UniLayDet: Simple Multi-dataset Document Layout Analysis

Prasidh Srikumar[0009−0007−5405−2008], Ajoy Mondal[0000−0002−4808−8860], and C. V. Jawahar[0000−0001−6767−7057]

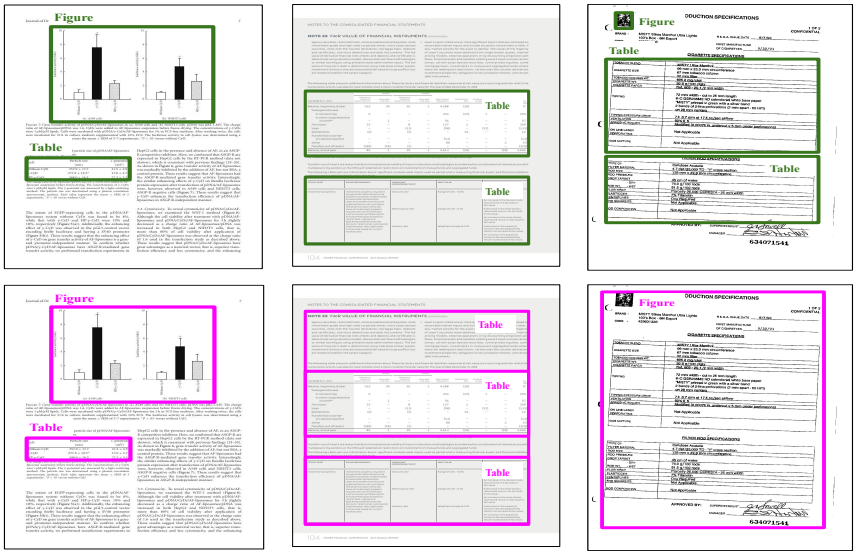
CVIT, International Institute of Information Technology, Hyderabad, India
prasidh.nitt@gmail.com, {ajoy.mondal,jawahar}@iiit.ac.in

Abstract. Information extraction from documents has become increasingly popular due to the rise of large language models (LLMs) and reaugmented generation (RAG) models. Document Layout Analysis (DLA) is a fundamental task in document AI, playing a crucial role in identifying semantically related elements within a document — a key step toward effective information extraction. Modern document layout analysis algorithms benefit from large-scale annotated datasets but suffer significant performance drops when tested across different datasets, limiting the generalization of models trained on a single source. To address this, we utilize a multi-dataset training approach for a Universal Layout Detection (UniLayDet) model utilizing a shared detection architecture with dataset-specific outputs and unifying the label space post-training through an automatic merging process. UniLayDet significantly improves generalization across datasets compared to models trained individually and also achieves competitive in-domain performance, notably attaining a mAP (@IoU[0.5-0.95]) of 68.9% on M^6Doc (partitioned setting), close to the existing SOTA of 69.9%, showing that our model is simple yet effective for the task. Code and models are available at github.com/Mobius1D/UniLayDet

Keywords: Document layout analysis · multi-dataset training · universal layout detector · object detection.

1 Introduction

Document Layout Analysis (DLA) [30] has emerged as a crucial pre-processing step in Visual Information Extraction (VIE) pipelines [44], serving as the foundation for extracting structured information from visually rich documents (VRDs) that combine text, graphics, tables, and other visual elements. Recently, DLA has gained significance with the rise of Large Language Models (LLMs) [19,43,1,15] and Retrieval-Augmented Generation (RAG) [51,23,34] systems, which depend on high-quality document data. Documents are critical in finance, healthcare, and law, where precise information extraction impacts decision-making. Given the intricate layouts of text, graphics, tables, and figures, robust DLA is essential for effective document processing and downstream applications.



(a) PubLayNet (b) DocLayNet (c) D^{4}LA

Fig. 1: Displays the predicted *Table* and *Figure* classes in PubLayNet, DocLayNet, and D^{4}LA, respectively, using VGT [10] trained on PubLayNet. Images in the first row indicate ground truth bounding boxes of *Table* and *Figure* classes. Images in the second row show the detected *Table* and *Figure* in respective datasets.

Method	**Train Set**	**Test set**			
		PubLayNet	DocLayNet	D^{4}LA	M^{6}Doc
VGT [10]	PubLayNet	**92.06**	50.63	6.05	-
	DocLayNet	77.02	**83.65**	17.44	-
	D^{4}LA	62.19	33.08	**68.13**	-
Cascade RCNN [6]	M^{6}Doc	57.51	23.64	9.83	**65.78**

Table 1: Shows the performance (overall mean average precision (mAP) @ IoU [0.5:0.95]) of state-of-the-art methods — VGT [10] and Cascade RCNN [6] with class aware sampling on cross-domain datasets. The model performance drastically reduces while evaluating cross-domain datasets. We consider labels which are common in both training and test sets for calculation of mAP.

Recent methods [32,57,25] formulate document layout analysis task as object detection problem, leveraging deep architectures [6,16,35]. This shift highlights the need for large, diverse datasets. Early datasets focused on scientific documents [25,57], limiting model generalization, while newer datasets [33,9,10,56] cover varied document types and languages with fine-grained annotations. Recent advances include pre-training [24,49], grid-based approaches [5,54,10], and multi-modal fusion [50,53,45]. However, most models are trained on single datasets,

fail to capture real-world document diversity, revealing a critical gap in current DLA research.

To validate the limitations of recent methods, we conducted cross-dataset evaluations on common classes of state-of-the-art DLA models. Our results reported in Table 1 and Fig. 1 show that models trained on a single dataset, either PubLayNet [57] or DocLayNet [33], suffer significant performance drops on unseen datasets. This issue persists even with newer datasets such as M^{6}Doc [9] and D^{4}LA [10]. Our experiment revealed that single DLA datasets constrain document types, languages, layouts, and label vocabularies, limiting general-purpose layout detection. *Can these limitations be mitigated by unifying diverse document layout analysis datasets?* Several recent works show that multi-dataset training approaches solve the limitation of the generalization capability of models trained on a single dataset in various computer vision tasks: segmentation [41,14] and object detection [59,8,42]. In this work, we also follow similar directions to solve this problem.

In this work, similar to previous [28,10,56,45], we define document layout analysis as the detection of objects or regions or elements within a document or page image. We simplify training a document layout detector across multiple datasets to be as straightforward as training on a single one. By maintaining separate outputs for each dataset and applying dataset-specific supervision, our approach mimics training multiple dataset-specific layout detectors within a shared network. It maximizes data utilization, enhances performance on training domains, and improves generalization to unseen data. However, it may produce duplicate outputs for overlapping classes across datasets.

