtain finer predictions.

Method	Model	Caption	Footnote	Formula	List-item	Page-footer	Page-header	Picture	Section-header	Table	Text	Title	mAP
Human [1]	-	89	91	85	88	94	89	71	84	81	86	72	83
Faster R-CNN [1]	R101	70.1	73.7	63.5	81.0	58.9	72.0	72.0	68.4	82.2	85.4	79.9	73.4
Mask R-CNN [1]	R101	71.5	71.8	63.4	80.8	59.3	70.0	72.7	69.3	82.9	85.8	80.4	73.5
YOLOv5 [1]	v5x6	77.7	77.2	66.2	86.2	61.1	67.9	77.1	74.6	86.3	88.1	82.7	76.8
Cascade Mask R-CNN	R101	73.2	75.3	66.9	83.9	61.7	71.3	75.0	70.1	85.9	87.1	81.5	75.6
TransDLANet [3]	R101	68.2	74.7	61.6	81.0	54.8	68.2	68.5	69.8	82.4	83.8	81.8	72.3
SwinDocSegmenter [4]	Swin	83.6	64.8	62.3	82.3	65.1	66.4	**84.7**	66.5	87.4	88.2	63.3	76.9
DINO [5]	R101	71.8	78.8	72.7	85.6	63.0	76.6	74.1	72.1	87.3	87.6	85.1	77.7
VSR [6]	R101	72.6	72.1	73.8	86.2	81.8	81.3	63.1	82.5	79.4	88.4	80.7	78.4
Ours (Cascade Mask R-CNN)	R101	**86.0**	83.6	87.1	92.8	86.7	85.6	76.3	89.1	86.4	92.7	87.8	86.7
Ours(DINO)	R101	85.3	**86.7**	**89.8**	**93.6**	**90.3**	**91.0**	78.4	**90.7**	**87.4**	**93.9**	**91.3**	**89.0**

Table 1: Performance comparisons on DocLayNet. Bold indicates SOTA and underline indicates the second best. ([1](Pfitzmann et al. 2022), [2](He et al. 2017), [3](Hiuyi et al. 2023), [4](Ayan et al. 2023), [5](Hao et al. 2023), [6](Peng et al. 2021))

Our experimentation has shown that utilizing a summation fusion method in the late fusion phase can yield superior results compared to the gate mechanism used in the early fusion phase. This difference can be attributed to the fact that the textual features in the early fusion phase are extracted by the backbone, while the textual features in the late fusion are provided directly by the pre-trained language model.

Two-Stage Detectors In contrast to end-to-end detectors, which employs a Transformer encoder-decoder and learnable queries to generate candidate bounding boxes, two-stage detectors uses the RPN to generate candidate bounding boxes called Region of Interests (ROIs), as shown in Fig. 3 (b). After obtaining the ROIs, two-stage detectors extracts the features using ROIAlign and feeds the features into the R-CNN network for further regression on the offsets of ROIs.

The multi-modal ROI feature can be obtained as follows:

$$\mathrm{RF}_i = \mathrm{RF}_j + \lambda_2 E_j \tag{8}$$

where RF_i is the feature in the ith ROI and λ_2 is a hyper-parameter controls trade-off between two modality features. E_j is the corresponding block-level textual features, which is obtained by using IoU matching and block-level textual feature transforming in Eq.(5) and Eq.(6). Finally, we send the multi-modal ROI features to R-CNN for better categorisation and precise regression.

Experimental Results

Datasets

We evaluate the effectiveness of our method on three layout analysis datasets: PubLayNet (Zhong, Tang, and Jimeno Yepes 2019), DocLayNet (Pfitzmann et al. 2022), and M^{6}Doc (Hiuyi et al. 2023).

