UniTabBank: A Large Scale Multi-Lingual, Multi-Layout, Multi-Type, Multi-Format Dataset for Table Detection

Ajoy Mondal Saumya Mundra Avijit Dasgupta C. V. Jawahar
CVIT, IIIT, Hyderabad, India
{ajoy.mondal, jawahar}@iiit.ac.in, {saumya.mundra, avijit.dasgupta}@research.iiit.ac.in

Abstract

*Tables play a key role in conveying structured data across documents. Accurate table detection is crucial for downstream tasks like structure recognition and information extraction. However, current datasets lack diversity in format, language, and layout, limiting real-world generalization. This underscores the need for well-annotated datasets that are **multi-lingual**, **layout-diverse**, **document-agnostic**, and **format-rich**.*

*To address these limitations, we introduce **UniTabBank**, a large scale, diverse table detection dataset designed to reflect realistic use cases. **UniTabBank** is characterized by five key attributes: (i) **Multi-Lingual** — supporting 28 languages (including Arabic, English, Hindi, etc.); (ii) **Multi-Layout** — encompassing both single-column and multi-column documents; (iii) **Multi-Type** — covering a wide range of document genres such as annual reports, books, newspapers, and magazines; (iv) **Multi-Format** — comprising scanned documents, photographed pages, and PDFs; and finally (v) **Scale and Annotation Quality** — consists of 55,443 document page images with 82,114 accurately annotated table instances, offering scale and annotation precision.*

*Additionally, we introduce **UniTabDet**, a YOLO-based model for table detection, which outperforms state-of-the-arts on eight out of nine table detection benchmarks. Cross-benchmark evaluation highlights the strong generalization capability of **UniTabBank** compared to existing benchmarks. The dataset and models are available here.*

1. Introduction

Tables are an essential component of structured documents such as reports, invoices, scientific articles, and government forms, where they convey dense, relational information in a compact layout [27]. Detecting tables accurately is critical for downstream tasks like table structure recognition [38, 39, 51], information extraction [25], and document understanding [2, 27, 56]. Recent deep learning-based methods [2, 16, 23, 36, 48] have achieved high accuracy in table detection tasks, using object detection frameworks such as Faster R-CNN [42], Cascade Mask R-CNN [3], and YOLO [40], respectively. More recently, Transformer-based models like DETR [4], ViT [8], and Deformable-DETR [60] have also been explored for table detection [1, 47, 55]. However, these models are often trained and evaluated on monolingual (mostly English), limited layouts, single domain-source datasets (e.g., *ICDAR-2013* [17], *TableBank* [29], and *PubTables-1M* [47]), which may not adequately represent the challenges posed by multi-lingual, multi-layout, multi-type, multi-format documents. These challenges include differences in script, font rendering, complex document and table layouts, etc. Due to the lim-

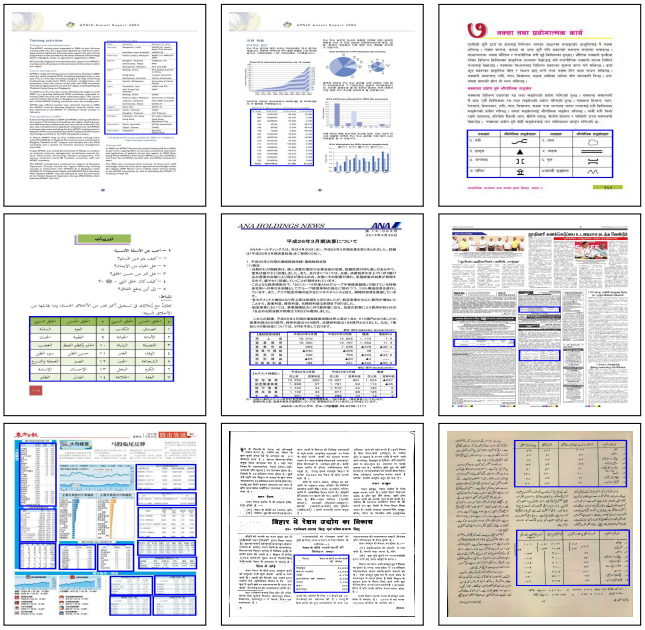

Figure 1. Examples of complex document pages with annotated table bounding boxes with blue colored rectangles across different document formats, types, layouts, and languages.

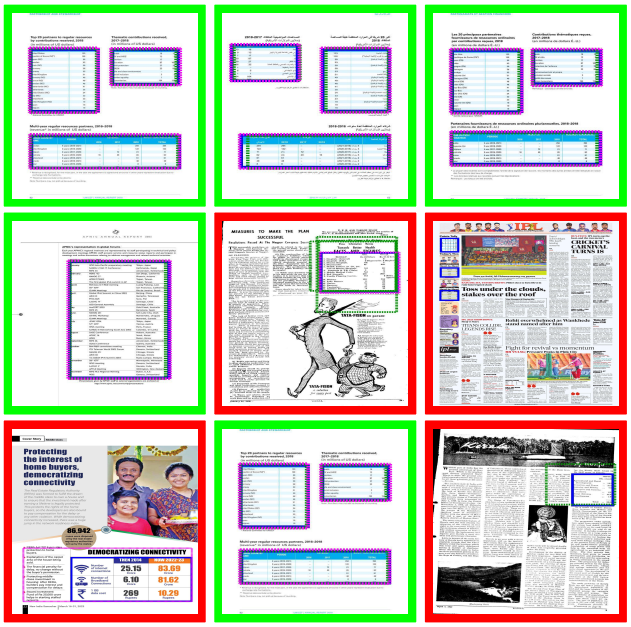

Figure 2. Illustrates the performance of two state-of-the-art models — Table-Transformer (TATR) [59] and SparseTableDet [56] — across varying conditions. The images with **red** boundaries indicate **failure** cases. The images with **green** boundaries indicate **success** cases. **The first row shows documents with the same layout but in different languages (English, Arabic, and French). The second row presents documents in English across multiple types: Annual Report, Magazine, and Newspaper. The third row displays English documents containing tables with diverse layouts: (i) bordered with separator lines, (ii) without borders but with separator lines, and (iii) without both borders and separator lines**. In the visualizations, pink and green dotted boxes represent the table detections by SparseTableDet and Table-Transformer, respectively, while blue boxes indicate the ground truth. Both models perform well when layout remains consistent across languages. However, they struggle to accurately detect tables in documents of varied types and especially under complex or minimal table layouts, highlighting the limitations of current approaches in real-world scenarios.

itations of these datasets, the state-of-the-art fails to detect tables accurately in such cases. Fig. 2 illustrates table detection outputs using Table-Transformer [47] (trained exclusively on the large-scale *PubTables-1M* [47]) and SparseTableDet [56] (trained on the domain-specific *ICT-TD* [57]). Both models often fail to detect tables accurately across diverse document types and complex layouts. Images with red boundaries highlight failure cases, while those with green boundaries indicate successful detections. These limitations highlight the challenges in generalizing table detection models beyond their training distributions.