A key challenge is unifying datasets into a common taxonomy for a layout detector that generalizes beyond dataset-specific classes. Traditional manual taxonomy creation [22,55] is labor-intensive and error-prone. We propose to use an automated approach [59] that leverages visual similarities between document layout detectors across datasets to optimize a unified taxonomy (label space). The method jointly determines the taxonomy and its mapping to each dataset using a 0-1 integer programming formulation. Unification, through this automatically generated vocabulary, incorporates concepts from all training datasets with minimal loss in performance.

We evaluate our unified document layout detector by training on four large and diverse datasets: PubLayNet [57], DocLayNet [33], M^{6}Doc [9] and D^{4}LA [10]. Our results demonstrate that this layout detector performs comparable to dataset-specific layout detectors in an in-domain setting. The unified document layout detector also generalizes new domains more effectively without the need for retraining and outperforms individual dataset layout detectors when evaluated in a cross-domain setting.

Our key contributions are:

- We perform a comprehensive evaluation of state-of-the-art document layout analysis models across multiple datasets, revealing important insights into their generalization abilities and effectiveness in real-world scenarios (Refer Table 1, Fig. 1, and Section 3.1).

- We utilize a multi-dataset training strategy and unify the label space across different datasets for document layout analysis tasks.
- We propose a unified document layout detector (UniLayDet) trained on multiple datasets: PubLayNet, DocLayNet, M^{6}Doc and D^{4}LA and demonstrate its effectiveness through extensive experiments on both in-domain and out-of-domain settings (Refer Table 2 and Table 3).

2 Related Work

2.1 Document Layout Analysis

Traditional document layout analysis (DLA) methods [31,21,20,38] have primarily relied on rule-based and template-driven approaches, using fixed heuristics to detect elements like text blocks, images, and tables, and organizing them based on predefined spatial relationships. While effective for simple and structured layouts, these methods struggle with complex designs. Moreover, adapting them to new layout styles requires significant manual effort, limiting their flexibility and general applicability.

With advancements in deep learning, document layout analysis has been formulated as an object detection problem, achieving significant improvements in accuracy and generalization for complex layouts. Early approaches [32,57,25] relied on visual inputs and employed object detectors, such as Faster R-CNN [36,37], Mask R-CNN [16], Cascade R-CNN [6], and YOLO [35]. More recently, Zhao *et al.* [56] has demonstrated promising results using YOLO based architecture. All these unimodal methods use only visual features and obtain reasonable accuracies.

In recent years, document pre-training has achieved significant success. Document Image Transformer (DiT) [24] leverages image-based pre-training, demonstrating strong performance in DLA. Given the multi-modal nature of documents, prior approaches such as LayoutLM [49,48,18,2] and Unidoc [13] have introduced multi-modal transformer models for document understanding. However, these methods still rely solely on visual features during DLA fine-tuning, which can limit performance and generalization capabilities. To effectively leverage both visual and textual information for DLA, grid-based methods [5,54,10] encode text with layout information and then combined with visual features, leading to improved performance, especially in textual elements, with particular improvements in text-related class identification. End-to-end transformer-based approaches [12,4,53,45] have also been developed for DLA. Methods [12,4] rely solely on image inputs, while models [53,45] incorporate both visual and language modalities.

Despite being trained on diverse datasets like D^{4}LA [10], DocLayNet [33], and M^{6}Doc [9] — which encompass various document types to reflect real-world scenarios, all these models struggle to generalize effectively to unseen or cross-domain datasets. We propose a unified document layout detector trained across multiple datasets to overcome this limitation to improve model robustness and generalization.

2.2 Multi-dataset Object Detection

In computer vision, models trained on a single dataset often face challenges in generalizing to unseen data due to domain-specific biases. This limitation affects various tasks, including segmentation [41,14] and object detection [59,8,42]. Recent methods have been developed to train on multiple datasets to enhance the generalization of models. These approaches demonstrate strong performance on in-domain datasets and exhibit greater robustness than models trained on a single dataset.

In object detection, Zhou *et al.* [59] introduced UniDet, which employs a straightforward approach using a Cascade R-CNN with multiple classification heads for different datasets. The method unifies label spaces through an automatic label merging process that learns an optimal mapping to minimize merging costs. ScaleDet [8] leverages visual-textual alignment within a unified label space to enable learning across datasets. Plain-Det [42] introduces a pluggable approach that enhances UniDet by incorporating sparse proposal generation and hardness-based sampling, improving both performance and training efficiency. In this work, we adopt a similar multi-dataset training strategy to develop a unified document layout detector.

3 Unified Document Layout Detector

3.1 Motivation

Several recent methods, such as Hybrid DLA [39], Detect-Order-Construct [46], M2Doc [53], DocLayout YOLO [56], and VGT [10] achieve high layout detection accuracy on in-domain datasets. However, most existing works, except for [47,11], focus solely on in-domain performance. To analyze cross-domain generalization, we evaluate VGT [10] on both in-domain and cross-domain datasets, with results presented in Table 1. Fig. 1 highlights that VGT when trained on PubLayNet, fails to detect common classes such as *Table* and *Figure* while tested on DocLayNet and D^4LA. Our analysis shows that this model experiences a drastic mAP drop of 33.02 and 62.08, respectively. A similar performance decline is observed when the model is trained on DocLayNet and tested on PubLayNet and D^4LA, as well as when trained on D^4LA and tested on PubLayNet and DocLayNet. These results indicate that while VGT performs well on in-domain datasets, it struggles significantly in cross-domain evaluations. A common strategy to address this issue is merging all existing datasets for training to create a more generalized detector. However, this approach is not straightforward due to inconsistent label space between datasets. These findings motivate our development of a multi-dataset training approach that merges diverse label spaces on semantically similar classes to create a robust, unified document layout detector.

Many studies [28,10,56,45] define document layout analysis as the detection of objects (regions or elements) within a document or page image. The objective is to predict a bounding box $b_i \in \mathbb{R}^4$ and an associated class-specific confidence score $d_i \in \mathbb{R}^{|L|}$ for each detected object i in a document image I. The confidence

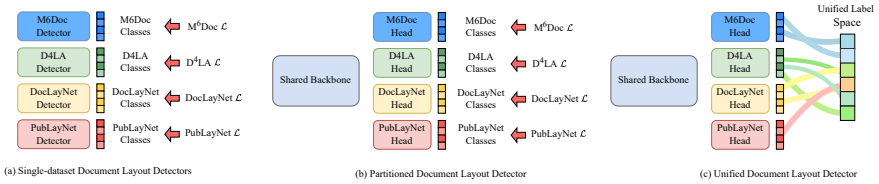


Fig. 2: Illustrate (a) standard document layout detectors are trained on a single dataset using a dataset-specific loss. (b) Our approach trains a single partitioned document layout detector across multiple datasets, utilizing a shared backbone with dataset-specific outputs. (c) Finally, we automatically unify the outputs of the partitioned layout detector into a common label space.

