

Layout-Aware Region Compression for Transformer Document Understanding

Authors

Haruto Sato^{1*}, Yuki Nakamura², Rina Takahashi³

Affiliations

Graduate School of Information Science and Technology, The University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan.

*Corresponding author: haruto.sato@i.u-tokyo.ac.jp

Abstract

Transformer models for document understanding must process text, tables, stamps, signatures, and layout structures at the page level. However, many document images contain repeated headers, blank margins, boilerplate clauses, and low-information background regions. Treating all regions equally increases memory usage and may reduce scalability for enterprise document processing. This study develops a layout-aware region compression method for document understanding. The proposed DocRegion-Slimmer first segments each page into semantic regions, including title blocks, paragraph zones, tables, key-value fields, signatures, seals, and background. It then compresses low-information regions with shared layout embeddings while preserving high-value regions such as numerical fields, table cells, dates, names, and approval marks. A region consistency loss is introduced to avoid losing cross-region dependencies between extracted entities. The experiments covered 128,000 scanned and digital-born pages from invoices, tax forms, purchase orders, insurance claims, contracts, bank statements, and clinical forms. The dataset contained 42 entity types, 16 table structures, and 3.7 million OCR text boxes. DocRegion-Slimmer reduced the average number of visual-layout tokens per page from 1,024 to 438. On entity extraction, micro-F1 changed only slightly from 91.6% to 91.1%, while memory consumption during inference decreased by 44.8%. For table understanding, cell-level F1 reached 88.4%, outperforming a simple patch-dropping baseline by 5.7 points. In long contract pages, the model reduced missing-field errors for dates and monetary values by 19.2% because key-value regions were preserved rather than compressed uniformly. Batch processing throughput increased from 2,900 to 5,180 pages per hour on one A100 GPU. The results indicate that layout-aware compression can improve the efficiency of transformer document understanding without sacrificing critical information extraction accuracy.

Keywords: Document understanding; layout analysis; region compression; information extraction; table recognition; transformer efficiency; enterprise automation

1. Introduction

Document understanding has become an important component of automated information processing in finance, insurance, healthcare, taxation, procurement, compliance, and contract management. Unlike plain-text analysis, document understanding must jointly interpret textual content, visual appearance, and spatial organization. A single page may contain paragraphs, tables, key-value fields, handwritten signatures, seals, checkboxes, logos, and other graphical elements, and the meaning of an item often depends on both its content and its location. Recent document intelligence models therefore combine textual, visual, and two-dimensional positional information within unified architectures. LayoutLMv2 introduced spatial-aware self-attention and joint text-image pre-training to improve document representation [1]. DocFormer further integrated visual, textual, and spatial features through a multimodal transformer architecture [2], while LayoutXLM extended layout-aware document modeling to multilingual scenarios [3]. Related research on complex AI systems has also shown that difficult computational tasks can benefit from structured decomposition and coordinated optimization among specialized components [4]. This broader design principle is relevant to document understanding because document pages are naturally composed of heterogeneous semantic regions that may require different levels of representation and computation. Subsequent document models, including LayoutLMv3, LiLT, BROS, and DiT, further improved multimodal pre-training, cross-lingual transfer, spatial encoding, and document image representation [5,6]. ERNIE-Layout additionally incorporated reading order and layout relationships to better model complex document structures [7,8].

The development of OCR-free and unified vision-language architectures has further expanded the scope of document understanding. Donut directly maps document images to structured outputs without relying on an external OCR engine, reducing error propagation from recognition pipelines [9,10]. UDOP unifies document understanding and generation tasks within a shared multimodal framework [11], while Pix2Struct learns visual language representations from screenshot-like inputs and structured visual content [12]. These approaches have improved the flexibility of document processing and reduced dependence on task-specific pipelines. However, stronger multimodal representation usually comes at the cost of increased computational complexity. High-resolution pages may produce large numbers of image patches, while OCR-based systems may generate hundreds or thousands of text boxes and associated layout tokens for a single document. When long contracts, dense financial statements, clinical forms, or multi-column reports are processed, the number of tokens grows rapidly. Since transformer attention and multimodal fusion operations are repeatedly applied across these tokens, memory consumption and inference latency can become substantial. This issue is particularly important in enterprise applications, where document systems are expected to process large batches of pages under practical constraints on GPU memory, processing time, and operating cost. Document structure further increases the difficulty of efficient representation. Many extraction tasks depend not only on recognizing individual words but also on preserving relationships among spatially

