Evaluation of YOLO Architectures for Document Layout Analysis of Historical Newspapers

Thomas Bianchi Todt
tbtodt@inf.ufpr.br
Departamento de Informática
Universidade Federal do Paraná
Curitiba, Paraná, Brazil

Eduardo Jaques Spinosa
spinosa@inf.ufpr.br
Departamento de Informática
Universidade Federal do Paraná
Curitiba, Paraná, Brazil

Abstract

Projects related to the preservation, democratization of access, and research of historical documents heavily benefit from Document Layout Analysis (DLA) and downstream tasks such as Optical Character Recognition. In recent years, neural networks developed for general-purpose object detection have seen immense interest and rapid development, resulting in improved accuracy and efficiency. Given the similarity between layout analysis and object detection, this research conducts a comparative study of the latest-generation object detection algorithms on DLA tasks. Specifically, we evaluate and compare the performance of YOLOv9, YOLOv10, YOLOv12, and a custom-designed hybrid model on both a large-scale modern dataset (PubLayNet) and a specialized historical collection (OCR-D). Furthermore, this work investigates the impact of transfer learning, analyzing the generalization from clean, contemporary documents to noisy, complex historical ones. The results demonstrate that these state-of-the-art models are highly effective for DLA and that fine-tuning a model pre-trained on a modern dataset significantly improves performance on historical documents, with YOLOv9 and YOLOv12 emerging as the top-performing architectures.

Keywords

Document Layout Analysis, Object Detection, Computer Vision, Deep Learning, YOLO

1 Introduction

The preservation of large amounts of historical documents as digital media has become a popular practice in recent years. The digitization makes it possible for multiple researchers in different places to access the same sources at the same time [16]. In this context, Computer Vision (CV) techniques have been applied in order to aid the efforts of researchers interested in these documents. One of the main contributions of CV to this field is automatic Optical Character Recognition (OCR), which extracts text in the images, making it searchable across the base of documents. The correct OCR of an image depends on several factors related to the preservation of the original files and of the manual operation of the digitization task. Therefore, it is usual that other steps take place before the OCR stage, like the deskew and binarization of images of pages. While deskewing involves rotating the image so that the scanned page alligns with the horizontal and the vertical axis, binarization is the process of converting the colored image to a new image with only two pixel values, aiming to simplify the image and trying to enhance contrast of key features [7].

Another stage that precedes OCR is the detection of text regions in the image. With this, regions of the image containing text can be cut out of the original image and analyzed individually. This is crucial for the process of OCR because it is easier to correctly identify text in input images if there is less unimportant information to be processed by the neural networks, which are usually the type of tools involved in this step [7].

The text regions detected in this stage can be further classified in different classes, like "paragraph" or "title". This is all part of the task known as Document Layout Analysis, which involves not only the classification of text regions, but the detection of image regions, text separators, tables, and others. Layout Analysis is also concerned with the reading order of the text and how the elements detected are related in general.

Besides its value for studies on humanities and language, DLA also provides useful metadata for higher-level semantic understanding. This structural data can be harnessed by modern AI systems, such as Large Language Models (LLMs), as valuable information that can improve the reasoning capabilities and context awareness of such models [11].

In this work, state-of-the-art computer vision techniques, specifically the latest iterations of the YOLO (You Only Look Once) architecture [17], were evaluated and compared on their effectiveness on the DLA task. The models were evaluated on both a modern benchmark and on a historical collection as well.

The primary contributions of this work are:

(1) A head-to-head comparison of object detectors of the latest generation (YOLOv9, YOLOv10, YOLOv12) for DLA, executed on PubLayNet, a dataset of widespread use on this field.
(2) An analysis of the generalization gap and the impact of transfer learning when transitioning from the domain of clean, contemporary scientific articles to that of noisy, complex historical documents.
(3) The evaluation of a custom-designed YOLO architecture, which explores a combination of features present in the other recent models of the family.

2 Related Works

Research on layout analysis on historical documents evolved in a tight relationship with other visual tasks. As such, early methods relied heavily on carefully constructed heuristics in order to classify regions of text, words, and characters. As testament to this, in the Historical Document Layout Analysis Competition [2] of the 2011 International Conference on Document Analysis and Recognition (ICDAR) every submission was based on a binarisation

pre-processing, followed by stages that implemented different heuristics taking information from white areas, the size and spacing of components in order to build polygons identifying text areas, separators, graphic elements, and images.

Contemporary methods, in turn, take advantage of deep learning approaches in order to automatically extract meaningful features from pages and guide the process of identifying regions of interest. On real tasks, Faster R-CNN [18], a two-stage object detector, showed good performance in layout analysis on early printed arabic documents [1], while Resnet-50 [6] was applied, with good results, to the task of segmenting historical proto-census documents of Ottoman origin [3]. The dominance of deep learning approaches was also observed on the ICDAR 2024 Competition on Few-Shot and Many-Shot Layout Segmentation of Ancient Manuscripts (SAM) [31]. The winner in that competition used an approach based on HookFormer [24], which leverages Cross-Attention Swin-Transformer and Cross-Interaction modules, facilitating the network's ability to model long-range dependencies, incorporating global contextual information, and performing detailed local interactions.