score represents the likelihood that the bounding box corresponds to a specific class $c \in L$, where L is the set of all document object classes (e.g., *text, table, figure, caption, equation*, etc.) in the dataset D. Training a document layout detector $\mathcal{H}$ on a dataset involves optimizing a loss function $\mathcal{L}$, typically formulated using a box-level log-likelihood expressed as $\mathcal{L} = \mathcal{L}_{cal} + \beta\mathcal{L}_{reg}$ [58], over a sampled document image $\hat{I}$ and its corresponding annotated bounding boxes $\hat{B}$ from dataset D. The objective of this optimization process is formally defined as:

$$\min_{\Theta} \ \mathbb{E}_{(\hat{I},\hat{B})\sim D} \, \mathcal{L}\big(\mathcal{H}(\hat{I};\Theta), \hat{B}\big), \tag{1}$$

where $\hat{B}$ represents class-specific bounding box annotations, the loss function $\mathcal{L}$ aligns the predicted outputs with the ground truth annotations using an overlap-based criterion and $\mathbb{E}$ is the expectation operator.

Since existing datasets [25,9,33,57] exhibit diverse document layouts (e.g., single-column, multi-column, and unstructured formats), domains (e.g., *research articles, newspapers, magazines*, etc.), and object categories (e.g., *text, tables, figures, equations, captions*, etc.), training a layout detector on a single dataset may not generalize well to others, leading to suboptimal performance [47,11]. A natural solution is to curate and integrate all existing datasets to train a unified document layout detector parameterized by Θ. By incorporating diverse samples, the combined dataset $D = D_1 \cup D_2 \cup \cdots \cup D_M$, where M is the total number of datasets and the unified label space $L = L_1 \cup L_2 \cup \ldots \cup L_M$, with repeated labels merged across datasets, improves generalization. This leads to a more robust and effective document layout detector across different domains.

$$\min_{\Theta} \ \mathbb{E}_{(\hat{I},\hat{B})\sim D_1\cup D_2\cup\cdots\cup D_M} \mathcal{L}\big(\mathcal{H}(\hat{I};\Theta), \hat{B}\big). \tag{2}$$

3.2 Training a Unified Document Layout Detector

Our goal is to train a single unified document layout detector $\mathcal{H}$, across M distinct datasets $D_1, D_2, \ldots, D_M$, each with its own label set $L_1, L_2, \ldots, L_M$, where

$m \in \{1, 2, \ldots, M\}$ denotes the index of a dataset. The key insight is that a unified document layout detector $\mathcal{H}$ with parameters Θ can be trained similarly to multiple dataset-specific document layout detectors as long as label spaces from different datasets are not merged. This approach effectively trains dataset-specific document layout detectors $\mathcal{H}_1, \mathcal{H}_2, \ldots, \mathcal{H}_M$ in parallel while sharing a common backbone architecture $\mathcal{H}$. Each dataset-specific model retains a unique classification head while sharing all other layers with the backbone. We refer to this structure as a *partitioned document layout detector* (Fig. 2(b)). The partitioned detector is trained across all M datasets by minimizing the box-level log-likelihood loss $\mathcal{L}$:

$$\min_{\Theta} \mathbb{E}_{D_m} \left[\mathbb{E}_{(I,\hat{B}) \sim D_m} \left[\mathcal{L}\Big(\mathcal{H}_m(\hat{I}; \Theta),\, \hat{B} \Big) \right] \right]. \tag{3}$$

Unification of Label Space Through Learning: For a detector to be useful, it should have one output per bounding box which the *partitioned document layout detector* does not achieve, predicting a class per dataset D_m per bounding box. Therefore we unify common labels across different datasets. Given multiple datasets $D_1, D_2, \ldots, D_M$, each with its own label space $L_1, L_2, \ldots, L_M$, our objective is to learn a unified label space L for all M datasets and define a Boolean mapping function between unified label space and dataset-specific labels $\mathcal{F}_m : L \to L_m$, $\mathcal{F}_m \in \{0,1\}^{|L_m| \times |L|}$. In this approach, each joint label $c \in L$ is mapped to at most one dataset-specific label $c_m \in L_m$: $\mathcal{F}_m^\top \mathbf{1} \leq \mathbf{1}$ and each dataset-specific label matches exactly one unified label: $\mathcal{F}_m \mathbf{1} = \mathbf{1}$.

Given a set of partitioned document layout detector outputs: $d_i^1 \in \mathbb{R}^{|L_1|}, d_i^2 \in \mathbb{R}^{|L_2|}, \ldots\, d_i^M \in \mathbb{R}^{|L_M|}$ for a bounding box b_i, we compute a joint detection score by averaging the outputs corresponding to common labels:

$$d_i = \frac{\sum_m \mathcal{F}_m^\top d_i^m}{\sum_m \mathcal{F}_m^\top 1}, \tag{4}$$

with element wise division (Fig. 2(c) provides an overview). For joint training, the dataset-specific scores are then recovered via $\tilde{d}_i^m = \mathcal{F}_m d_i$. Our objective is to find set of mappings $\mathcal{F}^\top = [\mathcal{F}_1^\top, \mathcal{F}_2^\top, \ldots, \mathcal{F}_M^\top]$ and thereby define joint label space L, so that the joint classifier maintains performance comparable to the individual ones.

Instead of relying on a hand-crafted mapping ($\mathcal{F}$) and label spaces (L) or language-based merging baseline, which may suffer performance degradation because of label ambiguities of datasets, we automatically optimize the joint label space using correlations in the outputs of a pretrained partitioned layout detector.

For a given output class c, let $\mathcal{L}_c$ be a loss function that evaluates the quality of the merged label space representation, d_i, and its reprojection, $\tilde{d}_i^m$, against the original disjoint label space d_i^m for a single bounding box i. Suppose $U^m = [d_1^m, d_2^m, \ldots]$ be the outputs of the partitioned layout detection head for dataset D_m. We define the merged detection scores as $U = \frac{\sum_m \mathcal{F}_m^\top U^m}{\sum_m \mathcal{F}_m^\top \mathbf{1}}$ and $\tilde{U}^m = \mathcal{F}_m U$

be the re-projection. Our goal is to optimize this loss over all detector outputs and overall optimization problem is formulated as:

$$\begin{aligned} \min_{L,\mathcal{F}} \quad & \mathbb{E}_{D_m}\left[\sum_{c\in L_m} \mathcal{L}_c\big(U_c^m, \tilde{U}_c^m\big)\right] \;+\; \lambda\,|L|, \\ \text{subject to} \quad & \mathcal{F}_m\mathbf{1} = \mathbf{1}, \quad \mathcal{F}_m^\top\mathbf{1} \le \mathbf{1}, \quad \forall m. \end{aligned} \tag{5}$$

The label size cost term $\lambda|L|$ encourages a smaller and more compact label space. The objective function defined in Eq. (5) combines combinatorial optimization over L with a binary integer program over $\mathcal{F}$. However, a straightforward reparameterization enables more efficient optimization.