PubLayNet PubLayNet is a widely used dataset that contains 360,000 document images. As all images in PubLayNet originate from PDF documents, the extraction of word-level OCR annotations can be facilitated through the use of PDFMiner (Shinyama 2015). PubLayNet is a physical layout analysis dataset that focuses on scientific articles and only classifies the basic units of document images, with 5 categories: *Text*, *Title*, *List*, *Table*, and *Figure*.

DocLayNet DocLayNet is a recently released logical layout analysis dataset that focuses on complex, challenging, and diverse layouts. It contains 80,863 manually annotated pages with sentence-level OCR annotations. DocLayNet mainly include 6 scenarios: Financial Reports, Manuals, Scientific Articles, Laws & Regulations, Patents, and Government. It distinguishes eleven categories in the layout, including *Caption*, *Footnote*, *Formula*, *List-item*, *Page-footer*, *Page-header*, *Picture*, *Section-header*, *Table*, *Text*, and *Title*.

M^{6}Doc M^{6}Doc is a newly released logical layout analysis dataset includes 9,080 document images from 7 scenarios: Scientific articles, Textbooks, Books, Test papers, Magazines, Newspapers, and Notes. M^{6}Doc contains PDF documents, scanned and photographed documents, and we get sentence-level OCR annotations using OCR engine. M^{6}Doc is the first dataset to consider the commonality and specificity of documents, it classify 74 categories, including *QR code*, *advertisement*, *figure*, and *algorithm*, etc.

Evaluation Metric and Implementation Details

To measure the performance of the document layout analysis models, we use the metric mean Average Precision (mAP) @ IoU [0.50:0.95:0.05], which is commonly used in the obeject detection task.

In main experiments, we employ DINO (Hao et al. 2023) and Cascade Mask R-CNN (Cai and Vasconcelos 2018) as representative end-to-end and two-stage detectors, respectively. We use ResNet-101 (Kaiming et al. 2016) with FPN (Tsung-Yi et al. 2017) to extract features. Considering that both DocLayNet and M^{6}Doc contain non-English texts, we use BERT-Base-Multilingual as the language model and load the pre-trained weights provided by HuggingFace[1]. We also load the pre-trained weights of DINO and Cascade Mask R-CNN detectors from the COCO 2017 dataset (Tsung-Yi et al. 2014) for initialisation. For the Cascade

[1]https://huggingface.co/bert-base-multilingual-cased

Method	Model	AP50	AP75	Recall	mAP
SOLOv2 [1]	R101	67.5	51.4	61.5	46.8
Faster R-CNN [1]	R101	67.8	57.2	57.2	49.0
Mask R-CNN [1]	R101	58.4	46.2	50.8	40.1
Cascade Mask R-CNN [1]	R101	70.5	62.9	62.1	54.4
HTC [1]	R101	74.3	67.2	68.1	58.2
SCNet [1]	R101	73.5	65.1	67.3	56.1
Deformable DETR [1]	R101	76.8	63.4	75.2	57.2
QueryInst [1]	R101	67.1	58.1	71.0	51.0
ISTR [1]	R101	80.8	70.8	73.2	62.7
TransDLANet [1]	R101	82.7	72.7	74.9	64.5
VSR [2]	R101	76.2	68.8	66.4	59.9
DINO [3]	R101	84.6	76.7	**82.9**	68.0
Ours(Cascade Mask R-CNN)	R101	78.0	70.7	67.9	61.8
Ours(DINO)	R101	**86.7**	**79.4**	82.5	**69.9**

Table 2: Performance comparisons on M^6Doc. ([1](Hiuyi et al. 2023), [2](Peng et al. 2021), [3](Hao et al. 2023))

