

Un-LOCC: Universal Lossy Optical Context Compression for Vision-Based Language Models

MaxDevv

January 2026

Abstract

The quadratic computational complexity of the Transformer attention mechanism [11, 10] makes large context LLM inferences prohibitively costly. This project, inspired by the research of DeepSeek-OCR [12], investigates a practical method for LLM context compression by rendering text as an image. The core hypothesis is that an image, which has a fixed token cost for a Vision-Language Model (VLM), can serve as a highly compressed representation of a much larger body of text.

I introduce the “Optical Needle in a Haystack” (O-NIH) evaluation framework to systematically optimize and quantify the performance of this technique. Through extensive experimentation with variables including font family, color contrast, image resolution, and font size, I’ve identified optimal configurations. These results show that it is possible to achieve compression ratios approaching 3:1 (e.g., **2.8:1**) while maintaining over **93% retrieval accuracy**. This demonstrates that Optical Compression is a viable, plug-and-play strategy for drastically extending the effective context length and reducing the operational cost of existing VLMs.

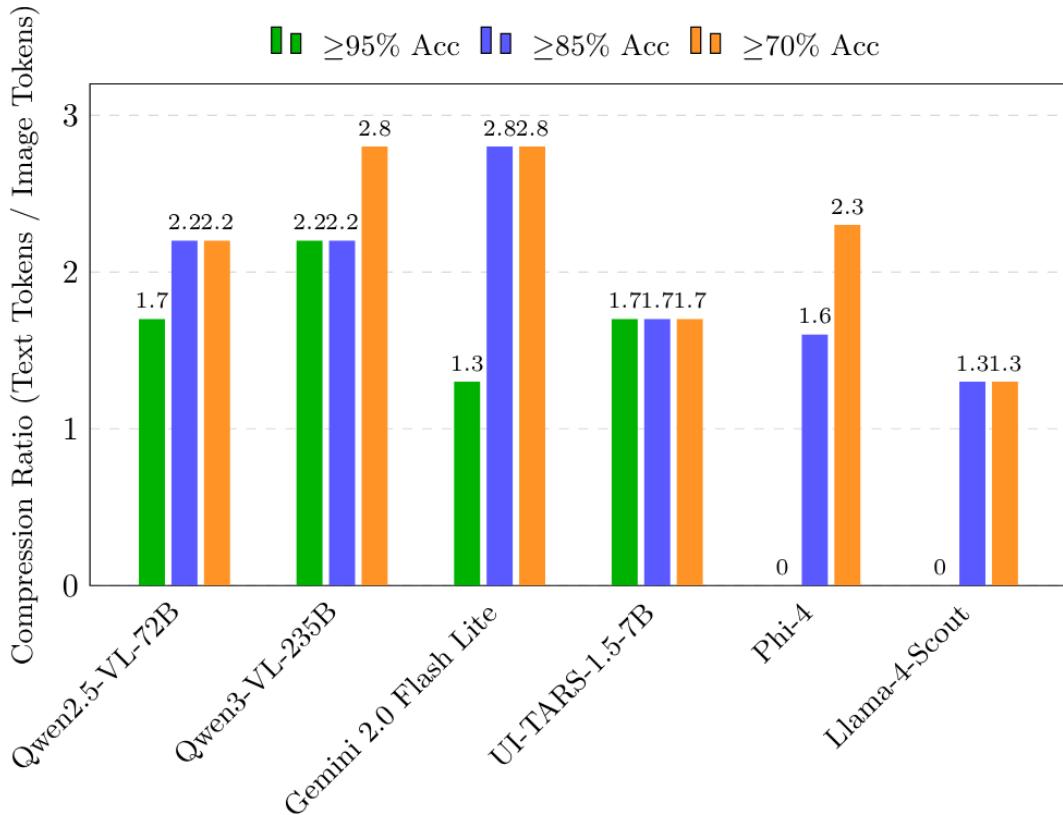


Figure 1: Maximum compression ratios achieved by top vision-language models at different accuracy thresholds.

1 Related Work

1.1 Optical Context Precedents

The concept of encoding text into a visual representation for processing by vision models is supported by several foundational works. The viability of using pixels for language understanding was established by models like **Pix2Struct** [6], which demonstrated the effectiveness of training on document screenshots. More recently, the feasibility of high-density text compression through rendering was directly explored by the **DeepSeek-OCR** [12] research, which served as the primary inspiration for this project. This work extends this concept by systematically optimizing rendering parameters across various state-of-the-art Vision-Language Models (VLMs).

1.2 Token-Based Compression

Most approaches to long-context efficiency focus on minimizing the input token count through text-level manipulation. Methods like **LLMLingua** [5] and context adaptation techniques [2] attempt to prune redundant or low-information tokens from the context, requiring a separate language model or heuristic to perform the compression. In contrast, the Optical Context Compression method presented here uses a zero-shot, plug-and-play approach by entirely side-stepping the costly text token input space in favor of a fixed-cost image token input.

