applied sciences

MDPI

Article

Rethinking Learnable Proposals for Graphical Object Detection in Scanned Document Images

Sankalp Sinha [1], Khurram Azeem Hashmi [1,2,3,*], Alain Pagani [3], Marcus Liwicki [4] and Didier Stricker [1,3] and Muhammad Zeshan Afzal [1,2,3]

[1] Department of Computer Science, Technical University of Kaiserslautern, 67663 Kaiserslautern, Germany
[2] Mindgarage, Technical University of Kaiserslautern, 67663 Kaiserslautern, Germany
[3] German Research Institute for Artificial Intelligence (DFKI), 67663 Kaiserslautern, Germany
[4] Department of Computer Science, Luleå University of Technology, 971 87 Luleå, Sweden
* Correspondence: khurram_azeem.hashmi@dfki.de

Abstract: In the age of deep learning, researchers have looked at domain adaptation under the pre-training and fine-tuning paradigm to leverage the gains in the natural image domain. These backbones and subsequent networks are designed for object detection in the natural image domain. They do not consider some of the critical characteristics of document images. Document images are sparse in contextual information, and the graphical page objects are logically clustered. This paper investigates the effectiveness of deep and robust backbones in the document image domain. Further, it explores the idea of learnable object proposals through Sparse R-CNN. This paper shows that simple domain adaptation of top-performing object detectors to the document image domain does not lead to better results. Furthermore, empirically showing that detectors based on dense object priors like Faster R-CNN, Mask R-CNN, and Cascade Mask R-CNN are perhaps not best suited for graphical page object detection. Detectors that reduce the number of object candidates while making them learnable are a step towards a better approach. We formulate and evaluate the Sparse R-CNN (SR-CNN) model on the IIIT-AR-13k, PubLayNet, and DocBank datasets and hope to inspire a rethinking of object proposals in the domain of graphical page object detection.

Keywords: graphical page object detection; deep learning; computer vision; proposals; document image analysis

check for updates

Citation: Sinha, S.; Hashmi, K.A.; Pagani, A.; Liwicki, M.; Stricker, D.; Afzal, M.Z. Rethinking Learnable Proposals for Graphical Object Detection in Scanned Document Images. *Appl. Sci.* **2022**, *12*, 10578. https://doi.org/10.3390/app122010578

Academic Editor: Jose Santamaria

Received: 30 August 2022
Accepted: 14 October 2022
Published: 20 October 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

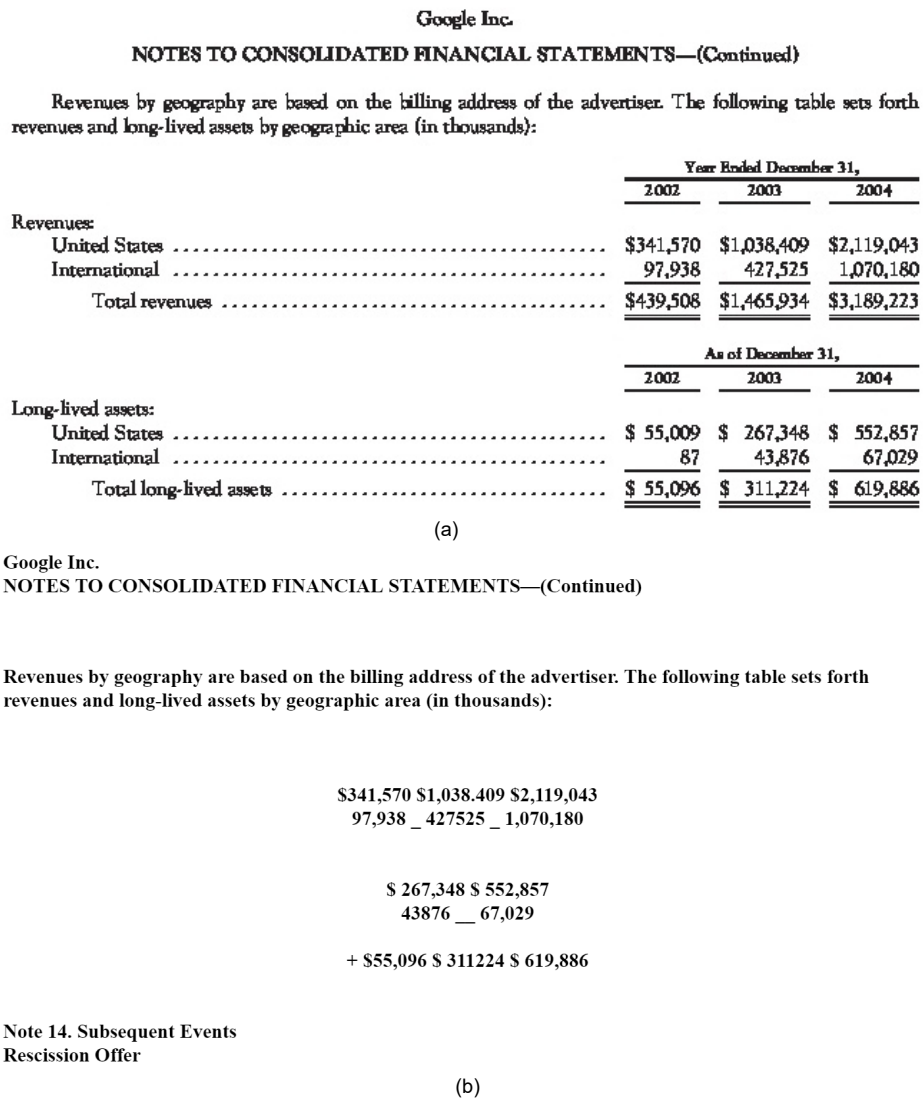
Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

We live amid the digital age, surrounded by the digital universe, which is expanding with the daily data we produce. In this digital age, we are creating, consuming, and replicating data like never before [1]. The estimates tell us that from 2005 to 2020, our digital universe has grown 300 times, from 130 exabytes to 40,000 exabytes of data. It will continue to double every year. It is also estimated that as much as 33% of the digital universe contains information that could be potentially valuable if analyzed. Further, almost 40% of the data produced will be touched by the cloud in some way or another. Manually processing such large amounts of data is laborious, time-consuming, and borderline infeasible. This presents an excellent opportunity for data mining and analysis. To this end, the field of document digitization has received considerable attention and made rapid strides in recent years [2]. Document digitization is the automatic extraction of meaningful information from scanned document images.

Information is stored in two major formats in scanned document images: text and graphical page objects. Graphical page objects can be tables, figures, natural images, equations, formulas, logos, etc. [2–4]. They contain valuable information, but this information can only be extracted when these objects are correctly identified in the respective images [5]. Figure 1 shows the importance of detecting graphical page objects before extracting information.

Graphical page objects have high intra-class variations in shape, type, and scale [2,6]. Tables come in different shapes, sizes, and layouts (bordered or borderless) [3,6–8]. There are also significant amounts of inter-class similarities [6]. Further, natural images can have superimpositions of vectors or diagrams. Figures can be arbitrary or abstractions of natural images. Equations and formulas can be single or multiple lines. A significant inter-mixing of classes can also occur, for instance, a logo inside a table or figure. Figure 2 illustrates the challenges in graphical page object detection.



Google Inc.

NOTES TO CONSOLIDATED FINANCIAL STATEMENTS—(Continued)

Revenues by geography are based on the billing address of the advertiser. The following table sets forth revenues and long-lived assets by geographic area (in thousands):

	Year Ended December 31,		
	2002	2003	2004
Revenues:			
United States	$341,570	$1,038,409	$2,119,043
International	97,938	427,525	1,070,180
Total revenues	$439,508	$1,465,934	$3,189,223

	As of December 31,		
	2002	2003	2004
Long-lived assets:			
United States	$ 55,009	$ 267,348	$ 552,857
International	87	43,876	67,029
Total long-lived assets	$ 55,096	$ 311,224	$ 619,886

(a)

Google Inc.
NOTES TO CONSOLIDATED FINANCIAL STATEMENTS—(Continued)

Revenues by geography are based on the billing address of the advertiser. The following table sets forth revenues and long-lived assets by geographic area (in thousands):

$341,570 $1,038.409 $2,119,043
97,938 _ 427525 _ 1,070,180

$ 267,348 $ 552,857
43876 __ 67,029

+ $55,096 $ 311224 $ 619,886

Note 14. Subsequent Events
Rescission Offer

(b)

Figure 1. This figure shows the importance of detecting graphical page objects: (**a**) shows a document image from the IIIT-AR-13k [9] dataset. In contrast, (**b**) shows the result of Tesseract-OCR [10] on the document image. Note that the naive application of Tesseract-OCR on the document image leads to poor extraction of information.

Figure 2. Challenges of graphical page object detection. Natural images are outlined in blue, logos in yellow, and signatures in orange. Natural images and signatures have text superimpositions in the second and third rows. The natural image in the fourth row has text and vector graphic superimposition.

Deep learning has been one of the most popular and explored solutions to object detection in the natural image domain. In recent years, it has seen rapid improvements. Graphical page object detection is a challenging downstream domain adaptation task of generic object detection in the natural image domain. This fact is not lost on many researchers, and they have come up with many adapted models for graphical object detection [2,11].

Document images are sparse in contextual information as compared to natural images. The graphical page objects of different classes, such as tables, figures, and text, are logically clustered together in document images. The same is not always true with natural images. The different graphical page objects show a wider variety of scale than objects in natural images. Further, most top-performing deep learning models on natural images have strong and deep backbones with texture-driven region proposal networks (RPNs) [4]. By design, these RPNs propose regions in natural images rich in texture, contrast, and color. Selective search in Fast R-CNN [12], and the RPN in Faster R-CNN [4,13] are examples of such.

The assumption of logical clustering may not always hold for natural images. Thus, it makes sense to have dense object candidates for the features maps to avoid missing objects in the image. When coupled with strong backbones, dense object candidates in document

images can confuse the detector head. Hence, naive adaptation of models designed for dense natural images to the document image domain may not yield good results.