It is to be noted that the label set L corresponds to the number of columns in $\mathcal{F}$. Additionally, each dataset D_m allows merging at most one label, ensuring $\mathcal{F}_m^\top\mathbf{1} \le \mathbf{1}$. Consequently, for each dataset D_m, a column $\mathcal{F}_m(c) \in \mathbb{F}_m$ can take one of $|\hat{L}_m| + 1$ possible values: $\mathbb{F}_m = \{0, \mathbf{1_1}, \mathbf{1_2}, \ldots\}$, where $\mathbf{1}_i \in \{0,1\}^{|L_m|}$ is an indicator vector for the i-th element. Each column $\mathcal{F}(c) \in \mathbb{F}$ is then selected from a limited set of possible values, $\mathbb{F} = \mathbb{F}_1 \times \mathbb{F}_2 \times \ldots$, where $\times$ denotes the Cartesian product. Instead of directly optimizing over the label set L and transformation $\mathcal{F}$, we reformulate the problem as a combinatorial optimization over the potential column values $p \in \mathbb{F}$. Let $x_p \in \{0,1\}$ be an indicator variable representing whether a particular combination $p \in \mathbb{F}$ is selected. If $x_p = 1$, the class combination specified by p is applied; otherwise, it is not. In this formulation, the constraint $\mathcal{F}_m\mathbf{1} = \mathbf{1}, \forall_m$ translates to $\sum_{p\in\mathbb{F}|p_c=1} x_p = 1$ for every dataset-specific label c and $p_c = 1$ represents the combinations for which class c is mapped. Then our overall objective becomes

$$\min_x \sum_{p\in\mathbb{F}} x_p c_p + \lambda \sum_{p\in\mathbb{F}} x_p, \tag{6}$$

where λ is the penalty term and c_p denotes the merge cost, can be precomputed for any subset of labels p and defined as:

$$c_p = \mathbb{E}_{D_m}\left[\sum_{c\in L_m|p_c=1} \mathcal{L}_c\big(U_c^m, \tilde{U}_c^m\big)\right], \tag{7}$$

where $\mathcal{L}_c$ is the loss function that compares the original detection outputs U_c^m to the reprojected outputs $\tilde{U}_c^m$. It is to be noted that c_p is precomputed for optimization. Therefore, the Eq. (6) can be rewritten as

$$\begin{aligned} \min_x \quad & \sum_{p\in\mathbb{F}} x_p\,(c_p + \lambda), \\ \text{subject to} \quad & \sum_{p\in\mathbb{F}|p_c=1} x_p = 1 \quad \forall\, c. \end{aligned} \tag{8}$$

This formulation results in a compact integer linear program (ILP) that can be efficiently solved using standard ILP solvers [27]. While the number of

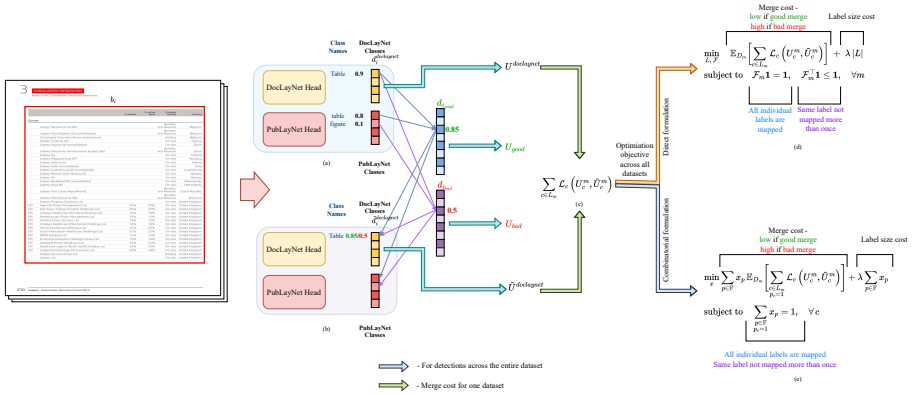


Fig. 3: Illustrates the intuition behind the process of merging labels across different datasets. (a) represents partitioned document layout detector's outputs and (b) represents the reprojected detection score.

combinations of $\mathbb{F}$ increases exponentially with the number of datasets, many elements in $\mathbb{F}$ lead to high merge costs and can be pruned using a simple greedy algorithm controlled by a hyperparameter τ for which the code will be made available234.

Loss function for merging: Average Precision To maintain detection performance, we use an Average Precision (AP) based loss defined in Eq. (9) for merging. For each class (c), let $\mathcal{J}_c(D_c^m)$ be the AP on the original dataset-specific output and $\mathcal{J}_c(\tilde{D}_c^m)$ be the AP with the reprojected detection scores. The AP-based loss is defined as:

$$\mathcal{L}_c^{\mathcal{J}}(U_c^m, \tilde{U}_c^m) = \frac{1}{|L_m|}\Big[\mathcal{J}_c(U_c^m) - \mathcal{J}_c(\tilde{U}_c^m)\Big]. \tag{9}$$

This metric ensures that the merging process preserves the detection accuracy and it is calculated in the validation sets of datasets.

Intuition Behind Label Space Merging Fig. 3 visually demonstrates the intuition behind merging label spaces across datasets. The process of merging DocLayNet's [33] `Table` class with a corresponding PubLayNet [57] class (`table` or `figure`) involves:

1. **Reprojection:** The reprojected score $\tilde{d}_i^{\text{doclaynet}}$ (Eq. 4) indicates compatibility, yielding high values for `Table` when merging with *semantically similar* classes (e.g., `table`) and low values for *dissimilar* classes (e.g., `figure`) as shown in (Fig. 3(b)).
2. **Merge Cost:** Aggregating reprojected scores across detections yields a dataset-level **merge cost** (Fig. 3(c)).

3. **Loss:** The loss (Eq. 9) for DocLayNet's `Table` class reflects this cost: high for dissimilar merges, low for similar merges. It propagates to the overall minimization objective (Fig. 3(d)–(e)).

Merging occurs only if the expected merge cost is sufficiently low and the reduction in label space size justifies it. Low label space cost favors merging, but a high merge cost (indicating semantic dissimilarity) prevents it.