separated elements. PubTables-1M demonstrated that table understanding requires reliable identification of rows, columns, cells, and structural relationships rather than simple text recognition [13,14]. TSRFormer further modeled table separators and cell relations using transformer-based representations [15,16]. GeoLayoutLM showed that geometric relationships between entities are important for document information extraction, especially when semantically related fields are located in different areas of a page [17,18]. These findings indicate that aggressive token reduction cannot be performed solely according to local appearance or isolated token importance. A small numerical field may be essential for an invoice total, a short date may determine contract validity, and a compact signature or approval mark may carry greater business significance than a long paragraph of standard text. Removing such regions may reduce computational cost while causing disproportionate degradation in extraction quality. More recent large document models have extended layout-aware reasoning to broader document understanding tasks. DocLLM introduces layout information into large language model reasoning for visually rich documents [19], while LayoutLLM integrates document layout and multimodal information for generalized document understanding [20]. These models increase the ability to interpret complex pages, but their computational requirements also grow with page resolution, sequence length, and the number of OCR elements. The problem becomes more pronounced for enterprise documents because information density is highly uneven across a page. Large blank margins, decorative elements, repeated headers and footers, page numbers, boilerplate clauses, and standardized instructions may occupy substantial visual and token space while contributing relatively little to the final extraction result. In contrast, names, monetary values, policy numbers, tax identifiers, dates, table cells, signatures, seals, and approval fields may occupy only small regions but determine the core output of the document-processing system. Treating these regions with the same computational resolution ignores the semantic imbalance inherent in document layouts. Existing efficiency strategies do not fully address this imbalance. Some methods reduce input image resolution, shorten sequences, merge tokens, or remove tokens according to general attention or importance scores. Although these approaches can lower computational cost, they may damage fine-grained document information. Resolution reduction can make small characters, decimal values, signatures, or seals difficult to distinguish, while global token pruning may remove short but semantically critical fields. Token-level importance measures may also ignore relationships between distant elements. For example, a field label and its corresponding value may appear in separate regions, and a table header may determine the interpretation of many cells below it. Compressing these regions independently can therefore break structural dependencies even when the individual tokens appear unimportant. In addition, document layouts vary considerably across document types. A region that is redundant in a contract may be essential in an insurance claim or tax form. Efficient document processing consequently requires a mechanism that considers semantic region type, information value, and inter-region relationships rather than applying uniform compression to all tokens [21,22]. Evaluation practices also leave an important

gap between public benchmarks and real enterprise document collections. Existing studies commonly evaluate forms, receipts, tables, invoices, or document question answering on separate public datasets [23,24]. These benchmarks are valuable for controlled comparison, but production document repositories often contain heterogeneous document types within the same processing pipeline. A single enterprise system may need to handle invoices, contracts, bank statements, tax documents, purchase orders, insurance claims, and clinical records. These documents differ substantially in page density, layout complexity, table frequency, visual quality, and the distribution of high-value fields. Scanned pages may contain noise, skew, stamps, or handwriting, whereas digital-born PDFs may contain highly regular text and table structures. As a result, efficiency methods optimized for a single benchmark or document type may not generalize well to large mixed document collections. A practical compression strategy should therefore reduce redundant computation while remaining sensitive to document semantics and preserving the structural relationships required for reliable extraction.

This study proposes DocRegion-Slimmer, a layout-aware region compression framework for efficient transformer-based document understanding. Rather than treating a page as a uniform sequence of visual and textual tokens, DocRegion-Slimmer represents the document as a collection of semantic regions with different information densities and task relevance. Each page is divided into regions such as titles, paragraphs, tables, key-value fields, signatures, seals, and background areas. Regions with low information density can be represented using compressed or shared layout embeddings, whereas numerical fields, dates, personal and organizational names, monetary values, table cells, signatures, and approval marks are retained at higher representational detail. A region consistency loss is introduced to preserve semantic and spatial relationships between associated regions after compression, reducing the risk that efficiency gains destroy important field-to-field or table-level dependencies. The experimental study is conducted on 128,000 scanned and digital-born pages collected from invoices, tax forms, purchase orders, insurance claims, contracts, bank statements, and clinical forms. The dataset contains 42 entity types, 16 table structures, and approximately 3.7 million OCR text boxes, providing a heterogeneous setting for evaluating document efficiency under realistic workloads. The study examines whether semantic region compression can substantially reduce token count, GPU memory consumption, and document processing time while maintaining entity extraction and table structure accuracy. It also evaluates whether explicitly preserving high-value regions can reduce missing-field errors in long and visually dense documents and whether the resulting reduction in computational demand can improve batch-processing throughput. Through these experiments, the study aims to establish whether document compression should be guided by semantic importance and layout structure rather than uniform token-level reduction. The broader significance of DocRegion-Slimmer is to provide a more practical balance between extraction reliability and computational efficiency, enabling multimodal document transformers to process

