SIIM CAIMI25
Conference on Artificial Intelligence in Medical Imaging
OCT 20-21 | UNIVERSITY OF CALIFORNIA, IRVINE

Unclogging the CPU Pipes: A Persistent Caching Pipeline to Speed Up Deep Learning Workflows in Radiology

Aditya V. Kulkarni, MSc, St. Jude Children's Research Hospital; Dharmam Y. Savani, MSc; Paul H. Yi, MD

Introduction/Background

Despite the large emphasis placed on GPUs for deep learning in radiology, the biggest bottlenecks often come from CPUs during image preprocessing. However, frameworks to optimize data loading are scarce. We present a high-performance data loading pipeline using persistent caching that outperforms the state-of-the-art in speed, while supporting hierarchical data access via metadata and seamless integration with augmentation libraries.

Methods/Intervention

Our pipeline uses persistent caching to reduce CPU overhead by storing deterministic transformations on disk and reloading preprocessed images during training (Figure 1). Cache files are saved as HDF5 files in a customizable hierarchical folder structure—mirroring typical medical data organization (e.g., patient/study/image). We evaluate two benchmarks using 10,000 chest radiographs from MIMIC-CXR-JPG: (1) parallelized data loading, measured by throughput (samples/second), and (2) end-to-end training for a contrastive learning task, measured by energy consumption (kWh), epoch time, and GPU idle time. For data loading, we compare against direct loading and MONAI's persistent caching pipeline. All experiments are run across varying batch sizes to simulate increasing data dimensionality. Each experiment was run for 5 epochs and repeated 3 times; results reflect mean across runs.

Results/Outcome

For data loading, our pipeline was 10-16× faster than direct loading across batch sizes (Figure 2). With augmentations, acceleration was 9-13×, increasing with batch size. Compared to MONAI, our method is 2.4–3.6× and 2.9–3.5× faster with and without augmentations, respectively. For training, our pipeline reduces GPU idle time, boosts throughput, and lowers energy use as data dimensionality grows (Figure 3). At batch size 128, it cuts energy by 40%, GPU idle time by 86%, and training time by 50% compared to direct loading. At the largest batch size, training time drops by up to 70%. No benefits are observed at small batch sizes (< 32), where data loading is not a major bottleneck.

Conclusion

Our persistent caching pipeline markedly reduces CPU data loading bottlenecks, accelerating deep learning training times for medical imaging. Our pipeline will be released publicly to benefit the radiology AI community.

Statement of Impact

Our data loading pipeline reduces large batch size training time by >3× while improving throughput by >16×, accelerating deep learning workflows.