2 Key Results

Analysis of over 90 experiments reveals that different models have unique “sweet spots” for optical compression¹. However, a general trend emerges: high-fidelity compression is achievable across multiple state-of-the-art VLMs.

Table 1: Model Performance Overview

Model	Peak Performance (Max Accuracy)	High-Fidelity Compression (Max Ratio @ $\geq 85\%$)	High-Density Compression (Max Ratio @ $\geq 70\%$)
qwen/qwen2.5-vl-72b-instruct	98.33% @ 1.7:1 (Exp 44)	2.2:1 @ 94.44% (Exp 81)	2.2:1 @ 94.44% (Exp 81)
qwen/qwen3-vl-235b-a22b	95.24% @ 2.2:1 (Exp 50)	2.2:1 @ 95.24% (Exp 50)	2.8:1 @ 82.22% (Exp 90)
google/gemini-2.0-flash-lite	100.0% @ 1.3:1 (Exp 46)	2.8:1 @ 93.65% (Exp 56)	2.8:1 @ 93.65% (Exp 56)
bytedance/ui-tars-1.5-7b	95.24% @ 1.7:1 (Exp 72)	1.7:1 @ 95.24% (Exp 72)	1.7:1 @ 79.71% (Exp 88)
microsoft/phi-4-multimodal	94.44% @ 1.1:1 (Exp 59, 85)	1.6:1 @ 91.11% (Exp 63)	2.3:1 @ 73.55% (Exp 61)
meta-llama/llama-4-scout	86.57% @ 1.3:1 (Exp 53)	1.3:1 @ 86.57% (Exp 53)	1.3:1 @ 86.57% (Exp 53)

2.1 Per-Model Optimization Summary

While optimal variables vary, this research provides a strong starting point for applying this technique:

- **Image Size:** The best results are often achieved when the input image resolution is slightly smaller than the model’s maximum supported single-tile resolution. An image size of **864x864 pixels** proved to be an excellent default. Aligning with the 16×16 pixel patching strategies employed by modern Vision Transformers [4, 3], where resolutions close to, but under, maximum single-tile limits optimize the VLM’s perception.
- **Font Size:** A font size between **12px** and **16px** is generally optimal. For `qwen/qwen2.5-vl-72b`, 13px was the clear winner, while `gemini-2.0-flash-lite` performed well with 12px.
- **Font Family:** High-legibility, sans-serif fonts are critical. Top performers included:
 1. **Atkinson Hyperlegible** (especially the Italic variant)
 2. **Lato**
 3. **Lexica Ultralegible**
- **Color Contrast:** Standard black-on-white or yellow-on-blue perform best. Low-contrast pairs (e.g., red-on-blue) fail completely.

¹Detailed raw logs of the 90+ experiments (the Appendix) have been excluded from this document for brevity. They can be found in the `README.md` of the actual GitHub repository: <https://github.com/MaxDevv/Un-LOCC>.

3 The O-NIH Methodology

The core challenge of this research was to create a reliable method for testing a model’s text processing capabilities across different image configurations. To solve this, I developed the **Optical Needle in a Haystack (O-NIH)** test. This approach is conceptually derived from standard long-context evaluation protocols used for models like **GPT-4** [8] and various large-scale VLMs [7].

The O-NIH test works by injecting a unique, non-contextual code (the “needle”) into a large, cohesive body of text (the “haystack”). The VLM is then tasked with retrieving only that code. The haystack and needle are prepared programmatically:

```
1 import random
2
3 def generate_needle(length=9):
4     """Generates a random, unique code (the 'needle')."""
5     chars = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ23456789'
6     result = ''
7     for i in range(length):
8         result += random.choice(chars)
9         if (i + 1) % 3 == 0 and i < length - 1:
10            result += '_'
11
12 return result
13
14 def prepare_text(source_text, word_count):
15     """Extracts a random chunk of text and injects the needle."""
16     words = source_text.split()
17     if len(words) < word_count:
18         raise ValueError(f"Source text is too short.")
19
20     start_index = random.randint(0, len(words) - word_count)
21     text_chunk = words[start_index : start_index + word_count]
22
23     needle = generate_needle()
24     # Inject the needle at a random position within the text chunk
25     injection_index = random.randint(1, len(text_chunk) - 1)
26     text_chunk.insert(injection_index, needle)
27
28     haystack = " ".join(text_chunk)
29     return haystack, needle
```