Many benchmarks exist for table detection shown in Table 1, including *ICDAR-2013* [17], *UNLV* [45], *DeepFigures* [46], *ICDAR-2019* [13], *Marmot* [10], *TNCR* [1], *STDW* [18], *ICT-TD* [57], *CamCap* [43], *CTE* [15], *TableBank* [29], and *PubTables-1M* [47]. However, despite their contributions, most of these datasets exhibit following **limitations**.

- **Limited Domain Diversity:** Most datasets are domain-specific — e.g., *ICDAR* [17] (government), *DeepFigures* [46] (scientific), *TableBank* [29] (Word/LaTeX), and *PubTables-1M* [47] (PMCOA) — limiting generalization to diverse real-world documents like invoices, magazines, and books (see Table 1).
- **Language Bias:** Most datasets are predominantly in English, with few exceptions like *Marmot* [10], *TableBank* [18], and *STDW* [18] (see Table 1), limiting progress in multi-lingual and low-resource settings.
- **Lack of Layout Variability:** Datasets like *TableBank* [29], *PubTables-1M* [47], and *DeepFigures* [46] focus on structured layouts (e.g., scientific articles), offering limited coverage of complex or irregular formats found in magazines, newspapers, and bank statements — hindering cross-layout generalization.
- **Document Source Homogeneity:** Datasets such as *DeepFigures* [46], *TableBank* [29], and *PubTables-1M* [47] rely heavily on sources like arXiv and PMCOA, resulting in stylistic and structural homogeneity that can lead to overfitting and poor generalization to out-of-distribution documents.
- **Lack of Diverse Table Styles:** Another common shortcoming is the lack of diverse table styles, such as borderless or irregular tables, which are underrepresented outside *TNCR* [1].
- **Data Volume Imbalance:** While datasets like *DeepFigures* [46], *TableBank* [29], and *PubTables-1M* [47] provide large-scale data, others such as *ICDAR-2013* [17] and *Marmot* [10] are small, limiting their effectiveness for training deep models (refer Table 1).
- **Lack of Real-world Noise:** Many datasets — *DeepFigures* [46], *TableBank* [29], and *PubTables-1M* [47] are curated from clean, digital sources (refer Table 1). Scenarios with scanning artifacts or degraded print (common in real-world documents) are often absent, affecting real-world applicability.

To overcome these limitations, we introduce a new large-scale dataset for table detection, called **UniTabBank**, designed to be Multi-Lingual, Multi-Layout, Multi-Type, and Multi-Format. **UniTabBank** offers several key advantages. **Language and Layout Diversity:** it includes documents in 28 languages — such as Arabic, Chinese, English, Hindi, Korean, Urdu, etc. and supports both single and multi-column layouts, covering a wide range of real-world layout scenarios. **Format and Type Coverage:** the dataset spans three document formats (scanned, photographed, and PDF) and four major document types (annual reports, books, magazines, and newspapers). Given the widespread use

Dataset	#Image	#Instance	A.M	Format	Document Type	Language
ICDAR-2013 [17]	238	150	Manual	PDF, Scanned	Government documents	English
ICDAR-2019 [13]	1,639	3,600	Manual	PDF, Scanned	Books, Scientific journals, Forms, Financial statements	English
UNLV [45]	2,889	558	Manual	Scanned	Technical reports, magazines, Business letters, Newspapers	English
DeepFigures [46]	5.5M	1.4M	Automatic	PDF	Research articles	English
Marmot [10]	2000	958	Semi-automatic	PDF	Books and Research articles	English, Chinese
TNCR [1]	6,621	9,428	Semi automatic	PDF and Scanned	-	English
STDW [18]	7,000	12,431	Manual	PDF	Invoices, Research papers, Books	English, German, Japanese, Hindi, etc.
ICT-TD [57]	5000	-	Manual	PDF	ICT commodities	English
TableBank [29]	-	417,234	Automatic	Word and Latex documents	-	English, Chinese, Japanese, Arabic
PubTables-1M [47]	1M	948K	Automatic	PDF	Scientific articles	English
UniTabBank (ours)	55,443	82,114	Semi automatic	PDF, Scanned, Photographed	Annual reports, Books, Magazines, Newspapers	28 languages — English, Arabic, Urdu, Hindi, etc.

Table 1. Shows table detection benchmark datasets along with **UniTabBank**. A.M. denotes the annotation mechanism.

of scanned and photographed documents, the **UniTabBank** dataset offers substantial diversity and closely reflects real-world document scenarios. **Scale and Annotation Quality:** **UniTabBank** contains 55,443 document page images with 82,114 accurately annotated table instances, offering scale and annotation precision. Several examples of the **UniTabBank** dataset are shown in Fig. 1. In addition, we introduce **UniTabDet**, a YOLO-based [28] model for accurate and efficient table detection.

The contributions of this paper are summarized as follows:

- **UniTabBank** is the first table detection dataset to combine real-world documents — captured through photography, scanning, and born-digital documents. It uniquely supports 28 languages, making it the most linguistically diverse resource. With coverage across four representative document types and multiple document layout structures,
  UniTabBank provides a comprehensive foundation for developing robust and generalizable table detection models.
- We demonstrate the generalization capability of **UniTabBank** by training **UniTabDet** on both benchmark-specific datasets and **UniTabBank** and evaluating across multiple test benchmarks. Models trained on **UniTabBank** consistently achieve strong performance across diverse datasets, highlighting its advantage over existing benchmarks.
- We evaluate the performance of our **UniTabDet** model across **nine table detection benchmarks** – *ICDAR-2013* [17], *ICDAR-2019* [13], *UNLV* [45], *Marmot* [10], *ICT-TD* [57], *TNCR* [1], *STDW* [18], *TableBank* [29], and *PubTables-1M* [47]. **UniTabDet** consistently outperforms state-of-the-art methods on all benchmarks, with the exception of *ICDAR-2019* [13].
- We conduct a detailed model performance analysis across different document types and languages, offering key insights into their generalization capabilities. Additionally, our ablation studies examine the choice of model architecture, the impact of language/script, varying IoU thresholds, and model sizes, highlighting trade-offs between detection accuracy and model complexity.