Besides that, other developments have arisen in the field of automatic layout analysis. Focusing specifically in the DLA task, [25] designed the Dynamic Residual Fusion (DRF) module in order to fully utilize low-dimensional information and maintain high-dimensional category information. The authors also propose the "dynamic select" mechanism to deal with overfitting issues on fine-tuning. In another work, [12] presented a neural network leveraging split attention and a Cascade RPN [20] head, arguing that since documents usually present mostly non-overlapping rectangular regions (ignoring the subdivisions of text-line, word, and character), the 2-stage detection coupled with adaptive convolution would be enough to detect elements. Also considering the nature of the shapes of target objects, [30] explored the potential of bounding boxes and positional encoding in guiding both pixel-wise segmentation and object detection, based on the idea that it could help encoding the continuity of the regions of interest.

Recognizing the relevance of bounding boxes for the accurate detection of regions of texts, the ICDAR 2023 Competition on Hierarchical Text Detection and Recognition [10] defined PQ (Panoptic Quality) as a target metric for their ranking, assuming that, since this metric prizes better tightness of the bounding boxes, it could reflect in better results for downstream tasks of text recognition. The methods that performed the best in this competition treated words, lines and paragraphs as generic objects, with a separate prediction branch for each, and then building hierarchy on post-processing with rules based on the intersection of detection areas. Interesting results overall were that two-stage methods achieved much better results than end-to-end ones, and also that the correlation between tightness scores and F1 scores was very low. Much like the solutions presented in this competition, the authors of [27] considered that regions of text and other layout elements can be understood as generic objects, and thus they built the FNO-YOLO. This model was based on the generic object detector Vit-YOLO [28], which combines the single-state detector YOLOv4 [14] and visual transformers [5]. The model was further improved by replacing the self-attention module with a novel token mixing module based on the Fast Fourier Transform neural operator, which resulted in better Precision and Recall, as well as a lower computational complexity.

Other works have also shown promising results for the application of transfomer-based architectures to the task of layout analysis. A fully transformer-based architecture was proposed by [26], comprising of a transformer backbone and a transformer encoder-decoder and utilizing visual features only. While also working with transformers, [4] opted for a multi-modal method, leveraging both visual and textual features, which are first processed in parallel branches and later fused together, with decoupled pre-training strategies for each branch.

The reviewed literature illustrates a clear evolutionary path for layout analysis, transitioning from early heuristic-based methods to the current dominance of deep learning. Moreover, purpose-build architectures have benefited largely from advancements in CV in general, which can also be seen by the adoption of transformers in more recent works. This points to the continued value of adapting novel, general-purpose computer vision methods to the specific domain of document layout analysis.

3 Approach

In this chapter, the tools and methods compared and developed in this paper are presented. The tools chosen represent the state of the art in object detection. The data used in the experiments is presented later in this same chapter.

3.1 YOLO Overview

YOLO [17] networks are known for their ability to make reliable real-time predictions, achieve good results in object detection tasks, and operate without requiring high computational power—characteristics.

The general architecture of any YOLO model can be described as comprising three parts: backbone, neck, and head:

- **Backbone:** The backbone network plays a crucial role in feature extraction from the input image and is typically structured as a Convolutional Neural Network (CNN). It is designed to capture features at different levels of abstraction: earlier layers detect basic elements such as edges and textures, while deeper layers capture more complex, high-level features like object parts and semantic information.
- **Neck:** The neck functions as an intermediary module that bridges the backbone and the head. Its primary role is to consolidate and refine the features extracted by the backbone, focusing on enhancing spatial and semantic information across various scales. This component may integrate additional convolutional layers and Feature Pyramid Networks (FPNs) [9], which operate by downsampling low-level feature maps and subsequently fusing them with higher-level feature maps. Each feature map of different scale in a FPN is known as a "level", where lower levels capture high-resolution, fine-grained details and higher levels capture more abstract, coarse information.
- **Head:** The head constitutes the final component of a YOLO network and is responsible for generating predictions based on the enriched features provided by the neck. It typically consists of one or more specialized subnetworks

designed for specific tasks. In an object detection framework, the head may be structured as a unified module that simultaneously performs bounding box prediction and classification. Alternatively, it can be decoupled into separate branches, with one branch dedicated to localization and the other to classification. This latter approach has been demonstrated to enhance both performance and training convergence. Finally, a post-processing step, such as Non-Maximum Suppression (NMS), is commonly employed to eliminate overlapping predictions and retain only the most confident detections.

3.2 YOLOv9

YOLOv9 [22] addresses mainly the issue of information loss during the feedforward process. This problem, known as information bottleneck, is inherent to Deep Learning Networks, since input data is processed and transformed through many layers.

In an attempt to mitigate this problem, the developers propose the concept of Programmable Gradient Information (PGI). PGI is composed by three components: main branch, auxiliary reversible branch, and multi-level auxiliary information. The main branch is the only one that is used during inference, therefore these additions don't result in additional inference cost. The auxiliary reversible branch is used to feed the main branch reliable gradient information during parameter learning. Multi-level auxiliary information works on a similar fashion, and is proposed as an improvement on deep supervision, aggregating gradient information concerning target objects of different sizes.

