DEEN : Detection Encoder For Document Layout Anlysis

Anonymous ACL submission

Abstract

Document Layout Analysis is typically formulated as an object detection task. However, most existing approaches are adapted from general-purpose detection frameworks and overlook the fundamental structural differences between document and natural images. To meet the needs of human reading habits, document images are two-dimensional and free from occlusion. Based on this observation, we propose DEtection ENcoder (DEEN), which reformulates document layout analysis as a graph connectivity prediction task, thereby eliminating the need for both Non-Maximum Suppression (NMS) and confidence thresholding in post-processing. To efficiently model high-resolution feature maps, DEEN combines global sparse and local dense attention for unified representation of overall layout and fine-grained details. Since DEEN does not rely on confidence scores, we evaluate it under two settings: one that favors confidence-based models, and another that simulates real-world usage scenarios. DEEN achieves competitive performance on three structurally diverse datasets, demonstrating strong generalization.

1 Introduction

Document layout analysis (DLA) is often formulated as an object detection task on document images. It plays a crucial role in enabling downstream tasks. For example, in Retrieval-Augmented Generation workflows, accurately reconstructing the original layout of a document helps produce well-structured information units (Ren et al., 2023; Zhang et al., 2022a). These structured units enable more precise grounding and context alignment during retrieval and generation, thereby improving the performance of large language models (Zhao et al., 2024a; Gao et al., 2024). Object detection emphasizes different aspects across domains: autonomous driving emphasizes speed, medical imaging focuses on accuracy, while the DLA task seeks a balanced trade-off among efficiency, accuracy, and usability.

Figure 1: A comparison between natural and document images: overlapping content frequently occurs in natural scenes, whereas document layouts are typically designed to avoid such overlap for human readability.

Existing DLA methods employ diverse modeling strategies, such as incorporating layout priors into general-purpose detectors(Zhao et al., 2024b), adapting document pre-trained models for detection tasks(Huang et al., 2022), and leveraging multimodal features to enhance context-aware layout representation(Da et al., 2023). However, they overlook a fundamental structural difference between document and natural images. As shown in Figure1, natural images, being 2D projections of 3D scenes, often involve occlusion and overlap, which constrain the design of object detectors and necessitate post-processing steps such as confidence-based ranking and Non-Maximum Suppression (NMS). In contrast, document images are inherently designed as two-dimensional layouts to support human reading habits, where layout elements are generally expected to be spatially non-overlapping and clearly bounded, except for background regions. This structural prior motivates a rethinking of how layout analysis should be modeled.

To this end, we reformulate the DLA task as a region connectivity prediction problem, thereby **eliminating the need for confidence thresholds and non-maximum suppression (NMS)**. Specifically, the image is divided into regular grids, and each grid cell is assigned two labels: a boundary label indicating whether it belongs to the background, edge, or interior, and a semantic label representing its category distribution. To capture fine-grained structural cues while maintaining computational efficiency, we propose the GSLD Attention module, which combines Mixture-of-Experts-based Global Sparse attention (GS) for modeling diverse spatial patterns and Local Dense attention (LD) for enhancing boundary continuity and local perception. Based on the boundary predictions, we construct a connectivity graph and then refine the regions using semantic information, including distinguishing densely distributed and entangled regions, and adjusting box boundaries according to semantic distributions. This leads to more complete structural predictions with improved boundary precision.

Since DEEN does not rely on confidence scores, it is incompatible with the standard Average Precision(Lin et al., 2015). To enable a comprehensive evaluation, we introduce two complementary settings: one that favors confidence-based models, and another that better reflects real-world deployment scenarios. We conduct experiments on three structurally diverse datasets: DocLayNet(Pfitzmann et al., 2022), PubLayNet(Zhong et al., 2019), and CDLA(Hang, 2021). While maintaining stable inference efficiency, DEEN demonstrates strong competitiveness under the first setting and achieves the highest number of best and second-best results under the second. These results confirm the practical effectiveness of our proposed formulation and attention mechanism.

Our main contributions are as follows:

- We propose DEEN, a novel framework that formulates the DLA task as a region connectivity prediction task, eliminating the need for confidence thresholds and NMS.
- We introduce GSLD Attention to capture both global structure and local details in high-resolution features, while maintaining inference efficiency.
- Extensive experiments on three structurally diverse datasets demonstrate the effectiveness and generalization capability of our method.

2 Related Work

2.1 Single-Modality Detection Methods

Most existing approaches adapt general-purpose detection frameworks such as R-CNN(Girshick et al., 2014; Ren et al., 2016) and the YOLO series(Redmon et al., 2016; Khanam and Hussain, 2024), relying on anchor boxes, confidence scores, and NMS for prediction and post-processing. While effective, manual adjustment of thresholds is often necessary to ensure reliable outputs. DETR series(Carion et al., 2020; Zhu et al., 2021; Zhang et al., 2022b) introduces an end-to-end paradigm that eliminates anchors and NMS via global matching, but still depends on a fixed number of object queries and confidence filtering at inference. Similar to our approach, SwinDocSegmenter(Banerjee et al., 2023) formulates layout prediction as instance segmentation, yet still depends on confidence scores during inference. Notably, DocLayout-YOLO(Zhao et al., 2024b) incorporates layout-specific designs and high-quality synthetic pretraining data, achieving a favorable trade-off between speed and accuracy.

2.2 Multi-modal Detection Methods

Some works draw inspiration from grid-based methods in vision information extraction(Katti et al., 2018; Denk and Reisswig, 2019; Lin et al., 2021), with several methods further constructing a text grid using textual and layout information(Yang et al., 2017; Zhang et al., 2021) to support document layout analysis. Recent models such as LayoutLMv3(Huang et al., 2022) and UniDoc(Feng et al., 2023) employ unified multimodal encoding during pretraining, but are typically fine-tuned with visual backbones. VGT(Da et al., 2023) adopts a dual-stream architecture that combines visual features and grid-based textual inputs to enhance both semantic and boundary modeling.

2.3 Autoregressive Detection Methods

Florence2(Xiao et al., 2023) discretizes continuous coordinates into tokens, enabling unified modeling of detection and generation tasks. DocFusion(Chai et al., 2024) addresses the inherent conflict between discrete tokens and continuous coordinates under multi-task training, allowing for accurate performance on a variety of document parsing tasks such as layout analysis and table recognition. However, these autoregressive models suffer from limited inference speed and error propagation, which hinder practical deployment.