3.3 Discussion

Fig. 2 illustrates the two step process of our UniLayDet model, highlighting the difference between a traditional single-dataset document layout detector (Fig. 2 (a)) and a multi-dataset document layout detector (Figs. 2 (b) and (c)). UniLayDet consists of four detection heads, each corresponding to a specific dataset — PubLayNet, DocLayNet, D^4LA, and M^6Doc — while sharing a common backbone. In step 1, each detection head is trained independently on its respective dataset. Once optimal training is achieved for all individual heads, we proceed to step 2, where we unify the label spaces of the four datasets. Finally, the trained UniLayDet model is evaluated.

4 Experiments

4.1 Datasets

PubLayNet [57]: It is a large-scale dataset comprising 360K page images annotated with various layout elements, including *text, titles, lists, figures, and tables.* It is generated through automated annotation of one million PubMed Central PDF articles, making it a valuable resource for document layout analysis.

DocLayNet [33]: comprises approximately 80K manually annotated pages spanning multiple domains, including finance, science, patents, tenders, law, and manuals. With eleven distinct layout labels, it introduces significant multi-domain variability, making it a valuable benchmark for document layout analysis.

D^4LA [10]: consists of approximately 11K manually annotated pages spanning 12 document types, including letters, forms, invoices, memos, and news articles. With 27 fine-grained labels, it captures diverse real-world layouts and inherent noise, enhancing model robustness for document analysis tasks.

M^6Doc [9]: contains 9K manually annotated pages across seven document types, featuring 74 labels. Uniquely, it includes scanned and born-digital documents in English and Chinese, making it a comprehensive dataset for layout analysis.

DSSE-200 [50]: comprises 200 manually annotated pages with intricate layouts from magazines, academic papers, and presentations. It includes six labels specifically designed to assess segmentation performance under challenging conditions.

IIIT-AR-13K [29]: comprises 13K manually annotated pages from annual reports, emphasizing graphical elements rather than text. It includes five labels — table, figure, logo, signature, and natural image — making it particularly useful for non-text document object detection.

PRImA [3]: comprises 1,240 semi-automatically annotated pages across 10 categories, including text, images, tables, charts, graphics, separators, math, captions, noise, and frames. It is a foundational dataset for document layout analysis, helping models recognize diverse structural elements. For evaluation, we focus on top-level regions — TextRegion, ImageRegion, TableRegion, and MathRegion — ensuring consistency in performance assessment across different layout structures.

We use *PubLayNet, D^4LA, DocLayNet, and M^6Doc* for multi-dataset training of the UniLayDet model as well as for the training of individual Cascade-RCNN models and evaluation of these models. We use validation sets of PubLayNet and DocLayNet and test sets of M^6Doc and D^4LA for evaluation purposes. Meanwhile, *DSSE-200*, *IIIT-AR-13K*, and *PRImA* are used only for cross-domain evaluation of individual and UniLayDet models.

4.2 Evaluation Metrics

We use the category-wise and overall mean average precision (mAP) [26] @intersection over union (IoU) $[0.50 : 0.95]$ of bounding boxes as the evaluation metric. When evaluating the UniLayDet model, we consider two settings: *partitioned* and *unified*. In the partitioned setting, we assess the outputs of each detection head separately, evaluating them on the same dataset they were trained on. The unified setting includes two scenarios: in-domain and cross-domain. In the in-domain scenario, we evaluate only the reprojected outputs ($\tilde{d}_i^m$) from the unified detection head. For the cross-domain scenario, we combine the labels from both the training and testing datasets and then evaluate the model on the common classes.

4.3 Implementation Details

For the UniLayDet model, which uses a *ResNet101* [17] backbone and a Cascade-RCNN [6] decoder, we resize the longest side of each input image to 1333 pixels. The model is trained with a base learning rate of 0.01 and a batch size of 32 across four datasets: (M^6Doc, D^4LA, DocLayNet, and PubLayNet. Training spans 180k iterations (equally distributed as 45k iterations per dataset), with the learning rate reduced by a factor of 10 at 120k and 160k iterations.

For the UniLayDet model with *ResNeSt101* [52] as backbone, we use a base learning rate of 0.02 and a batch size of 16 for 450k iterations across all datasets. The learning rate is lowered at 300k and 400k iterations. We apply uniform sampling across datasets and class-aware sampling for M^6Doc, as it enhances performance in that setting. We apply dataset-specific scaling to the logits from different model heads based on their performance. For the DSSE and PRImA

datasets, we multiply the logits of both M^6Doc and DocLayNet by 1.5, while for the IIIT13k dataset, we scale the logits of DocLayNet by a factor of 3 before unification.

All training uses the AdamW optimizer. For individual models, we train Cascade-RCNN with a ResNet101 backbone for 180k iterations, using a base learning rate of 0.01 and a batch size of 16. The learning rate is reduced at 120k and 160k iterations. A linear warm-up is applied for the first 1% of iterations in all models. Class-aware sampling [40] is applied for the training of models on M^{6}Doc dataset as emperically the models perform better when it is applied. Training is conducted using the Detectron2 framework.

Method	PubLayNet	DocLayNet	D^{4}LA	M^{6}Doc
Hybrid DLA [39]	**97.3**	81.6	-	-
Detect-Order-Construct [46]	<u>96.5</u>	81.0	-	-
M2Doc [53]	95.5	**89.0**	-	**69.9**
M2Doc CNN [53]	94.5	<u>86.7</u>	61.8	-
DocLayout YOLO [56]	-	79.7	**70.3**	-
VGT [10]	96.2	83.7	<u>68.8</u>	-
TransDLANet [9]	94.5	72.3	-	64.5
LayoutLMv3 [18]	95.1	76.8	60.5	-
DiT [24]	94.9	80.3	67.7	-
VSR [54]	95.7	-	-	-
DINO [7]	95.5	74.3	-	-
Cascade RCNN [6] (our)	94.9	75.0	64.5	65.8
Partitioned UniLayDet (our)	95.2	76.9	67.8	<u>68.9</u>
Unified UniLayDet (our)	94.8	76.6	67.3	68.7

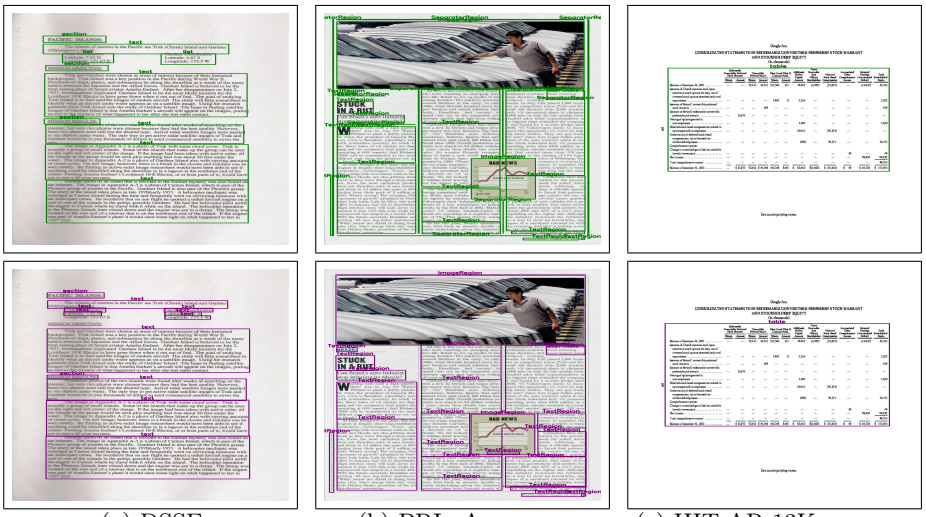
Table 2: Presents the performance of both existing methods and our approach on in-domain datasets. The values reported represent mAP at IoU [0.5:0.95] over all labels. The bold and underline text indicates the best performance and the second-best performance, respectively.