YOLOv9 developers also propose a new network architecture, GELAN (Generalized Efficient Layer Aggregation Network), which combines previous network achitectures CSPNet (Cross Stage Partial Network) and ELAN, essentially improving the latter by making it able to use any computational blocks, since the original was limited to convolutional layers. The YOLOv9-M model achieved an AP(50:95) (Average Precision, averaged over IoU (Intersection over Union) thresholds from 0.50 to 0.95) of 51.5% on the validation set of the COCO [8] dataset.

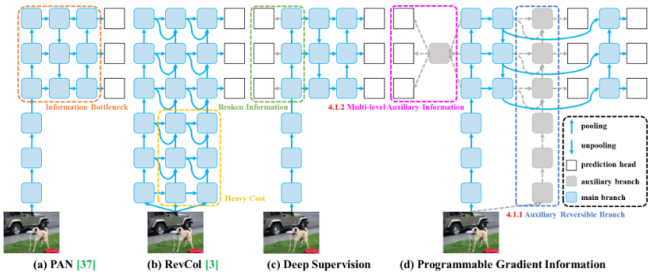

Figure 1: PGI graphic from [22]

3.3 YOLOv10

YOLOv10 [21] developers identified two characteristics of YOLO models that hindered performance and speed: the NMS post-processing, and the existence of considerable computational redundancy within architecture designs.

To tackle the post-processing issue, the authors developed a training strategy with dual label assignment. By jointly optimizing both a one-to-one and a one-to-many head and discarding the latter during inference, they avoid the NMS post-processing, while still leveraging plentiful supervisory signals during training.

To solve the issue related to the architecture designs, the authors propose a series of improvements that they called Holistic Efficiency-Accuracy Driven Model Design. Also aiming to improve efficiency, they implemented a lightweight classification head, since they discovered through experimentation that the regression head had more impact on the performance of the model and so the classification head could be simplified. The authors also propose to decouple spatial reduction from channel increase during the downsampling stage, which they achieved by swapping standard convolutions for a sequence of pointwise and depthwise convolutions, which leads to lower computational cost and parameter count.

The paper also introduced a strategy to tackle redundant computation based on first calculating the intrinsic rank of each stage to identify redundancy, and then progressively swap the basic blocks of the stages with the lowest calculated values for a new compact block they present, doing this for every stage while there is no performance degradation.

To improve performance the authors presented a Partial Self-Attention (PSA) module design, feeding just part of the features to the multi-head self-attention module. With this, and being placed only in deeper states with the lowest resolution, it is possible to leverage the global modeling capability of self-attention, while avoiding excessive overhead. Additionally, seeking to enlarge the receptive field in small models, the authors explored the use of large-kernel convolutions, increasing the kernel size of convolutions within deep states from 3x3 to 7x7. YOLOv10-M achieved an AP(50:95) of 51.1% on the validation set of the COCO dataset.

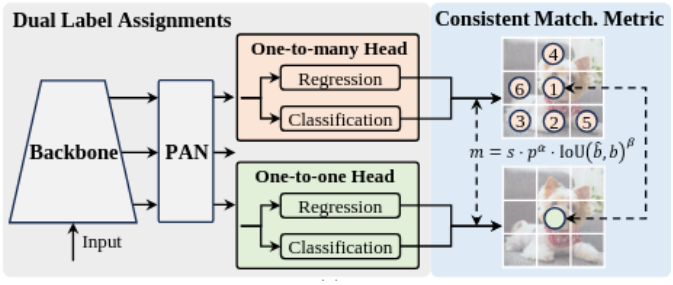

Figure 2: Dual label assignment graphic from [21]

3.4 YOLOv12

YOLOv12 [19] was proposed as an attention-centric YOLO architecture. The main drawback when using methods based on attention mechanisms is that such mechanisms usually show very high computational complexity, making training and inference more costly on time and resources.

In order to tackle this problem and still leverage the capabilities of attention to extract global information based on long-range dependencies, the authors developed the Area Attention (A2)

module, reducing complexity by dividing the input maps in sections and then calculating attention over these smaller areas, while still maintaining large receptive fields. A "position perceiver" component was also added to the attention mechanism, which applies large kernel convolutions to encode some positional information and add it to attention values. In addition to this, the ratio of hidden dimension to the embedding dimension is reduced in this part of the network, shifting the computational towards the attention mechanism and highlighting its importance.

Moreover, seeking to solve problems of bad gradients, convergence and optimization, the authors proposed Residual Efficient Layer Aggregation Networks (R-ELAN), which not only includes A2 modules, but also builds upon ELAN by introducing a residual shortcut from input to output throughout the block, and also promotes less parameter and memory usage by getting rid of an early split of the feature maps inside the block, favoring a bottleneck structure.

Other optimizations aimed at reducing computational cost include the use of FlashAttention and ditching of positional encoding. Five scaled versions of YOLOv12 are provided, and the YOLOv12-M model achieved an AP(50:95) of 52.5% on the validation set of the COCO dataset.