Method	Model	Text	Title	List	Table	Figure	mAP
Faster R-CNN [1]	X101	91.0	82.6	88.3	95.4	93.7	90.2
Mask R-CNN [1]	X101	91.6	84.0	88.6	96.0	94.9	91.0
Cascade Mask R-CNN	R101	93.9	88.4	94.7	97.6	96.9	94.2
UDoc† [3]	R50	93.9	88.5	93.7	97.3	96.4	93.9
DiT† [4]	ViT	94.4	88.9	94.8	97.6	96.9	94.5
LayoutLMv3† [5]	ViT	94.5	90.6	95.5	97.9	97.0	95.1
StructTextv2† [6]	ViT	-	-	-	-	-	95.5
SwinDocSegmenter [7]	Swin	94.6	87.2	93.0	97.9	97.3	93.7
TransDLANet [8]	R101	94.3	89.2	95.2	97.2	96.6	94.5
VSR [9]	X101	**96.7**	**93.1**	94.7	97.4	96.4	**95.7**
DINO [10]	R101	94.8	89.4	97	**98.3**	**97.6**	95.4
Ours(Cascade Mask R-CNN)	R101	94.3	88.7	95.2	97.3	96.7	94.5
Ours(DINO)	R101	95.6	89.7	**96.6**	98.1	97.3	95.5

Table 3: Performance comparisons on PubLayNet. ("†" denotes pre-trained methods, [1](Zhong, Tang, and Jimeno Yepes 2019), [2](He et al. 2017), [3](Jiuxiang et al. 2021), [4](Li et al. 2022), [5](Yupan et al. 2022), [6](Yu et al. 2023), [7](Ayan et al. 2023), [8](Hiuyi et al. 2023), [9](Peng et al. 2021), [10](Hao et al. 2023))

Mask R-CNN, we use 10 anchors [0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 1.0, 2, 5, 10] to adapt to different scales of input. For the DINO, we use DINO-4Scale, and set the query numbers to 900 following the default settings of DINO.

We train our model based on MMDetection (Chen et al. 2019). We adopt the same setting of models trained on both the M^6Doc dataset and the DocLayNet dataset., Cascade Mask R-CNN uses the SGD optimiser with an initialised learning rate of 2e-2 to train for 36 epochs, while learning rate decays to 2e-3 on 27th epoch and decays to 2e-4 on 33rd epoch; DINO uses the AdamW optimiser (Ilya and Frank 2019) with an initialised learning rate of 1e-4 to train for 36 epochs, while learning rate decays to 3.3e-5 on 27th epoch and decays to 1e-5 on 33rd epoch. For PubLayNet dataset, both Cascade Mask R-CNN and DINO training 6 epochs with the same initialized learning rate in DocLayNet, and both learning rates divided by 10 on 5th epoch.

Results and Discussion

The performance of all methods on DocLayNet is summarized in Table 1. The first row represents the human performance baseline provided by the DocLayNet. Notably, our model is the first to significantly outperform the human baseline. Our multi-modal models achieve significant improvements over the performance of their previous unimodal models. DINO was improved by 11.3% mAP (from 77.7% mAP to 89.0% mAP), and Cascade Mask R-CNN also obtained an 11.1% mAP gain (from 75.6% mAP to 86.7% mAP).

It's worth noting that VSR as the only multi-modal model in Table1 except for our method, and it still performs better than other unimodal detectors. Notably, VSR and our method have both demonstrated tremendous improvements in certain categories (*Page-header*, *Page-footer*, *Section-header*) compared to other detectors. Instances in these categories have semantically distinct elements, thus detectors can get better classification accuracy by integrating textual features. Therefore, these categories can be better enhanced from multi-modal modeling methods.

Additionally, it is interesting to note that the only category with a worse mAP than the unimodal detectors is *Figure*, where most instances do not have texts. Thus it doesn't gain mAP improvements using multi-modal models.

As shown in Table 2, traditional two-stage detectors have a recall around 60%, and their mAP is below 60%. In contrast, the recall of end-to-end detectors is around 70%, and their mAP is mostly above 60%. Notably, DINO's recall can reach 82.9% because of its large number of queries, thus its mAP is close to 70%. The high correlation between the mAP and Recall metrics is mainly due to several difficulties in the M^6Doc dataset, including a great variation in input image scale, complex data scenario distribution, and the need to distinguish 74 categories for each instance. These difficulties lead to a low recall of the model for some scenarios, which in turn limits the detection performance. Although our method does not solve the problem of low recall on M^6Doc, we can still improve the performance of the model at the original recall level and achieve state-of-the-art result 69.9% mAP.

