INDEX-PRESERVING LIGHTWEIGHT TOKEN PRUNING FOR EFFICIENT DOCUMENT UNDERSTANDING IN VISION-LANGUAGE MODELS

Jaemin Son **Sujin Choi** **Inyong Yun**
Hana Institute of Technology, Hana TI
{woalsdnd, sujin_choi, iyyun}@hanafn.com

ABSTRACT

Recent progress in vision-language models (VLMs) has led to strong accuracy on document understanding tasks such as parsing and key information extraction, but processing high-resolution document images remains computationally expensive. We propose a lightweight pre-encoder token pruning framework that removes non-informative background patches using a binary text-region classifier with a max-pooling refinement step. The framework preserves token indices to maintain the spatial correspondence required for layout-sensitive recognition. Experiments on real-world document benchmarks show 40–60% FLOPs reduction while maintaining comparable accuracy.

1 INTRODUCTION

Recent advancements in vision-language models (VLMs) (Team et al., 2024; Bai et al., 2025; Grattafiori et al., 2024; Touvron et al., 2023; Achiam et al., 2023; Lu et al., 2024; Beyer et al., 2024) have broadened their applications in document understanding tasks, including document layout parsing (Yang et al., 2024), key information extraction (Park et al., 2019), and visual question answering (Mathew et al., 2021). By jointly modeling text and vision features, these models achieve strong performance with minimal task-specific tuning. However, their high computational cost remains a critical obstacle to practical deployment.

To address this challenge, token pruning and merging techniques have been proposed in the vision domain (Kim et al., 2024; Rao et al., 2021; Bolya et al., 2022; Zheng et al., 2023; Kong et al., 2022). These methods reduce redundant visual tokens, improving inference efficiency with minimal accuracy degradation. Yet, such strategies are underexplored in document understanding tasks, where large background regions allow aggressive filtering without affecting recognition performance.

In this work, we introduce a lightweight and effective token pruning method tailored for document understanding, aiming to reduce unnecessary computation while preserving accuracy[1]. In summary, the key findings are as follows:

- **Lightweight Patch Classification** effectively filters out background regions in document images, enabling substantial computational savings.
- **Preserving the original indices** of the remaining tokens after pruning has a significant impact on performance in document understanding tasks.

2 RELATED WORK

Existing methods for ViTs aim to improve efficiency by removing or merging uninformative tokens in classification and detection tasks. DynamicViT (Rao et al., 2021) inserts lightweight prediction modules between transformer blocks to dynamically determine which tokens to retain. Focus-DETR (Zheng et al., 2023) selects foreground-relevant tokens from multi-scale feature maps for object detection, pruning away background regions early. SPViT (Kong et al., 2022) compresses

[1] https://github.com/jaeminSon/index-preserving-lightweight-token-pruning

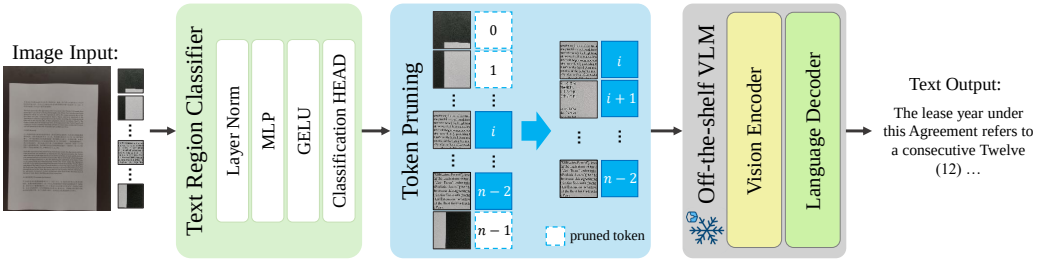


Figure 1: **Illustration of the Index-Preserving Lightweight Token Pruning framework.** The proposed framework consists of three components: a binary text-region classifier (green), index-preserving token pruning (blue) and a frozen off-the-shelf VLM (gray). Text-region patches are selected and fed into an off-the-shelf VLM.

uninformative tokens into a single package token to retain global context while reducing count. ToMe (Bolya et al., 2022) merges spatially and semantically similar tokens using cosine similarity in each transformer block, while Token Fusion (Kim et al., 2024) modifies this by weighted merging that preserves magnitude to retain more informative content.