Figure 3: A2 graphic from [19]

3.4.1 Combined-YOLO. Analysis of the YOLO model codebases reveals that their development often proceeds in parallel rather than in a streamlined, cumulative fashion. For example, YOLOv10 and YOLOv12 do not incorporate the PGI module proposed in YOLOv9, and YOLOv12 similarly does not adopt the NMS-free strategy introduced in YOLOv10.

With this, we present a custom YOLO model that combines characteristics of the three other models experimented with in this work. YOLOv12 was used as the base to build the backbone and the main branch. The PGI strategy from YOLOv9 was adapted, with modules similar to those used in YOLOv12. Finally, the efficient head and NMS-free strategy from YOLOv10 were also incorporated in the model. This model is further referred as C-YOLO in this work.

3.5 Datasets

3.5.1 OCR-D. OCR-D [13] is a project developed with the goal to provide full-text recognition, initially with the specific goal of digitizing a set of early modern texts originating from German-speaking countries. The OCR-D project provides a set of annotated instances of historical documents [15]. An instance consists of the image of a page, and a corresponding XML file that encodes layout information and contains the text that can be seen in the image. This dataset consists of 217 instances, encompassing 9 classes for detection across 45314 delimited regions.

Index	From	Module	Arguments	From
Backbone				
0	-1	Silence	[]	
1	-1	Conv	[64, 3, 2]	v12
2	-1	Conv	[128, 3, 2, 1, 2]	v12
3	-1	C3k2	[256, False, 0.25]	v12
4	-1	Conv	[256, 3, 2, 1, 4]	v12
5	-1	C3k2	[512, False, 0.25]	v12
6	-1	Conv	[512, 3, 2]	v12
7	-1	A2C2f	[512, True, 4]	v12
8	-1	Conv	[1024, 3, 2]	v12
9	-1	A2C2f	[1024, True, 1]	v12
Neck				
10	-1	nn.Upsample	[None, 2, "nearest"]	v12
11	[-1,7]	Concat	[1]	v12
12	-1	A2C2f	[512, False, -1]	v12
13	-1	nn.Upsample	[None, 2, "nearest"]	v12
14	[-1,5]	Concat	[1]	v12
15	-1	A2C2f	[256, False, -1]	v12
16	-1	Conv	[256, 3, 2]	v12
17	[-1,12]	Concat	[1]	v12
18	-1	A2C2f	[512, False, -1]	v12
19	-1	Conv	[512, 3, 2]	v12
20	[-1,9]	Concat	[1]	v12
21	-1	C3k2	[1024, True, 0.5]	v12
Auxiliar Branch				
22	5	CBLinear	[[240]]	v9
23	7	CBLinear	[[240, 360]]	v9
24	9	CBLinear	[[240, 360, 480]]	v9
25	0	Conv	[32, 3, 2]	v9 and v12
26	-1	Conv	[64, 3, 2]	v9 and v12
27	-1	A2C2f	[512, False, -1]	v9 and v12
28	-1	Conv	[240, 3, 2]	v9 and v12
29	[22,23,24,-1]	CBFuse	[[0,0,0]]	v9
30	-1	A2C2f	[256, False, -1]	v9 and v12
31	-1	Conv	[360, 3, 2]	v9 and v12
32	[23,24,-1]	CBFuse	[[1,1]]	v9
33	-1	A2C2f	[512, False, -1]	v9 and v12
34	-1	Conv	[480, 3, 2]	v9 and v12
35	[24,-1]	CBFuse	[[2]]	v9
36	-1	C3k2	[1024, True, 0.5]	v9 and v12
Head				
37	[15,18,21,30,33,36]	HybridDetect	[nc]	v9 and v10

Table 1: C-YOLO architecture: backbone and head modules with their configurations.

Model	Parameters (Training)	Parameters (Inference)
YOLOv9	32.8 M	20.1 M
YOLOv10	16.5 M	15.4 M
YOLOv12	19.6 M	19.6 M
C-YOLO	27.9 M	18.1 M

Table 2: Number of parameters for each model during training and inference.

Region Type	Count
TextRegion	1648
TextLine	6609
Word	36685
PrintSpace	173
Border	36
ImageRegion	1
SeparatorRegion	141
NoiseRegion	17
MusicRegion	4

Table 3: Count of instances of each class in the documents in the OCR-D dataset

It is relevant to notice that some classes have too few instances to provide adequate training for the neural network. Also, the class Word is usually contained within TextLine, which itself is usually contained within TextRegion.

The dataset was split with 80% instances for training, 10% for validation and 10% for test.

3.5.2 PubLayNet. This dataset consists of over 360k images corresponding to pages of PDF articles and it is one of the most popular datasets in the field of DLA. The pages were automatically annotated by matching the XML representation of the PDF original files to the images, and there are 5 layout categories that can describe a region: Text, Title, List, Table, and Figure [29]. Due to the massive size of this dataset, in this work we opted to work with a subset of this dataset, with a total of 47958 images.