On the PubLayNet dataset, as presented in Table 3, we also compare pre-trained models, including LayoutLMv3 (Yupan et al. 2022), DiT (Li et al. 2022), UDoc (Jiuxiang et al. 2021), and StrucTextv2 (Yu et al. 2023). Although they utilize a well-pre-trained ViT (Dosovitskiy et al. 2020) backbone, VSR with ResNeXt-101 (Saining et al. 2017) as the backbone achieves the best performance on the PubLayNet dataset. Our proposed method also achieves a comparable result 95.5% mAP. However, we observed that our method does not significantly outperform DINO itself. We speculate that this is because PubLayNet is a simple physical layout analysis dataset, which only distinguishes five basic categories unrelated to semantic information for scientific articles. Therefore, a good enough unimodal detector such as DINO can perform well on this dataset.

Furthermore, we find that the mAP gain for the Table and Figure categories was not as significant as for other categories after using multi-modal modeling(M2Doc or VSR),

Method	Early	Late	AP50	AP75	Recall	mAP
DINO	✔	✔	**86.7**	**79.4**	82.5	**69.9**
	✔	✕	85.4	77.4	82.9	68.5
	✕	✔	85.3	77.2	82.8	68.4
	✕	✕	84.6	76.7	**82.9**	68.0
Cascade Mask R-CNN	✔	✔	**78.0**	**70.7**	**67.9**	**61.8**
	✔	✕	76.0	69.5	67.4	60.5
	✕	✔	76.3	69.3	67.3	60.9
	✕	✕	74.9	68.6	65.7	59.7

Table 4: Main ablation results on M^{6}Doc test set

Module	Strategies	AP50	AP75	Recall	mAP
Early-fusion	Concate	86.5	78.9	82.4	69.4
	Sum	86.2	78.8	82.1	69.1
	Gate	**86.7**	**79.4**	**82.5**	**69.9**
Late-fusion	Concate	85.6	78.1	**82.6**	69.2
	Sum	**86.7**	**79.4**	82.5	**69.9**
	Gate	85.5	76.9	82.3	68.5

Table 5: Ablation results on M^{6}Doc test set using DINO with different fusion strategies in two modules.

and even exhibited a decline. Such intriguing experimental phenomenon was observed in Tabel 3 and 5. We attribute this phenomenon to the presence of text content within these categories, which can potentially degrade the detection quality of detectors. For instance, the determination of boundaries becomes challenging for multi-modal detectors when dealing with pictures containing overlaid texts.

Ablation and Effectiveness Analysis

To validate the effectiveness of our proposed two level modality fusion strategy, we conducted ablation studies on the M^{6}Doc test set, and the results are presented in Table 4.

To determine whether the early-fusion module indeed improves detector performance, we compared the results with and without the early-fusion module using DINO and Cascade Mask R-CNN detectors, respectively. As shown in Table 4, the removal of the early-fusion led to a 0.5% absolute mAP drop in DINO and a 0.8% drop in Cascade Mask R-CNN. The removal also resulted in a drop of approximately 0.8% in AP50 and AP75 for DINO and a 1.0% drop for Cascade Mask R-CNN. These results demonstrate the benefits of the early-fusion. Similarly, we verified the effectiveness of the late-fusion following the same process, and we can see the mAP decrease with the removal of the late fusion, which demonstrates the benefits of the late fusion.And when we use both fusion modules, Cascade Mask R-CNN and DINO can get 2.1% and 1.9 % mAP gain respectively. These results indicate that either early-fusion or late-fusion module is beneficial to both detectors and provides a relatively large boost when used together due to the different fusion levels.