In document understanding, DocKylin (Zhang et al., 2025) applies Sobel filtering to remove white backgrounds and merges tokens in a language embedding space. Unlike the prior work, our method prunes before the vision encoder to maximize the effectiveness of pruning. Additionally, we retain the spatial indices of the remaining tokens, which is crucial for processing pruned tokens within the vision encoder.

3 METHOD

We propose an index-preserving pruning method that discards background patches at the earliest stage, before any computation occurs in a VLM. See Figure 1 for the overview.

3.1 LIGHTWEIGHT TEXT-REGION CLASSIFIER

A lightweight binary classifier operates on image patches to predict whether each contains text (foreground) or can be discarded (background). Unlike existing methods, our approach applies pruning prior to any visual encoding or language decoding. This eliminates non-informative regions early, reducing the input size for computationally expensive components.

3.2 TOKEN PRUNING WITH INDEX PRESERVATION

To maintain spatial coherence, we preserve the original indices of the selected patches after pruning. Each patch is paired with its index, and those recognized as foreground text are forwarded to the VLM along with their original indices. Maintaining the original indices is crucial for document understanding tasks, where positional information encodes critical semantics such as text content and spatial layout. Without index preservation, the decoder receives irregularly stitched patches with misaligned positional information, which severely degrades text recognition.

3.3 FOREGROUND REFINEMENT WITH MAX-POOLING

Patch-level classification often produces fragmented foreground masks, where parts of text regions are missed. This is because the classifier does not explicitly account for spatial continuity.

To address this, we apply a max-pooling operation to the binary mask, which enhances spatial continuity and recovers adjacent missed regions. This refinement significantly improves the coherence of selected text areas, as illustrated in Figure 2.

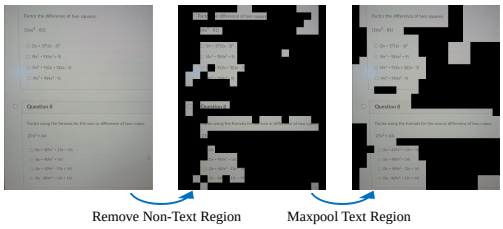
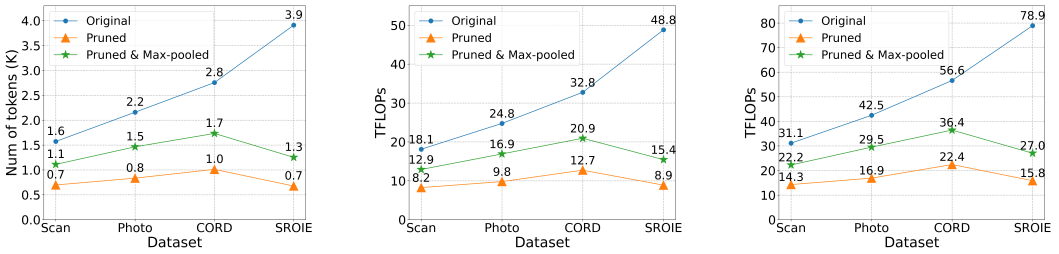


Figure 2: **Text recovery with max-pooling.**



(a) Visual token count. (b) TFLOPs for Qwen2.5-VL 3B. (c) TFLOPs for Qwen2.5-VL 7B.

Figure 3: **Visual token and TFLOPs statistics**. Metrics are averaged over all images in each dataset. TFLOPs are counted end-to-end including the computation of the text-region classifier.

4 Experiments

4.1 Evaluation setup

We evaluated our pruning approach (Section A.1) with Qwen2.5-VL on CC-OCR dataset (Yang et al., 2024). We retained the evaluation logic in the official implementations (Alibaba Research, 2024) and focused on English documents with ample marginal background — particularly, document parsing tasks (LaTeX formulas and HTML tables) and key information extraction tasks (receipts).

Two images in the CORD subset were fully pruned, meaning all patches were removed by the classifier. We excluded them from the FLOPs computation and assigned 0 score for F1 and the normalized TED accuracy.

4.2 Token and FLOPs reduction

As shown in Figure 3(a), token pruning alone reduced the number of visual tokens by an average of 65.7% across datasets, while the combination of pruning and max-pooling yielded a reduction of 41.6%. Max-pooling expands the foreground mask, retaining additional neighboring patches. Receipt-style images often have abundant backgrounds and smaller text regions compared to document images. This allows for more aggressive token pruning in CORD and SROIE datasets.