Most state-of-the-art graphical page object detection models suffer from the aforementioned systemic challenges and issues. They fail to account for the differences in the domains in an intrinsic manner. Instead, they rely on extrinsic methods such as heavy pre/post-processing of images that are not genuinely end-to-end. In an attempt to address these systemic issues, this paper investigates the feasibility of sparse and learnable object candidates in scanned images of documents and attempts to justify the applicability of the same through extensive experimentation. This paper also investigates the relationship between meaningful gains and the feature extraction capabilities of deep backbones. The scope of the paper has been limited to investigating the effectiveness of deep backbones and learnable sparse proposals for the downstream task of graphical page object detection. Our experiments show that sparse proposals-based methods such as Sparse R-CNN come close to state-of-the-art methods while being computational cheaper.

The main contribution of the paper can be summarized as follows:

- Empirically shows that detectors based on dense object candidate priors are not well suited for graphical page object detection. Detectors that reduce the number of object candidate priors while also making them learnable are a better approach;
- Investigates the effectiveness of deep and robust backbones using the CBNetV2 [14] architecture with different detection heads and backbones, such as Dual Swin-Transformer tiny, small, base, and large;
- Experimentally shows that throwing deep and strong backbones or ensemble versions of these backbones does not lead to better detection results;
- Empirically show that Sparse R-CNN achieves comparable performance to state-of-the-art while being computationally efficient by a large margin;
- Investigates the generalization abilities of the ResNet-101 Sparse R-CNN [15] model by performing exhaustive cross-validation experiments.

The remainder of this paper is structured as follows. Section 2 categorizes the prior work in the field into traditional approaches and deep learning approaches. Section 3 describes the deep backbones investigated, such as CBNetV2 (Section 3.1). Further, this section also describes the detection heads investigated, such as Faster R-CNN (Section 3.2.1), Mask R-CNN (Section 3.2.2), Cascade Mask R-CNN (Section 3.2.3), and Sparse R-CNN (Section 3.2.4). Section 4 describes the datasets employed in our experiments, the evaluation protocol, the implementation details of the models, and qualitative and quantitative analysis of our experiments, and it concludes with the cross-dataset evaluation for the models. Section 5 concludes the paper and provides a brief look into future work.

2. Related Work

The problem of graphical page object detection is quite challenging. The approaches to solving the problem can be broadly divided into two categories: traditional and deep-learning approaches. The traditional approaches can be further divided into rule-based and learning-based methods [2,11]. The rule-based methods rely on manually defined rules and heuristics for the detection of graphical page objects in document images, whereas learning-based methods rely on statistical learning from the data and standard machine learning algorithms. With the advent of deep learning, many deep learning-based approaches to graphical page object detection have been developed. The field has seen rapid progress in recent years.

2.1. Traditional Approaches

Some of the earliest approaches exploited the metadata from the output of optical character recognition (OCR) systems. Green et al. [16] used the metadata to construct custom grammars and pre-defined heuristics to detect graphical page objects. Tupaj et al. also exploited the metadata from OCR to preserve the geometry of white spaces in document images. The geometrical information was then used alongside heuristics to detect

tabular structures [17]. Kieninger et al. used the metadata to extract text-level geometry, which was then used to locate and extract tables from document images [18]. Others, such as Hu et al. [19], used image processing techniques to calculate the correlation between white spaces in a document and performed a connected component analysis to predict the presence of tables and other graphical page objects in documents.

Some early work did not use document images. Costa e Silva et al. [20] employed statistical models such as Hidden Markov Models (HMMs) for table detection in documents. The method works by first parsing text from document files and then computing feature vectors by analyzing the white gaps between different objects in the document. Similarly, Shigarov et al. [21] proposed a method that does not utilize document images. It first extracts the metadata from documents. Next, it treats each word as an individual text block with bounding boxes. These bounding boxes are used to extract the boundaries for tables. Chandran and Kasturi [22] proposed a method for table structure recognition. First, both horizontal and vertical lines and white streams are extracted from the documents. Next, a structure interpretation step is followed to build a structure for the table form the extracted features. The traditional methods can detect tables in documents that are simple and do not have complex patterns or structure. Traditional methods rely on manual handcrafted features and rules to detect graphical page objects. Hence, these methods are highly susceptible to noise in the data.

2.2. Deep Learning-Based Approaches

In deep learning, graphical object detection is seen as a downstream task of generic object detection in natural images. One of the first works to adapt deep learning models from the natural image domain to the downstream task of graphical object detection was by Gilani et al. [23]. They applied Faster R-CNN to detect tables in document images. The document images are first transformed per pixel using a distance transformation mechanism. These transformed images are then processed by the Faster R-CNN model to aid in table structure recognition. Schreiber et al. [24] also apply Faster R-CNN [13] for detecting tables in scanned document images. They also proposed a novel deep learning approach for table structure recognition. In their work, they employ pre-trained backbones and showed that object detectors from the natural image domain could be successfully adapted for the document image domain on the ICDAR-13 dataset [25]. Following this in 2018, Vo et al. [26] proposed to combine Fast R-CNN [12] and Faster R-CNN using an ensemble paradigm to capitalize on the benefits of both object detectors. They achieved this by taking the region proposal from both Fast R-CNN and Faster R-CNN and applying the bounding box regression on their combination. They applied their ensemble model to the graphical page object detection problem and achieved promising results. By now, domain adaptation from the natural image domain to the document image domain had become quite popular. Siddiqui et al. [27] proposed DeCNT, which employed deformable convolutions to detect tables in document images. The authors empirically showed that the dynamic receptive field of the network due to the deformable convolutions can adapt better to the dynamic layout of tables in documents. Sun et al. [28] also applied Faster R-CNN for table detection, where the tabular area is retrieved by refining the coarse table detection and corner location produced by Faster R-CNN. This refining is achieved by grouping the corner belonging to the same table by coordinate matching and filtering out unreliable corners. In 2019, Saha et al. [29] proposed the GOD framework in which they investigated the performance of Faster R-CNN and Mask R-CNN [30]. They compared the two detectors for graphical page object detection in the document images. After exhaustively evaluating the two detectors, they concluded that Mask R-CNN performs better for graphical page object detection than Faster R-CNN.

Detection is the basis for more complex tasks such as table structure recognition, in which tables are first detected and then the structure is also extracted. To this end, Shigarov et al. [31] proposed TabbyPDF, which is a web-based system for detecting and extracting structures of tables located in PDF documents. The system uses a heuristic-based

approach for table detection and structure recognition. Bordered tables are detected using rule lines that compose a rectangular frame, whereas border-less tables are detected using a bottom-up segmentation approach. The authors evaluate their approach on the ICDAR-13 dataset. Chi et al. [32] proposed GraphTSR, a graph neural network for recognizing the table structure in PDF files. First, through pre-processing, the cell contents and their corresponding bounding boxes are extracted from the PDF, then an undirected graph is produced using the cells, the edges of which are filtered using adjacent relation prediction through the attention mechanism. Finally, the predicted table structure is extracted from the labeled graph. TabStruct-Net, proposed by Raja et al. [33], approaches table structure detection as a relationship problem between the row and column of the detected cell. First, a ResNet backbone is used to detect cells from the table. Next, an alignment loss function is introduced to ensure that detected cells belong to the same row or column, followed by the post-processing step of producing an XML output of the table structure. One limitation of this method is its inability to handle tables with a large number of empty cells. Zang et al. [34] proposed SEM (split, embed, and merge), which exploits the multimodality of tables, as they contain both visual and text features. First, a fine grid structure of the potential rows and columns of the table is obtained. Next, using a visual module and text module, the two modalities are fused; the fused features are used to predict the table structure. Prasad et al. [35] proposed CascadeTab-Net, which tries to solve the table detection and table structure recognition together. Unlike most other detection networks that treat the problem of table detection and structure representation separately, CascadeTab-Net uses a single-shot method for both. They utilized the Cascade Mask R-CNN [36] detection head. More recently, CDeC-Net [37] utilized the dual-backbone paradigm introduced by CBNetv1 [38]. It has high-resolution dual ResNeXt101 backbones with deformable convolutions. The backbone is paired with a Cascade Mask R-CNN detector head. It allows the network to detect objects at different resolutions and achieve high accuracy, even at higher IoU thresholds.

As our work focuses on graphical page object detection, to this end, Table 1 shows a comparison of the different approaches we investigate in our work with prior work in the domain. The table lists the key characteristics of each approach along with its limitations and advantages that are drawn from the experiments we carry out for each approach and its corresponding methods.

Table 1. Comparison of the key characteristics of the explored approaches with prior work in the domain, along with their advantages, limitations, and results.