4.4 Evaluation on In-domain Datasets

Table 2 compares the performance of UniLayDet (ResNeSt-101 [52]), partitioned detector, and state-of-the-art dataset-specific models. In M^{6}Doc, UniLayDet achieves a mAP close to the best-performing model M2Doc [53]. We also reach competitive performances in D^{4}LA and PubLayNet as well. However, for DocLayNet, there is a noticeable gap of 12.4 between UniLayDet and the top-performing models. Unlike dataset-specific models that perform well only on a single dataset, UniLayDet shows good performance on multiple datasets at once.

4.5 Evaluation on Cross-domain Datasets

We use DSSE [50], PRImA [3], and IIIT-AR-13K [29] to evaluate the performance of dataset-specific layout detector as cascade RCNN [6] and UniLayDet in



(a) DSSE (b) PRImA (c) IIIT-AR-13K

Fig. 4: Presents UniLayDet's detection results on cross-domain datasets — DSSE, PRImA, and IIIT-AR-13K. The first row displays ground truth annotations, while the second row shows UniLayDet's predictions.

cross-domain scenarios. The cascade RCNN with ResNet101 as backbone trained on individual datasets. Table 3 shows the individual detector and UniLayDet results on the mentioned cross-domain datasets. It shows that our document layout unified model — UniLayDet obtains better results than individual detectors for all datasets although we are evaluating it on more fine-grained labels. This shows that UniLayDet generalizes better than models trained on single datasets. Fig. 4 shows predicted regions in sample pages from DSSE, PRImA, and IIIT-AR-13K, using UniLayDet[1].

4.6 Qualitative study

Fig. 5 shows that compared to other models which fail when tested on other datasets, UniLayDet makes detections that closely match the ground truth. It is also to be noted that UniLayDet provides more finegrained and accurate labels because of training across multiple datasets.

4.7 Ablation Study

Effect of Unification Parameters The number of labels in the unified detector and its performance depends on the parameters λ and τ (see Table 4).

[1] Additional class-wise quantitative and qualitative results on in-domain and cross-domain datasets and mapping labels from training to test sets are given in supplementary material.

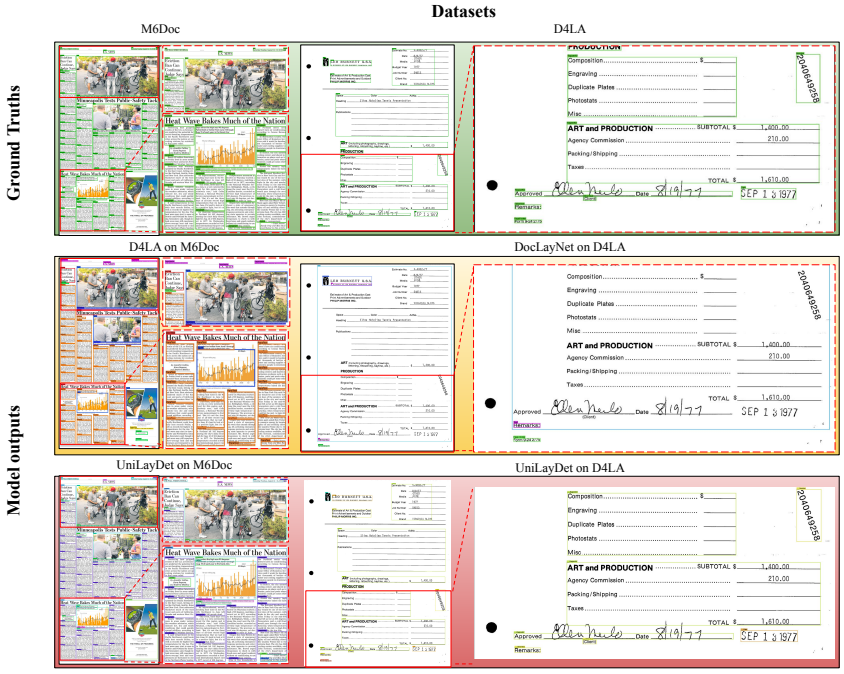


Fig. 5: Qualitative comparison of UniLayDet against other existing models.

Model	Trained on	Tested on		
		DSSE	PRImA	IIIT-AR-13K
Cascade RCNN [6]	PubLayNet	28.30	12.70	40.28
Cascade RCNN [6]	DocLayNet	31.89	35.0	<u>59.06</u>
Cascade RCNN [6]	D^4LA	30.40	37.06	35.30
Cascade RCNN [6]	M^6Doc	<u>32.45</u>	<u>38.20</u>	51.20
UniLayDet	all four	**33.16**	**40.32**	**59.24**

Table 3: Shows the performance (mAP at IoU [0.5:0.95] over all classes) of the individual detector trained on individual datasets and the UniLayDet trained on four datasets — PubLayNet, DocLayNet, D^4LA, and M^6Doc. We consider labels common in training and test sets for mAP calculation.

By varying λ in increments of 0.1 from 0.1 to 1 and τ between 0.1 and 0.3, we observe that the pruning parameter τ has a more significant impact on the size of the resulting label space. We select the label space that achieves the highest mean mAP during evaluation.

Manual vs Automatic Label Merging In addition to automatic merging, manual merging based on label semantics is also possible. Experimental results in Table 5 show that automatic merging produces slightly better results comparable

τ	λ	$\|L\|$	PubLayNet	DocLayNet	D^4LA	M^6Doc	Avg.mAP
0.1	**0.25**	**106**	**94.17**	**73.85**	**66.50**	**68.60**	**75.78**
0.1	0.5	106	94.17	73.85	66.50	68.60	75.78
0.2	0.5	104	94.13	73.63	66.34	68.60	75.67
0.3	0.5	103	93.74	73.63	66.36	68.60	75.58
0.6	0.5	101	93.91	73.37	65.93	68.60	75.45
-	-	-	94.38	74.35	67.14	68.72	76.15

Table 4: Show the effect of unification parameters τ and λ in the performance of our UniLayDet model. The last row represents the baseline performance of the partitioned detector Cascade RCNN [6] with ResNet101 backbone.