Region Type	Count
Text	286900
Title	89349
List	11431
Table	14646
Figure	15555

Table 4: Count of instances of each class in the documents in the subset of the PubLayNet dataset

4 Experiments and Results

Results were obtained after training 200 epochs on the OCR-D dataset, while for the PubLayNet subset, due to its massive size, 10 epochs were used, which is also in line with the literature [4][30]. Training was executed on a machine with an NVIDIA RTX A4000 GPU (16GB VRAM), CUDA 12.8, and driver version 570.133.20. All hyperparameters were set to default values.

Table 5 shows the results obtained at the end of training on the subset of PubLayNet dataset [29]. It is worth mentioning that every model achieved state-of-the-art scores [23] [4]. These results, however, cannot be taken as conclusive, since the validation set used in this work is but a sample of the original. YOLOv9 and YOLOv12 were the best models overall in this set, with the former achieving higher Recall, while the latter scored highest in Precision and mean Average Precision (mAP). Among classes, *Figure* scores were consistently higher, which might reflect the fact that they are very different from the other categories, which are textual in nature. *List*, showed the lowest scores, which could indicate that this category is particularly hard to automatically distinguish from plain *Text* or *Table*.

The results for the models trained on the OCR-D dataset are shown in table 6. Results for classes with very low occurrence are most probably not representative of the capacity of the models, therefore OverallText results for Precision, Recall, and mAP are calculated considering only the *Word*, *TextLine*, and *TextRegion* classes (and likewise in table 7). In a similar manner to the experiment on the PubLayNet subset, we again observe better results by YOLOv9 and YOLOv12, but this time the highest Precision was by YOLOv9, with YOLOv12 achieving the highest Recall. YOLOv12 also achieved, again, the best overall mAP results. It should be also noticed that results for *TextRegion* were on average significantly worse than for *Word* and *TextLine*.

The results for models pre-trained on the PubLayNet set and later fine-tuned to the OCR-D dataset are shown in table 7. This evaluation shows that every model achieved better results with this strategy. YOLOv9 and YOLOv12 are again the best scorers, but the pre-training affected their results in such a way that YOLOv12 shows the best Precision score, with YOLOv9 achieving the best results overall for Recall and mAP.

In relation to the good Precision score of YOLOv12, fruit of significant gains with the pre-training strategy, it is interesting to notice that it was accompanied by a substantial lowering of its Recall score (in relation to the version trained from scratch), as evidenced in table 8. This table also shows that C-YOLO had the highest and most consistent gains overall when compared to the other models.

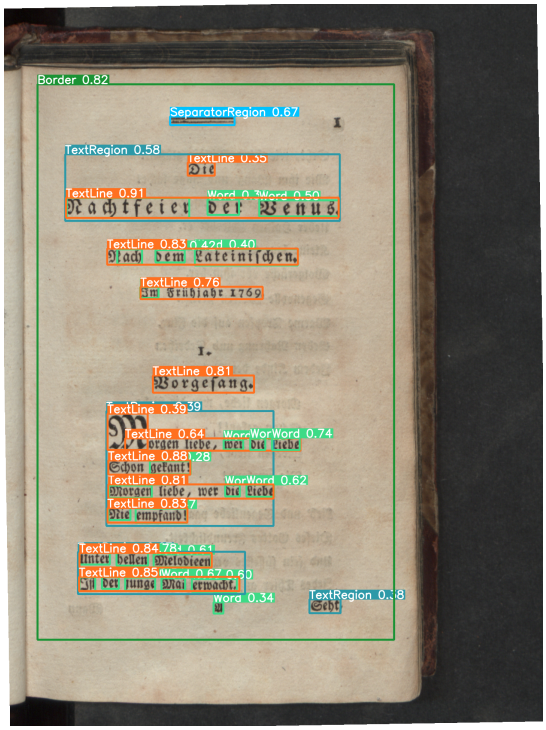

Figure 5: Detections by YOLOv9 on OCR-D images

5 Discussion

The results achieved by the modern object detectors on the DLA task confirm the effectiveness of these methods when applied to this more specific domain, which strengthens the notion that DLA can be treated as analogous to object detection. This doesn't discredit, however, the pixel-wise segmentation approaches.

The transfer learning technique applied when fine-tuning the YOLO models proved effective for the task of layout analysis of historical documents. The results were achieved by first training on a large modern dataset and then fine-tuning on a smaller dataset of historical documents. This approach is advantageous because

Table 5: Per-class and overall Precision, Recall, and mAP@50 for all models on the PubLayNet subset.