Approach	Methods	Advantages	Limitations	Results
Dual Backbone Architectures	Faster R-CNN, Swin-T Mask R-CNN, and Swin-S Cascade Mask R-CNN	Ability to capture different features with the use of two parallel backbones	Features extracted by the parallel backbones are not informatively distinct	Parallel backbones do not lead to better performance in document images
Transformer based Architecture	Sparse R-CNN paired with Swin Tiny, Swin Small, Swin Base (1k), Swin Base (22k), and Swin Large	Ability to extract fine features and patterns with the help of attention that scales linearly w.r.t. image size	Extraction of fine features and patterns by the strong backbones end up confusing the detection heads. Slow, memory intensive and high GFLOPs	Transformer based backbones do not lead to better performance in document images
YOLOF [39]	SSD with a dilated encoder and uniform matching	Fast and low GFLOPs	Large-sized objects raise problems as YOLOF has limited range scale, results in poor performance	Performance is worse than standard two-stage object detection methods such as Faster R-CNN and Mask R-CNN

Table 1. *Cont.*

Approach	Methods	Advantages	Limitations	Results
CDeCNet [37]	ResNext backbone with Cascade Mask R-CNN detection head and deformable convolutions	High-resolution backbone, cascading detection head that is robust to false positives, and deformable convolutions that address the issue of the limited range scale	Slow, memory, intensive and high GFLOPs	State-of-the-art results at high computational costs
SR-CNN r101	Sparse R-CNN with a ResNet-101 backbone	Has a balanced, strong backbone in comparison to ResNext, DB, or Transformer based detection backbones. Learnable proposals	As the proposal boxes are learnable it needs sufficient data for each class to learn the learnable proposal feature	Close to state-of-the-art with 6× less GFLOPs than CDeCNet

2.3. Related Datasets

Table detection and structure recognition is a well studied area, and many standard benchmark datasets are available: UNLV [40], Marmot [41], ICDAR-13 [25], ICDAR-POD-17 [5], ICDAR-19 [42], and TableBank [43]. UNLV is one of the earliest datasets in the document object detection domain. The dataset consists of almost 10,000 document images. However, only 427 of them contain tables, and hence we only list the images containing tables in Table 2. Marmot, on the other hand, is one the largest early datasets for table detection in document images. It was introduced by Peking University and consists of 2000 images for which a ratio of almost 1:1 is maintained between the positive and negative samples. The original dataset suffered from incorrect ground truth annotations, hence we list the corrected version of the dataset from [24], which has 1967 images, in Table 2. ICDAR-2013 [25] is another popular dataset for both table detection and structure recognition. The dataset was constructed by converting PDF files to images. The dataset has 229 training images and 233 validation images. ICDAR-2017-POD (Page Object Detection) [5] is another dataset that, along with tables, also has information for the boundaries of formulas and figures. The dataset consists of 2417 images in total, where 1600 are for training and 817 are for validation. ICDAR-19 [42] is the latest dataset to come out of the ICDAR in 2019. The dataset consists of two types of document images: modern and historical. The modern set of documents is collected from scientific and commercial documents, whereas the historical documents are a collection of images of handwritten documents. TableBank, introduced by Li et al. [43], is one of the most well known datasets in the table detection literature. The dataset has 417,000 training document images pooled from word and latex documents. It has 2000 images for both validation and test.

Table 2. Statistics of related datasets. T: indicates table, F: indicates figure, L: indicates list, S: indicates signature, NI: indicates natural images, Ti: indicates titles, Tx: indicates text, Ab: indicates abstract, Au: indicates author, C: indicates caption, D: indicates date, E: indicates equation, Fo: indicates footer, P: indicates paragraph, R: indicates reference, S: indicates sections.

Datasets	Labels	Training	Validation	Test
UNLV	T	427	-	-
Marmot	T	1967	-	-
ICDAR-13	T	229	233	-
ICDAR-POD-17	T	1600	817	-
ICDAR-19	T	1200	439	-
TableBank (word + latex)	T	417 K	2 K	2 K
IIIT-AR-13k	T, F, L, S, and NI	9 K	2 K	2 K
PubLayNet	T, F, Ti, Tx, and L	340 K	11 K	11 K
DocBank	T, Ab, Au, C, D, E, F, Fo, L, P, R, S, and Ti	400 K	50 K	50 K

More recently, datasets for graphical page object detection have also emerged that, alongside tables, also have other page objects, such as figures, text, signatures, dates, equations, etc. Such datasets include IIIT-AR-13K [9], PubLayNet [44], and DocBank [45]. Table 2 shows the statistics of the various datasets. As our goal is to investigate graphical page object detection in document images, we restrict ourselves to IIIT-AR-13K, PubLayNet, and DocBank datasets, as they facilitate the general page object detection task (including tables).

3. Method

In recent years, many novel backbone networks have been developed that have consistently improved the performance of object detection [14]. First, the effectiveness of deep and robust backbones and their ensemble versions in the document image domain is investigated. To this end, Section 3.1 discusses the CBNetv2 compositing architecture this work investigates. Different detector heads are employed upon these deep backbones, which use novel methods for detection and segmentation. Section 3.2 discusses the different detection heads this study investigates.

3.1. CBNetv2: Composite Backbone Network v2

A key driver of advancements in object detection is the improvement in the backbones of the detectors that extract the features [14]. These improvements are often structural in nature. They also require high computing resources and expertise to achieve and validate. Another approach would be to use pre-existing, well-designed, validated deep backbones and design efficient ensemble techniques. CBNetv2 [14] is a step in this direction. The key idea of CBNetv2 is that it finds novel composition styles to ensemble identical existing pre-trained backbones [14], which allows for greater flexibility and generalization capabilities as different detector designs can be explored to suit the task.

As seen in Figure 3, in CBNetv2 there are K identical backbones of which $K-1$ are assisting backbones. Each backbone has its own feature pyramid network (FPN) consisting of L stages, and the output of each stage is denoted as x^l. Let a composition connection be denoted as $h^l(x)$, which takes $x = \{x^i | i \in [1, L]\}$ as inputs that come from each stage of the FPN of the assisting backbone and outputs a feature map y^l. Further, let g represent a 1×1 convolution layer and a batch-normalization layer. The proposed composition connections are discussed in Section 3.1.1.

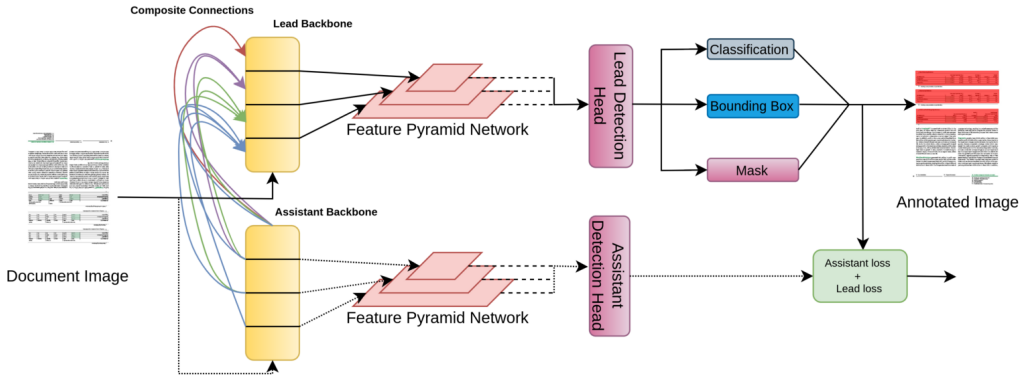


Figure 3. The detection pipeline and architecture of CBNetV2 consists of the lead network and the assistant network. The lead and assistant networks are used during training, where the assistant backbone and detection head provide additional supervision. During inference, only the lead network is used.

3.1.1. Composite Connections

There are many ways to achieve feature fusion across backbones. The most simple is fusing features of the same level across the backbones. Equation (1) formulates this as the Same Level Composition. In a deep backbone, the higher-level features amass greater semantic meaning. Adjacent Higher-Level Composition, as seen in Equation (2), fuses

the higher-level feature of the adjacent backbone with that of the lower-level feature of the current backbone. The converse of Adjacent Higher-Level Composition is Adjacent Lower-Level Composition, as formulated in Equation (3). Dense features can be built by combining the features from all the higher levels of the assisting backbones with that of the lower ones of the lead backbone. Termed as Dense Higher-Level Composition, it is formulated in Equation (4). Equation (5) describes the Fully Connected Composition, which tries to build comprehensive features by connecting features from all levels in the assisting backbones to that of the lead backbone. The authors in [14] show that Dense-Higher-Level Composition and Fully Connected Composition achieve the best results when tested on the COCO dataset [46]. Taking computational complexity into account, the authors in [14] use Dense Higher-Level Composition. Our implementation of the model also utilize the same.

$$h^l(x) = g(x^l) \quad where \quad l \geq 2 \tag{1}$$

$$h^l(x) = U(g(x^{l+1})) \quad where \quad l \geq 1 \tag{2}$$

$$h^l(x) = D(g(x^{l-1})) \quad where \quad l \geq 1 \tag{3}$$

$$h^l(x) = \sum_{i=l+1}^{L} U(g_i(x^i)) \quad where \quad l \geq 1 \tag{4}$$

$$h^l(x) = \sum_{i=2}^{L} I(g_i(x^i)) \quad where \quad l \geq 1 \tag{5}$$

3.1.2. Assistant Supervision

The authors in [14] recognize the positive correlation between depth and increased performance. However, increased depth leads to a regularization problem, requiring novel methods to improve convergence [47]. To tackle regularization and also increase depth, the authors propose assistant supervision. All the backbones and their detection heads are used during training, and the total loss minimized is given in Equation (6). In contrast, only the lead backbone is used during the inference, as given in Equation (7).

$$L_{Train} = L_{Lead} + \sum_{i=1}^{K-1} \alpha_i \dot{L}^i_{Assistant} \tag{6}$$

$$L_{Test} = L_{Lead} \tag{7}$$

3.2. Detection Heads

Object detection aims to determine the locations of objects in a given image and their classes. This task can be broken down into two main steps: Target Region Selection and Classification and Regression. The methods applied to address these main steps lead to different detection head designs. Section 3.2 and its subsections discuss the different detection heads, such as Faster R-CNN (Section 3.2.1), Mask R-CNN (Section 3.2.2), Cascade Mask R-CNN (Section 3.2.3), and Sparse R-CNN (Section 3.2.4).

3.2.1. Faster R-CNN

Faster R-CNN [13] is a two-stage detector. It has two networks: a regional proposal network (RPN) and a detection network. The RPN proposes regions of interest (RoI) hence performing the target region. The detection network uses the ROIs proposed by the RPN for object bounding box regression and classification. Faster R-CNN builds upon Fast R-CNN [12]. In Fast R-CNN, there is a decoupling of the selective search and the detection network. The false negatives of the selective search directly affect the network's detection accuracy. Hence, this decoupling is not a great idea. It would be better if there were a correlation between the two.

Therefore, the core idea is that Faster R-CNN does not utilize selective search for RoI proposals. As seen in Figure 4, it utilizes a small RPN instead. It hence drastically reduces the time cost for generating region proposals. The RPN outputs a set of rectangular object proposals, each with its objectiveness score. The regressor and the classifier then take these object proposals and objectiveness scores from the RPN. Finally, the classifier outputs the probability of predicted ROI as an object (foreground) or background. In contrast, the regressor outputs the predicted bounding boxes.