Method	PubLayNet	DocLayNet	D^4LA	M^6Doc	Avg. mAP
Manual Merging	94.17	73.96	**65.89**	65.29	74.82
Automatic Merging	**94.91**	**74.98**	64.54	**65.78**	**75.41**

Table 5: Presents the performance of automatic and manual merging label spaces in a multi-dataset training approach.

to manual merging, yielding a smaller label space size of 106 labels compared to 107 — the optimal size based on parameters that achieve the best performance.

5 Conclusions

In conclusion, the proposed simple yet effective UniLayDet model addresses the challenge of generalization in Document Layout Analysis tasks by leveraging a multi-dataset training approach. By employing a shared detection architecture with dataset-specific outputs and an automatic label space merging process, UniLayDet effectively integrates diverse datasets while mitigating taxonomy inconsistencies. The results demonstrate its superior generalization capabilities, outperforming models trained on individual datasets and paving the way for more robust and adaptable document AI systems.

6 Limitations

While our approach benefits from training across multiple datasets and improved generalization, it has some limitations:

- Automatically merged label spaces may still group a few semantically distinct classes, which may not be intuitive despite improving in-domain mAP.
- The unified detector may produce different labels for similar semantics across datasets (e.g., _Equation_Formula_ vs. formula_ _ _).
- We do not address incremental learning for extending the unified detector to new datasets without retraining.

Acknowledgments

This work is supported by the MeitY Government of India, through the NLTM Bhashini (https://bhashini.gov.in/) project.

References

1. Abhimanyu, D., et al.: The llama 3 herd of models. arXiv preprint arXiv:2407.21783 (2024) 1
2. Adnan, W., Tang, J., Zouggari, Y.B.K., Laatiri, S.E., Lam, L., Caspani, F.: A layoutlmv3-based model for enhanced relation extraction in visually-rich documents. In: ICDAR. pp. 160–174 (2024) 4
3. Antonacopoulos, A., Pletschacher, S., Bridson, D., Papadopoulos, C.: Icdar 2009 page segmentation competition. In: ICDAR. pp. 1370–1374 (2009) 11, 12
4. Banerjee, A., Biswas, S., Lladós, J., Pal, U.: Swindocsegmenter: An end-to-end unified domain adaptive transformer for document instance segmentation. In: ICDAR. pp. 307–325 (2023) 4
5. Barman, R., Ehrmann, M., Clematide, S., Oliveira, S.A., Kaplan, F.: Combining visual and textual features for semantic segmentation of historical newspapers. Journal of Data Mining & Digital Humanities (HistoInformatics) (2021) 2, 4
6. Cai, Z., Vasconcelos, N.: Cascade r-cnn: Delving into high quality object detection. In: CVPR. pp. 6154–6162 (2018) 2, 4, 11, 12, 14, 15
7. Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., Joulin, A.: Emerging properties in self-supervised vision transformers. In: ICCV. pp. 9650–9660 (2021) 12
8. Chen, Y., Wang, M., Mittal, A., Xu, Z., Favaro, P., Tighe, J., Modolo, D.: Scaledet: A scalable multi-dataset object detector. In: CVPR. pp. 7288–7297 (2023) 3, 5
9. Cheng, H., Zhang, P., Wu, S., Zhang, J., Zhu, Q., Xie, Z., Li, J., Ding, K., Jin, L.: M^6doc: A large-scale multi-format, multi-type, multi-layout, multi-language, multi-annotation category dataset for modern document layout analysis. In: CVPR. pp. 15138–15147 (2023) 2, 3, 4, 6, 10, 12
10. Da, C., Luo, C., Zheng, Q., Yao, C.: Vision grid transformer for document layout analysis. In: ICCV. pp. 19462–19472 (2023) 2, 3, 4, 5, 10, 12
11. De Nardin, A., Zottin, S., Piciarelli, C., Foresti, G.L., Colombi, E.: In-domain versus out-of-domain transfer learning for document layout analysis. IJDAR pp. 1–15 (2024) 5, 6
12. Deng, Q., Ibrayim, M., Hamdulla, A., Luo, H., Zhang, C.: Doc-DINO: A transformer model for complex logical document layout analysis. In: ICDAR. pp. 76–89 (2024) 4
13. Gu, J., Kuen, J., Morariu, V.I., Zhao, H., Jain, R., Barmpalios, N., Nenkova, A., Sun, T.: Unidoc: Unified pretraining framework for document understanding. In: NeurIPS. pp. 39–50 (2021) 4
14. Gu, X., Cui, Y., Huang, J., Rashwan, A., Yang, X., Zhou, X., Ghiasi, G., Kuo, W., Chen, H., Chen, L.C., et al.: Dataseg: Taming a universal multi-dataset multi-task segmentation model. NeurIPS pp. 67329–67354 (2023) 3, 5
15. Haoran, W., et al.: General ocr theory: Towards ocr-2.0 via a unified end-to-end model. arXiv preprint arXiv:2409.01704 (2024) 1
16. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask R-CNN. In: ICCV. pp. 2961–2969 (2017) 2, 4

17. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR. pp. 770–778 (2016) 11
18. Huang, Y., Lv, T., Cui, L., Lu, Y., Wei, F.: LayoutLMv3: Pre-training for document ai with unified text and image masking. In: ACM MM. pp. 4083–4091 (2022) 4, 12
19. et al. Hugo Touvron: Llama: Open and efficient foundation language models. CoRR (2023) 1
20. Jain, A.K., Zhong, Y.: Page segmentation using texture analysis. Pattern recognition **29**(5), 743–770 (1996) 4
21. Journet, N., Eglin, V., Ramel, J.Y., Mullot, R.: Text/graphic labelling of ancient printed documents. In: ICDAR. pp. 1010–1014 (2005) 4
22. Lambert, J., et al.: Mseg: A composite dataset for multi-domain semantic segmentation. IEEE Trans. on PAMI **45**(1), 796–810 (2023) 3
23. Lewis, P., et al.: Retrieval-augmented generation for knowledge-intensive nlp tasks. In: NeurIPS. pp. 2200–2209 (2020) 1
24. Li, J., Xu, Y., Lv, T., Cui, L., Zhang, C., Wei, F.: DiT: Self-supervised pre-training for document image transformer. In: ACM MM. pp. 3530–3539 (2022) 2, 4, 12
25. Li, M., Xu, Y., Cui, L., Huang, S., Wei, F., Li, Z., Zhou, M.: DocBank: A benchmark dataset for document layout analysis. arXiv preprint arXiv:2006.01038 (2020) 2, 4, 6
26. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: ECCV. pp. 740–755 (2014) 11
27. Linderoth, J.T., Ralphs, T.K.: Noncommercial software for mixed-integer linear programming. In: Integer programming, pp. 269–320. CRC Press (2005) 8
28. Mondal, A., Agarwal, M., Jawahar, C.: Dataset agnostic document object detection. Pattern Recognition **142**, 109698–109713 (2023) 3, 5
29. Mondal, A., Lipps, P., Jawahar, C.: IIIT-AR-13K: A new dataset for graphical object detection in documents. In: DAS. pp. 216–230 (2020) 11, 12
30. Namboodiri, A.M., Jain, A.K.: Document structure and layout analysis. In: Digital Document Processing: Major Directions and Recent Advances, pp. 29–48. Springer (2007) 1
31. O'Gorman, L.: The document spectrum for page layout analysis. IEEE Trans. on PAMI **15**(11), 1162–1173 (1993) 4
32. Oliveira, D.A.B., Viana, M.P.: Fast cnn-based document layout analysis. In: ICCVW. pp. 1173–1180 (2017) 2, 4
33. Pfitzmann, B., Auer, C., Dolfi, M., Nassar, A.S., Staar, P.: DocLayNet: A large human-annotated dataset for document-layout segmentation. In: ACM SIGKDD. pp. 3743–3751 (2022) 2, 3, 4, 6, 9, 10
34. Ram, O., et al.: In-context retrieval-augmented language models. Trans. of ACL pp. 1316–1331 (2023) 1
35. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: CVPR. pp. 779–788 (2016) 2, 4
36. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: Towards real-time object detection with region proposal networks. NeurIPS **28** (2015) 4
37. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. IEEE trans. on PAMI **39**(6), 1137–1149 (2016) 4
38. Shafait, F., Breuel, T.M.: The effect of border noise on the performance of projection-based page segmentation methods. IEEE Trans. on PAMI **33**(4), 846–851 (2010) 4

39. Shehzadi, T., Stricker, D., Afzal, M.Z.: A hybrid approach for document layout analysis in document images. In: ICDAR. pp. 21–39 (2024) 5, 12
40. Shen, L., Lin, Z., Huang, Q.: Relay backpropagation for effective learning of deep convolutional neural networks. In: ECCV. pp. 467–482 (2016) 12
41. Shi, B., Zhang, X., Xu, H., Dai, W., Zou, J., Xiong, H., Tian, Q.: Multi-dataset pretraining: A unified model for semantic segmentation. arXiv preprint arXiv:2106.04121 (2021) 3, 5
42. Shi, C., Zhu, Y., Yang, S.: Plain-det: A plain multi-dataset object detector. In: ECCV. pp. 210–226 (2025) 3, 5
43. Touvron, H., et al.: Llama 2: Open foundation and fine-tuned chat models. CoRR **abs/2307.09288** (2023) 1
44. Wang, B., Xu, C., Zhao, X., Ouyang, L., Wu, F., Zhao, Z., Xu, R., Liu, K., Qu, Y., Shang, F., et al.: Mineru: An open-source solution for precise document content extraction. arXiv preprint arXiv:2409.18839 (2024) 1
45. Wang, J., Hu, K., Huo, Q.: DLAFormer: An end-to-end transformer for document layout analysis. In: ICDAR. pp. 40–57 (2024) 2, 3, 4, 5
46. Wang, J., Hu, K., Zhong, Z., Sun, L., Huo, Q.: Detect-order-construct: A tree construction based approach for hierarchical document structure analysis. PR **156**, 110836–110842 (2024) 5, 12
47. Wu, X., Xiao, L., Du, X., Zheng, Y., Li, X., Ma, T., Jin, C., He, L.: Cross-domain document layout analysis using document style guide. Expert Systems with Applications **245**, 123039–123049 (2024) 5, 6
48. Xu, Y., Xu, Y., Lv, T., Cui, L., Wei, F., Wang, G., Lu, Y., Florencio, D., Zhang, C., Che, W., et al.: Layoutlmv2: Multi-modal pre-training for visually-rich document understanding. arXiv preprint arXiv:2012.14740 (2020) 4
49. Xu, Y., Li, M., Cui, L., Huang, S., Wei, F., Zhou, M.: Layoutlm: Pre-training of text and layout for document image understanding. In: ACM SIGKDD. pp. 1192–1200 (2020) 2, 4
50. Yang, X., Yumer, E., Asente, P., Kraley, M., Kifer, D., Lee Giles, C.: Learning to extract semantic structure from documents using multimodal fully convolutional neural networks. In: CVPR. pp. 5315–5324 (2017) 2, 10, 12
51. Yunfan, G., et al.: Retrieval-augmented generation for large language models: A survey. arXiv preprint arXiv:2312.10997 **2** (2023) 1
52. Zhang, H., Wu, C., Zhang, Z., Zhu, Y., Lin, H., Zhang, Z., Sun, Y., He, T., Mueller, J., Manmatha, R., et al.: Resnest: Split-attention networks. In: CVPR. pp. 2736–2746 (2022) 11, 12
53. Zhang, N., Cheng, H., Chen, J., Jiang, Z., Huang, J., Xue, Y., Jin, L.: M2doc: A multi-modal fusion approach for document layout analysis. In: AAAI. pp. 7233–7241 (2024) 2, 4, 5, 12
54. Zhang, P., Li, C., Qiao, L., Cheng, Z., Pu, S., Niu, Y., Wu, F.: VSR: a unified framework for document layout analysis combining vision, semantics and relations. In: ICDAR. pp. 115–130 (2021) 2, 4, 12
55. Zhao, X., et al.: Object detection with a unified label space from multiple datasets. In: ECCV. pp. 178–193 (2020) 3
56. Zhao, Z., Kang, H., Wang, B., He, C.: DocLayOut-YOLO: Enhancing document layout analysis through diverse synthetic data and global-to-local adaptive perception. arXiv preprint arXiv:2410.12628 (2024) 2, 3, 4, 5, 12
57. Zhong, X., Tang, J., Yepes, A.J.: PubLayNet: largest dataset ever for document layout analysis. In: ICDAR. pp. 1015–1022 (2019) 2, 3, 4, 6, 9, 10
58. Zhong, Z., Sun, L., Huo, Q.: An anchor-free region proposal network for faster r-cnn-based text detection approaches. IJDAR **22**, 315–327 (2019) 6

59. Zhou, X., Koltun, V., Krähenbühl, P.: Simple multi-dataset detection. In: CVPR. pp. 7571–7580 (2022) 3, 5