large and heterogeneous enterprise document collections without sacrificing the small but business-critical regions on which downstream decisions often depend.

2. Materials and Methods

2.1 Document Samples and Data Composition

The study used 128,000 scanned and digital-born pages from invoices, tax forms, purchase orders, insurance claims, contracts, bank statements, and clinical forms. The documents contained 42 entity types, 16 table structures, and about 3.7 million OCR text boxes. The samples covered single-page forms, long contracts, dense tables, and pages with signatures or seals. Scanned documents included differences in resolution, rotation, background noise, and print quality. Digital-born files usually had clearer text and page layouts. Pages with unreadable content, duplicate records, or incomplete labels were removed. The remaining documents were divided into training, validation, and test sets by data source to reduce overlap between related pages.

2.2 Experimental Design and Control Models

DocRegion-Slimmer was compared with a full-region transformer and a simple patch-dropping model under the same training and test settings. The full-region model kept all visual-layout tokens and served as the main baseline. The patch-dropping model removed low-score patches without using semantic region labels. DocRegion-Slimmer divided each page into titles, paragraphs, tables, key-value fields, signatures, seals, and background regions. Dates, names, numerical fields, table cells, and approval marks were kept at higher detail. Repeated headers, blank margins, and other low-information areas were compressed. Ablation tests removed semantic region labels, high-value region protection, or region consistency loss. All models used the same document splits, OCR input, image size, training schedule, and evaluation rules.

2.3 Measurement Methods and Quality Control

Model performance was measured using entity-level micro-F1, table cell-level F1, missing-field rate, token count, inference memory, and page-processing throughput. Entity extraction was counted as correct only when both the entity class and extracted content matched the reference label. Table evaluation compared predicted cells and their content with labeled table structures. Missing-field errors were calculated separately for dates and monetary values on long contract pages. GPU memory use was measured during inference with a fixed batch size. Throughput was recorded as the number of pages processed per hour on one NVIDIA A100 GPU. Warm-up runs were completed before timing tests. OCR boxes and region labels were checked for invalid coordinates, repeated text, and missing labels. Pages with unclear table boundaries or uncertain entity links were reviewed before final testing.

2.4 Data Processing and Model Formulation

Each page was converted into OCR text boxes, visual features, two-dimensional positions, and semantic region labels. Region importance was calculated before compression. For region r , the retention score was defined as:

$$S_r = \alpha I_r + \beta E_r + \gamma R_r$$

where I_r is the information score of region r , E_r is the entity importance score, and R_r is the relation score between region r and linked regions. The parameters α , β , and γ control the weights of the three terms. Regions with high S_r values kept detailed tokens, while low-score regions were represented by shared layout embeddings. To keep relations between compressed and retained regions, the total training loss was defined as:

$$L = L_{\text{task}} + \lambda L_{\text{cons}}$$

where L_{task} is the loss for entity and table prediction, L_{cons} is the region consistency loss, and λ controls the weight of the consistency term. This term limits changes in region relations after compression.

2.5 Model Training and Statistical Analysis

All models were trained with the same optimizer, batch size, learning-rate schedule, and stopping rule. Model selection was based on validation micro-F1, and the test set was used only for final evaluation. Results were reported for the full dataset and for individual document types, including contracts, invoices, and table-heavy forms. Entity F1, table F1, token count, memory use, and processing throughput were compared across models. Long-document results were analyzed separately for dates and monetary values. The main experiments were repeated in independent runs, and mean values were reported. Paired tests were used when two models were evaluated on the same pages. Statistical significance was set at $p < 0.05$.