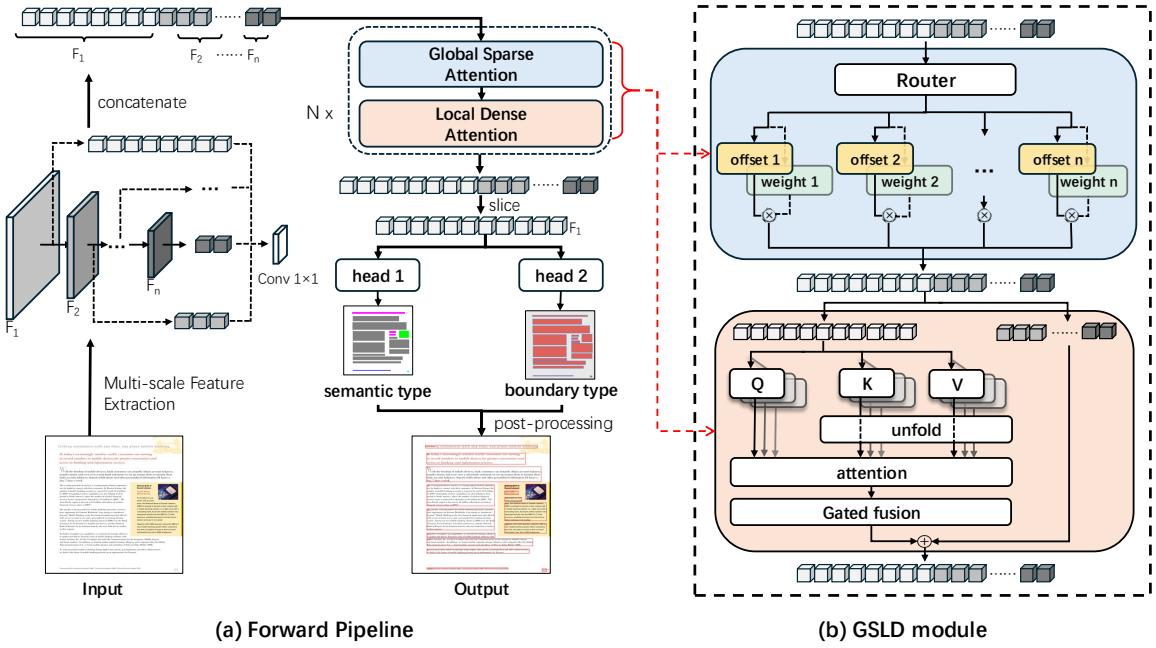

Figure 2: Illustration of the proposed DEEN. The input image is first processed into multi-scale feature maps, which are then refined by structure-aware GSLD Attention. The highest-resolution feature map is passed to two output heads for boundary and semantic prediction. Each GSLD block combines Global Sparse and Local Dense Attention.

3 DEEN

We describe how open-sourced raw annotations are converted into grid-level semantic and boundary labels (Section 3.1), followed by an overview of the encoder-based architecture that enables layout understanding (Section 3.2). The core GSLD attention mechanism is detailed in Section 3.3. We then outline the post-processing step that transforms grid predictions into structured outputs (Section 3.4). For clarity, we omit the batch dimension in tensor shapes throughout this section.

3.1 Data Construction

To avoid unnecessary annotation effort, we design automated scripts to convert region-level annotations from existing public datasets into the supervision format required by DEEN. Specifically, for each input image $\mathbf{I}$, we divide it into a spatial grid of size $H_g \times W_g$, where the resolution is dynamically determined by the highest-resolution feature map produced by the feature extractor. Each grid cell corresponds to a visual token and serves as the fundamental unit for token-level classification. For each cell, we construct two types of supervision labels: a semantic label, indicating the content category covered by the region, and a boundary label, capturing its geometric role (e.g., inside, edge, or background) within the annotated layout element.

Semantic labels: For each annotated region (e.g., a bounding box or segmentation mask), we compute its overlapping area with each grid cell and assign a class probability distribution. When a cell overlaps with multiple labeled regions, its semantic label becomes a weighted distribution based on the area proportions of the overlapping classes. This results in a dense supervision tensor $\mathbf{y}^{\text{sem}} \in \mathbb{R}^{H_g \times W_g \times C}$, which serves as the target for KL-divergence-based loss, allowing the model to learn semantic concepts in a soft and flexible manner.

Boundary labels: To capture structural boundaries, each grid cell is also assigned a discrete label from $\{0, 1, 2\}$, indicating background, edge, or interior. These labels are determined based on the cell's relative position to the annotated region—cells fully inside are labeled as 2, those near region boundaries as 1, and the rest as 0. This produces a label map $\mathbf{y}^{\text{bnd}} \in \mathbb{R}^{H_g \times W_g \times 3}$ used for cross-entropy supervision. We do not use semantic labels directly for connectivity prediction for the following reasons. First, when the number of classes is large, constructing class-specific connectivity graphs significantly increases the complexity of the task. Second, in dense or closely packed layouts, it is difficult to separate regions accurately using only semantic labels. Instead, using boundary labels simplifies the task and improves robustness in structurally complex documents.

3.2 Architecture

As illustrated in Figure 2, given an input image $\mathbf{I} \in \mathbb{R}^{H \times W \times 3}$, the model first extracts multi-scale feature maps $\mathbf{F}_1, \mathbf{F}_2, \ldots, \mathbf{F}_n$ using convolutional operations (or patch-based embedding). Each feature map at scale n is denoted as $\mathbf{F}_n \in \mathbb{R}^{H_n \times W_n \times C_n}$. After unifying channel dimensions, the multi-scale features are flattened and concatenated into $\mathbf{X} \in \mathbb{R}^{N \times D}$ for further modeling.

To capture both long-range dependencies and local spatial structures in document layouts, we adopt the GSLD (Global Sparse and Local Dense) Attention mechanism. The resulting representation is denoted as $\mathbf{X}' \in \mathbb{R}^{N \times D}$.

Following the GSLD attention blocks, the tokens at the highest-resolution level are selected to form $\mathbf{G} \in \mathbb{R}^{N_g \times D}$ corresponding to the highest-resolution feature map, where $N_g = H_g \times W_g$. Each grid token $\mathbf{g}_i$ is then processed by two classification heads:

Semantic classification head $\mathbf{W}^{\text{sem}} \in \mathbb{R}^{D \times C}$, which predicts the semantic category distribution over C classes:

$$\mathbf{p}_i^{\text{sem}} = \text{softmax}(\mathbf{W}^{\text{sem}^\top} \mathbf{g}_i) \in \mathbb{R}^C \tag{1}$$

Boundary classification head $\mathbf{W}^{\text{bnd}} \in \mathbb{R}^{D \times 3}$, which predicts structural boundary:

$$\mathbf{p}_i^{\text{bnd}} = \text{softmax}(\mathbf{W}^{\text{bnd}^\top} \mathbf{g}_i) \in \mathbb{R}^3 \tag{2}$$

During training, we supervise the semantic predictions $\mathbf{p}_i^{\text{sem}}$ using KL divergence against the soft label distribution $\mathbf{y}_i^{\text{sem}}$ constructed in Section 3.1. For the boundary predictions $\mathbf{p}_i^{\text{bnd}}$, we apply a standard cross-entropy loss with the discrete boundary label $\mathbf{y}_i^{\text{bnd}} \in \{0, 1, 2\}$. Both losses are computed only over valid tokens using spatial masks, and the total loss is a weighted sum of the two terms:

$$\mathcal{L}_{\text{total}} = \lambda_{\text{sem}} \cdot \mathcal{L}_{\text{KL}} + \lambda_{\text{bnd}} \cdot \mathcal{L}_{\text{CE}} \tag{3}$$

where λ_{sem} and λ_{bnd} control the relative importance of semantic and boundary supervision.