2. Related Work

2.1. Table Detection Methods

Early research on table detection in document images began in 1993 with rule-based methods. Itonori [24] proposed using text-block arrangements and ruled lines, while Chandran and Kasturi [6] relied on horizontal and vertical line detection. Subsequent works [14, 21, 22, 34, 52] refined these heuristics, but such methods required extensive manual tuning and lacked generalization across diverse layouts. In 1997, Pyreddy and Croft [37] introduced techniques using character alignment, holes, and gaps, while Seo *et al.* [44] and Kasar *et al.* [26] leveraged junction detection. Kasar *et al.* further improved accuracy by integrating junction features with an SVM classifier, signaling a shift toward machine-learning approaches for more robust and scalable table detection.

Hao *et al.* [19] first applied convolutional neural networks (CNNs) to classify heuristic-based table-like regions from PDFs as table or non-table. However, their approach is limited by its dependence on heuristic region extraction and its applicability only to non-raster PDFs. TableNet [33] used FCN [32] to detect table and row and column of tables. TableSense [7] is a CNN-based model specifically enhanced for detecting tables in spreadsheet documents, incorporating tailored modifications to handle their unique structure

effectively.

Many recent studies have explored the table detection problem. A common approach involves treating tables in visually rich documents as visual objects and applying standard object detection methods to identify them. Gilani *et al.* [16] applied Faster R-CNN [42] for table detection, using distance-transformed images instead of raw documents better to adapt the pre-trained model across diverse document types. Sun *et al.* [48] refined table boundaries by combining corner information with Faster R-CNN outputs, reducing false positives. Due to the limited number of training samples for the table detection problem, transfer learning methods are widely used. Multi-Type-TD-TSR [12] used Faster R-CNN [42] to extract tables from documents. In [5], the authors show that fine-tuning object detection models (Mask R-CNN [20], RetinaNet [30], SSD [31], YOLO [41]) on a closely related domain helps prevent overfitting and improves performance across tasks. CDeC-Net [2] used Cascade Mask R-CNN [3] that incorporates a dual-backbone architecture for table detection. CascadeTabNet [36], built on Cascade Mask R-CNN [3] with an HRNet [54] backbone, employs two-stage transfer learning and data augmentation. Similarly, TableDet [11], based on Cascade R-CNN [3], introduces Table-Aware Cutout augmentation and a two-step transfer learning strategy to boost performance. Xiao *et al.* [56] employed SparseR-CNN [49] as the base model and enhanced it with Noise-Augmented Region Proposal Generation, Many-to-One Label Assignment, and a Decoupled IoU strategy to improve the accuracy of table detection.

In addition to two-stage detectors, one-stage methods like YOLO [41] and its variants have also been applied to table detection. YOLOv3-TD [23] builds on YOLOv3 [40], introducing adaptive modifications such as optimized anchor selection and an improved post-processing pipeline. Beyond one-stage and two-stage methods, transformer-based models like DETR [4], ViT [8], and Deformable-DETR [60] have also been explored for table detection in Table-Transformer [47], TransTab [55], and TNCR [1].

2.2. Datasets for Table Detection

Several benchmarks support table detection research. The *ICDAR-2013* dataset [17] contains 238 pages from 67 government PDFs (EU and US), with 150 tables annotated as rectangular regions. The *UNLV* dataset [45] offers 1,639 scanned images from diverse sources, with 558 table zones annotated at both table and cell levels. *DeepFigures* [46] provides 5.5M scientific pages from arXiv and PubMed, including 1.4M tables and 4M figures, supporting large-scale analysis. The *ICDAR-2019* dataset [13] includes 2,439 images from historical and modern documents, with annotations for table regions and cell structures in scanned and digital formats.

The *Marmot* dataset [10] features 2,000 bilingual PDF pages (Chinese and English) with diverse layouts and table styles, supporting both detection and structure recognition. The *TNCR* dataset [1] contains a mix of scanned and digital documents across multiple domains, annotated for bordered and borderless tables — highlighting the challenge of detecting tables lacking visible boundaries. The *STDW* [18] dataset comprises 7,294 document images containing tables sourced from a wide range of domains, including electronic component datasheets, material safety data sheets, product safety sheets, billing invoices, research papers, financial reports, and books. The *ICT-TD* dataset [57] comprises 5,000 PDF images collected from documents related to Information and Communication Technology (ICT) products and services. The *CamCap* [43] comprises 85 camera-captured images, testing detection robustness on curved and flat surfaces.

The *TableBank* [29] is a large-scale dataset with 417K labeled tables from Word and LaTeX documents (2014–2018), with 145K pages offering structure-level annotations. The *PubTables-1M* [47] comprises nearly one million tables with detailed header and spatial annotations, supporting various input modalities and addressing annotation inconsistencies through canonicalization. The *CTE* dataset [15] includes 75K annotated scientific pages (35K tables), combining annotations from *PubTables-1M* [47] and *PubLayNet* [59], and extending them with new CTE-specific classes.

3. UniTabBank Dataset

The **UniTabBank** dataset comprises a total of **55,443 document images**, organized into four primary categories based on content and layout: **Annual Report (55%), Book (21%), Magazine (18%), and Newspapers (6%)**. These document images are available in three formats: **PDFs, photographed documents, and scanned documents**. The dataset spans **28 languages**, including *Arabic, Assamese, Bengali, Bodo, Chinese, English, Farsi, French, Gujarati, Hindi, Indonesian, Japanese, Kannada, Korean, Malayalam, Manipuri, Marathi, Nepali, Oriya, Punjabi, Sanskrit, Sinhala, Spanish, Tamil, Telugu, Thai, Urdu*, and *Vietnamese*. In total, the dataset contains **82,114 annotated table instances**. Tables exhibit a wide variety of table layout structures, including (i) bordered tables with complete row and column separators, (ii) bordered tables without row and column separators, (iii) borderless tables with row and column separators, (iv) bordered tables with partial separators, (v) tables containing merged cells, and (vi) tables without merged cells.