We measured end-to-end Tera Floating-point Operations (TFLOPs) using the Calflops Python module (Ye, 2023). FLOPs were computed over the forward pass of the complete VLM pipeline and averaged across all images in each dataset. Due to memory constraints, we present results for the 3B and 7B variants only, as shown in Figure 3(b) and 3(c), respectively.

Our pruning approach consistently achieved over 60% FLOPs reduction. On the SROIE dataset, FLOPs dropped by ∼80%. When combined with max-pooling, FLOPs are reduced by 40–60% on all datasets.

4.3 Impact of pruning and max-pooling

As shown in Table 1, applying text-region pruning alone led to a noticeable degradation in model performance. In document parsing tasks, performance dropped up to ∼17%p in ANLS. Similarly, key information extraction datasets showed severe decreases in F1-score, up to ∼31%p, underscoring the sensitivity of downstream models to missing text regions.

Introducing max-pooling after pruning mitigated much of this degradation. In document parsing tasks, ANLS scores remained largely unaffected after max-pooling, showing at most ∼3%p decrease. In some cases, performance even improved, possibly because the model focused more on informative regions. This indicates that max-pooling can effectively recover some of the missed regions discarded by the classifier.

Similar patterns emerged in key information extraction tasks. On SROIE, F1-score and the accuracy remained nearly unchanged after applying pruning and max-pooling. In CORD, minor decreases of up to ∼5%p were observed, which is a significant improvement over the pruning-only results.

Params	Method	Document Parsing		Key Information Extraction			
		Scan	Photo	CORD		SROIE	
		ANLS	ANLS	F1	Acc	F1	Acc
3B	Original	62.4	73.7	87.2	94.7	88.7	97.5
	Pruned (Δ)	-13.6	-13.0	-30.8	-17.1	-15.4	-2.7
	Pruned & Max-pooled (Δ)	-0.6	-2.7	-4.2	-4.4	-0.8	+0.3
7B	Original	64.7	69.9	89.5	96.3	90.7	98.1
	Pruned (Δ)	-11.0	-10.2	-24.7	-15.9	-13.7	-1.9
	Pruned & Max-pooled (Δ)	-2.7	-1.3	-5.3	-4.8	-0.4	+0.1
32B	Original	60.9	67.5	87.0	95.2	88.5	97.5
	Pruned (Δ)	-9.5	-16.6	-26.0	-14.8	-14.4	-2.3
	Pruned & Max-pooled (Δ)	+1.7	-1.9	-3.3	-4.0	-0.0	+0.1
72B	Original	67.7	70.0	92.8	97.6	91.6	98.6
	Pruned (Δ)	-7.0	-4.5	-24.3	-13.5	-13.8	-1.8
	Pruned & Max-pooled (Δ)	-0.4	+1.5	-3.8	-4.0	-0.5	-0.1

Table 1: **Performance degradation due to pruning and the effect of max-pooling.** Evaluation results are shown for the original Qwen2.5-VL models on document understanding tasks in the CC-OCR dataset. Relative differences (Δ) are shown for the pruned variants with respect to the original model of the same parameter size. Acc represents normalized TED accuracy.

4.4 Comparison with existing methods

We integrated official implementations of existing methods in Qwen2.5-VL-Instruct-3B. For ToMe (Bolya et al., 2022), we merged 1% of tokens at each ViT layer. For DocKylin (Zhang et al., 2025), we integrated only Dynamic Token Slimming (DTS) excluding Adaptive Pixel Slimming, which produces a new compact image.

Method	Doc. Parsing		Key Info. Extract.	
	Scan	Photo	CORD	SROIE
Original	62.4	73.7	87.2 (94.7)	88.7 (97.5)
ToME	8.8	11.1	6.0 (13.5)	0.0 (9.9)
DocKylin	34.3	47.7	73.1 (84.1)	69.6 (83.5)
Ours	**61.8**	**71.0**	**83.0 (90.3)**	**87.9 (97.8)**

Table 2: **Comparison with existing methods.** Metrics are ANLS for document parsing and F1 and normalized TED accuracy for key information extraction.