To convert grid-level predictions into structured layout elements, we apply a post-processing step that generates a set of M predicted boxes:

$$\hat{\mathcal{Y}} = \{(x_{1j}, y_{1j}, x_{2j}, y_{2j}, \hat{c}_j, \hat{s}_j)\}_{j=1}^{M} \tag{4}$$

where $(x_{1j}, y_{1j}, x_{2j}, y_{2j})$ are the top-left and bottom-right coordinates of the j-th box, $\hat{c}_j$ is the predicted class, and $\hat{s}_j$ is the associated confidence score (used to assist in confirming $\hat{c}$, not for filtering).

3.3 GSLD Attention

Our goal is to generate bounding boxes and class labels from grid-level classification. However, when targets are small and densely distributed, accuracy heavily depends on the spatial resolution of the feature map. Standard self-attention has quadratic complexity $O(N^2)$, making it impractical for high-resolution inputs due to memory and computation costs.

To address this, we propose GSLD attention, which combines Global Sparse (GS) attention for long-range dependencies and Local Dense (LD) attention for fine-grained local patterns. This hybrid design balances resolution and efficiency while improving representation quality.

3.3.1 Global Sparse

To efficiently capture long-range dependencies on high-resolution feature maps, we extend the standard multi-scale deformable attention mechanism(Zhu et al., 2021) into a sparse Mixture-of-Experts (MoE) architecture, referred to as Global Sparse Attention (GS). This module introduces token-level expert routing to enhance structural diversity in attention patterns while maintaining similar computational complexity to the original formulation.

In the multi-scale deformable attention module, each query q with normalized reference point $\hat{\mathbf{p}}_q \in [0, 1]^2$ aggregates features from a sparse set of sampling locations across multiple feature levels. The output feature is computed as:

$$\mathbf{y}_q = \sum_{m=1}^{M} W_m \left[\sum_{l=1}^{L} \sum_{k=1}^{K} A_{mlqk} \cdot W'_m \, \mathbf{x}_l(\phi_l(\hat{\mathbf{p}}_q) + \Delta \mathbf{p}_{mlqk}) \right] \tag{5}$$

where m, l, and k index the attention head, feature level, and sampling point, respectively. $\Delta \mathbf{p}_{mlqk} \in \mathbb{R}^2$ denotes the sampling offset, and A_{mlqk} is the attention weight normalized such that $\sum_{l,k} A_{mlqk} = 1$ for each head m. The function $\phi_l(\hat{\mathbf{p}}_q)$ maps the normalized reference point to the coordinate system of level l, and $\mathbf{x}_l(\cdot)$ performs bilinear interpolation over the l-th feature map. W_m and W'_m are learnable projection matrices specific to each head.

However, in the standard formulation, all query tokens share a single set of linear layers for predicting offsets and weights, regardless of their position or semantic context. In practice, we observe that this shared predictor tends to learn an averaged sampling pattern biased toward the

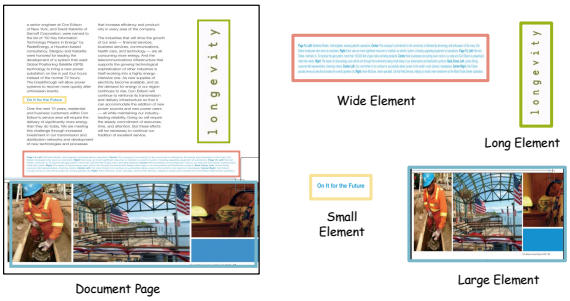

Figure 3: Different layout elements on a document page exhibit significant differences in shape.

majority structures in the training data, which often consist of horizontally elongated regions. As shown in Figure 3, document pages typically contain layout elements with significantly diverse shapes, making it difficult for a unified sampling strategy to adapt to all regions.

To address this limitation, we replace the shared sampling projections with expert-specific layers. Each query token $\mathbf{q}_i$ is routed to one of E experts using a Gumbel-Softmax-based gating network with hard sampling:

$$\boldsymbol{\pi}_i = \mathrm{GumbelSoftmax}(\mathbf{W}_g \mathbf{q}_i + \mathbf{b}_g;\, \tau,\, \text{hard=True}) \quad (6)$$

Here, $\mathbf{W}_g$ and $\mathbf{b}_g$ are the projection weights and bias of the gating network, and τ is the temperature controlling the softness of the output. The hard=True setting produces a one-hot vector $\boldsymbol{\pi}_i$ in the forward pass, while allowing gradient flow through a soft distribution during backpropagation via the straight-through estimator.

The selected expert index $e_i = \arg\max(\boldsymbol{\pi}_i)$ is used only for grouping tokens in the forward pass. The corresponding expert-specific layers are used to generate the full set of sampling offsets and attention weights:

$$\Delta_i = \mathrm{Linear}^{(e_i)}_{\mathrm{offset}}(\mathbf{q}_i), \quad (7)$$

$$\alpha_i = \mathrm{Linear}^{(e_i)}_{\mathrm{weight}}(\mathbf{q}_i) \quad (8)$$

The subsequent attention computation follows the standard deformable attention formulation, using Δ_i and α_i for sampling and aggregation.

GS enables sparse token-level routing, encouraging structural specialization across experts and improving adaptability to diverse layouts, especially for rare structures. As each token activates only one expert, the design remains computationally efficient while retaining strong representational capacity.

3.3.2 Local Dense

To suppress jagged artifacts in boundary classification and improve edge continuity, we introduce the Local Dense Attention (LD). It operates exclusively on the highest-resolution feature map (level-0) and applies gated multi-head attention within fixed 3×3 spatial neighborhoods. This enables each token to aggregate local context in a content-aware and spatially coherent way, producing smoother and more plausible boundary predictions.

Let the level-0 feature tokens be denoted as $\mathbf{X}_0 \in \mathbb{R}^{N_g \times D}$, where $N_g = H_g \times W_g$ is the spatial resolution and D is the hidden dimension. The feature map is first projected to queries, keys, and values via shared projection matrices $\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_V \in \mathbb{R}^{D\times D}$ and reshaped for M attention heads of dimension $d = D/M$:

$$\mathbf{Q} = \mathrm{reshape}(\mathbf{X}_0 \mathbf{W}_Q), \quad (9)$$

$$\mathbf{K} = \mathrm{unfold}_{3\times3}(\mathrm{reshape}(\mathbf{X}_0 \mathbf{W}_K)), \quad (10)$$

$$\mathbf{V} = \mathrm{unfold}_{3\times3}(\mathrm{reshape}(\mathbf{X}_0 \mathbf{W}_V)) \quad (11)$$

Here, for each head $h \in \{1, \ldots, M\}$ and each token $i \in \{1, \ldots, N_g\}$, we denote $\mathbf{q}_i^{(h)} \in \mathbb{R}^d$ as the query vector, and $\{\mathbf{k}_{i,j}^{(h)}\}_{j=1}^{9}$ and $\{\mathbf{v}_{i,j}^{(h)}\}_{j=1}^{9}$ as the keys and values extracted from the 3×3 neighborhood around token i.