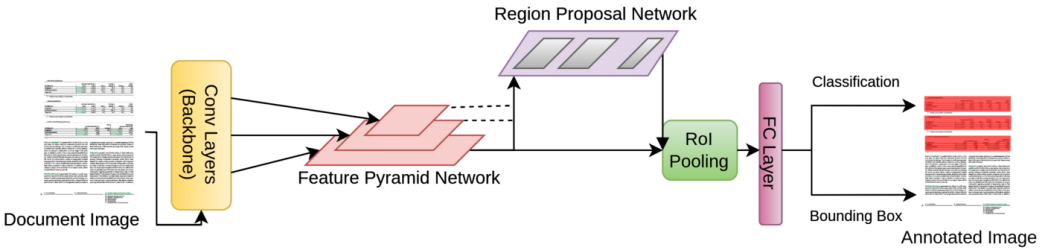


Figure 4. The detection pipeline for the Faster R-CNN detection head. Faster R-CNN builds upon Fast R-CNN by replacing the slow selective search with a small RPN, which reduces the time and cost of generating object proposals.

3.2.2. Mask R-CNN

Semantic segmentation is the process of classifying each pixel of an image as belonging to a particular label. This labelling does not differentiate among instances. Hence, it can be considered a classification problem on a per-pixel basis. Mask R-CNN [30] is a detection head for semantic and instance segmentation. Mask R-CNN builds upon the groundwork laid by Faster R-CNN.

As seen in Figure 5, Mask R-CNN extends Faster R-CNN with the addition of two more convolution layers after the RoI align layer. The additional convolutional layers produce the segmentation masks. The RoI align layer is a novel addition in Mask R-CNN. The layer does not digitize the cell boundaries of the target feature maps like the RoI pooling layer in Faster R-CNN. Instead, it utilizes interpolation to calculate the cell boundaries.

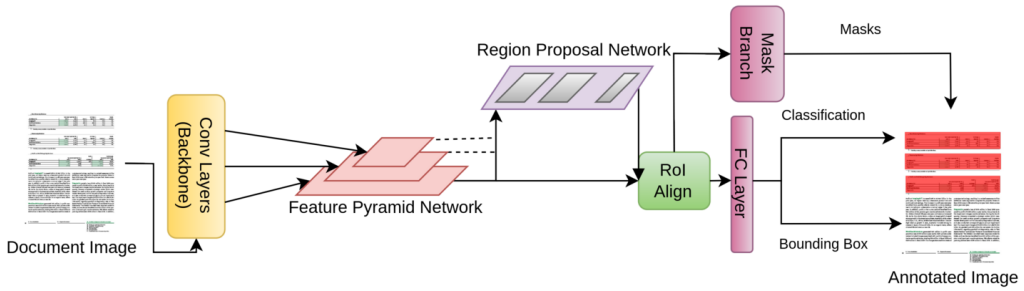


Figure 5. The detection pipeline for the Mask R-CNN detection head. Mask R-CNN extends Faster R-CNN to the image segmentation domain by adding the mask branch, which produces masks for semantic segmentation.

3.2.3. Cascade Mask R-CNN

Most deep learning detectors, such as Fast R-CNN, Faster R-CNN, and Mask R-CNN as previously discussed, tend to show a degradation in performance with increasing intersection over union (IoU) thresholds. This occurs for two main reasons: the overfitting during training due to exponentially vanishing positive samples and the mismatch during inference between the IoUs set for training and the ones given as input at inference time.

Cascade R-CNN [36] attempts to address these problems. Figure 6 shows that the core idea is refinement through cascading. A series of heads take the RoI features from the region proposal network (RPN). Each head has its classification and regression networks. The first head (Head 1) takes the RoI features and performs the first round of classification

and regression. The output of the first head is treated as an input for the subsequent heads, producing a cascade effect.

This cascading architecture is essentially a resampling procedure. The architecture provides good positive samples to the next head, as later heads are more prone to false positives. This enables each subsequent head to operate at a higher IoU threshold. Further, this also tackles the mismatch between training and inference as the architecture and IoU thresholds are the same during training and inference. Finally, the last head's classification and bounding box predictions are used for optimal predictions. The addition of a segmentation mask branch extends it to Cascade Mask R-CNN.

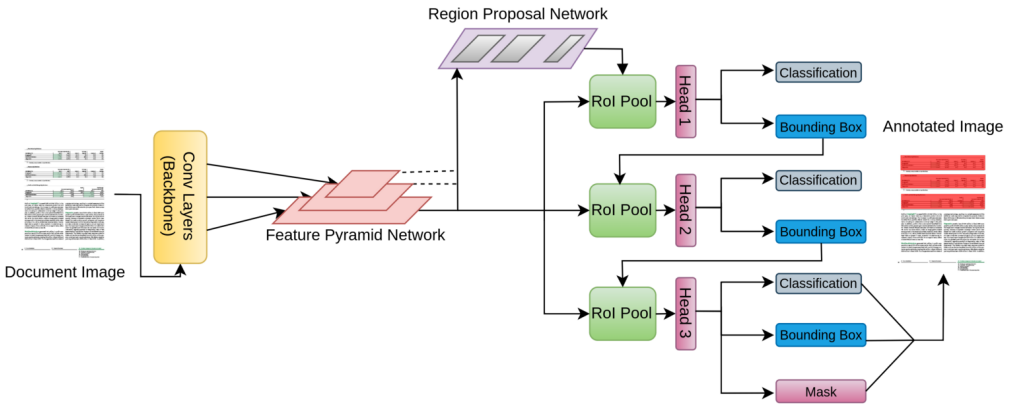


Figure 6. The detection pipeline for the Cascade Mask R-CNN detection head. Cascade Mask R-CNN has multiple heads and utilizes cascading among these heads to refine the bounding box predictions.

3.2.4. Sparse R-CNN

The famous deep learning object detectors such as Fast R-CNN, Faster R-CNN, Mask R-CNN, and Cascade Mask R-CNN all rely on dense object priors. These detectors have many anchor points containing anchor boxes that densely cover different spatial positions, aspect ratios, and scales. The dense coverage helps to predict the classes and bounding box coordinates. In the case of single-shot detectors, let W be the width and H be the height of a feature map. If each anchor point is responsible for predicting k bounding boxes, then we have a large number $W \times H \times k$ of bounding boxes for each feature map.

Further, in the case of two-stage detectors such as Faster R-CNN, Mask R-CNN, and Cascade Mask R-CNN, dense object priors are generated first and then, using the region proposal network (RPN), a sparse set of foreground proposal boxes are obtained from these dense object priors. Finally, these are fed into the classification and regression branches to produce bounding box and category predictions.

The authors of Sparse R-CNN [15] propose a different paradigm that moves towards thoroughly sparse object priors. As seen in Figure 7, the authors avoid using an RPN. They instead use a small set of proposal boxes, usually 100 or 300. These proposal boxes are learnable. The authors introduce the dynamic head to obtain them.

First, the initial set of randomly initialized proposal boxes in the dynamic head is used with the RoI align operation in extracting features from the network backbone. Next, each extracted RoI feature is convolved with a learnable proposal feature. The result is fed into its detection and classification head to predict the class and bounding box. Each head is essentially conditioned on these learnable proposal features. The proposal features essentially act as weights for the convolutions and give rise to the learnable proposal boxes.

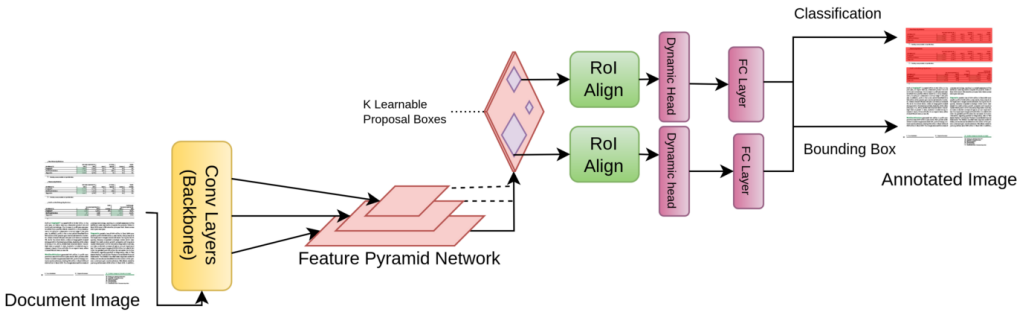


Figure 7. The detection pipeline for the Sparse R-CNN detection head. Sparse R-CNN proposes a paradigm shift by utilizing sparse and learnable proposal boxes and features instead of region proposal networks (RPNs).

4. Experimental Results

4.1. Datasets

4.1.1. IIIT-AR-13K

IIIT-AR-13K [9] is a new public and open source dataset for graphical page object detection. The dataset has a total of 13k images with manually annotated bounding boxes of graphical objects in five popular and different categories: table, figure, signature, natural images, and logos [9]. The dataset consists of annual reports of 29 different corporations and companies. These reports span a large time interval of over ten years. These reports are also in many languages such as English, German, French, Russian, and Japanese.

The authors argue that such diversity in document images helps train highly effective and practical object detectors in business documents and academic articles [9]. They demonstrate that it is possible to achieve good performance by training on a smaller dataset that is more diverse than one that has a large number of examples [9]. As seen in Table 3, one key drawback of the dataset is that it is heavily skewed towards tables. The dataset contains a few examples of the other classes, such as logos, signatures, and natural images.

Table 3. Statistics of training, validation, and testing sets in the IIIT-AR-13k [9] dataset.