As shown in Table 2, ToMe (Bolya et al., 2022) showed low ANLS and accuracy. As it shuffles and rearranges tokens at each merge step, the index structure collapses leading to poor text recognition. This suggests that, unlike classification tasks, text recognition requires proper token indexing to attain reasonable performance. Also, DocKylin (Zhang et al., 2025) with DTS showed limited performance. Its merging strategy treats tokens highly correlated with others as background, under the assumption that backgrounds share similar visual patterns. When this assumption does not hold, the method becomes less effective.

4.5 Ablation on indexing strategy

We tested alternative indexing strategies using Qwen2.5-VL-Instruct-3B: setting all indices to zero (constant), assigning indices randomly (random), or incrementally assigning integers from 0 to $L-1$ (ordered).

Indexing Strategies	Doc. Parsing		Key Info. Extract.	
	Scan	Photo	CORD	SROIE
Constant	9.1	5.8	3.5 (19.7)	0.0 (10.9)
Random	16.0	13.7	8.5 (27.9)	0.2 (11.7)
Ordered	36.2	49.2	38.8 (58.0)	40.7 (65.1)
Preserved	**61.8**	**71.0**	**83.0 (90.3)**	**87.9 (97.8)**

Table 3: **Ablation of indexing strategies.** Metrics are ANLS for document parsing and F1 and normalized TED accuracy for key information extraction.

Table 3 shows that constant and random indexing perform poorly, and ordered indexing remains far below preserving the original indices, indicating that index preservation is essential for maintaining spatial correspondence.

5 Conclusion

We proposed a straightforward yet effective token pruning strategy tailored for vision-language models in document understanding tasks. Across multiple document text datasets, pruning and max-pooling substantially reduce computation with only minor performance degradation. These findings demonstrate the promise of early-stage pruning for efficient and effective document analysis.

6 ACKNOWLEDGMENTS

This research was supported by Hana Financial Group.

REFERENCES

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Alibaba Research. CC-OCR codebase. `https://github.com/AlibabaResearch/AdvancedLiterateMachinery/tree/main/Benchmarks/CC-OCR`, 2024. Accessed: 2025-09-16.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

Lucas Beyer, Andreas Steiner, André Susano Pinto, Alexander Kolesnikov, Xiao Wang, Daniel Salz, Maxim Neumann, Ibrahim Alabdulmohsin, Michael Tschannen, Emanuele Bugliarello, et al. Paligemma: A versatile 3b vlm for transfer. *arXiv preprint arXiv:2407.07726*, 2024.

Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your vit but faster. *arXiv preprint arXiv:2210.09461*, 2022.

Tri Dao. Flashattention-2: Faster attention with better parallelism and work partitioning. *arXiv preprint arXiv:2307.08691*, 2023.

Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers. *arXiv preprint arXiv:2309.16588*, 2023.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Minchul Kim, Shangqian Gao, Yen-Chang Hsu, Yilin Shen, and Hongxia Jin. Token fusion: Bridging the gap between token pruning and token merging. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 1383–1392, 2024.

Zhenglun Kong, Peiyan Dong, Xiaolong Ma, Xin Meng, Wei Niu, Mengshu Sun, Xuan Shen, Geng Yuan, Bin Ren, Hao Tang, et al. Spvit: Enabling faster vision transformers via latency-aware soft token pruning. In *European conference on computer vision*, pp. 620–640. Springer, 2022.

Yi Li, Zhe Wu, Shuang Zhao, Xian Wu, Yehong Kuang, Yangtian Yan, Shen Ge, Kai Wang, Wei Fan, Xiang Chen, et al. Psenet: Psoriasis severity evaluation network. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 800–807, 2020.

Haoyu Lu, Wen Liu, Bo Zhang, Bingxuan Wang, Kai Dong, Bo Liu, Jingxiang Sun, Tongzheng Ren, Zhuoshu Li, Hao Yang, et al. Deepseek-vl: towards real-world vision-language understanding. *arXiv preprint arXiv:2403.05525*, 2024.

Minesh Mathew, Dimosthenis Karatzas, and CV Jawahar. Docvqa: A dataset for vqa on document images. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pp. 2200–2209, 2021.

Seunghyun Park, Seung Shin, Bado Lee, Junyeop Lee, Jaeheung Surh, Minjoon Seo, and Hwalsuk Lee. Cord: a consolidated receipt dataset for post-ocr parsing. In *Workshop on Document Intelligence at NeurIPS 2019*, 2019.

Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. Dynamicvit: Efficient vision transformers with dynamic token sparsification. *Advances in neural information processing systems*, 34:13937–13949, 2021.

Andreas Steiner, André Susano Pinto, Michael Tschannen, Daniel Keysers, Xiao Wang, Yonatan Bitton, Alexey Gritsenko, Matthias Minderer, Anthony Sherbondy, Shangbang Long, et al. Paligemma 2: A family of versatile vlms for transfer. *arXiv preprint arXiv:2412.03555*, 2024.

Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

Thomas Wolf et al. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 38–45, 2020.

Zhibo Yang, Jun Tang, Zhaohai Li, Pengfei Wang, Jianqiang Wan, Humen Zhong, Xuejing Liu, Mingkun Yang, Peng Wang, Yuliang Liu, et al. Cc-ocr: A comprehensive and challenging ocr benchmark for evaluating large multimodal models in literacy. *arXiv preprint arXiv:2412.02210*, 2024.

Xiaoju Ye. Calflops: A flops and params calculate tool for neural networks in pytorch framework. `https://github.com/MrYxJ/calculate-flops.pytorch`, 2023. Accessed: 2025-09-16.

Jiaxin Zhang, Wentao Yang, Songxuan Lai, Zecheng Xie, and Lianwen Jin. Dockylin: A large multimodal model for visual document understanding with efficient visual slimming. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 9923–9932, 2025.

Dehua Zheng, Wenhui Dong, Hailin Hu, Xinghao Chen, and Yunhe Wang. Less is more: Focus attention for efficient detr. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 6674–6683, 2023.

A APPENDIX

A.1 IMPLEMENTATION DETAILS

We implemented our method using Qwen2.5-VL-Instruct (Bai et al., 2025) of various sizes, available via Hugging Face Transformers module (Wolf et al., 2020). Qwen2.5-VL models accept variable-length image tokens without resizing input images, enabling seamless integration of our index-preserving pruning. This flexibility is typically limited in fixed-length models like PaliGemma (Steiner et al., 2024). Furthermore, unlike DINOv2 (Darcet et al., 2023), Qwen2.5-VL does not rely on register tokens in transformer layers within the vision encoder, which are prone to performance degradation when the number of tokens is reduced.

After training the classifier with binary cross entropy loss, we prepended it to the vision encoder. Image patches with classifier logits greater than zero are retained as text regions and max-pooled by 3×3. Then, the indices of the selected tokens were preserved and subsequently passed to the vision encoder and the language decoder. We experimented with Bfloat16 precision and flash-attention-2 (Dao, 2023). We set the maximum generation length to 2048 tokens, which we found sufficient in our experimental settings.

Note that our lightweight classifier contains a minimal number of parameters, enabling fine-tuning even with minimal resources.

A.2 PATCH CLASSIFIER DATASET

We sampled 800 OCR document images from AI-Hub[2]. Using PSENet (Li et al., 2020), we extracted text bounding boxes and assigned binary labels to square image patches: foreground (text) if the patch overlapped any bounding box, and background (non-text) otherwise. Based on ablation results (Section A.3), we used patch size 28 for all experiments.

A.3 ABLATION ON PATCH SIZE

Square image patches were generated with side lengths of 14, 28, 56, and 112 pixels. Each patch-size-specific dataset contains approximately 99,600 training samples and 400 validation samples, with a balanced split between foreground and background. Ambiguous validation images were manually removed. We fixed the linear projection dimension to 256 and evaluated the best-performing weights for each patch size.

Patch Size	Params(K)	AP
14	51	0.96
28	**203**	**0.99**
56	810	0.98
112	3000	0.98

Table 4: Average Precision across patch sizes.

As shown in Table 4, models with patch sizes of 28 and above performed similarly, while the smallest size of 14 showed degraded performance. This can be explained by character sizes in documents. An A4 page scanned at 300 DPI (Dots Per Inch) produces a 2481×3507 pixel image. A 12-point font amounts to $\sim$50 pixels and has effective character heights of 30–35 pixels, considering spacing. A patch size of 14 is too small to capture complete character structures and may fail to distinguish texts from noise.

[2]This research used datasets from The Open AI Dataset Project (AI-Hub, S. Korea). All data information can be accessed through AI-Hub (`https://www.aihub.or.kr`)