The attention weights over the local neighborhood are computed via scaled dot-product attention:

$$\alpha_{i,j}^{(h)} = \frac{\exp\left(\langle \mathbf{q}_i^{(h)}, \mathbf{k}_{i,j}^{(h)} \rangle / \sqrt{d}\right)}{\sum_{j'=1}^{9} \exp\left(\langle \mathbf{q}_i^{(h)}, \mathbf{k}_{i,j'}^{(h)} \rangle / \sqrt{d}\right)} \quad (12)$$

The updated token representation is obtained by aggregating value vectors weighted by attention coefficients:

$$\tilde{\mathbf{x}}_i = \mathrm{Concat}_{h=1}^{M} \left(\sum_{j=1}^{9} \alpha_{i,j}^{(h)}\, \mathbf{v}_{i,j}^{(h)} \right) \in \mathbb{R}^D \quad (13)$$

To avoid over-updating and preserve useful features from the original input, we introduce a learnable gate $\mathbf{g}_i \in (0,1)^D$ to adaptively fuse the original and refined token representations:

$$\mathbf{g}_i = \sigma(\mathrm{MLP}(\mathbf{x}_i)) \quad (14)$$

$$\mathbf{x}'_i = \mathbf{g}_i \odot \tilde{\mathbf{x}}_i + (1 - \mathbf{g}_i) \odot \mathbf{x}_i \quad (15)$$

After refinement, the updated level-0 tokens $\{\mathbf{x}'_i\}_{i=1}^{N_g}$ are concatenated with the lower-resolution features to form the complete sequence for downstream processing. By focusing attention on the most detailed feature map and enabling token-specific local aggregation, the LD module enhances the model's capability to represent fine layout boundaries and subtle structural variations without introducing significant overhead.

3.4 Edge-Enhanced Post-processing

To convert grid-level semantic and boundary predictions into structured layout elements, we adopt a connectivity-aware post-processing procedure. Given boundary logits $\mathbf{p}^{\text{bnd}} \in \mathbb{R}^{H\times W\times 3}$ (with classes 0 = background, 1 = edge, 2 = interior) and semantic logits $\mathbf{p}^{\text{sem}} \in \mathbb{R}^{H\times W\times C}$, the algorithm first generates a hard boundary map $\hat{b}_{i,j}$ via argmax, extracts connected interior regions, and merges nearby edge pixels. For each region, we compute a bounding box, aggregate the semantic distribution within it, and assign a predicted class $\hat{c}$ with confidence score $\hat{s}$ (used to assist in confirming $\hat{c}$, not for filtering).

Algorithm 1: Connectivity Post-processing

1 **Input:** Boundary logits $\mathbf{p}^{\text{bnd}} \in \mathbb{R}^{H\times W\times 3}$, semantic logits $\mathbf{p}^{\text{sem}} \in \mathbb{R}^{H\times W\times C}$

2 **Process:** Identify connected regions, merge edges, predict class, and refine boxes.

- Compute hard boundary map: $\hat{b}_{i,j} = \arg\max_c \mathbf{p}^{\text{bnd}}_{i,j,c}$
- Remove isolated class-2 pixels
- Extract connected class-2 components (8-connectivity)
- Dilate class-2 regions to absorb adjacent class-1
- Cluster remaining class-1 pixels via BFS
- For each region:
  - Get minimum bounding box
  - Average $\mathbf{p}^{\text{sem}}$ in box $\rightarrow \hat{c}$ and $\hat{s}$
  - Expand non-boundary edges using adjacent semantic scores
  - Clamp and rescale box coordinates
- **Return:** Final boxes $\hat{\mathcal{Y}} = \{(x_1, y_1, x_2, y_2, \hat{c}, \hat{s})\}$

4 Experiments

4.1 Evaluation Metrics

We report FPS (frames per second) to measure inference speed. Standard mAP (Lin et al., 2015) evaluates detection by integrating F1 scores over varying confidence thresholds. However, since DEEN does not rely on confidence scores during inference, mAP is not applicable. Instead, we adopt two F1-based metrics for other methods:

- **F1@BestThreshold:** Reports the highest F1 score obtained by sweeping over all thresholds, representing upper-bound performance.
- **F1@SampledConf:** Selects a global threshold based on 50 randomly sampled training examples and applies it to the entire test set, simulating real-world deployment.

4.2 Datasets and Comparison DLA methods

To comprehensively evaluate our model, we use three representative datasets with different characteristics. **DocLayNet**(Pfitzmann et al., 2022)contains 80,863 manually annotated pages across 7 document types and 11 layout categories, and serves as a challenging benchmark for structural generalization. **PubLayNet**(Zhong et al., 2019)includes approximately 340,000 pages with five common layout types, offering large-scale, automatically annotated data suitable for pretraining and scalability evaluation. **CDLA** (Hang, 2021) provides 5,000 training and 1,000 validation images. Despite its smaller size, it features high-quality annotations over diverse layout elements, making it well-suited for fine-grained modeling and complementary to larger datasets.

We conduct comparisons with models spanning multiple architectural paradigms. Multimodal approaches include LayoutLMv3(Huang et al., 2022), DiT-Cascade(Li et al., 2022), and VGT(Da et al., 2023). For unimodal baselines, we consider DINO(Zhang et al., 2022b) and Deformable-DETR(Zhu et al., 2021), both based on the DETR framework, as well as DocLayout-YOLO(Zhao et al., 2024b), which is specifically designed for document understanding, and the latest general-purpose YOLOv12(Tian et al., 2025). Additionally, we include SwinDocSegmenter(Banerjee et al., 2023), a recent instance segmentation-based method for document layout analysis. The implementation details are provided in Appendix A.