Class	YOLOv9			YOLOv10			YOLOv12			C-YOLO		
	P	R	mAP	P	R	mAP	P	R	mAP	P	R	mAP
Figure	98.0	**97.2**	**99.2**	96.9	96.5	98.6	**98.1**	**97.2**	**99.2**	97.3	96.6	98.8
List	90.7	**85.8**	91.4	92.4	79.9	90.5	92.6	83.1	**91.7**	**93.6**	81.2	91.1
Table	**97.1**	**98.4**	98.1	96.0	96.3	97.8	96.9	97.7	**98.5**	95.4	96.1	98.2
Text	96.5	94.2	97.9	95.4	93.7	97.9	**96.7**	**94.6**	**98.5**	95.9	93.4	98.1
Title	**96.1**	95.4	**97.9**	92.8	94.3	97.1	95.9	**95.7**	97.8	94.1	93.8	97.4
Overall	95.7	**94.2**	96.9	94.7	92.1	96.4	**96.1**	93.6	**97.1**	95.3	92.2	96.7

Table 6: Per-class and overall Precision, Recall, and mAP@50 for all models on the OCR-D dataset

Class	YOLOv9			YOLOv10			YOLOv12			C-YOLO		
	P	R	mAP	P	R	mAP	P	R	mAP	P	R	mAP
Word	**89.5**	**83.3**	**86.9**	84.3	79.2	83.2	83.0	83.3	86.0	77.5	79.9	81.4
TextLine	72.2	**81.7**	84.0	**75.9**	75.3	80.2	71.7	89.9	**89.8**	60.2	76.2	73.5
TextRegion	67.9	38.0	49.4	**68.8**	39.2	47.8	43.6	**65.5**	**58.4**	64.4	30.4	40.2
PrintSpace	**90.9**	94.4	95.2	72.8	**100**	93.1	83.0	**100**	**99.5**	71.5	**100**	92.2
Separator	33.4	7.69	16.3	85.1	7.69	17.9	26.3	**30.8**	**22.5**	**100**	0.0	0.08
Border	26.8	**100**	86.6	**62.1**	98.7	88.0	42.2	**100**	93.8	53.9	**100**	83.9
Noise	0.0	0.0	0.0	100	0.0	0.0	**100**	**69.6**	**99.5**	100	0.0	0.0
ImageRegion	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Music	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Overall	42.3	45.0	46.5	**61.0**	44.5	45.6	50.0	**59.9**	**61.1**	58.6	43.0	42.2
OverallText	**76.5**	67.7	73.4	76.3	66.3	70.4	66.1	**79.6**	**78.1**	67.4	62.2	65.0

Table 7: Per-class and overall Precision, Recall, and mAP@50 for all models on the OCR-D dataset, pre-trained on the PubLayNet subset

Class	YOLOv9			YOLOv10			YOLOv12			C-YOLO		
	P	R	mAP	P	R	mAP	P	R	mAP	P	R	mAP
Word	89.7	**86.1**	**89.4**	83.6	82.7	85.3	**89.9**	83.6	87.3	80.7	81.3	84.4
TextLine	82.5	82.3	85.8	85.9	84.1	88.5	**92.3**	**84.8**	**91.5**	76.4	81.4	85.2
TextRegion	69.7	**57.9**	**65.6**	60.1	54.4	56.1	68.6	49.8	59.7	**71.1**	53.2	57.2
PrintSpace	91.1	**100**	**99.5**	77.9	**100**	95.8	**92.1**	**100**	**99.5**	70.6	94.4	95.7
Separator	62.7	**51.9**	**59.8**	41.3	30.8	43.4	**83.7**	38.5	56.9	73.7	43.4	50.2
Border	81.9	80.0	96.2	65.6	80.0	72.0	**89.2**	**100**	**99.5**	52.4	80.0	69.5
Noise	**100**	0.0	0.0	**100**	0.0	0.0	**100**	0.0	0.0	**100**	0.0	0.0
ImageRegion	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Music	100	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Overall	**75.3**	**50.9**	**55.1**	57.2	48.0	49.0	68.4	50.7	54.9	58.3	48.2	49.1
OverallText	80.6	**75.4**	**80.3**	76.5	73.7	76.6	**83.6**	72.7	79.5	76.1	72.0	75.6

Figure 4: Detections by YOLOv12 on PubLayNet images

Table 8: Per-class and overall Precision, Recall, and mAP@50 changes with transfer learning strategy, selected classes

Class	YOLOv9			YOLOv10			YOLOv12			C-YOLO		
	P	R	mAP	P	R	mAP	P	R	mAP	P	R	mAP
Word	+0.2	+2.8	+2.5	-0.7	**+3.5**	+2.1	**+6.9**	+0.3	+1.3	+3.2	+1.4	+3.0
TextLine	+10.3	+0.6	+1.8	+10.0	+8.8	+8.3	**+22.6**	-5.1	+1.7	+16.2	+5.2	**+11.7**
TextRegion	+1.8	**+27.9**	+16.2	-8.7	+15.2	+8.3	**+25.0**	-15.7	+1.3	+6.7	+22.8	**+17.0**
OverallText	+4.1	+7.7	+6.9	+0.2	+7.4	+6.2	**+17.5**	-6.9	+1.4	+8.7	**+9.8**	**+10.6**

modern document datasets, such as PubLayNet, are abundant and can be automatically annotated using existing PDF XML data. In contrast, historical document datasets are typically much smaller and require manual annotation, making it essential to leverage knowledge learned from modern data to improve performance on historical materials.