Table 5 presents a comparison of the performance of early-fusion and late-fusion using three different fusion strategies. The results indicate that the best result is achieved by utilizing the gate mechanism in Eq.(4) for the early-fusion module and the summation for the late-fusion module. We think this may be attributed to the unique textual feature distributions in two module, as previously mentioned.

Method	M2Doc	AP50	AP75	Recall	mAP
DINO	✕	84.6	76.7	82.9	68.0
	✔	86.7	79.4	82.5	69.9
	Δ	+2.1	+2.7	-0.4	+1.9
Cascade Mask R-CNN	✕	74.9	68.6	65.7	59.7
	✔	78.0	70.7	67.9	61.8
	Δ	+3.1	+2.1	+2.2	+2.1
Mask R-CNN	✕	73.2	64.7	63.7	55.9
	✔	77.5	69.2	66.1	58.8
	Δ	+4.3	+4.5	+2.4	+2.9
Faster R-CNN	✕	72.3	64.6	62.6	55.3
	✔	77.3	68.6	65.1	57.9
	Δ	+5.0	+4.0	+2.5	+2.6
Deformable DETR	✕	81.4	70.1	73.8	62.3
	✔	83.7	72.1	75.1	63.9
	Δ	+2.3	+2.0	+1.3	+1.6

Table 6: Comparison between detectors before and after plugging M2Doc on M^{6}Doc test set. Due to the different experimental setting, the baseline results we reproduce are higher than the results provided by M^{6}Doc.

Pluggablity of M2Doc

To further validate the pluggablity of M2Doc, we also combine it with other detectors. As shown in Table 6, we conduct experiments on M^{6}Doc dataset using Mask R-CNN (He et al. 2017), Faster R-CNN (Ren et al. 2015), and Deformable DETR (Xizhou et al. 2021) besides DINO and Cascade Mask R-CNN. The experimental setting of Mask R-CNN and Faster R-CNN basically refer to the setting of Cascade Mask R-CNN mentioned above, and Deformable DETR uses the default setting. In Table 6, with the use of M2Doc, all detectors get significant improvements across all metrics. These qualitative results demonstrate the excellent generality and robustness of M2Doc.

Conclusion

In this paper, we propose an effective and pluggable multi-modal fusion approach M2Doc for document layout analysis. M2Doc aims to endow existing unimodal detectors with multi-modal capablities for DLA tasks. We have demonstrated the broad applicability of M2Doc by implementing it on top of both two-stage and end-to-end detectors. Extensive experiments on three benchmark datasets, DocLayNet, M^{6}Doc and PubLayNet validate that M2Doc significantly boosts the performance over baseline unimodal detectors. While promising progress has been made, some limitations persist such as marginal gains on simple datasets where unimodal methods suffice. Future work can explore adaptive fusion techniques and incorporate structural and semantic relationships between document entities. Nonetheless, we believe M2Doc provides an important step towards developing more unified multi-modal models for advanced document layout understanding.

Acknowledgements

This research is supported in part by NSFC (Grant No.: 61936003) and Alibaba DAMO Innovative Research Foundation. We thank the support from the Alibaba-South China University of Technology Joint Graduate Education Program.

References

Antonacopoulos, A.; Bridson, D.; Papadopoulos, C.; and Pletschacher, S. 2009. A Realistic Dataset for Performance Evaluation of Document Layout Analysis. In *ICDAR*, 296–300.

Ayan, B.; Sanket, B.; Josep, L.; and Umapada, P. 2023. SwinDocSegmenter: An End-to-End Unified Domain Adaptive Transformer for Document Instance Segmentation. In *ICDAR*.

Cai, Z.; and Vasconcelos, N. 2018. Cascade R-CNN: Delving Into High Quality Object Detection. In *CVPR*, 6154–6162.