Model	Size	DocLayNet		PubLayNet		CDLA		FPS↑
		F_1^{75}	$F_1^{75:95}$	F_1^{75}	$F_1^{75:95}$	F_1^{75}	$F_1^{75:95}$	
YOLOv12m(2025)	20M	87.1 (86.6)	83.2 (82.0)	96.1 (96.0)	92.4 (92.0)	92.8 (92.4)	87.0 (86.6)	126.7
YOLO-Doc(2024b)	20M	89.5 (87.8)	84.4 (82.8)	96.9 (96.5)	91.7 (91.6)	93.8 (93.5)	86.1 (85.9)	76.3
SwinDoc(2023)	218M	86.2 (85.7)	82.4 (82.0)	96.5 (96.3)	94.7 (94.6)	-	-	2.6
DINO(2022b)	46M	89.5 (89.3)	84.3 (84.1)	96.7 (96.6)	96.1 (96.0)	93.7 (93.5)	92.7 (92.5)	26.7
LayoutLMv3(2022)	133M	88.2 (88.0)	80.3 (80.1)	96.5 (96.2)	94.3 (94.1)	93.6 (93.3)	91.0 (90.7)	8.3
DiT- Cascade(2022)	141M	89.0 (88.6)	80.8 (80.4)	96.5 (96.4)	94.1 (93.8)	93.1 (92.9)	91.2 (91.0)	8.6
VGT(2023)	266M	89.4 (89.3)	85.9 (85.9)	96.6 (96.4)	96.3 (96.0)	-	-	-
DEEN-T	21M	81.2	75.3	94.1	91.7	90.5	86.5	28.4
DEEN-S	26M	85.6	79.7	96.5	94.7	93.1	91.4	23.5
DEEN-B	34M	88.7	85.8	97.2	97.0	93.7	92.6	19.7

Table 1: Each cell reports F1@BestThreshold (outside the parentheses) as the primary comparison metric, and F1@SampledConf (inside the parentheses) as a reference for real-world applicability. Here, F_1^{75} denotes the F1 score at an IoU threshold of 0.75, while $F_1^{75:95}$ averages F1 over thresholds from 0.75 to 0.95 (step size 0.05). Detailed definitions of both metrics are provided in Section 4.1. Best and second best results are highlighted.

Type	~~NMS~~	~~Conf~~	~~OCR~~	Parallel
YOLO	✗	✗	✓	✓
DETR	✓	✗	✓	✓
Multi-modal	✗	✗	✗	✓
Autoregressive	✓	✓	✓	✗
DEEN (Ours)	✓	✓	✓	✓

Table 2: Comparison of types by functional properties. **~~NMS~~**: no need for Non-Maximum Suppression; **~~Conf~~**: no confidence-based filtering; **~~OCR~~**: does not rely on Optical Character Recognition modules; **Parallel**: supports parallel prediction. *Note: This table reflects standard behavior, though exceptions exist. For example, YOLOv10(2024) is designed without the need for NMS.*

4.3 Main Results

4.3.1 Confidence-Free Prediction

The highlighted comparisons in Table 1 use the F1@BestThreshold metric, which inherently favors confidence-based models. Under this setting, the advantages of DEEN are not fully reflected. For example, on DocLayNet's F_1^{75}, DEEN ranks at a mid level (see analysis in Appendix B).

However, The advantages of DEEN become clear when the evaluation metric is switched to F1@SampledConf, which more accurately reflects real-world deployment scenarios. Confidence-based models typically require extensive threshold tuning to approach their optimal performance (as shown in Figure 4), which is impractical in real applications. For instance, on DocLayNet's $F_1^{75:95}$, even after 50 different confidence threshold samples (already considered frequent in practice), the YOLO series still lags over 1% behind theirs upper bound. In contrast, DEEN achieves the most best and second-best results across multiple datasets without any tuning.

4.3.2 Low Sensitivity to IoU Variations

DEEN also exhibits stronger performance under stricter evaluation metrics. For example, on CDLA, models in the YOLO series show a significant performance drop of nearly 5 points when moving from F_1^{75} to $F_1^{75:95}$, highlighting their sensitivity to threshold changes and localization precision. In contrast, DEEN's performance only drops by only 1.1 points, indicating stronger consistency under stricter IoU constraints.

This advantage stems from DEEN's fundamentally different modeling approach. Instead of using anchor boxes, DEEN classifies visual tokens and builds connectivity graphs to generate structured region predictions. This leads to more precise and layout-consistent outputs.

4.3.3 Deployment Simplicity

Usability and inference efficiency are also critical factors in real-world deployment. As shown in Table 2, DEEN demonstrates clear advantages in terms of system simplicity. For example, while VGT achieves strong performance, it relies heavily on external OCR modules. These modules often require complex adaptation to different deployment environments, significantly increasing complexity.

DEEN's overall inference speed ranks in the middle among current methods. Although it is slower than the ultra-lightweight YOLO series, it is comparable to other vision-only models and significantly faster than multimodal approaches. Moreover, its post-processing requires no manual threshold tuning and has a computational cost that is at least an order of magnitude lower than the main inference pipeline, making it virtually negligible.

Conv	GS	LD	DocLayNet		PubLayNet		CDLA		FPS
			F_1^{75}	$F_1^{75:95}$	F_1^{75}	$F_1^{75:95}$	F_1^{75}	$F_1^{75:95}$	
			80.7	78.1	92.4	90.1	87.7	85.6	**24.5**
✓			85.1	82.7	96.1	92.8	91.6	89.7	21.7
✓	✓		87.5	83.9	97.0	93.4	92.0	90.9	17.2
✓	✓	✓	**88.7**	**85.8**	**97.2**	**97.0**	**93.7**	**92.6**	19.7

Table 3: Ablation study on the model architecture. **Conv** denotes using a convolutional backbone instead of patch embedding; **GL** indicates replacing odd-numbered attention blocks with Global Sparse Attention; **LD** replaces even-numbered attention blocks with Local Dense Attention.

4.4 Ablation Study

4.4.1 Feature Extraction Strategy

Although the final output depends on the highest-resolution feature map, the DEEN encoder still requires multi-scale inputs for effective feature extraction. Unlike traditional object detection models that predict bounding boxes, our model directly classifies tokens to construct a connectivity graph. This motivates the use of attention-based fusion over fixed convolutional aggregation, which may introduce redundant or misleading structure information before encoding.

To test this, we explored a convolution-free alternative by dividing the image into multi-scale patches and applying linear projections. This design aims to reduce redundant receptive fields with a comparable parameter budget, and leaves feature integration to the encoder. However, as shown in Table 3, this patch-based design leads to a clear performance drop on structurally complex datasets. For example, there is a 5.1% decrease in F_1^{75} on DocLayNet. These results suggest that, under our setup, convolution provides more robust multi-scale representations for structure-aware modeling.

4.4.2 Effect of GS and LD in GSLD

We compare the two main components of GSLD module—GS Attention and LD Attention against the baseline Deformable Attention. The GS is designed to address the significant variations in layout element shapes commonly found in complex documents (as discussed in Section 3.3). By routing tokens to different experts for diverse sampling, GS introduces more flexible attention patterns. As shown by the comparison between the second and third rows in Table 3, GS has minimal impact on PubLayNet, where layouts are relatively uniform, with improvements of less than 1% on both metrics. In contrast, on the more structurally complex DocLayNet, GS brings substantial gains of 2.8% and 1.4% in F_1^{75} and $F_1^{75:95}$ respectively, demonstrating its effectiveness in handling layout diversity and structural complexity.

The comparison between the third and fourth rows in Table 3 further shows that introducing the LD consistently improves DEEN's performance across all three datasets. This improvement primarily stems from the gated local perception mechanism, which allows each token to dynamically incorporate surrounding information through attention. This leads to finer-grained feature updates and is particularly beneficial for boundary modeling. LD helps eliminate irregular, jagged artifacts often observed in boundary predictions, resulting in smoother contours that better align with the true document structure. The improvement is particularly pronounced under the stricter IoU criterion of $F_1^{75:95}$, with gains of 2.2% on DocLayNet and 1.8% on CDLA.