The fact that YOLOv10 and C-YOLO did not perform as well as the other models can be hypothesized to result from the Dual Label Assignment (DLA) strategy implemented by both, along with the efficient lightweight head of YOLOv10, which may impair the learning of important high-level features required for the DLA task. This hypothesis is further supported by the discrepancy between YOLOv10 and YOLOv12 results, given that both employ similar, albeit different, attention mechanisms in their architectures. Nonetheless, both YOLOv10 and C-YOLO are the models with the fewest parameters utilized during inference, as seen in table 2, which could also contribute to their lower performance.

Conversely, the significant improvement observed for C-YOLO with the pre-training strategy may indicate the effectiveness of YOLOv9's PGI strategy in this context. This is also supported by the strong overall results achieved by YOLOv9 itself, which are comparable to YOLOv12, the latter benefiting from its powerful attention mechanism.

6 Conclusion

The results obtained in this study indicate that novel computer vision techniques can be successfully applied to document layout analysis (DLA) of historical documents. Combining transfer learning with state-of-the-art architectures yielded promising results not only on the historical dataset but also on the modern dataset.

Given the strong performance achieved on the PubLayNet subset, future work could extend these experiments to the full dataset to further validate the effectiveness of the proposed techniques.

Additionally, the results suggest that an architecture combining YOLOv9's PGI module with YOLOv12's Attention Area mechanism, while excluding YOLOv10's NMS-free strategy and lightweight head, could potentially achieve even better performance. Ideally, this could be investigated through an ablation study systematically evaluating the contributions of each architectural component.

Finally, recognizing that DLA is often a preliminary stage in OCR workflows, future research could explore how character and word recognition accuracy might provide additional useful metrics for assessing layout analysis performance, investigating this multimodal approach particularly in the context of historical document digitization.

References

[1] Latifa Aljiffry, Hassanin Al-Barhamtoshy, Amani Jamal, and Felwa Abukhodair. 2022. Arabic Documents Layout Analysis (ADLA) using Fine-tuned Faster RCN. In *2022 20th International Conference on Language Engineering (ESOLEC)*, Vol. 20. 66–71. doi:10.1109/ESOLEC54569.2022.10009375

[2] A. Antonacopoulos, C. Clausner, C. Papadopoulos, and S. Pletschacher. 2011. Historical Document Layout Analysis Competition. In *2011 International Conference on Document Analysis and Recognition*. 1516–1520. doi:10.1109/ICDAR.2011.301

[3] Yekta Said Can and M. Erdem Kabadayı. 2020. Developing an Automatic Layout Analysis System for Ottoman Population Registers. In *2020 28th Signal Processing and Communications Applications Conference (SIU)*. 1–4. doi:10.1109/SIU49456.2020.9302464

[4] Cheng Da, Chuwei Luo, Qi Zheng, and Cong Yao. 2023. Vision Grid Transformer for Document Layout Analysis. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*. 19405–19415. doi:10.1109/ICCV51070.2023.01783

[5] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net. https://openreview.net/forum?id=YicbFdNTTy

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 770–778. doi:10.1109/CVPR.2016.90

[7] Seong-Whan Lee and Dae-Seok Ryu. 2001. Parameter-free geometric document layout analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 23, 11 (2001), 1240–1256. doi:10.1109/34.969115

[8] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. Microsoft COCO: Common Objects in Context. In *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V (Lecture Notes in Computer Science, Vol. 8693)*, David J. Fleet, Tomás Pajdla, Bernt Schiele, and Tinne Tuytelaars (Eds.). Springer, 740–755. doi:10.1007/978-3-319-10602-1_48

[9] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. 2017. Feature Pyramid Networks for Object Detection. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 936–944.

[10] Shangbang Long, Siyang Qin, Dmitry Panteleev, Alessandro Bissacco, Yasuhisa Fujii, and Michalis Raptis. 2023. ICDAR 2023 Competition on Hierarchical Text Detection and Recognition. In *Document Analysis and Recognition - ICDAR 2023*, Gernot A. Fink, Rajiv Jain, Koichi Kise, and Richard Zanibbi (Eds.). Springer Nature Switzerland, Cham, 483–497.

[11] Chuwei Luo, Yufan Shen, Zhaoqing Zhu, Qi Zheng, Zhi Yu, and Cong Yao. 2024. LayoutLLM: Layout Instruction Tuning with Large Language Models for Document Understanding. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 15630–15640. doi:10.1109/CVPR52733.2024.01480

[12] Mohammad Minouei, Mohammad Reza Soheili, and Didier Stricker. 2021. Document Layout Analysis with an Enhanced Object Detector. In *2021 5th International Conference on Pattern Recognition and Image Analysis (IPRIA)*. 1–5. doi:10.1109/IPRIA53572.2021.9483509

[13] Clemens Neudecker, Konstantin Baierer, Maria Federbusch, Matthias Boenig, Kay-Michael Würzner, Volker Hartmann, and Elisa Herrmann. 2019. OCR-D: An end-to-end open source OCR framework for historical printed documents. *Proceedings of the 3rd International Conference on Digital Access to Textual Cultural Heritage* (2019). https://api.semanticscholar.org/CorpusID:204976059

[14] Mang Ning, Yao Lu, Wenyuan Hou, and Mihhail Matskin. 2021. YOLOv4-object: an Efficient Model and Method for Object Discovery. In *2021 IEEE 45th Annual Computers, Software, and Applications Conference (COMPSAC)*. 31–36. doi:10.1109/COMPSAC51774.2021.00016

[15] OCR-D. 2024. GT structure text. https://github.com/OCR-D/gt_structure_text/releases/tag/v1.5.0 Accessed: 2024-12-21.