Chen, K.; Wang, J.; Pang, J.; Cao, Y.; Xiong, Y.; Li, X.; Sun, S.; Feng, W.; Liu, Z.; Xu, J.; et al. 2019. MMDetection: Open MMLab Detection Toolbox and Benchmark. *arXiv preprint arXiv:1906.07155*.

Denk, T. I.; and Reisswig, C. 2019. BERTgrid: Contextualized Embedding for 2D DocumentRepresentation and Understanding. In *Document Intelligence Workshop at NeurIPS*.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. 2020. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *ICLR*.

Grill, J.-B.; Strub, F.; Altché, F.; Tallec, C.; Richemond, P.; Buchatskaya, E.; Doersch, C.; Avila Pires, B.; Guo, Z.; Gheshlaghi Azar, M.; et al. 2020. Bootstrap your own latent-a new approach to self-supervised learning. In *NeurIPS*, 21271–21284.

Hangbo, B.; Li, D.; Songhao, P.; and Furu, W. 2022. BEiT: BERT Pre-Training of Image Transformers. In *ICLR*.

Hao, Z.; Feng, L.; Shilong, L.; Lei, Z.; Hang, S.; Jun, Z.; Lionel, M. N.; and Heung-Yeung, S. 2023. DINO: DETR with Improved DeNoising Anchor Boxes for End-to-End Object Detection. In *ICLR*.

He, K.; Gkioxari, G.; Dollar, P.; and Girshick, R. 2017. Mask R-CNN. In *ICCV*, 2961–2969.

Hiuyi, C.; Peirong, Z.; Sihang, W.; Jiaxin, Z.; Qiyuan, Z.; Zecheng, X.; Jing, L.; Kai, D.; and Lianwen, J. 2023. M^{6}Doc: A Large-Scale Multi-Format, Multi-Type, Multi-Layout, Multi-Language, Multi-Annotation Category Dataset for Modern Document Layout Analysis. In *CVPR*, 15138–15147.

Ilya, L.; and Frank, H. 2019. Decoupled Weight Decay Regularization. In *ICLR*.

Jacob, D.; Ming-Wei, C.; Kenton, L.; and Kristina, T. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL*, 4171–4186.

Jiuxiang, G.; Jason, K.; Morariu, V. I.; Handong, Z.; Nikolaos, B.; Rajiv, J.; Ani, N.; and Tong, S. 2021. Unified Pretraining Framework for Document Understanding. In *NeurIPS*, 39–50.

Kaiming, H.; Xiangyu, Z.; Shaoqing, R.; and Jian, S. 2016. Deep Residual Learning for Image Recognition. In *CVPR*, 770–778.

Katti, A. R.; Reisswig, C.; Guder, C.; Brarda, S.; Bickel, S.; Hohne, J.; and Faddoul, J. B. 2018. Chargrid: Towards Understanding 2D Documents. In *EMNLP*, 4459–4469.

Lee, J.; Hayashi, H.; Ohyama, W.; and Uchida, S. 2019. Page Segmentation using a Convolutional Neural Network with Trainable Co-Occurrence Features. In *ICDAR*, 1023–1028.

Li, J.; Xu, Y.; Lv, T.; Cui, L.; Zhang, C.; and Wei, F. 2022. DiT: Self-supervised Pre-training for Document Image Transformer. In *ACM Multimedia*, 3530–3539.

Li, K.; Wigington, C.; Tensmeyer, C.; Zhao, H.; Barmpalios, N.; Morariu, V. I.; Manjunatha, V.; Sun, T.; and Fu, Y. 2020a. Cross-Domain Document Object Detection: Benchmark Suite and Method. In *CVPR*, 12915–12924.

Li, M.; Xu, Y.; Cui, L.; Huang, S.; Wei, F.; Li, Z.; and Zhou, M. 2020b. DocBank: A Benchmark Dataset for Document Layout Analysis. In *ICCL*, 949–960.