5 Conclusion

We propose DEEN, which eliminates the need for confidence filtering and NMS in layout prediction. To achieve this without compromising inference speed, we designed the GSLD module to capture both global and local structural patterns. Compared to traditional confidence-based methods, DEEN produces deterministic outputs, offering greater stability and controllability.While other methods perform competitively under idealized metrics (F1@BestThreshold), their performance drops under more realistic conditions (F1@SampledConf), highlighting the robustness of DEEN's design. Further analysis shows that GS and LD improve the modeling of layout diversity and boundary precision, highlighting the value of structure-aware design for complex document parsing.

Limitations

Although DEEN's post-processing does not rely on Non-Maximum Suppression or confidence thresholds, allowing direct use of model outputs without additional tuning, its core logic is based on graph connectivity construction. This makes it more complex than the post-processing of many existing methods and requires extra graph libraries. Nonetheless, as discussed in our main experiments, even with a Python implementation, it runs significantly faster than model inference, typically taking only one-tenth of the inference time. We therefore consider its overhead negligible in most practical cases. Still, further optimization with C or C++ could be beneficial in latency-sensitive scenarios.

References

Ayan Banerjee, Sanket Biswas, Josep Lladós, and Umapada Pal. 2023. *SwinDocSegmenter: An End-to-End Unified Domain Adaptive Transformer for Document Instance Segmentation*, page 307–325. Springer Nature Switzerland.

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. 2020. End-to-end object detection with transformers. *Preprint*, arXiv:2005.12872.

Mingxu Chai, Ziyu Shen, Chong Zhang, Yue Zhang, Xiao Wang, Shihan Dou, Jihua Kang, Jiazheng Zhang, and Qi Zhang. 2024. Docfusion: A unified framework for document parsing tasks. *Preprint*, arXiv:2412.12505.

Cheng Da, Chuwei Luo, Qi Zheng, and Cong Yao. 2023. Vision grid transformer for document layout analysis. *Preprint*, arXiv:2308.14978.

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

Hao Feng, Zijian Wang, Jingqun Tang, Jinghui Lu, Wengang Zhou, Houqiang Li, and Can Huang. 2023. Unidoc: A universal large multimodal model for simultaneous text detection, recognition, spotting and understanding. *Preprint*, arXiv:2308.11592.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. 2024. Retrieval-augmented generation for large language models: A survey. *Preprint*, arXiv:2312.10997.

Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. *Preprint*, arXiv:1311.2524.

Hang. 2021. Cdla: A chinese document layout analysis dataset. https://github.com/buptlihang/CDLA.

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

Anoop Raveendra Katti, Christian Reisswig, Cordula Guder, Sebastian Brarda, Steffen Bickel, Johannes Höhne, and Jean Baptiste Faddoul. 2018. Chargrid: Towards understanding 2d documents. *Preprint*, arXiv:1809.08799.

Rahima Khanam and Muhammad Hussain. 2024. Yolov11: An overview of the key architectural enhancements. *Preprint*, arXiv:2410.17725.

Junlong Li, Yiheng Xu, Tengchao Lv, Lei Cui, Cha Zhang, and Furu Wei. 2022. Dit: Self-supervised pre-training for document image transformer. *Preprint*, arXiv:2203.02378.

Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. 2015. Microsoft coco: Common objects in context. *Preprint*, arXiv:1405.0312.

Weihong Lin, Qifang Gao, Lei Sun, Zhuoyao Zhong, Kai Hu, Qin Ren, and Qiang Huo. 2021. Vibertgrid: A jointly trained multi-modal 2d document representation for key information extraction from documents. *Preprint*, arXiv:2105.11672.

Birgit Pfitzmann, Christoph Auer, Michele Dolfi, Ahmed S Nassar, and Peter W J Staar. 2022. Doclaynet: A large human-annotated dataset for document-layout segmentation. page 3743–3751.

Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. 2016. You only look once: Unified, real-time object detection. *Preprint*, arXiv:1506.02640.

Ruiyang Ren, Yingqi Qu, Jing Liu, Wayne Xin Zhao, Qiaoqiao She, Hua Wu, Haifeng Wang, and Ji-Rong Wen. 2023. Rocketqav2: A joint training method for dense passage retrieval and passage re-ranking. *Preprint*, arXiv:2110.07367.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2016. Faster r-cnn: Towards real-time object detection with region proposal networks. *Preprint*, arXiv:1506.01497.

Yunjie Tian, Qixiang Ye, and David Doermann. 2025. Yolov12: Attention-centric real-time object detectors. *Preprint*, arXiv:2502.12524.

Ao Wang, Hui Chen, Lihao Liu, Kai Chen, Zijia Lin, Jungong Han, and Guiguang Ding. 2024. Yolov10: Real-time end-to-end object detection. *Preprint*, arXiv:2405.14458.

Bin Xiao, Haiping Wu, Weijian Xu, Xiyang Dai, Houdong Hu, Yumao Lu, Michael Zeng, Ce Liu, and Lu Yuan. 2023. Florence-2: Advancing a unified representation for a variety of vision tasks (2023). *URL https://arxiv. org/abs/2311.06242*.

Xiao Yang, Ersin Yumer, Paul Asente, Mike Kraley, Daniel Kifer, and C. Lee Giles. 2017. Learning to extract semantic structure from documents using multimodal fully convolutional neural network. *Preprint*, arXiv:1706.02337.

Hang Zhang, Yeyun Gong, Yelong Shen, Jiancheng Lv, Nan Duan, and Weizhu Chen. 2022a. Adversarial retriever-ranker for dense text retrieval. *Preprint*, arXiv:2110.03611.

Hao Zhang, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun Zhu, Lionel M. Ni, and Heung-Yeung Shum. 2022b. Dino: Detr with improved denoising anchor boxes for end-to-end object detection. *Preprint*, arXiv:2203.03605.

Peng Zhang, Can Li, Liang Qiao, Zhanzhan Cheng, Shiliang Pu, Yi Niu, and Fei Wu. 2021. Vsr: A unified framework for document layout analysis combining vision, semantics and relations. *Preprint*, arXiv:2105.06220.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. 2024a. A survey of large language models. *Preprint*, arXiv:2303.18223.

Zhiyuan Zhao, Hengrui Kang, Bin Wang, and Conghui He. 2024b. Doclayout-yolo: Enhancing document layout analysis through diverse synthetic data and global-to-local adaptive perception. *Preprint*, arXiv:2410.12628.

Xu Zhong, Jianbin Tang, and Antonio Jimeno Yepes. 2019. Publaynet: largest dataset ever for document layout analysis. *Preprint*, arXiv:1908.07836.

Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. 2021. Deformable detr: Deformable transformers for end-to-end object detection. *Preprint*, arXiv:2010.04159.