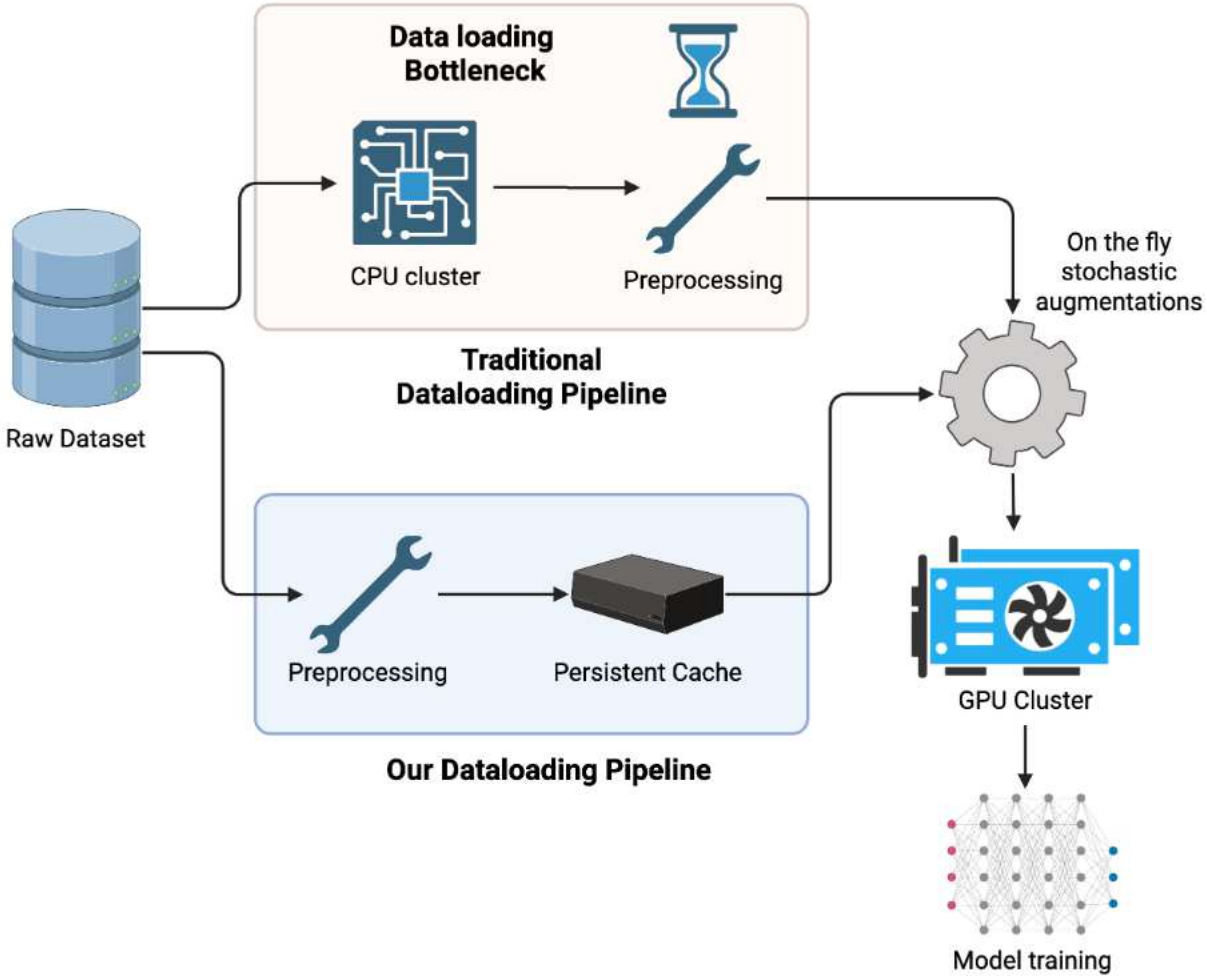

Figure 1. Schematic diagram illustrating how our pipeline circumvents CPU bottlenecks commonly observed in traditional machine learning workflows by leveraging persistent caching mechanism