The **UniTabBank** dataset is constructed through a carefully stratified sampling strategy to maximize diversity and generalization. It spans 28 languages, four document categories (annual reports, books, magazines, and newspa-

Document Type	Image				Document Type	Table			
	Total number	Training number	Validation number	Test number		Total number	Training number	Validation number	Test number
Annual Report	30573	18344	3078	9151	Annual Report	44456	26639	4459	13358
Book	11894	7148	1215	3531	Book	15827	9453	1574	4800
Magazine	9927	5955	1012	2960	Magazine	13020	7784	1310	3926
Newspaper	3049	1786	308	955	Newspaper	8811	4862	838	3111
Total	55443	33233	5613	16597	Total	82114	48738	8181	25195

Table 2. Shows summary of table frequencies across different document categories and their distribution within each dataset split of **UniTabBank** dataset.

pers), and three acquisition modes (PDFs, scanned copies, and photographed documents). For each language–category combination, documents were sourced from multiple public repositories and publishers to minimize domain bias[1]. This systematic approach ensures that **UniTabBank** captures a broad spectrum of layouts, visual qualities, and content types, making it a robust and representative benchmark for cross-domain table detection research. The origin and composition of each subset are detailed below.

- The **Annual Report** subset comprises **30,573 document images** across **19 languages**: Arabic, Bengali, Chinese, English, French, Gujarati, Hindi, Indonesian, Japanese, Kannada, Korean, Malayalam, Marathi, Oriya, Sinhala, Spanish, Tamil, Thai, and Vietnamese. The corresponding PDFs were sourced from various public repositories and archives. These PDFs were subsequently converted into document images. This subset contains a total of **44,456 annotated table instances**.
- The **Book** subset comprises **11,894 scanned document images** from textbooks spanning four educational levels — elementary, middle school, high school, and college and covering **29 subjects** such as Mathematics, Physics, Chemistry, Biology, History, Geography, Politics, Economics, Social Science, and Computer Science. The subset represents content in **19 languages**: Arabic, Assamese, Bengali, Bodo, English, Farsi, Gujarati, Hindi, Kannada, Malayalam, Manipuri, Marathi, Nepali, Oriya, Punjabi, Sanskrit, Tamil, Telugu, and Urdu. In total, it includes **15,827 annotated table instances**.
- The **Magazine** subset contains **9,927** document images sourced from both PDF-format magazines and scanned pages, collected from multiple platforms. It includes 13,020 annotated table instances and content in **13 languages**: Assamese, Bengali, English, Gujarati, Hindi, Kannada, Malayalam, Marathi, Oriya, Punjabi, Tamil, Telugu, and Urdu.
- The **Newspaper** subset includes **3049** document images in PDF format, sourced from various publishers and photographed samples. It covers **12 languages** — Bengali, English, Gujarati, Hindi, Kannada, Malayalam, Marathi, Oriya, Punjabi, Tamil, Telugu, and Urdu and contains a total of **8,811 annotated table instances**.

To ensure fair evaluation, we divided the dataset into training, validation, and test sets following a **7:1:2 ratio**. We maintained a balanced distribution of label (*table*) across all three sets. Table 2 provides a summary of table frequencies across different document categories and their distribution within each dataset split[2].

3.1. Data Annotation

We adopt a two-stage framework for annotating tables in document images: (i) automatic detection and (ii) manual verification and correction. Initially, we employ DocLayOut-YOLO [58] to automatically detect tables within the document images. The manual verification stage is carried out by five trained annotators with prior experience in document annotation. Each annotator follows a detailed guideline that specified how to correct bounding boxes, handle borderline cases (e.g., tables without borderlines), and resolve ambiguous table boundaries. To ensure quality, we adopt a two-pass verification protocol: an independent second annotator re-checks a random 20% of the samples. The measured inter-annotator agreement (IoU $>$ 0.9) is consistently high, confirming reliability and the high quality of the annotations. The annotation files follow the Pascal VOC annotation format [9] for object detection.

4. Experiments

Baseline and Implementation Details: Our baseline **UniTabDet** is built on the **YOLOv11 architecture** [28]. YOLOv11 is the latest advancement in the YOLO family, designed for efficient and accurate real-time object detection. Its architecture introduces key innovations such as the C3k2 block, SPPF (Spatial Pyramid Pooling), and C2PSA (Parallel Spatial Attention), which improve feature extraction and attention to important regions. The model supports multiple tasks beyond detection, including instance segmentation, pose estimation, image classification, and ori-

[1] *A complete list of sources is provided in Appendix A of the supplementary material.*

[2] *For detailed language-wise table frequencies and additional visual examples, refer to Appendix B in the supplementary material.*

ented object detection. With variants ranging from nano to extra-large, YOLOv11 balances speed, accuracy, and parameter efficiency, making it versatile for edge devices and high-performance computing applications. All experiments are conducted using four NVIDIA GeForce RTX 2080 Ti GPUs (each with 12 GB memory). We train the model for 300 epochs using an input resolution of 1280 and a batch size 8. The training setup includes a learning rate 0.01, weight decay of 0.0005, and momentum of 0.937.

Datasets: We use nine table detection (TD) benchmarks — *ICDAR-2013* [17], *UNLV* [45], *ICDAR-2019* [13], *Marmot* [10], *TNCR* [1], *STDW* [18], *ICT-TD* [57], *TableBank* [29], and *PubTables-1M* [47].

Evaluation Metrices: We assess model performance using standard evaluation metrics: precision (P), recall (R), and F1 score [2, 13, 55], calculated at various Intersection-over-Union (IoU) thresholds. To provide a holistic view of detection quality, we also report the weighted average F1 score [13, 56] and average precision metrics — AP_{50}, AP_{75}, and mean AP over the IoU range [0.50–0.95] [18, 47][3].