3. Results and Discussion

3.1 Entity Extraction After Region Compression

DocRegion-Slimmer reduced the average number of visual-layout tokens from 1,024 to 438 per page, while entity-level micro-F1 decreased only from 91.6% to 91.1%. This means that a 57.2% reduction in token count caused only a 0.5 percentage-point drop in extraction accuracy. As shown in Fig. 1, most removed tokens came from blank margins, repeated headers, and standard text regions, while names, dates, amounts, signatures, and key-value fields were kept at higher detail. This result shows that document regions do not need the same level of processing. LayoutLMv3 also confirmed the value of text, image, and layout information for document understanding, but it does not reduce regions according to their task importance [25,26]. Recent token-pruning studies have also shown that low-value document patches can be removed with limited accuracy loss. However, most of these methods operate at patch level rather than semantic region level.

3.2 Table Structure and Cell Information

DocRegion-Slimmer achieved a cell-level F1-score of 88.4% for table understanding, which was 5.7 percentage points higher than the simple patch-dropping baseline. This result shows that direct patch removal can damage row, column, and cell relations. Table cells often depend on nearby headers, labels, and neighboring values. Therefore, local visual importance alone is not enough for selecting tokens. PubTables-1M also showed that table recognition depends on both cell content and table structure [27,28]. In DocRegion-Slimmer, the region consistency loss helps keep these relations after compression. This is useful for tables with merged cells, repeated headers, and small numerical fields. The results in Fig. 1 show that region-based compression can better preserve structured document information than simple patch removal.

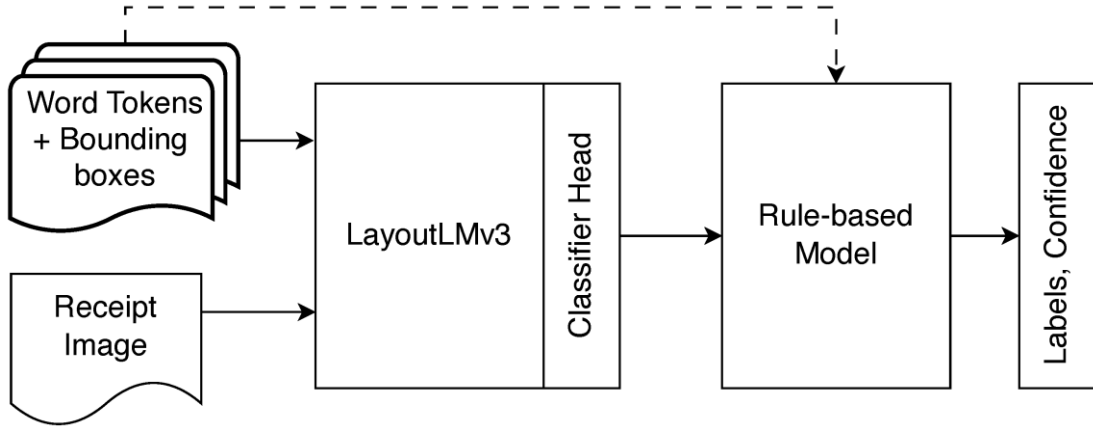


Fig. 1. Entity extraction and table recognition results of DocRegion-Slimmer and the baseline models.

3.3 Memory Use and Processing Throughput

Inference memory decreased by 44.8% after region compression, while throughput increased from 2,900 to 5,180 pages per hour on one A100 GPU. This corresponds to a throughput increase of about 78.6%, as shown in Fig. 2. The main reason is the lower token count, which reduces the cost of transformer attention. Long pages, large OCR box numbers, and complex layouts usually increase both memory use and processing time. DocRegion-Slimmer reduces this cost without changing the main transformer structure. Compared with general patch pruning, the method keeps more computation for important document regions and removes more repeated content from low-value areas. This balance is useful for enterprise systems that need to process large document batches under fixed GPU memory.

3.4 Long-Page Field Preservation

On long contract pages, missing-field errors for dates and monetary values decreased by 19.2% compared with uniform compression. These fields are often small and may be far from their related labels. They can therefore be removed when token selection depends only on local visual scores. DocRegion-Slimmer keeps these fields and preserves their links with nearby labels, tables,

and approval regions. GeoLayoutLM also showed that geometric relations are important for relation extraction in visually rich documents [29,30]. The current results follow the same idea but apply it to document compression. Important fields and their relations should be kept during token reduction. As shown in Fig. 2, this can be achieved together with lower memory use and higher processing speed. However, the current tests cover seven document types, and other industries may contain different page structures. Further studies should include more document sources and longer multi-page files.