A Implementation Details

	L	H	P	Params
DEEN-T	6	64	4	21M
DEEN-S	9	128	4	26M
DEEN-B	12	256	4	34M

Table 4: Configurations of DEEN variants. L: number of encoder layers; H: hidden dimension; P: number of sampling points in the GS attention module.

In all experiments, the largest-scale feature maps are uniformly downsampled by a factor of 4 across all datasets, and ResNet-50 is adopted as the convolutional backbone. The GS attention module is configured with 4 MoE experts. Model training is conducted using the AdamW optimizer and a cosine learning rate scheduler, with an initial learning rate of 1e-4 and a batch size of 64. Other configuration details for the three model sizes are summarized in Table 4. All experiments are conducted using 8 NVIDIA A100 GPUs.

Training is performed for 60 epochs on DocLayNet, 8 epochs on PubLayNet, and 40 epochs on CDLA. During the first half of training, the KL divergence for semantic labels is computed with uniform token-wise weighting. In the second half, the weights are dynamically adjusted according to the class distribution within each sample: the most frequent class is assigned a weight of 1, while other classes are scaled proportionally. This strategy is designed to prevent overrepresented categories from dominating the gradient updates.

B Data Noise

As shown in the results of Experiment 4.3.1, DEEN does not exhibit a significant advantage over traditional state-of-the-art baselines under the F1@Best Threshold metric, particularly on the DocLayNet dataset. Further analysis suggests that this is largely due to label noise introduced during the automated annotation process.

As described in Section 3.1, we employ a script to convert bounding box annotations into training labels for DEEN by computing their intersection with grid cells. While this automation reduces manual effort, it implicitly assumes that all pixels within a bounding box belong to the same semantic region. In practice, however, many layout elements—such as titles, tables, and figures—contain substantial white space or padding, leading to a mismatch between the annotated bounding box and the actual visual content (see Figure 5).

This misalignment introduces two key issues: (1) it degrades boundary learning by treating empty regions as foreground, and (2) it distorts semantic distributions across grids, especially when blank areas dominate. Together, these factors compromise label quality and negatively affect performance, particularly in terms of boundary precision and semantic consistency.

C Breadth-First Search

To refine boundary predictions during post-processing, we apply Breadth-First Search (BFS) to cluster residual edge pixels labeled as class-1. These pixels typically lie between disconnected or fragmented semantic regions and are not part of the interior components extracted earlier.

BFS is a classical graph traversal algorithm that explores all neighbors of a node before moving to the next level, ensuring a layer-wise search order. In our setting, we treat each class-1 pixel as a node in an 8-connected undirected grid graph. For each unvisited class-1 pixel, we initiate a BFS traversal to collect all reachable class-1 pixels into a single connected component. This process is repeated until all such pixels have been visited and clustered.

Formally, for a given binary mask $\mathbf{M} \in \{0,1\}^{H\times W}$ indicating class-1 pixels, the algorithm proceeds as follows:

- Iterate over all (i, j) where $\mathbf{M}_{i,j} = 1$ and not yet visited;
- Initialize a queue with (i, j) and mark as visited;
- While the queue is not empty:
  - Pop the front pixel (u, v) and add it to the current component;
  - For each 8-connected neighbor (u', v') of (u, v):
    - If $\mathbf{M}_{u',v'} = 1$ and unvisited, enqueue and mark as visited;
- Store the full connected component for further box refinement.

This clustering step allows us to preserve fragmented yet semantically meaningful edge regions, which are subsequently incorporated into bounding box expansion and class estimation. Compared to fixed morphological operations, BFS offers more flexible and content-aware connectivity modeling.

D F1@SampledConf

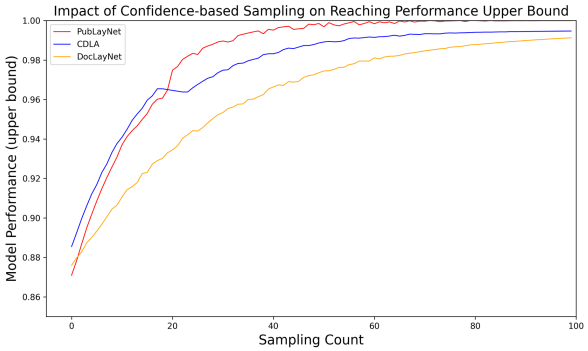

Figure 4: Average model performance on three datasets under confidence-based sampling.

This process approximates the way confidence thresholds are automatically set based on a limited number of validation samples in real-world deployments, thereby offering a more objective evaluation of the model's adaptability and stability in practical scenarios. As shown in Figure 4, as the number of sampled validation examples increases, the evaluation performance progressively approaches the model's upper bound, indicating that more samples help achieve a more accurate threshold setting. However, beyond a certain point, performance gains saturate, and further sampling yields diminishing returns. This suggests that the method achieves high stability and practicality even with a relatively small sampling scale. Nevertheless, due to sample diversity and the complexity of real-world applications, there may still be cases where the selected threshold does not perfectly match the true optimal setting. Future work may explore more robust threshold adjustment strategies to further improve generalization.

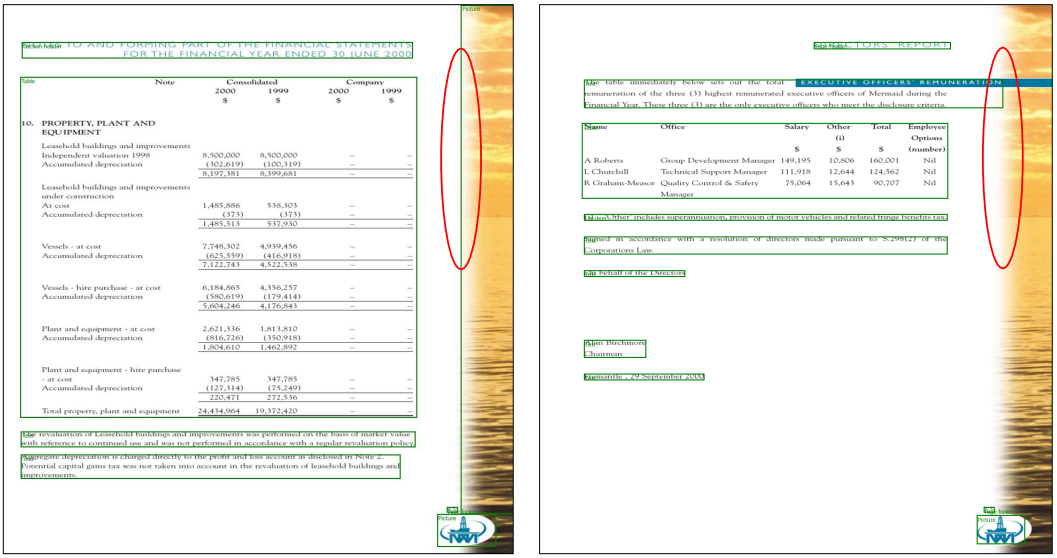

Figure 5: Labeling inconsistencies may lead to unexpected behaviors.