5. Result Analysis

5.1. Generalization Capability of UniTabBank

Training on benchmark-specific datasets typically yields the best in-domain accuracy, but such models fail to generalize to unseen domains. **UniTabBank** is designed to overcome this limitation by offering document linguistic, layout, type, and format diversity. To validate this, we trained **UniTabDet** (YOLOv11-based) on both benchmark-specific datasets and **UniTabBank**, and evaluated across multiple test benchmarks. Results in Table 3 show that while in-domain training achieves near-perfect accuracy (e.g., *PubTables* → *PubTables*: AP=0.989, *TableBank* → *TableBank*: AP=0.958, **UniTabBank** → **UniTabBank**: AP=0.959), these models perform poorly on unseen datasets (e.g., *PubTables* → *UNLV*: AP=0.417, *TableBank* → *UNLV*: AP=0.288). In contrast, **UniTabBank**-trained models achieve consistently high performance across diverse benchmarks (e.g., AP=0.826 on *PubTables*, AP=0.899 on *TableBank*, AP=0.773 on *UNLV*, AP=0.928 on *STDW*), often outperforming single-benchmark models. Other datasets, such as *ICT-TD*, *TNCR*, and *ICDAR-2019*, provide moderate cross-domain robustness (AP≈0.80 - 0.88) but do not match **UniTabBank**'s breadth. These findings highlight the generalization capability of **UniTabBank**[4].

[3] *For further details, see Appendix C in the supplementary material.*

[4] *Additional cross-benchmark dataset results are provided in Appendix D of the supplementary material.*

Training Set	Test Set	AP_{50}	AP_{75}	AP
PubTables	*PubTables*	**0.994**	**0.994**	**0.989**
TableBank		0.863	0.734	0.665
UniTabBank		0.993	0.947	0.826
ICT-TD		0.981	0.933	0.828
TNCR		0.985	0.916	0.810
ICDAR-2019		0.985	0.924	0.821
PubTables	*TableBank*	0.840	0.719	0.606
TableBank		**0.980**	**0.973**	**0.958**
UniTabBank		0.933	0.921	0.899
ICT-TD		0.921	0.898	0.865
TNCR		0.916	0.895	0.871
ICDAR-2019		0.916	0.893	0.859
PubTables	**UniTabBank**	0.599	0.523	0.439
TableBank		0.742	0.694	0.661
UniTabBank		**0.981**	**0.971**	**0.959**
ICT-TD		0.877	0.830	0.797
TNCR		0.851	0.793	0.767
ICDAR-2019		0.878	0.828	0.794
PubTables	*UNLV*	0.604	0.498	0.417
TableBank		0.391	0.314	0.288
UniTabBank		**0.914**	**0.854**	**0.773**
ICT-TD		0.663	0.568	0.500
TNCR		0.806	0.723	0.635
ICDAR-2019		0.729	0.653	0.568
PubTables	*STDW*	0.699	0.594	0.519
TableBank		0.675	0.642	0.632
UniTabBank		**0.964**	**0.949**	**0.928**
ICT-TD		0.926	0.895	0.875
TNCR		0.888	0.853	0.830
ICDAR-2019		0.929	0.897	0.879

Table 3. Cross-benchmark evaluation of **UniTabDet** trained on different datasets and tested across multiple benchmarks. Models trained on benchmark-specific datasets achieve high in-domain accuracy but generalize poorly, whereas the models trained with **UniTabBank** achieve consistently strong cross-domain performance. Bold and underlined values represent the best and second-best results, respectively.

5.2. Comparison with SOTA on TD Benchmarks

Table 4 shows that **UniTabDet**†, fine-tuned using only 20,000 images from the *TableBank* dataset, outperforms the leading method CascadeTabNet [36] by 2.1%, demonstrating strong performance with limited supervision. Table 5 presents results on the *PubTables-1M* dataset, showing that our fine-tuned **UniTabDet**† — trained with only 20,000 samples — outperforms the state-of-the-art Table-Transformer [47] by 2.4%[5].

[5] *Additional quantitative results are provided in Appendix E, and visual samples in Appendix F of the supplementary material.*

Method	Train		Test: *TableBank*		
	Dataset	#Image	Word+Latex		
			P	R	F1
Li *et al.* [29]	*TableBank*	260582	**0.966**	0.899	0.931
CTabNet [36]	*TableBank*	260582	0.929	0.957	0.943
CDeC-Net [2]	*TableBank*	260582	0.934	0.924	0.929
UniTabDet	**UniTabBank**	55,443	0.909	0.965	0.936
UniTabDet†	*TabelBank*	20000	0.949	**0.979**	**0.964**

Table 4. Performance evaluation on *TableBank* using precision (P), recall (R) and F1 score at IoU=0.5. † models fine-tuned with only on 20K samples from *TableBank*. Bold and underlined values indicate the best and second-best results, respectively.

Model	Train		Test: *PubTables*		
	Dataset	#Image	AP_{50}	AP_{75}	AP
Table-Transformer [47]	*PubTables*	460,589	**0.995**	0.989	0.970
TabSniper [50]	*BankTabNet*	9724	0.939	0.906	0.852
ClusterTabNet [35]	*PubTables*	460,589	0.990	-	0.989
UniTabDet	**UniTabBank**	55,443	0.993	0.947	0.826
UniTabDet†	*PubTables*	20,000	**0.995**	**0.995**	**0.994**

Table 5. Performance evaluation on *PubTables-1M* using object detection metrics. † models fine-tuned on the *PubTables-1M* dataset. Bold and underlined values indicate the best and second-best results, respectively.

5.3. Analysis on Failure Cases

Although **UniTabDet** trained on **UniTabBank** demonstrates strong performance across multiple benchmarks, Fig. 4 presents several representative failure cases that reveal its current limitations. In datasets like *UNLV*, *PubTables*, and *TNCR*, errors frequently occur when tables lack clear row and column separators or well-defined headers, making it challenging for the model to infer the structure accurately. Similarly, in *ICDAR-2019*, the model struggles with archival documents, where degraded scan quality, faded ink, or full-page spanning tables reduce detection accuracy. The *ICT-TD* dataset poses additional challenges due to its unconventional or highly complex table layouts, which often deviate from standard structures and cause boundary localization errors. These examples highlight that while **UniTabDet** achieves robust overall performance, it remains sensitive to ambiguous, noisy, or irregular table formats. Addressing these limitations is important for improving the model's robustness and generalization in future work.

5.4. Ablation Study

Choice of Architecture: We selected **YOLOv11** (**UniTabDet**) as a baseline because it provides an efficient, scalable, and robust baseline for table detection while ensuring fast inference and easy deployment. Our aim is to highlight the value of **UniTabBank** rather than propose a novel architecture. As shown in Table 6, **UniTabDet** achieves the highest accuracy (AP = 0.959), outperforming both DocLayOut built on YOLOv10 [53] (AP = 0.956) and TATR (AP = 0.723). This demonstrates that **UniTabDet** offers the best balance of accuracy, robustness, and practical usability.