Sets	Table	Figure	Signature	Natural Images	Logo
Training	11,163	2004	420	1987	379
Validation	2596	463	92	455	135
Test	2222	481	108	438	67

4.1.2. PubLayNet

The PubLayNet [44] dataset is the largest public and open source dataset for document layout analysis. The automatic matching of the XML representations of PDF articles with their content helps create such a large dataset. These articles are from the medical domain. The dataset has over 360,000 document scanned images [44]. The dataset annotates the most popular and typical graphical objects, such as text, titles, lists, tables, and figures. The dataset goes a long way in the training of real-world object detectors. The PubLayNet dataset is frequently used as a corpus to train detectors to detect document layouts, especially in scientific articles. Table 4 shows the statistics of the dataset's training, validation, and test split.

Table 4. Statistics of training, validation, and testing sets in PubLayNet [44] dataset.

Sets	Text	Title	Lists	Tables	Figures
Training	2,376,702	633,359	81,850	103,057	116,692
Validation	93,528	19,908	4561	4905	4913
Test	95,780	20,340	5156	5166	5333

4.1.3. DocBank

DocBank [45] is a unique dataset that utilizes weak supervision in its construction. The dataset is large, public, and open source, consisting of 500,000 document pages and having 13 classes. Table 5 shows the statistics of the most relevant classes to the graphical page object detection task in the training, validation, and test splits of the DocBank dataset. DocBank has both textual and layout information to facilitate document layout analysis and textual analysis using natural language processing (NLP) [45]. Most other datasets contain scanned images of documents and do not have fine-grained annotations. DocBank has token-level annotations and ample semantic annotations for figures, tables, and equations to facilitate common computer vision and natural language processing tasks (NLP).

Table 5. Statistics of the most relevant classes to the task of graphical page object detection in the training, validation, and testing sets in DocBank [45] dataset.

Set	Abstract	Equation	Figure	List	Paragraph	Section	Table
Training	25,387	161,140	90,429	44,927	398,086	180,774	19,638
Validation	3164	20,154	11,463	5609	49,759	22,666	2374
Test	3176	20,244	11,378	5553	49,762	22,384	2505

4.2. Evaluation Protocol

Analogous to standard work in the field of graphical page object detection, the performance of the Sparse R-CNN model is assessed on the metrics of recall, precision, F1-score, and mean average precision (mAP) [46]. The IoU thresholds of the achieved results are also reported to facilitate comparison with other approaches.

4.2.1. Intersection over Union

Intersection over Union (IoU) [48] is a standard method for computing the intersecting region between the predicted and ground truth regions. Equation (8) expresses Intersection over Union (IoU) mathematically.

$$\text{IoU}(X, Y) = \frac{X \cap Y}{X \cup Y} \tag{8}$$

4.2.2. Recall

Let TP represent the true positives, and FN represent the false negatives. Recall [48] is defined as the ratio of true positives over the sum of true positives and false negatives. Equation (9) represents Recall mathematically.

$$\text{Recall} = \frac{TP}{TP + FN} \tag{9}$$

4.2.3. Precision

Let TP represent the true positives, and FP represent the false positives. Precision [48] is defined as the ratio of true positives over the sum of true positives and false positives. Equation (10) represents Precision mathematically.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{10}$$

4.2.4. F1 Score

The F1-Score [48] is essentially the harmonic mean of the Recall and Precision. Equation (11) represents F1-Score mathematically.

$$\text{F1-Score} = \frac{2 \times \text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}} \tag{11}$$

4.3. Implementation Details

The Sparse R-CNN, baseline, and CBNetv2 models are all implemented by utilizing the MMDetection framework [49]. All the models utilize COCO pre-trained weights for the different detector backbones. All the models follow the pre-training fine-tuning paradigm with the first stage of the detector backbone being frozen. All input document images are resized to a maximum size of 1200 × 800 while preserving the actual aspect ratio of these images. The SR-CNN model utilizes multi-scaling to scale the width of images between 480 to 800 pixels. The baseline models are trained for 20 epochs, and the Sparse R-CNN-based models are trained for 36 epochs. We employ early stopping based on the validation accuracy for model selection for all models. We use the Adam optimizer for all Sparse R-CNN models and SGD for all baseline and CBNetv2 based models with a linear learning rate schedule with 500 warmup iterations, with a step weight decay of 0.0001 at 8 and 11 epochs, respectively.

4.4. Result and Discussion

This paper investigates the effectiveness of deep and robust backbone architectures with different detection heads in the document image domain. Further, the idea of learnable proposals, especially in the sparse paradigm, is also investigated. All this is set under the pre-training fine-tuning paradigm. Towards this end, in Section 4.4 and its subsequent sub-sections, we discuss the experimental results to support the validity of our hypothesis. This validation is achieved by experiments on individual datasets as noted in Section 4.1 and cross-dataset evaluation on the same datasets. The subsections of Section 4.4 are structured as follows: In Section 4.4.1, the IIIT-AR-13K dataset is utilized to investigate the effectiveness of deep backbones and the idea of learnable proposals in the sparse paradigm. In Section 4.4.2, the findings from the Section 4.4.1 are applied to the standard PubLayNet dataset, and the results are discussed. Section 4.4.3 investigates the cross-dataset generalizability of the models.

4.4.1. IIIT-AR-13K

Section 3.1 discussed the compositing architecture for deep learning backbones, and Section 3.2 discussed the different types of detection heads this study aims to investigate. This section discusses the experimental results of the IIIT-AR-13K dataset.

Establishing Baselines

The first thing needed for any comparison is the establishment of strong baselines. To this end, under the pre-training fine-tuning paradigm baseline, Faster R-CNN, Mask R-CNN, and Cascade Mask R-CNN models are trained on the IIT-AR-13K dataset. The training is per the implementation details noted in Section 4.3.

Table 6 shows the mean average precision (mAP) at the Intersection over Union (IoU) threshold from 0.5 to 0.95 in accordance with the COCO evaluation protocol [46]. The table also shows the average precision (AP) at Intersection over Union (IoU) thresholds of 0.75 and 0.5. It is evident that Cascade Mask R-CNN establishes the strongest baseline for the IIIT-AR-13K dataset, reaching an mAP of 0.76.

This strong performance is due to the cascading nature of the detection head. Cascade Mask R-CNN allows each subsequent head to be enhanced by the localizing distribution of the previous head, hence giving better performance on the higher IoU threshold values. Mask R-CNN and Faster R-CNN perform equivalently. The reasoning is that in document images, the masks for graphical objects are rectangular and hence do not add much additional information compared to bounding boxes. Figure 8 shows the detection results as true-positives and false-positives for the baseline models in Table 6.

Table 6. The established baseline results on the IIIT-AR-13k [9] dataset.

Method	bbox mAP (IoU = 0.50:0.95)	bbox AP (IoU = 0.75)	bbox AP (IoU = 0.50)	FLOPS (GFLOPs)
Baseline Faster R-CNN	0.747	0.864	0.930	206.68
Baseline Mask R-CNN	0.745	0.845	0.929	258.23
Baseline Cascade Mask R-CNN	0.769	0.879	0.928	389.16

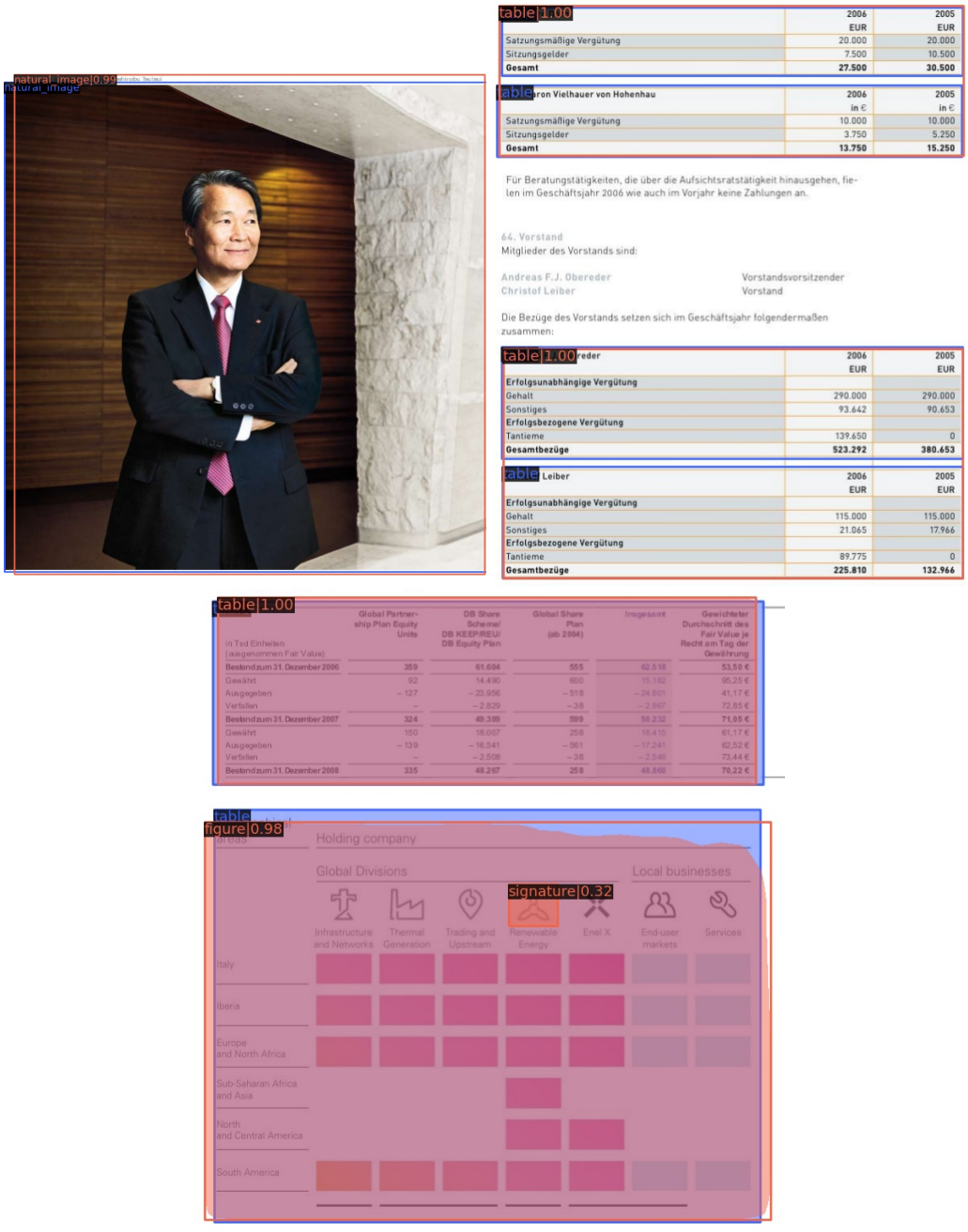


Figure 8. Visualizing some detection results for the models in Table 6. First row: The first and second images show true positive and false positive detection results for the baseline Faster R-CNN, respectively. Second row: The top and bottom images show the true positive and false positive detection results for the baseline Cascade Mask R-CNN [36].