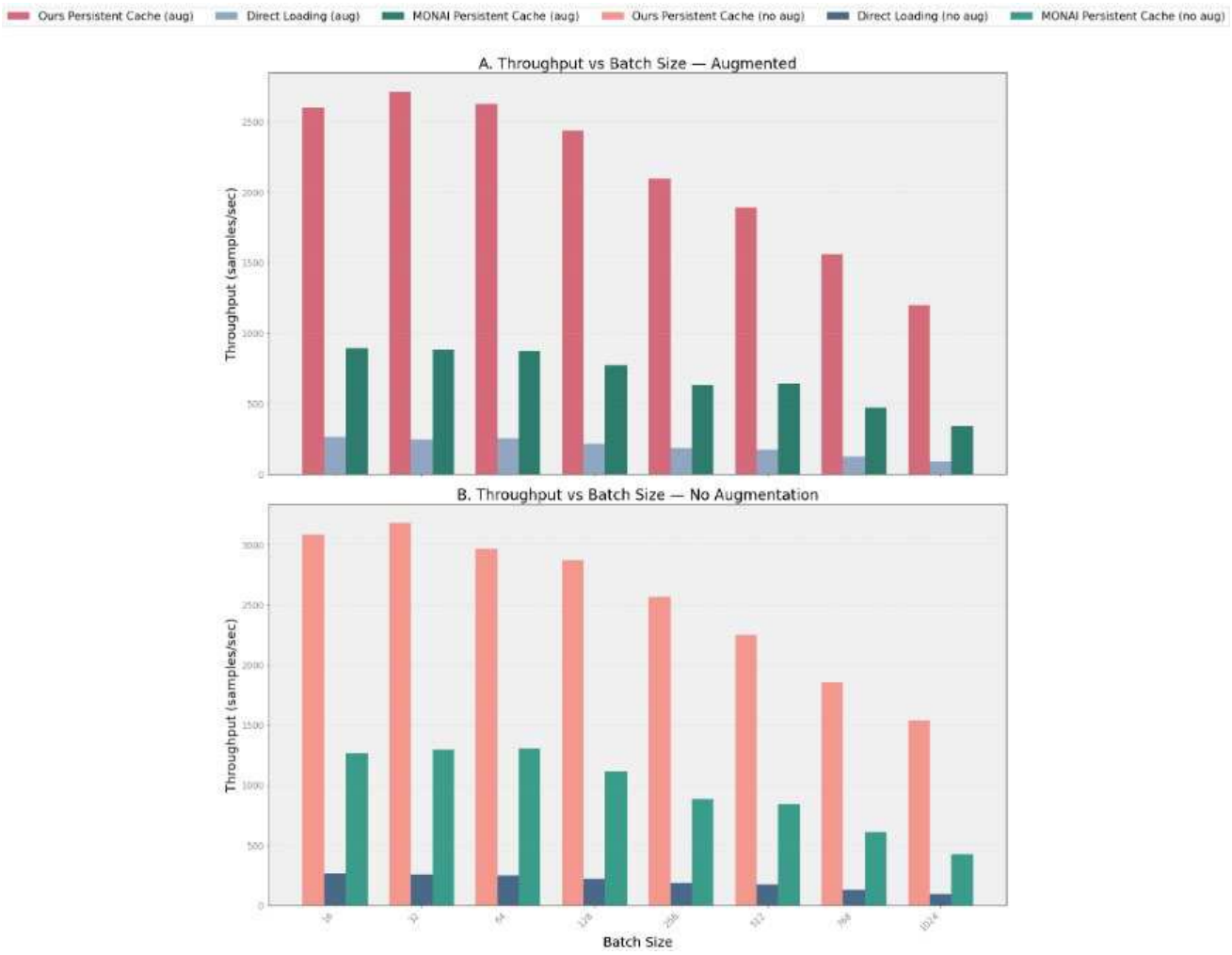

Figure 2 Our Pipeline (red) consistently outperforms direct loading and MONAI caching across all batch sizes, both with and without augmentations. (a). Comparison of throughput(samples/sec) between the MONAI-based pipeline and our custom pipeline used in the training benchmark procedure. Graph illustrates the throughput performance when data augmentations are included in pipeline. (b). Comparison of throughput between the MONAI-based pipeline and our custom pipeline used in the training benchmark procedure. This graph illustrates throughput performance without data augmentations, resulting in significantly higher throughput compared to Figure 1(a).

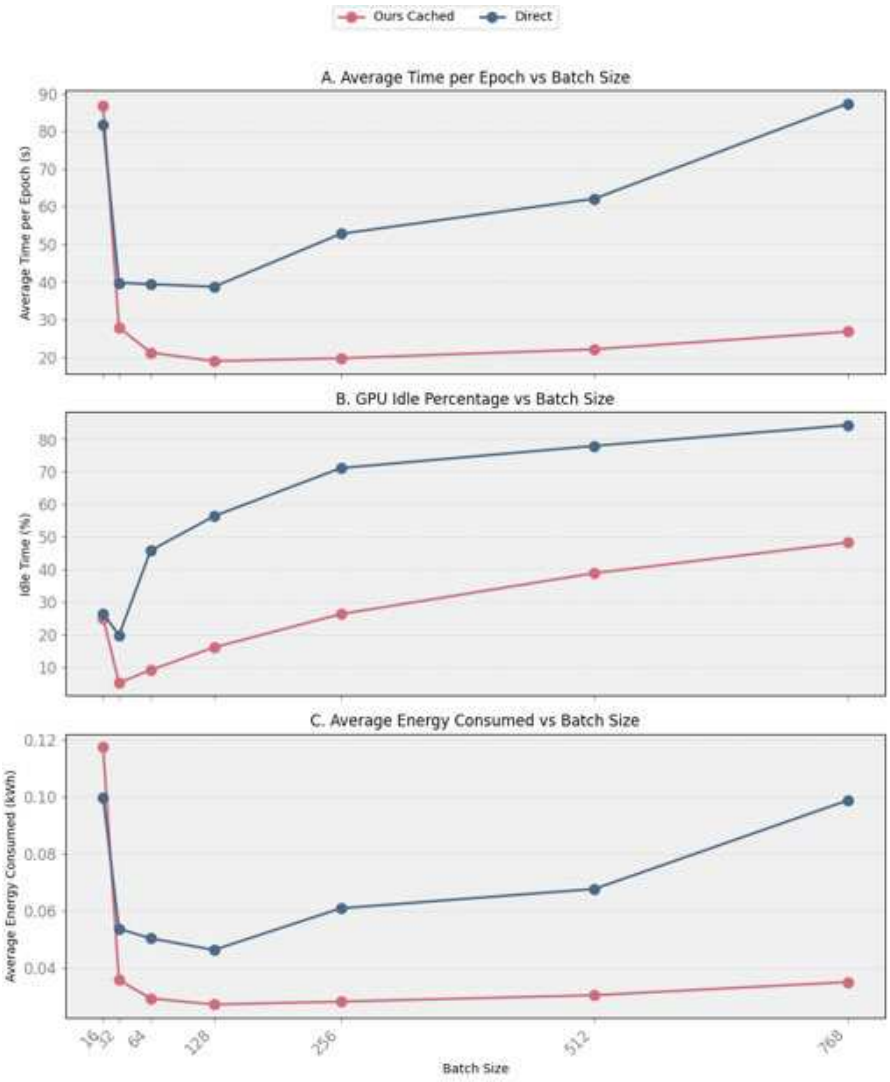

Figure 3 Our Pipeline maintains low values of epoch time, GPU idle percentage and energy consumed as batch size increases. Note that direct loading scales poorly as batch size increases. (a). Comparison of Average Time required for model to complete the epochs across multiple batch sizes. (b). Comparison for Percentage of time GPU remains Idle (does not perform any computations) across multiple batch size (c). Energy Consumed(kWh) by the training benchmark procedure across multiple batch sizes

Keywords

Deep Learning; Optimization Pipeline; Energy-efficient training; Medical Imaging