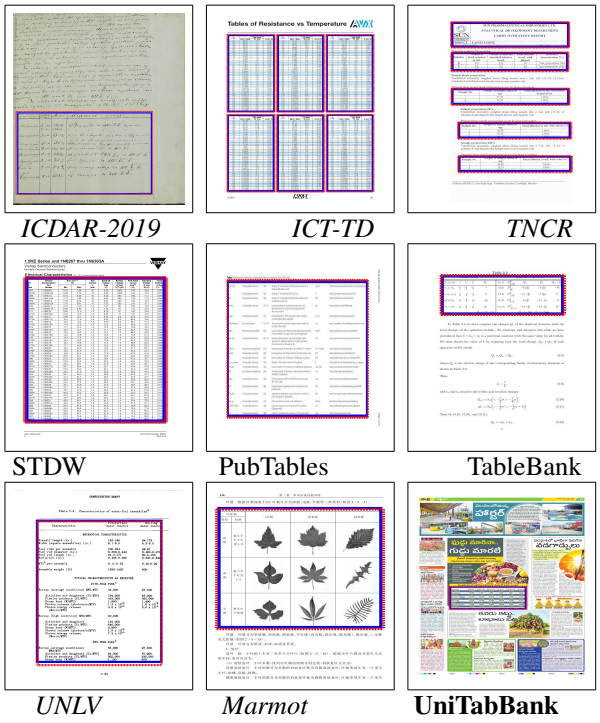

Figure 3. Shows visual results of multiple benchmarks using **UniTabDet**. The red dotted box represents the detected tables by **UniTabDet**, while the blue boxes indicate corresponding ground truths.

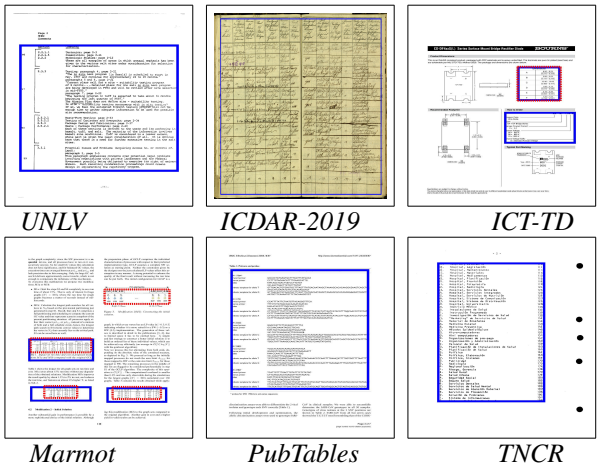

Figure 4. Shows a few failure results on multiple benchmarks using **UniTabDet**. The red dotted box represents the detected tables by **UniTabDet**, while the blue boxes indicate corresponding ground truths.

Model	AP_{50}	AP_{75}	AP
DocLayOut [58]	**0.984**	**0.972**	0.956
TATR [47]	0.895	0.780	0.723
SparseTableDet [56]	0.927	0.899	0.874
UniTabDet	0.981	0.971	**0.959**

Table 6. Comparison of **UniTabDet** with DocLayOut, TATR, and SparseTableDet on **UniTabBank**. Results show that **UniTabDet** achieves the highest accuracy (AP), justifying its choice as the backbone for our benchmark. Bold and underlined values represent the best and second-best results, respectively.

Model Parameters: Table 7 presents the relationship between model size and performance. The largest **UniTabDet** model, with 56.9M parameters, achieves the highest AP of 0.959. However, a significantly smaller model with only 2.6M parameters shows just a minor drop of 0.4% in AP. This indicates that lightweight models can deliver competitive performance while being more suitable for deployment in real-world applications with limited computational resources.

Model	#Params (M)	Test: UniTabBank		
		AP_{50}	AP_{75}	AP
UniTabDet (n)	2.6	0.984	0.972	0.955
UniTabDet (s)	9.4	0.982	0.971	0.953
UniTabDet (m)	20.1	0.981	0.970	0.954
UniTabDet (l)	25.3	**0.985**	**0.973**	0.955
UniTabDet (x)	56.9	0.981	0.971	**0.959**

Table 7. Performance comparison on the UniTabBank dataset using different UniTabDet model variants. Here, n, s, m, l, and x represent the tiny, small, medium, large, and extra-large configurations, respectively.

Impact of IoU: Table 3 presents the performance of **UniTabDet** trained with **UniTabBank**, across multiple benchmarks at different IoU thresholds. The results show that the Average Precision (AP) at IoU 0.75 and the mean AP over IoU [0.5–0.95] are closely aligned with the AP at IoU 0.5 (**UniTabBank**, *TableBank*, *PubTables*, *STDW*). This consistency across thresholds indicates the robustness and reliability of the model in accurately detecting tables under varying evaluation criteria.

Importance of Language: To assess the influence of language or script on table detection, we apply three different blurring techniques to the training images — Gaussian blur (0,5), Median blur (7×7), and Average blur (9×9) — to suppress textual details while preserving structural layout. Examples of the original and blurred images are shown in Fig. 5. Using these modified training sets, we train three models: **UniTabDet**$^{\alpha}$, **UniTabDet**$^{\beta}$, and **UniTabDet**$^{\gamma}$, corresponding to the respective blur types. Table 8 shows performance drops of 0.8%, 3.7%, and 2.2% (AP at IoU [0.5-0.95]) for **UniTabDet**$^{\alpha}$, **UniTabDet**$^{\beta}$, and **UniTabDet**$^{\gamma}$, respectively, compared to the original **UniTabDet**, indicating that table detection depends little on the script or language of the table content.

(a) (b) (c)

Figure 5. Example of (a) a training image and its corresponding blurred versions (with content in a mixture of English and Telugu languages) using (b) Gaussian blur and (c) average blur techniques. Better view in Zoom.

Model	Blur	Test: UniTabBank		
		AP_{50}	AP_{75}	AP
UniTabDet$^{\alpha}$	Gaussian	**0.985**	**0.972**	0.951
UniTabDet$^{\beta}$	Median	0.972	0.954	0.922
UniTabDet$^{\gamma}$	Average	0.983	0.965	0.937
UniTabDet	-	0.981	0.971	**0.959**

Table 8. Performance comparison between the original **UniTabDet** and its blurred variants to evaluate the impact of table content language on detection accuracy.