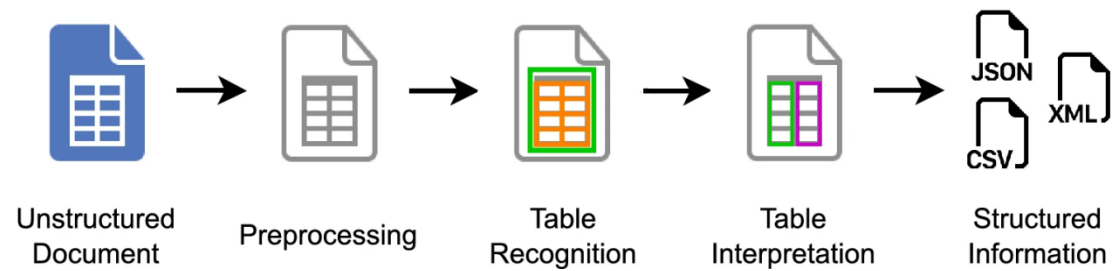


Fig. 2. Memory use, processing speed, and missing-field errors before and after region compression.

4. Conclusion

This study developed DocRegion-Slimmer to reduce the cost of transformer-based document understanding. The method reduced the average number of visual-layout tokens from 1,024 to 438 per page, while entity-level micro-F1 changed only from 91.6% to 91.1%. Inference memory decreased by 44.8%, and processing throughput increased from 2,900 to 5,180 pages per hour on one A100 GPU. For table recognition, cell-level F1 reached 88.4%, which was 5.7 percentage points higher than that of the simple patch-dropping model. Missing-field errors for dates and monetary values on long contract pages were also reduced by 19.2%. These results show that region-based compression can lower memory and processing cost while keeping important document information. DocRegion-Slimmer keeps key fields and their layout links while compressing repeated or low-value regions. This approach is suitable for large-scale processing of invoices, contracts, bank statements, insurance forms, and other business documents. However, the current study includes only seven document types, and throughput was mainly tested on one GPU platform. Future work should cover more document layouts, longer multi-page files, multilingual data, and different hardware platforms.

References

1. Lvov, D., Smirnov, I., Volokha, V., Laushkina, A., & Boukhanovsky, A. (2026). Enhancing Technical Question Answering Quality Through Multimodal Document Segmentation. IEEE Access.
2. Zhang, Z. (2026). A Study on the Impact of Cost Presentation on Decision-making Bias and Cancellation Behavior in Online Subscriptions. Available at SSRN 7349839.

3. Du, Y., Liu, Q., Dong, Y., & Chen, Z. (2026). Cross-Domain Transfer and Few-Shot Recognition of Defect Images in Complex Industrial Settings.
4. Zhang, Z., Wang, J., Li, Z., Wang, Y., & Zheng, J. (2025). AnnCoder: A multi-agent-based code generation and optimization model. *Symmetry*, 17(7), 1087.
5. Singh, A., Shah, M., Sampath, D., & Jarvis AI Team. (2025). Spatial ModernBERT: Spatial-Aware Transformer for Table and Key-Value Extraction in Financial Documents at Scale. arXiv preprint arXiv:2507.08865.
6. Xiong, W., Guo, Y., & Zeng, Y. (2026). A Study on the Energy Efficiency of MEP Systems and Coordinated Dispatch Mechanisms for Multi-energy Systems in High-density Urban Buildings. Available at SSRN 7120518.
7. Liang, S., Hsu, K. H., & Liu, Z. (2026). Rolling Scheduling and Capacity Balancing for Server Assembly Under Engineering Change Disturbances.
8. Bao, Y., Qiu, Y., & Wang, H. (2026). FLARE: A Real-Time Framework for Detecting Malicious Account Activities at Internet Scale. Available at SSRN 7185659.
9. Trivedi, D., & Agarwal, R. (2026). Improving Optical Character Recognition (OCR) Reliability on Noisy Document Images Using Hybrid Deep Learning and Post-Processing Techniques. *Cureus Journals*, 3(1).
10. Xu, D., Chen, H., & Gui, H. (2026, March). Unified Online Estimation Method for SOC, SOH, and Power Capacity Considering Safety Boundary Consistency in Battery Management Systems. In 2026 International Conference on Electrical, Control and Information Technology (ECITech) (pp. 133-137). IEEE.
11. Zhao, Z., & Welsch, R. E. (2026). Point-in-Time Financial RAG with Frozen LLMs and Market-Feedback Adaptive Retrieval. arXiv preprint arXiv:2605.31201.
12. Yuan, Y., Xu, T., Yin, J., & Huang, J. (2026). Modeling Loading Anomalies and Identifying Root Causes in Media Consumption Workflows on Large Social Media Platforms. Available at SSRN 7393318.
13. Prajapati, A., & Maurya, S. (2025, October). Table structure recognition: a review of current approaches and future opportunities. In International Conference on Advancing Technology in Engineering and Science (ICATES 2025) (Vol. 2025, pp. 86-91). IET.
14. Yuan, Y., Yin, J., Huang, J., & Xu, T. (2026). Weakly Supervised Anomaly Detection and Privacy Risk Scoring in Community Message Threads.
15. Hong, Z., Liang, S., Xu, T., & Chen, H. (2026). A Study on Failover Verification and Recovery Objective Prediction for Cross-Region Cloud Services.
16. Bao, Y., & Wang, H. (2026). Orion: A High-Throughput, Fault-Tolerant Event Routing Architecture for Hyperscale Data Streams. Fault-Tolerant Event Routing Architecture for Hyperscale Data Streams (January 01, 2026).

