L3: Accelerator-Friendly Lossless Image Format for High-Resolution, High-Throughput DNN training

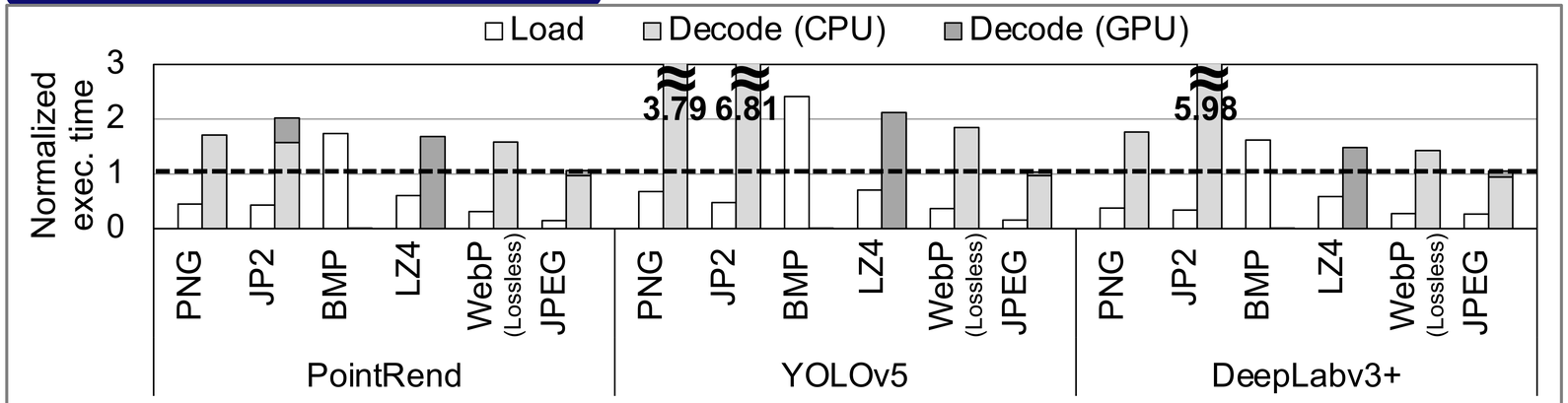
Jonghyun Bae, Woohyeon Baek, Tae Jun Ham, and Jae W. Lee

Seoul National University, Seoul, Korea

Abstract

The training process of deep neural networks (DNNs) is usually pipelined with stages for data preparation on CPUs followed by gradient computation on accelerators like GPUs. In an ideal pipeline, the end-to-end training throughput is eventually limited by the through- put of the accelerator, not by that of data preparation. In the past, the DNN training pipeline achieved a near-optimal throughput by utilizing datasets encoded with a lightweight, lossy image format like JPEG. However, as high-resolution, losslessly-encoded datasets become more popular for applications requiring high accuracy, a performance problem arises in the data preparation stage due to low-throughput image decoding on the CPU. Thus, we propose L3, a custom lightweight, lossless image format for high-resolution, high-throughput DNN training. The decoding process of L3 is effectively parallelized on the accelerator, thus minimizing CPU intervention for data preparation during DNN training. L3 achieves a 9.29x higher data preparation throughput than PNG, the most popular lossless image format, for the Cityscapes dataset on NVIDIA A100 GPU, which leads to 1.71x higher end-to-end training throughput. Compared to JPEG and WebP, two popular lossy image formats, L3 provides up to 1.77x and 2.87x higher end-to-end training throughput for ImageNet, respectively, at equivalent metric performance.

Data Preparation Bottleneck



`Load` and `Decode` execution time normalized to the `Compute` time

- Spending most of time for **Decode on CPU** in case of PNG, JP2, and WebP by complexed and sequentialized decoding algorithm
- Spending most of time for **Load** in case of BMP format by fetching uncompressed raw data from disk
- The use of the lossy format results in **degradation of the test set accuracy** in object detection and semantic segmentation apps

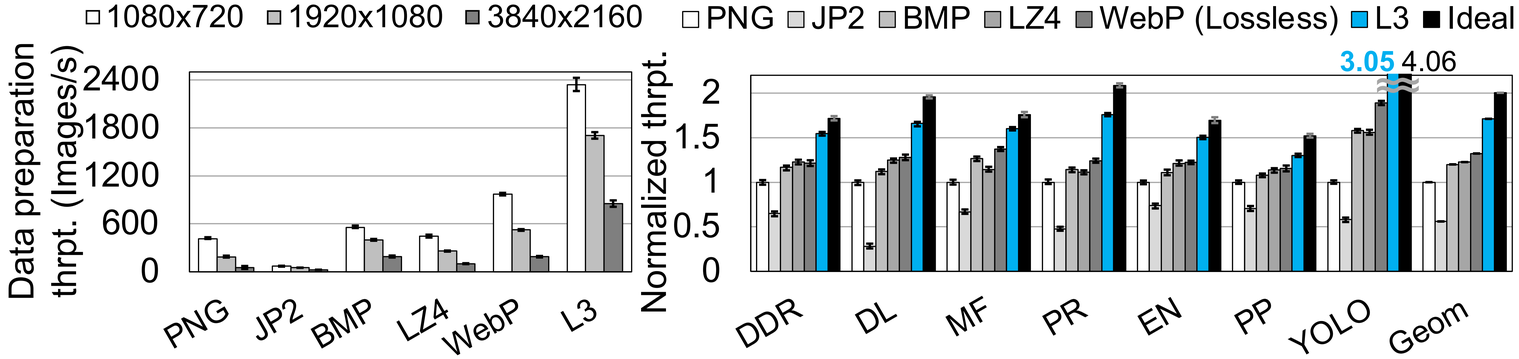
Design Goals

- Specialized for ML/DL training with a **lightweight**, **lossless** algorithm
- Maximizing decoding throughput by **leveraging the GPU**
- Providing a **good-enough compression ratio** not to introduce a new bottleneck in `Load` stage

Design Comparison

	Algorithm		Lossless?	GPU-support?
	Filter	Compression		
PNG	None, Sub, Up, Avg, Paeth	Deflate	O	X
JPEG	DCT, Quantization	Run-length + Huffman coding	X	△
L3	Custom Paeth	Base-delta coding	O	O

Evaluation



(a) Data preparation throughput (b) Normalized end-to-end training iteration throughput

Throughput comparison with lossless-encoded dataset

- **5.67x, 9.29x**, and **15.71x** higher decoding throughput for the HD, FHD, and UHD dataset than the lossless PNG format
- **1.71x** and **1.29x** higher geomean end-to-end training throughput than PNG and WebP