[16] Brian Ogilvie. 2016. Scientific Archives in the Age of Digitization. *Isis* 107, 1 (2016), 77–85. arXiv:https://doi.org/10.1086/686075 doi:10.1086/686075

[17] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. 2016. You Only Look Once: Unified, Real-Time Object Detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 779–788. doi:10.1109/CVPR.2016.91

[18] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In *Advances in Neural Information Processing Systems*, C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett (Eds.), Vol. 28. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2015/file/14bfa6bb14875e45bba028a21ed38046-Paper.pdf

[19] Yunjie Tian, Qixiang Ye, and David S. Doermann. 2025. YOLOv12: Attention-Centric Real-Time Object Detectors. *CoRR* abs/2502.12524 (2025). arXiv:2502.12524 doi:10.48550/ARXIV.2502.12524

[20] Thang Vu, Hyunjun Jang, Trung X. Pham, and Chang D. Yoo. 2019. *Cascade RPN: delving into high-quality region proposal network with adaptive convolution*. Curran Associates Inc., Red Hook, NY, USA.

[21] Ao Wang, Hui Chen, Lihao Liu, Kai Chen, Zijia Lin, Jungong Han, and Guiguang Ding. 2024. YOLOv10: Real-Time End-to-End Object Detection. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (Eds.). http://papers.nips.cc/paper_files/paper/2024/hash/c34ddd05eb089991f06f3c5dc36836e0-Abstract-Conference.html

[22] Chien-Yao Wang, I-Hau Yeh, and Hong-Yuan Mark Liao. 2025. YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information. In *Computer Vision – ECCV 2024*, Aleš Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol (Eds.). Springer Nature Switzerland, Cham, 1–21.

[23] Papers with Code. 2024. Document Layout Analysis on PubLayNet (Val) | Papers with Code. https://paperswithcode.com/sota/document-layout-analysis-on-publaynet-val. Accessed: 2025-07-01.

[24] Fei Wu, Nora Gourmelon, Thorsten Seehaus, Jianlin Zhang, Matthias Braun, Andreas Maier, and Vincent Christlein. 2024. Contextual HookFormer for Glacier Calving Front Segmentation. *IEEE Transactions on Geoscience and Remote Sensing* 62 (2024), 1–15. doi:10.1109/TGRS.2024.3368215

[25] Xingjiao Wu, Ziling Hu, Xiangcheng Du, Jing Yang, and Liang He. 2021. Document Layout Analysis via Dynamic Residual Feature Fusion. In *2021 IEEE International Conference on Multimedia and Expo (ICME)*. 1–6. doi:10.1109/ICME51207.2021.9428465

[26] Huichen Yang and William Hsu. 2022. Transformer-Based Approach for Document Layout Understanding. In *2022 IEEE International Conference on Image Processing (ICIP)*. 4043–4047. doi:10.1109/ICIP46576.2022.9897491

[27] Yahui Yang, Shanhong He, Lihong Tang, Chenhui Dong, Tian Wan, and Rui Li. 2024. Enhancing Object Detection in Layout Analysis: Leveraging Vision Transformer and Fourier Neural Operator. In *2024 5th International Seminar on Artificial Intelligence, Networking and Information Technology (AINIT)*. 2236–2240. doi:10.1109/AINIT61980.2024.10581855

[28] Zixiao Zhang, Xiaoqiang Lu, Guojin Cao, Yuting Yang, Licheng Jiao, and Fang Liu. 2021. ViT-YOLO:Transformer-Based YOLO for Object Detection. In *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*. 2799–2808. doi:10.1109/ICCVW54120.2021.00314

[29] Xu Zhong, Jianbin Tang, and Antonio Jimeno-Yepes. 2019. PubLayNet: Largest Dataset Ever for Document Layout Analysis. In *2019 International Conference on Document Analysis and Recognition, ICDAR 2019, Sydney, Australia, September 20-25, 2019*. IEEE, 1015–1022. doi:10.1109/ICDAR.2019.00166

[30] Ejian Zhou, Xingjiao Wu, Luwei Xiao, Xiangcheng Du, Tianlong Ma, and Liang He. 2022. Document Layout Analysis Via Positional Encoding. In *2022 IEEE International Conference on Image Processing (ICIP)*. 1156–1160. doi:10.1109/ICIP46576.2022.9897330

[31] Silvia Zottin, Axel De Nardin, Gian Luca Foresti, Emanuela Colombi, and Claudio Piciarelli. 2024. ICDAR 2024 Competition on Few-Shot and Many-Shot Layout Segmentation of Ancient Manuscripts (SAM). In *Document Analysis and Recognition - ICDAR 2024*, Elisa H. Barney Smith, Marcus Liwicki, and Liangrui Peng (Eds.). Springer Nature Switzerland, Cham, 315–331.