Listing 1: Needle Generation and Injection

The model’s accuracy is then graded using a “fuzzy” comparison that accounts for common OCR errors (e.g., mistaking ‘I’ for ‘1’) by calculating the Levenshtein distance.

```
1 def normalize_for_ocr(text):
2     """Normalizes a string to account for common OCR errors."""
3     replacements = {
4         '0': '0', 'o': '0', 'I': '1', 'l': '1', 'S': '5', 's': '5',
5         'B': '8', 'A': '4', '_': ' ', ' ': ''
6     }
7     text = text.upper()
8     for old, new in replacements.items():
9         text = text.replace(old, new)
10    return text
11
12 def fuzzy_similarity(s1, s2):
13     """Calculates a normalized similarity score based on Levenshtein distance."""
14     norm_s1 = normalize_for_ocr(s1)
15     norm_s2 = normalize_for_ocr(s2)
16     # Standard Levenshtein implementation omitted for brevity
17     distance = levenshtein_distance(norm_s1, norm_s2)
18     max_len = max(len(norm_s1), len(norm_s2))
19     if max_len == 0: return 1.0
20     return (max_len - distance) / max_len
```

Listing 2: Fuzzy Similarity Scoring

4 Limitations

- **Contextual Understanding:** This technique tests perceptual retrieval, not deep contextual understanding. While a model can *find* text, its ability to reason over the optically compressed context may be different than with standard text tokens. The challenge of complex reasoning over visual documents is a known limitation across the domain of VLM-based document understanding [1, 9].
- **Prompt Injection Risk:** This technique could potentially be used to circumvent safety filters by rendering harmful text within an image.

5 Vision & Future Work

The field of Lossy Optical Context Compression offers an immediate path to reducing costs and increasing context length for developers without fine-tuning.

The strong performance of smaller models like **Phi-4 Multimodal** opens a particularly exciting avenue for **on-device context expansion**. By offloading historical context to an optically compressed image, it may be possible to run agents with vast memory on edge devices with limited VRAM.

I foresee a series of libraries that wrap around existing API and SDK interactions to make large-context, low-cost VLM inference easily accessible.

6 Acknowledgements

This project would not have been possible without the foundational ideas presented in the **DeepSeek-OCR** paper (“Contexts Optical Compression”). Their work provided the initial spark that demonstrated the feasibility of representing large amounts of text in a compressed optical format.

7 Code and Data Availability

To facilitate reproducibility and further research, all resources are open-source:

- **Research Data & Logs:** The full dataset of 90+ experiments and original research notes can be found at <https://github.com/MaxDevv/Un-LOCC>.
- **Python Implementation:** A usable wrapper library for implementing O-NIH tests and optical compression is available at <https://github.com/MaxDevv/Un-LOCC-Wrapper>.

References

- [1] Lukas Blecher, Guillem Cucurull, Thomas Scialom, and Robert Stojnic. Nougat: Neural optical understanding for academic documents. *arXiv preprint arXiv:2308.13418*, 2023.
- [2] Alexis Chevalier, Alexander Wettig, Anirudh Ajith, and Danqi Chen. Adapting language models to compress contexts. *ArXiv preprint abs/2305.14788*, 2023.
- [3] Mostafa Dehghani, Basil Mustafa, Josip Djolonga, Jonathan Heek, Matthias Minderer, Mathilde Caron, Andreas Steiner, Joan Puigcerver, Robert Geirhos, Ibrahim Alabdulmohsin, Avital Oliver, Piotr Padlewski, Alexey Gritsenko, Mario Lučić, and Neil Houlsby. Patch n' pack: Navit, a vision transformer for any aspect ratio and resolution, 2023.
- [4] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021.
- [5] Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. Llmlingua: Compressing prompts for accelerated inference of large language models, 2023.
- [6] Kenton Lee, Mandar Joshi, Iulia Turc, Hexiang Hu, Fangyu Liu, Julian Eisenschlos, Urvashi Khandelwal, Peter Shaw, Ming-Wei Chang, and Kristina Toutanova. Pix2struct: Screenshot parsing as pretraining for visual language understanding, 2022.
- [7] Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024.
- [8] OpenAI. Gpt-4 technical report, 2023.
- [9] Jake Poznanski, Aman Rangapur, Jon Borchardt, Jason Dunkelberger, Regan Huff, Daniel Lin, Christopher Wilhelm, Kyle Lo, and Luca Soldaini. olmocr: Unlocking trillions of tokens in pdfs with vision language models. *arXiv preprint arXiv:2502.18443*, 2025.
- [10] Yi Tay, Mostafa Dehghani, Dara Bahri, and Donald Metzler. Efficient transformers: A survey. *ACM Computing Surveys*, 55(6):1–28, 2022.
- [11] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.
- [12] Haoran Wei, Yaofeng Sun, and Yukun Li. Deepseek-ocr: Contexts optical compression, 2025.