Analyzing Effects of Strong Backbones

Using strong backbones for object detection in natural images generally increases detection accuracy. Deep backbones can extract fine features and patterns. These fine features and patterns then help construct a high-level semantic representation of the data by the network. These semantic representations play a crucial role in object detection in natural images.

Nevertheless, the extraction of high-level semantic features is highly dependent on the depth of the network. Increasing the depth of the network and tackling the accuracy degradation problem is not as straightforward. To achieve this, many novel general-purpose deep backbones such as ResNet [47] and even backbones from natural language processing (NLP) such as Swin-Transformers have been proposed. More recently, ensemble methods that combine two or more general-purpose backbones such as CBNet [38] and CBNetv2 [14] have been proposed.

The effects of strong backbones in the document image domain are investigated by applying Dual-backbone Swin-Transformer, and Dual-backbone ResNet50 backbones with different detection heads such as Faster R-CNN, Mask R-CNN, and Cascade Mask R-CNN. Table 7 shows the different combinations of backbones and detection heads explored. The core idea behind CBNetv2 was that using two identical backbones could lead to improved performance, as the two backbones will learn different features and aid in the detection of objects. Tables 6 and 7 show that the Dual-backbone version of Faster R-CNN performs marginally better than the baseline version. This can be caused by document images being simpler in context than natural images, and the two parallel and identical backbones do not learn different features, hence do not aid each other significantly in detection.

As seen in Table 7, the Dual-backbone architecture of CBNetv2 with different Swin-Transformers as backbones is explored. Apparently, with both Mask R-CNN and Cascade Mask R-CNN as detection heads, the results are considerably worse than the baselines. This can be caused by the fact that the extraction of fine features and patterns by the strong backbones ends up confusing the detection heads of the network. This means that, in the document domain, where the images are much less contextually rich than natural images, the naive use of deep and strong backbones degrades performance. Figure 9 shows the detection results for the models in Table 7.

Learnable Proposals (Sparse R-CNN)

The subsection Analyzing Effects of Strong Backbones in Section 4.4.1 empirically establishes that the naive use of deep and strong backbones in the document image domain does not lead to better detection results. Compared to natural images, document images are essentially sparse in contextual information. Most detection heads, such as Faster R-CNN, Mask R-CNN and Cascade Mask R-CNN, were designed for natural images.

These detection heads' region proposal network (RPN) depends heavily on texture, color, and contrast information to predict the Region of interest (RoIs). Features such as texture, contrast, and color are abundant in natural images. However, document images lack these features; hence, these RPN can struggle when applied to document images.

Table 7. The table shows the results for different detection heads on the Dual-backbone architecture with backbones of different depths on the IIIT-AR-13k [9] dataset.

Method	bbox mAP (IoU = 0.50:0.95)	bbox AP (IoU = 0.75)	bbox AP (IoU = 0.50)	GFLOPs
DB-Faster R-CNN	0.757	0.863	0.924	283.33
DB-Swin-T Mask R-CNN	0.704	0.805	0.909	355.47
DB-Swin-S Cascade Mask R-CNN	0.720	0.805	0.896	1010.54

Figure 9. Visualizing some detection results for the models in Table 7. First row: The two images show the detection results for Dual-Backbone Swin Transformer Tiny Mask R-CNN. Second row: The first image shows the detection results for Dual-Backbone Faster R-CNN, and the second shows the detection results for Dual-Backbone Swin-Transformer Small Cascade Mask R-CNN.

Further, due to the sparse nature of the document images, the application of detection heads that rely on dense object priors can lead to degraded performance. A better approach would be if the object priors were sparse and learnable. Sparse R-CNN achieves this. Table 8 shows the different ResNet backbones used along with the Sparse R-CNN detection head. The table also shows the Sparse R-CNN (SR-CNN) model. Even with a relatively shallower backbone of ResNet50, the Sparse R-CNN model outperforms the strongest baseline in Table 6.

With a deeper backbone such as ResNet101, the best results on the IIIT-AR-13K dataset are achieved. This may be because Sparse R-CNN employs learnable proposal boxes using the learnable proposal features in the dynamic head of the Sparse R-CNN detection head. At the same time, the deep backbone extracts high-level features relevant for detection.

As seen in Table 8, the authors in [39] evaluate the new one-stage YOLOF method on the IIIT-AR-13K. Their experimental YOLOF model achieves 0.588 mAP on the IIIT-AR-13K dataset. In comparison, the SR-CNN model achieves 35% higher raw mAP and considerably outperforms the model in the pre-training and fine-tuning paradigm.

Further, Table 9 shows the effects of deep and robust backbones on the SR-CNN model. To this end, the Sparse R-CNN head is paired with the Swin-Transformer general-purpose backbone. The results saturate with the Swin-Transformer Base backbone, which is pre-trained on the ImageNet-1k dataset. Using more robust backbones trained on the ImageNet-22k dataset, such as Swin-Transformer Base and Swin-Transformer Large, results in degradation of the mean average precision (mAP).

Figure 10 shows the graph of the F1 score vs. IoU thresholds for the SR-CNN model on the validation and test set split of the IIIT-AR-13K dataset. From the figure, it is evident that the the SR-CNN model performs well even on higher IoU thresholds for both the validation and test split of the dataset.

Table 8. The table shows experimentation of Sparse R-CNN detection head with different backbones. We also compare the SR-CNN model with other models on the IIIT-AR-13k dataset.

Method	bbox mAP (IoU = 0.50:0.95)	bbox AP (IoU = 0.75)	bbox AP (IoU = 0.50)	GFLOPs
Nguyen et. al YOLOF [39]	0.588	0.649	0.812	99.98
Sparse R-CNN r50	0.771	0.888	0.944	146.01
SR-CNN r50	0.771	0.868	0.932	149.90
SR-CNN r101	**0.795**	**0.893**	**0.954**	**225.97**

Table 9. These experiments show that even with deep and strong backbones such as the Swin-Transformer, the performance of the SR-CNN model does not improve.

Method	bbox mAP (IoU = 0.50:0.95)	bbox AP (IoU = 0.75)	bbox AP (IoU = 0.50)	GFLOPs
SR-CNN Swin Tiny	0.713	0.797	0.796	153.55
SR-CNN Swin Small	0.716	0.804	0.890	243.54
SR-CNN Swin Base (1k)	0.741	0.835	0.909	386.32
SR-CNN Swin Base (22k)	0.726	0.819	0.895	386.32
SR-CNN Swin Large	0.731	0.831	0.896	793.28

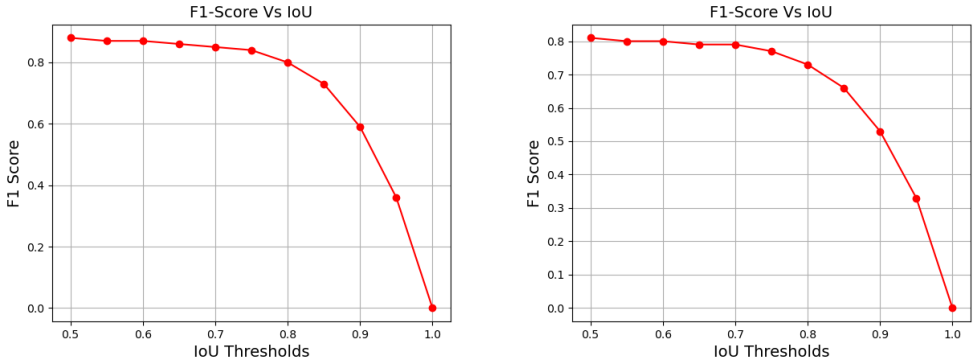


Figure 10. The F1-Score vs. IoU thresholds for the SR-CNN model. The first image shows results for the validation set and the second image for the test set of the IIIT-AR-13K [9] dataset.

4.4.2. PubLayNet

Motivated by the results on the IIT-AR-13K dataset, the SR-CNN model is applied to the PubLayNet [44] dataset. Table 10 shows the results for the same. To the best of our knowledge, the state-of-the-art results on the PubLayNet dataset are achieved by

CDeC-Net [37]. The CDeC-Net model has a Dual-ResNeXt101 backbone with deformable convolutions. The detection head is a Cascade Mask R-CNN. The model achieves a mean average precision (mAP) of 0.96 [37]. The SR-CNN model achieves an mAP of 0.93. The model can achieve these results without using memory-intensive deformable convolutions and a strong high-resolution backbone such as ResNeXt101. The results are close to the current state-of-the-art as achieved by the CDeC-Net model. At the same time, the SR-CNN model is simpler than CDeC-Net and utilizes readily available COCO weights. Figure 11 shows the detection results for the SR-CNN model on the PubLayNet dataset. Figure 12 shows the F1-score vs. IoU thresholds for the SR-CNN model.