6. Conclusion

This paper presents **UniTabBank**, a large-scale table detection dataset comprising four document types collected through PDF rendering, scanning, and photographing. Designed to reflect real-world scenarios, **UniTabBank** offers diverse formats, languages, layouts, and acquisition conditions, making it a strong benchmark for robust table detection. We introduce **UniTabDet**, a YOLO-based table detection model, and evaluate it on nine widely used table detection benchmarks. Across all benchmarks, **UniTabDet** consistently outperforms existing state-of-the-art methods. We train the model on both individual benchmark datasets and on **UniTabBank**, then test it across multiple benchmarks. Models trained on **UniTabBank** achieve consistently strong results across diverse datasets, demonstrating superior cross-dataset generalization compared to existing benchmarks.

Acknowledgment

This work is supported by MeitY, Government of India, through the NLTM-Bhashini project.

References

[1] Abdelrahman Abdallah, Alexander Berendeyev, Islam Nuradin, and Daniyar Nurseitov. TNCR: Table net detection and classification dataset. *Neurocomputing*, 473:79–97, 2022. 1, 2, 3, 4, 6

[2] Madhav Agarwal, Ajoy Mondal, and CV Jawahar. CDeC-Net: Composite deformable cascade network for table detection in document images. In *ICPR*, pages 9491–9498, 2021. 1, 4, 6, 7

[3] Zhaowei Cai and Nuno Vasconcelos. Cascade R-CNN: Delving into high quality object detection. In *CVPR*, pages 6154–6162, 2018. 1, 4

[4] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, pages 213–229, 2020. 1, 4

[5] Ángela Casado-García, César Domínguez, Jónathan Heras, Eloy Mata, and Vico Pascual. The benefits of close-domain fine-tuning for table detection in document images. In *DAS*, pages 199–215, 2020. 4

[6] Surekha Chandran and Rangachar Kasturi. Structural recognition of tabulated data. In *ICDAR*, pages 516–519, 1993. 3

[7] Haoyu Dong, Shijie Liu, Shi Han, Zhouyu Fu, and Dongmei Zhang. TableSense: Spreadsheet table detection with convolutional neural networks. In *AAAI*, pages 69–76, 2019. 3

[8] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 1, 4

[9] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 88(2):303–338, 2010. 5

[10] Jing Fang, Xin Tao, Zhi Tang, Ruiheng Qiu, and Ying Liu. Dataset, ground-truth and performance metrics for table detection evaluation. In *DAS*, pages 445–449, 2012. 2, 3, 4, 6

[11] Johan Fernandes, Murat Simsek, Burak Kantarci, and Shahzad Khan. TableDet: An end-to-end deep learning approach for table detection and table image classification in data sheet images. *Neurocomputing*, 468:317–334, 2022. 4

[12] Pascal Fischer, Alen Smajic, Giuseppe Abrami, and Alexander Mehler. Multi-type-td-tsr–extracting tables from document images using a multi-stage pipeline for table detection and table structure recognition: From ocr to structured table representations. In *KI*, pages 95–108, 2021. 4

[13] Liangcai Gao, Yilun Huang, Hervé Déjean, Jean-Luc Meunier, Qinqin Yan, Yu Fang, Florian Kleber, and Eva Lang. Icdar 2019 competition on table detection and recognition (ctdar). In *ICDAR*, pages 1510–1515, 2019. 2, 3, 4, 6

[14] Basilios Gatos, Dimitrios Danatsas, Ioannis Pratikakis, and Stavros J Perantonis. Automatic table detection in document images. In *ICAPR*, pages 609–618, 2005. 3

[15] Andrea Gemelli, Emanuele Vivoli, and Simone Marinai. Cte: A dataset for contextualized table extraction. *arXiv preprint arXiv:2302.01451*, 2023. 2, 4

[16] Azka Gilani, Shah Rukh Qasim, Imran Malik, and Faisal Shafait. Table detection using deep learning. In *ICDAR*, pages 771–776, 2017. 1, 4

[17] Max Göbel, Tamir Hassan, Ermelinda Oro, and Giorgio Orsi. Icdar 2013 table competition. In *ICDAR*, pages 1449–1453, 2013. 1, 2, 3, 4, 6

[18] Mrinal Haloi, Shashank Shekhar, Nikhil Fande, Siddhant Swaroop Dash, et al. Table detection in the wild: A novel diverse table detection dataset and method. *arXiv preprint arXiv:2209.09207*, 2022. 2, 3, 4, 6

[19] Leipeng Hao, Liangcai Gao, Xiaohan Yi, and Zhi Tang. A table detection method for pdf documents based on convolutional neural networks. In *DAS*, pages 287–292, 2016. 3

[20] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *ICCV*, pages 2961–2969, 2017. 4

[21] Yuki Hirayama. A method for table structure analysis using dp matching. In *ICDAR*, pages 583–586, 1995. 3

[22] Jianying Hu, Ramanujan S Kashi, Daniel P Lopresti, and Gordon Wilfong. Medium-independent table detection. In *DRR*, pages 291–302, 1999. 3

[23] Yilun Huang, Qinqin Yan, Yibo Li, Yifan Chen, Xiong Wang, Liangcai Gao, and Zhi Tang. A yolo-based table detection method. In *ICDAR*, pages 813–818, 2019. 1, 4

[24] Katsuhiko Itonori. Table structure recognition based on textblock arrangement and ruled line position. In *ICDAR*, pages 765–768, 1993. 3

[25] Pongsakorn Jirachanchaisiri, Nam Tuan Ly, and Atsuhiro Takasu. Trh2tqa: Table recognition with hierarchical relationships to table question-answering on business table images. In *WACV*, pages 8844–8852, 2025. 1

[26] Thotreingam Kasar, Philippine Barlas, Sebastien Adam, Clément Chatelain, and Thierry Paquet. Learning to detect tables in scanned document images using line information. In *ICDAR*, pages 1185–1189, 2013. 3

[27] Mahmoud salaheldin Kasem, Abdelrahman Abdallah, Alexander Berendeyev, Ebrahem Elkady, Mohamed Mahmoud, Mahmoud Abdalla, Mohamed Hamada, Sebastiano Vascon, Daniyar Nurseitov, and Islam Taj-eddin. Deep learning for table detection and structure recognition: A survey. *ACM Computing Surveys*, 56(12), 2024. 1