Li, X.-H.; Yin, F.; and Liu, C.-L. 2020. Page segmentation using convolutional neural network and graphical model. In *Document Analysis Systems: 14th IAPR International Workshop, DAS 2020*, 231–245. Springer.

Nair, V.; and Hinton, G. E. 2010. Rectified linear units improve restricted boltzmann machines. In *ICML*, 807–814.

Namboodiri, A. M.; and Jain, A. K. 2007. *Document Structure and Layout Analysis*. Springer London.

Peng, Z.; Can, L.; Liang, Q.; Zhanzhan, C.; Shiliang, P.; Yi, N.; and Fei, W. 2021. VSR: A Unified Framework for Document Layout Analysis combining Vision, Semantics and Relations. In *ICDAR*, 115–130.

Pfitzmann, B.; Auer, C.; Dolfi, M.; Nassar, A. S.; and Staar, P. W. J. 2022. DocLayNet: A Large Human-Annotated Dataset for Document-Layout Analysis. In *ACM SIGKDD*, 3743–3751.

Ren, S.; He, K.; Girshick, R.; and Sun, J. 2015. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In *NeurIPS*, 91–99.

Ronneberger, O.; Fischer, P.; and Brox, T. 2015. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 234–241. Springer.

Saining, X.; Ross, G.; Piotr, D.; Zhuowen, T.; and Kaiming, H. 2017. Aggregated Residual Transformations for Deep Neural Networks. In *CVPR*, 1492–1500.

Shinyama, Y. 2015. PDFMiner: Python PDF Parser and Analyzer. *Retrieved on*.

Subhajit, M.; Sanket, B.; Siladittya, M.; Ayan, B.; Josep, L.; Saumik, B.; and Umapada, P. 2023. SelfDocSeg: A Self-Supervised vision-based Approach towards Document Segmentation. In *ICDAR*.

Tsung-Yi, L.; Michael, M.; Serge, B.; James, H.; Pietro, P.; Deva, R.; ar, P. D.; and Zitnick, C. L. 2014. Microsoft COCO: Common Objects in Context. In *ECCV*, 740–755.

Tsung-Yi, L.; Piotr, D.; Ross, G.; Kaiming, H.; Bharath, H.; and Serge, B. 2017. Feature Pyramid Networks for Object Detection. In *CVPR*, 2117–2125.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is All You Need. In *NeurIPS*, 6000—6010.

Xizhou, Z.; Weijie, S.; Lewei, L.; Bin, L.; Xiaogang, W.; and Jifeng, D. 2021. Deformable DETR: Deformable Transformers for End-to-End Object Detection. In *ICLR*.

Yang, X.; Yumer, E.; Asente, P.; Kraley, M.; Kifer, D.; and Lee Giles, C. 2017. Learning to extract semantic structure from documents using multimodal fully convolutional neural networks. In *CVPR*, 5315–5324.

Yu, Y.; Yulin, L.; Chengquan, Z.; Xiaoqiang, Z.; Zengyuan, G.; Xiameng, Q.; Kun, Y.; Junyu, H.; Errui, D.; and Jingdong, W. 2023. StrucTexTv2: Masked Visual-Textual Prediction for Document Image Pre-training. In *ICLR*.

Yupan, H.; Tengchao, L.; Lei, C.; Yutong, L.; and Furu, W. 2022. LayoutLMv3: Pre-training for Document AI with Unified Text and Image Masking. In *ACM Multimedia*, 4083–4091.

Zhao, Y.; Jiaqi, W.; Yansong, T.; Kai, C.; Hengshuang, Z.; and Torr, P. H. 2022. LAVT: Language-Aware Vision Transformer for Referring Image Segmentation. In *CVPR*, 18155–18165.

Zhong, X.; Tang, J.; and Jimeno Yepes, A. 2019. PubLayNet: Largest Dataset Ever for Document Layout Analysis. In *ICDAR*, 1015–1022.