17. Giovannini, S., & Marinai, S. (2025, October). A survey on reading order, table of contents, and structure extraction in document analysis. In 2025 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW) (pp. 7635-7644). IEEE.
18. Yang, Q., & Du, Y. (2026). Predicting Rollback Risks and Controlling Gradual Rollout Traffic for Online Ranking Feature Deployments.
19. Wu, J., Zhang, J., Wu, D., & Peng, Y. (2026). Visual Transformation Mechanisms for Integrating Digital Cultural Heritage Resources into Junior High School Art Curricula.
20. Shihab, I. F., Akter, S., & Sharma, A. (2026). LLM-Guided Probabilistic Fusion for Label-Efficient Document Layout Analysis. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 3574-3583).
21. Huang, J., Xu, T., Yin, J., & Yang, J. (2026). Data Quality Monitoring and Intelligent Scheduling Optimization in Financial Data Workflow Automation. Available at SSRN 7179338.
22. Yao, G., Zheng, J., Wang, Z., Zhang, W., Han, R., Zhao, C., ... & Liu, R. (2026, March). V-pruner: A fast and globally-informed token pruning framework for vision transformer. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 40, No. 40, pp. 34396-34404).
23. Abdalla, M., Kasem, M. S., Mahmoud, M., Yagoub, B., Senussi, M. F., Abdallah, A., ... & Kang, H. S. (2025). Receiptqa: A question-answering dataset for receipt understanding. *Mathematics*, 13(11), 1760.
24. Gu, X. (2026). Identifying Causal Effects and Analyzing Heterogeneity of User Growth Interventions on Digital Platforms: Evidence from Large-Scale Behavioral Data. Available at SSRN 6809181.
25. Zhao, J., Fan, J., & Li, L. (2026). A Study on an Explainable Causal-Enhanced LLM Agent for Predicting the Forming Quality of Automotive Component Materials.
26. Minouei, M. (2026). Structural Information Extraction from Document Images: Addressing Challenges in Layout Analysis, Table Detection, and Classification (Doctoral dissertation, Rheinland-Pfälzische Technische Universität Kaiserslautern-Landau).
27. You, S. (2026). Verifiable Audit Mechanisms in AI Compliance Automation: Scalability. Available at SSRN 6547458.
28. Jiao, Y., Shi, T., Zhao, B., & Wang, A. (2026). Retrieval-Guided Structured Reasoning and Interpretable Representation Learning for Large-Scale Video–Language Models.
29. Liang, S., Du, Y., Chen, W., & Liu, Z. (2026). Order Allocation and Emergency Transportation Decisions Amid Multi-Source Procurement Disruptions.
30. Giovannini, S., & Marinai, S. (2025, October). A survey on reading order, table of contents, and structure extraction in document analysis. In 2025 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW) (pp. 7635-7644). IEEE.