[28] Rahima Khanam and Muhammad Hussain. Yolov11: An overview of the key architectural enhancements. *arXiv preprint arXiv:2410.17725*, 2024. 3, 5

[29] Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, Ming Zhou, and Zhoujun Li. TableBank: Table benchmark for image-based table detection and recognition. In *COLING*, pages 1918–1925, 2020. 1, 2, 3, 4, 6, 7

[30] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, pages 2980–2988, 2017. 4

[31] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. SSD: Single shot multibox detector. In *ECCV*, pages 21–37, 2016. 4

[32] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, pages 3431–3440, 2015. 3

[33] Shubham Singh Paliwal, D Vishwanath, Rohit Rahul, Monika Sharma, and Lovekesh Vig. TableNet: Deep learning model for end-to-end table detection and tabular data extraction from scanned document images. In *ICDAR*, pages 128–133, 2019. 3

[34] Ricardo Wandré Dias Pedro, Fátima LS Nunes, and Ariane Machado-Lima. Using grammars for pattern recognition in images: a systematic review. *ACM CSUR*, 46(2):1–34, 2013. 3

[35] Marek Polewczyk and Marco Spinaci. ClusterTabNet: Supervised clustering method for table detection and table structure recognition. In *ICDAR*, pages 334–349, 2024. 7

[36] Devashish Prasad, Ayan Gadpal, Kshitij Kapadni, Manish Visave, and Kavita Sultanpure. CascadeTabNet: An approach for end to end table detection and structure recognition from image-based documents. In *CVPRW*, pages 572–573, 2020. 1, 4, 6, 7

[37] P Pyreddy and WB Croft. Tinti: A system for retrieval in text tables title2, 1997. 3

[38] Sachin Raja, Ajoy Mondal, and CV Jawahar. Visual understanding of complex table structures from document images. In *WACV*, pages 2299–2308, 2022. 1

[39] Sachin Raja, Ajoy Mandal, and CV Jawahar. Treading towards privacy-preserving table structure recognition. In *WACV*, pages 2311–2321, 2025. 1

[40] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018. 1, 4

[41] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, pages 779–788, 2016. 4

[42] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. *NeurIPs*, 28, 2015. 1, 4

[43] Wonkyo Seo, Hyung Il Koo, and Nam Ik Cho. Junction-based table detection in camera-captured document images. *IJDAR*, 18:47–57, 2015. 2, 4

[44] Wonkyo Seo, Hyung Il Koo, and Nam Ik Cho. Junction-based table detection in camera-captured document images. *IJDAR*, 18:47–57, 2015. 3

[45] Faisal Shafait and Ray Smith. Table detection in heterogeneous documents. In *DAS*, pages 65–72, 2010. 2, 3, 4, 6

[46] Noah Siegel, Nicholas Lourie, Russell Power, and Waleed Ammar. Extracting scientific figures with distantly supervised neural networks. In *ACM/IEEE JCDL*, pages 223—-232, 2018. 2, 3, 4

[47] B Smock, R Pesala, and R Abraham. Pubtables-1m: Towards comprehensive table extraction from unstructured documents. In *CVPR*, pages 4624–4632, 2021. 1, 2, 3, 4, 6, 7, 8

[48] Ningning Sun, Yuanping Zhu, and Xiaoming Hu. Faster r-cnn based table detection combining corner locating. In *ICDAR*, pages 1314–1319, 2019. 1, 4

[49] Peize Sun, Rufeng Zhang, Yi Jiang, Tao Kong, Chenfeng Xu, Wei Zhan, Masayoshi Tomizuka, Lei Li, Zehuan Yuan, Changhu Wang, et al. Sparse R-CNN: End-to-end object detection with learnable proposals. In *CVPR*, pages 14454–14463, 2021. 4

[50] Abhishek Trivedi, Sourajit Mukherjee, Rajat Kumar Singh, Vani Agarwal, Sriranjani Ramakrishnan, and Himanshu S Bhatt. TabSniper: Towards accurate table detection & structure recognition for bank statements. *arXiv preprint arXiv:2412.12827*, 2024. 7

[51] David Tschirschwitz and Volker Rodehorst. Cisol: An open and extensible dataset for table structure recognition in the construction industry. In *WACV*, pages 7605–7613, 2025. 1

[52] Scott Tupaj, Zhongwen Shi, C Hwa Chang, and Hassan Alam. Extracting tabular information from text files. *EECS Department, Tufts University, Medford, USA*, 1, 1996. 3

[53] Ao Wang, Hui Chen, Lihao Liu, Kai Chen, Zijia Lin, Jungong Han, et al. Yolov10: Real-time end-to-end object detection. In *NeurIPS*, pages 107984–108011, 2024. 7

[54] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, et al. Deep high-resolution representation learning for visual recognition. *IEEE trans. on PAMI*, 43(10):3349–3364, 2020. 4

[55] Yongzhou Wang, Wenliang Lv, Weijie Wu, Guanheng Xie, BiBo Lu, ChunYang Wang, Chao Zhan, and Baishun Su. TransTab: A transformer-based approach for table detection and tabular data extraction from scanned document images. *Machine Learning with Applications*, pages 100665–100678, 2025. 1, 4, 6

[56] Bin Xiao, Murat Simsek, Burak Kantarci, and Ala Abu Alkheir. Table detection for visually rich document images. *Knowledge-Based Systems*, 282:111080–111115, 2023. 1, 2, 4, 6, 8

[57] Bin Xiao, Murat Simsek, Burak Kantarci, and Ala Abu Alkheir. Revisiting table detection datasets for visually rich documents. *IJDAR*, pages 1–20, 2025. 2, 3, 4, 6

[58] Zhiyuan Zhao, Hengrui Kang, Bin Wang, and Conghui He. Doclayout-yolo: Enhancing document layout analysis through diverse synthetic data and global-to-local adaptive perception. *arXiv preprint arXiv:2410.12628*, 2024. 5, 8

[59] Xu Zhong, Jianbin Tang, and Antonio Jimeno Yepes. Publaynet: largest dataset ever for document layout analysis. In *ICDAR*, pages 1015–1022, 2019. 2, 4

[60] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable DETR: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020. 1, 4