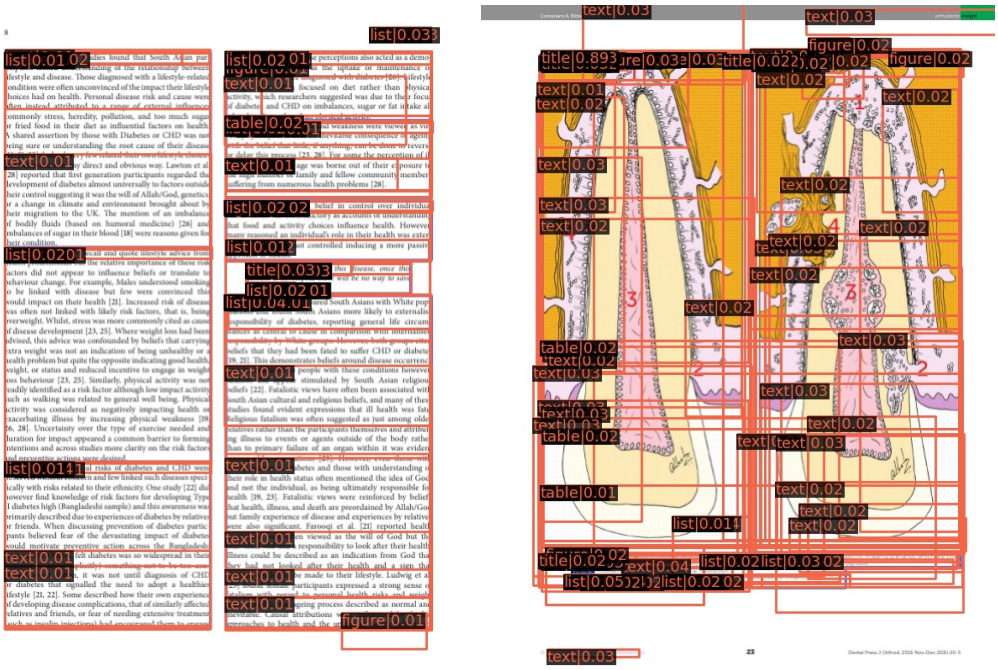


Figure 11. The detection results for the SR-CNN model on the PubLayNet [44] dataset. The first image highlights a true positive example, and the second image highlights a false positive example.

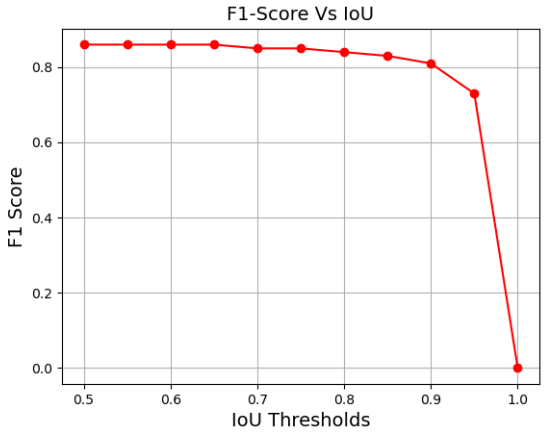


Figure 12. The image shows the F1-Score vs. IoU thresholds for the SR-CNN model on the test set for the PubLayNet [44] dataset.

Table 10. The results of the SR-CNN model on the PubLayNet [44] dataset, along with the comparison with other state-of-the-art methods.

Method	bbox mAP (IoU = 0.50:0.95)	bbox AP (IoU = 0.75)	bbox AP (IoU = 0.50)	GFLOPs
F-RCNN [44]	0.900	-	-	206.68
M-RCNN [44]	0.907	-	-	258.23
SR-CNN r101	**0.937**	**0.962**	**0.981**	**225.97**
CDeC-Net [37]	0.967	-	-	1375.32

4.4.3. Cross-Dataset Evaluation

Deep learning methods for object detection in document images are preferred over rule-based methods due to the fact that these methods generalize better over different and distinct datasets. To investigate its generalization capabilities, the SR-CNN model is evaluated over three datasets: IIIT-AR-13k, PubLayNet, and DocBank. Only the common classes among the datasets were utilized for the cross-dataset evaluation. Baseline Faster R-CNN is also evaluated for comparison. The results are presented in Table 11.

Table 11 shows that the SR-CNN model gives a mean average precision of 0.56 and 0.45 on the PubLayNet and DocBank, respectively, when trained on the IIIT-AR-13K dataset. This may be because IIIT-AR-13K has a diverse collection of real-world images of company reports, which allows the model to learn a more general representation of document images and performs reasonably well on the much simpler dataset with document images of academic articles.

When the SR-CNN model is trained on the PubLayNet, it gives 0.510 and 0.323 mean average precision on the DocBank and IIIT-AR-13K datasets. The model's performance suffers on the IIIT-AR-13K dataset as the model is trained on a dataset that contains primarily academic and research articles. The IIIT-AR-13K offers an entirely different distribution of document images. However, when tested on the DocBank dataset, which consists mainly of document images from the academic domain, better performance is observed than for IIIT-AR-13k.

Finally, when the model is trained on DocBank and tested on IIIT-AR-13K and PubLayNet, it gives a mean average precision of 0.184 and 0.735 on the IIIT-AR-13k and PubLayNet, respectively. Hence, it can be inferred that the distribution of document images in the DocBank and PubLayNet datasets is similar. Therefore, the model achieves relatively good results when trained on one dataset and tested on the other.

Table 11. The cross-dataset evaluation results for the SR-CNN and baseline Faster R-CNN models. We evaluate the models on the common classes of IIIT-AR-13k, PubLayNet, and DocBank.

Model		SR-CNN			Faster R-CNN		
		IoU [0.50:0.95]					
Train Dataset	Test Dataset	Table (AP)	Figure (AP)	mAP	Table (AP)	Figure (AP)	mAP
IIIT-AR-13K	PublayNet	0.770	0.351	0.562	0.704	0.258	0.481
	DocBank	0.560	0.353	0.456	0.557	0.316	0.436
PublayNet	DocBank	0.545	0.476	0.510	0.547	0.500	0.522
	IIIT-AR-13K	0.538	0.109	0.323	0.497	0.103	0.299
DocBank	IIIT-AR-13K	0.293	0.075	0.184	0.399	0.083	0.241
	PublayNet	0.790	0.681	0.735	0.780	0.717	0.748

5. Conclusions and Future Work

This paper investigates the effectiveness of deep and robust backbones in the document image domain. Further, it also explores the idea of learnable object proposals through

Sparse R-CNN. To this end, the Analyzing Effects of Strong Backbones subsection in Section 4.4.1 shows that naively throwing the best and strongest deep learning backbones at the object detection problem in documents will not improve results. The same is shown by experimenting with strong backbones such as Swin-Transformers and an ensemble method of combining two or more strong backbones via the CBNetV2 framework. These backbones are paired with various detection heads, such as Faster R-CNN [13], Mask R-CNN [30], and Cascade Mask R-CNN, and the degradation in results is evident in Table 7.

Turning our attention to the detection head, where the argument is that it would be better if the object priors in the detection head were sparse and learnable. The Sparse R-CNN [15] detection head achieves this, formulating the SR-CNN model. The Learnable Proposals (Sparse R-CNN) subsection substantiates the above argument through experiments on the IIIT-AR-13k dataset and demonstrates that sparse and learnable proposals in the document image domain can lead to better performance. Table 9 also shows that the performance of Sparse R-CNN suffers when paired with a more robust and deeper backbone. This supports our claim that the naive use of strong backbones is not a great idea in document images.

Inspired by this insight in Section 4.4.2, the SR-CNN model is trained on the PubLayNet dataset and achieves a mean average precision of 0.936. To the best of our knowledge, this is close to the current state-of-the-art of 0.96 (mAP) achieved by CDeC-Net [37]. The SR-CNN model is more computationally efficient (six times less GFLOPs as compared to CDeCNet) and uses the readily available COCO pre-trained weights. Table 1 shows the comparison of the key characteristics of the explored approaches along with their advantages and limitations to facilitate understanding. In future work, we plan to extend this idea to the more challenging domain of document structure recognition in document images. There is also a possibility of exploring the role of different attention mechanisms in the sparse and learnable proposal paradigm.

Author Contributions: Writing—original draft preparation, S.S., K.A.H. and M.Z.A.; writing—review and editing, K.A.H., M.Z.A. and M.L.; supervision and project administration, A.P. and D.S. All authors have read and agreed to the submitted version of the manuscript.

Funding: The work leading to this publication has been partially funded by the European project INFINITY under Grant Agreement ID 883293.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Gantz, J.; Reinsel, D. The digital universe in 2020: Big data, bigger digital shadows, and biggest growth in the far east. *IDC iView IDC Anal. Future* **2012**, *2007*, 1–16.
2. Bhatt, J.; Hashmi, K.A.; Afzal, M.Z.; Stricker, D. A survey of graphical page object detection with deep neural networks. *Appl. Sci.* **2021**, *11*, 5344. [CrossRef]
3. Li, X.H.; Yin, F.; Liu, C.L. Page object detection from pdf document images by deep structured prediction and supervised clustering. In Proceedings of the 2018 24th International Conference on Pattern Recognition (ICPR), Beijing, China, 20–24 August 2018; pp. 3627–3632.
4. Yi, X.; Gao, L.; Liao, Y.; Zhang, X.; Liu, R.; Jiang, Z. CNN based page object detection in document images. In Proceedings of the 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), Kyoto, Japan, 9–15 November 2017; Volume 1, pp. 230–235.
5. Gao, L.; Yi, X.; Jiang, Z.; Hao, L.; Tang, Z. ICDAR2017 competition on page object detection. In Proceedings of the 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), Kyoto, Japan, 9–15 November 2017; Volume 1, pp. 1417–1422.
6. Ajij, M.; Pratihar, S.; Roy, D.S.; Hanne, T. Robust Detection of Tables in Documents Using Scores from Table Cell Cores. *SN Comput. Sci.* **2022**, *3*, 1–19. [CrossRef]