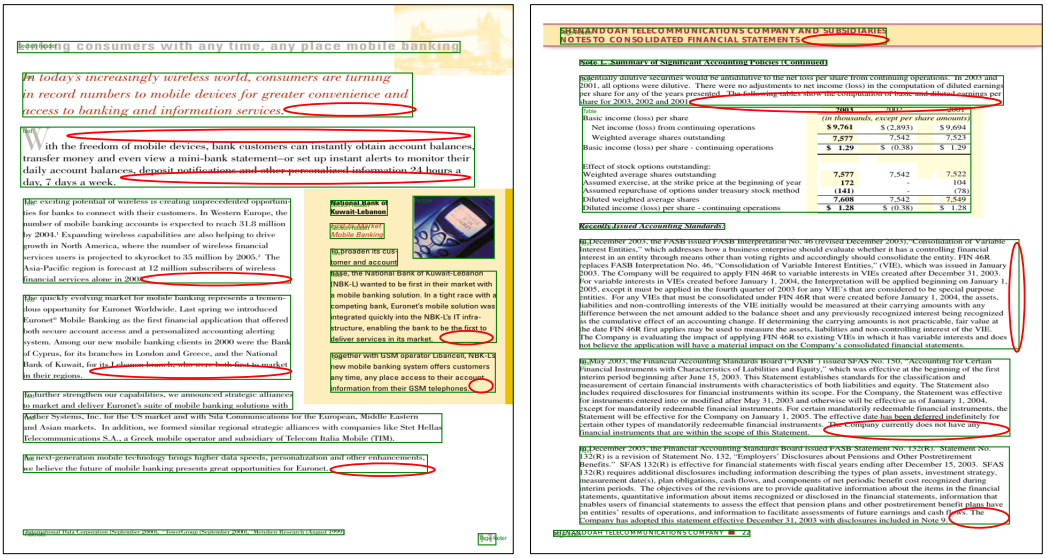

Figure 6: Potential noise introduced by automatic annotation

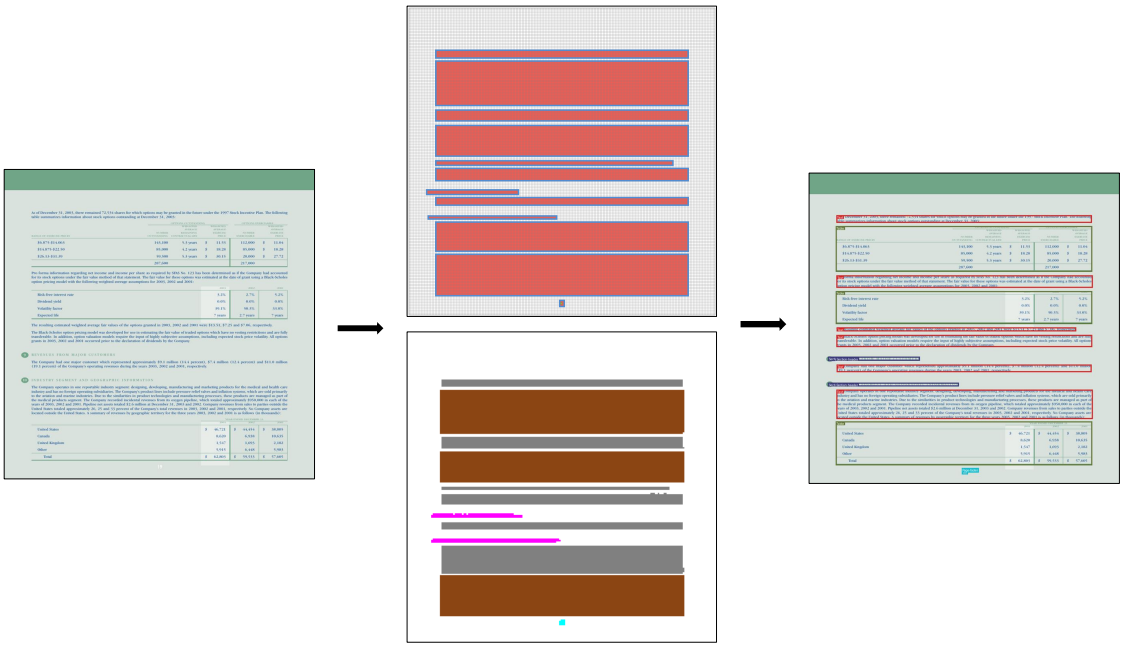

Figure 7: Example 1

Figure 8: Example 2

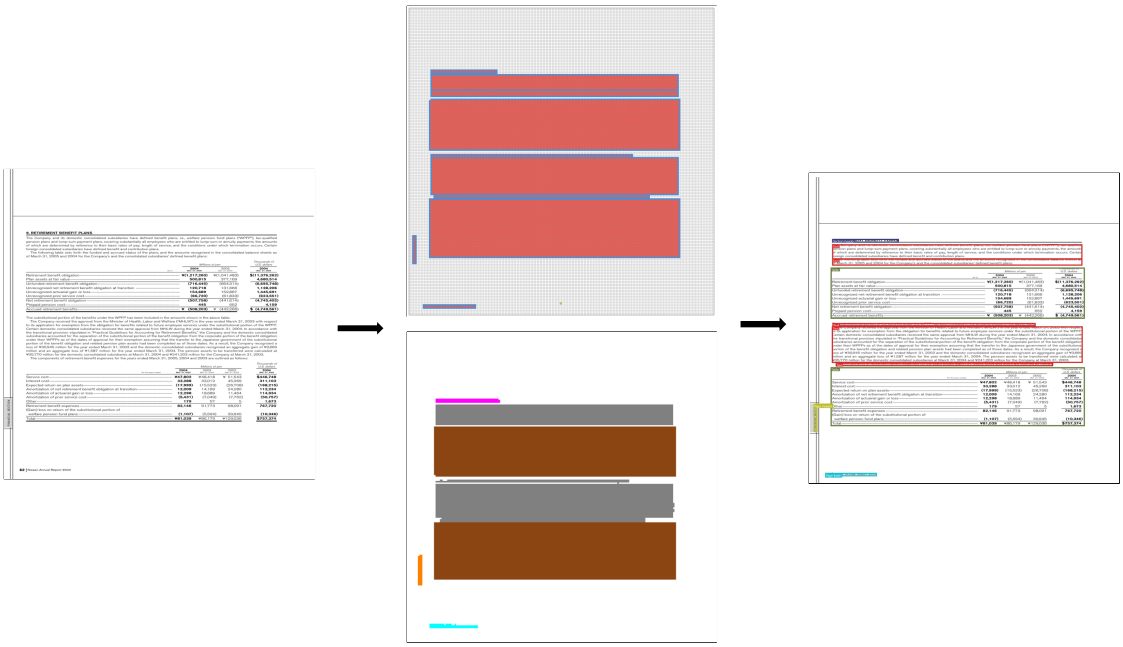

Figure 9: Example 3