7. Hashmi, K.A.; Liwicki, M.; Stricker, D.; Afzal, M.A.; Afzal, M.A.; Afzal, M.Z. Current Status and Performance Analysis of Table Recognition in Document Images with Deep Neural Networks. *IEEE Access* **2021**, *9*, 87663–87685. [CrossRef]
8. Nazir, D.; Hashmi, K.A.; Pagani, A.; Liwicki, M.; Stricker, D.; Afzal, M.Z. HybridTabNet: Towards better table detection in scanned document images. *Appl. Sci.* **2021**, *11*, 8396. [CrossRef]
9. Mondal, A.; Lipps, P.; Jawahar, C. IIIT-AR-13K: A new dataset for graphical object detection in documents. In *International Workshop on Document Analysis Systems*; Springer: Cham, Switzerland, 2020; pp. 216–230.
10. Smith, R. An overview of the Tesseract OCR engine. In Proceedings of the Ninth International Conference on Document Analysis and Recognition (ICDAR 2007), Curitiba, Brazil, 23–26 September 2007; Volume 2, pp. 629–633.
11. Hashmi, K.A.; Pagani, A.; Liwicki, M.; Stricker, D.; Afzal, M.Z. CasTabDetectoRS: Cascade Network for Table Detection in Document Images with Recursive Feature Pyramid and Switchable Atrous Convolution. *J. Imaging* **2021**, *7*, 214. [CrossRef] [PubMed]
12. Girshick, R. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
13. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 7–12 December 2015; Volume 28.
14. Liang, T.; Chu, X.; Liu, Y.; Wang, Y.; Tang, Z.; Chu, W.; Chen, J.; Ling, H. Cbnetv2: A composite backbone network architecture for object detection. *arXiv* **2021**, arXiv:2107.00420.
15. Sun, P.; Zhang, R.; Jiang, Y.; Kong, T.; Xu, C.; Zhan, W.; Tomizuka, M.; Li, L.; Yuan, Z.; Wang, C.; et al. Sparse r-cnn: End-to-end object detection with learnable proposals. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 14454–14463.
16. Green, E.; Krishnamoorthy, M. Recognition of tables using table grammars. In Proceedings of the Fourth Annual Symposium on Document Analysis and Information Retrieval, Las Vegas, NA, USA, 24–26 April 1995; pp. 261–278.
17. Tupaj, S.; Shi, Z.; Chang, C.H.; Alam, H. *Extracting Tabular Information from Text Files*; EECS Department, Tufts University: Medford, OR, USA, 1996; Volume 1.
18. Kieninger, T.; Dengel, A. Applying the T-RECS table recognition system to the business letter domain. In Proceedings of the Sixth International Conference on Document Analysis and Recognition, Seattle, WA, USA, 13 September 2001; pp. 518–522.
19. Hu, J.; Kashi, R.S.; Lopresti, D.P.; Wilfong, G. Medium-independent table detection. In *Document Recognition and Retrieval VII*; SPIE: Bellingham, DC, USA, 1999; Volume 3967, pp. 291–302.
20. e Silva, A.C. Learning rich hidden markov models in document analysis: Table location. In Proceedings of the 2009 10th International Conference on Document Analysis and Recognition, Barcelona, Spain, 26–29 July 2009; pp. 843–847.
21. Shigarov, A.; Mikhailov, A.; Altaev, A. Configurable table structure recognition in untagged PDF documents. In Proceedings of the 2016 ACM Symposium on Document Engineering, Vienna, Austria, 13–16 September 2016; pp. 119–122.
22. Chandran, S.; Kasturi, R. Structural recognition of tabulated data. In Proceedings of the 2nd International Conference on Document Analysis and Recognition (ICDAR'93), Tsukuba, Japan, 20–22 October 1993; pp. 516–519.
23. Gilani, A.; Qasim, S.R.; Malik, I.; Shafait, F. Table detection using deep learning. In Proceedings of the 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), Kyoto, Japan, 9–15 November 2017; Volume 1, pp. 771–776.
24. Schreiber, S.; Agne, S.; Wolf, I.; Dengel, A.; Ahmed, S. Deepdesrt: Deep learning for detection and structure recognition of tables in document images. In Proceedings of the 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), Kyoto, Japan, 9–15 November 2017; Volume 1, pp. 1162–1167.
25. Göbel, M.; Hassan, T.; Oro, E.; Orsi, G. ICDAR 2013 table competition. In Proceedings of the 2013 12th International Conference on Document Analysis and Recognition, Washington, DC, USA, 25–28 August 2013; pp. 1449–1453.
26. Vo, N.D.; Nguyen, K.; Nguyen, T.V.; Nguyen, K. Ensemble of deep object detectors for page object detection. In Proceedings of the 12th International Conference on Ubiquitous Information Management and Communication, Langkawi, Malaysia, 5–7 January 2018; pp. 1–6.
27. Siddiqui, S.A.; Malik, M.I.; Agne, S.; Dengel, A.; Ahmed, S. Decnt: Deep deformable cnn for table detection. *IEEE Access* **2018**, *6*, 74151–74161. [CrossRef]
28. Sun, N.; Zhu, Y.; Hu, X. Faster R-CNN based table detection combining corner locating. In Proceedings of the 2019 International Conference on Document Analysis and Recognition (ICDAR), Sydney, NSW, Australia, 20–25 September 2019; pp. 1314–1319.
29. Saha, R.; Mondal, A.; Jawahar, C. Graphical object detection in document images. In Proceedings of the 2019 International Conference on Document Analysis and Recognition (ICDAR), Sydney, NSW, Australia, 20–25 September 2019; pp. 51–58.
30. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2961–2969.
31. Shigarov, A.; Altaev, A.; Mikhailov, A.; Paramonov, V.; Cherkashin, E. TabbyPDF: Web-based system for PDF table extraction. In *International Conference on Information and Software Technologies*; Springer: Cham, Switzerland, 2018; pp. 257–269.
32. Chi, Z.; Huang, H.; Xu, H.D.; Yu, H.; Yin, W.; Mao, X.L. Complicated table structure recognition. *arXiv* **2019**, arXiv:1908.04729.
33. Raja, S.; Mondal, A.; Jawahar, C. Table structure recognition using top-down and bottom-up cues. In *European Conference on Computer Vision*; Springer: Cham, Switzerland, 2020; pp. 70–86.
34. Zhang, Z.; Zhang, J.; Du, J.; Wang, F. Split, embed and merge: An accurate table structure recognizer. *Pattern Recognit.* **2022**, *126*, 108565. [CrossRef]

35. Prasad, D.; Gadpal, A.; Kapadni, K.; Visave, M.; Sultanpure, K. CascadeTabNet: An approach for end to end table detection and structure recognition from image-based documents. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Seattle, WA, USA, 14–19 June 2020; pp. 572–573.
36. Cai, Z.; Vasconcelos, N. Cascade r-cnn: Delving into high quality object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 6154–6162.
37. Agarwal, M.; Mondal, A.; Jawahar, C. Cdec-net: Composite deformable cascade network for table detection in document images. In Proceedings of the 2020 25th International Conference on Pattern Recognition (ICPR), Milan, Italy, 10–15 January 2021; pp. 9491–9498.
38. Liu, Y.; Wang, Y.; Wang, S.; Liang, T.; Zhao, Q.; Tang, Z.; Ling, H. Cbnet: A novel composite backbone network architecture for object detection. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 11653–11660.
39. Nguyen, P.; Ngo, L.; Truong, T.; Nguyen, T.T.; Vo, N.D.; Nguyen, K. Page Object Detection with YOLOF. In Proceedings of the 2021 8th NAFOSTED Conference on Information and Computer Science (NICS), Hanoi, Vietnam, 21–22 December 2021; pp. 205–210.
40. Shahab, A.; Shafait, F.; Kieninger, T.; Dengel, A. An open approach towards the benchmarking of table structure recognition systems. In Proceedings of the 9th IAPR International Workshop on Document Analysis Systems, Boston, MA, USA, 9–11 June 2010; pp. 113–120.
41. Fang, J.; Tao, X.; Tang, Z.; Qiu, R.; Liu, Y. Dataset, ground-truth and performance metrics for table detection evaluation. In Proceedings of the 2012 10th IAPR International Workshop on Document Analysis Systems, Gold Coast, QLD, Australia, 27–29 March 2012; pp. 445–449.
42. Gao, L.; Huang, Y.; Déjean, H.; Meunier, J.L.; Yan, Q.; Fang, Y.; Kleber, F.; Lang, E. ICDAR 2019 competition on table detection and recognition (cTDaR). In Proceedings of the 2019 International Conference on Document Analysis and Recognition (ICDAR), Sydney, NSW, Australia, 20–25 September 2019; pp. 1510–1515.
43. Li, M.; Cui, L.; Huang, S.; Wei, F.; Zhou, M.; Li, Z. Tablebank: Table benchmark for image-based table detection and recognition. In Proceedings of the 12th Language Resources and Evaluation Conference, Marseille, France, 11–16 May 2020; pp. 1918–1925.
44. Zhong, X.; Tang, J.; Yepes, A.J. Publaynet: Largest dataset ever for document layout analysis. In Proceedings of the 2019 International Conference on Document Analysis and Recognition (ICDAR), Sydney, NSW, Australia, 20–25 September 2019; pp. 1015–1022.
45. Li, M.; Xu, Y.; Cui, L.; Huang, S.; Wei, F.; Li, Z.; Zhou, M. DocBank: A Benchmark Dataset for Document Layout Analysis. *arXiv* **2020**, arXiv:cs.CL/2006.01038.
46. 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 *European Conference on Computer Vision*; Springer: Cham, Switzerland, 2014; pp. 740–755.
47. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NA, USA, 27–30 June 2016; pp. 770–778.
48. Blaschko, M.B.; Lampert, C.H. Learning to localize objects with structured output regression. In *European Conference on Computer Vision*; Springer: Berlin/Heidelberg, Germany, 2008; pp. 2–15.
49. Chen, K.; Wang, J.; Pang, J.; Cao, Y.; Xiong, Y.; Li, X.; Sun, S.; Feng, W.; Liu, Z.; Xu, J.; et al. MMDetection: Open MMLab Detection Toolbox and Benchmark. *arXiv* **2019**, arXiv:1906.07155.