Optimizing Transformer Models for Image Segmentation on the Edge

Sriram Sai Ganesh[1]*, Srinivasan Parthasarathy[1]

[1]Department of Computer Science, The Ohio State University; *corresponding author: saiganesh.3@osu.edu

ABSTRACT

Vision transformer-based (**ViT**) models have achieved state-of-the-art performance in computer vision tasks by effectively capturing nuanced global relationships between image features. Despite their success, these models are **computationally intensive** and challenging to deploy on **resource-constrained** edge devices **without dedicated GPUs**, such as drones and satellites.

This work addresses the optimization of ViT models for enhanced resource efficiency, enabling high-performance inference on low-powered edge devices in a **disaster response** setting.

We focus on Meta's state-of-the-art **Segment Anything Model (SAM)** for semantic image segmentation, implementing a series of architectural and hardware-specific modifications to improve its viability for edge deployment. Our optimizations include (1) a faster **attention mechanism**, (2) post-training weight and activation **quantization**, and (3) **PyTorch tensor** operation optimizations. These enhancements result in an over 40% speedup in inference time with minimal degradation in segmentation quality on the SA-1B image dataset.

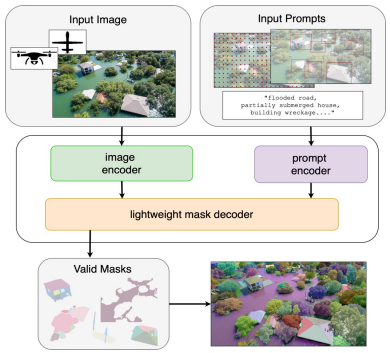
Motivation

- **Rapid automatic assessment of disaster impact** using **computer vision** is crucially important to first responders **mitigating damage to lives & property.**
- Drones, satellites & other UAV **edge devices cannot employ powerful transformer models** on-device due to energy & performance constraints.

Goal: Improve the resource efficiency of one such semantic segmentation model.

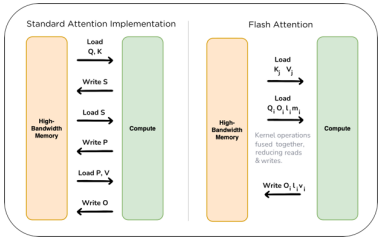
Segment Anything

Segment Anything Model (SAM): Vision Transformer (ViT)-based **semantic image segmentation** model.



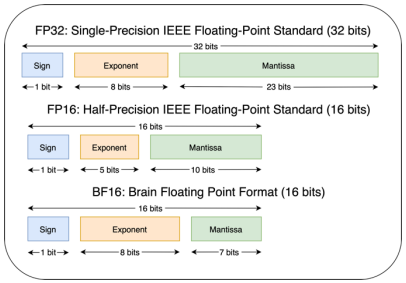
Flash Attention

- **Attention mechanism:** Enables ViT-based image encoder to capture complex image features.
- Computation bottleneck: **over 90%** SAM runtime.
- Naïve Scaled Dot Product Attention (SDPA) → **kernel-fused Triton implementation** of **Flash Attention V2** – compute tensors **on demand**.



Post-Training Quantization

- Trade-off model weight & activation precision for higher math operation throughput on hardware.
- **Post-Training Dynamic Quantization:**
- **Weights**: quantized en masse prior to inference.
  FP32 (32-bit IEEE Floating Point decimals) → **BF16** (16-bit Brain Float decimals).
- **Activations: Dynamically computed** BF16 decimals for arithmetic **during inference**.



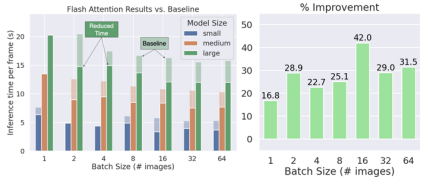
Optimized Tensors

- Image, prompt, and label tensors are of **variable batch sizes**; kernels must process **sequentially**.
- PyTorch's `NestedTensor`: Pack a jagged list of tensors into a **single efficient data structure**.
- Optimize SAM's **prompt encoder** & **mask decoder**, bypassing padding and enabling specialized kernels.

Ablation Study

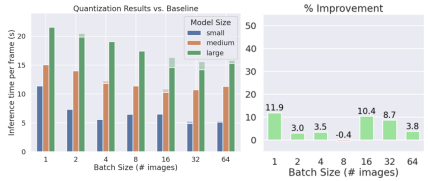
Flash Attention

- SAM image encoder ViT attention mechanism: **Most significant fraction** of inference runtime.
- Tensor multiplication encompasses **over 90%** of computation time for all three model sizes.
- **FlashAttention V2**: Kernel fusion mitigates memory read & write overhead.
- Results in **up to 42% image encoder speed-up.**



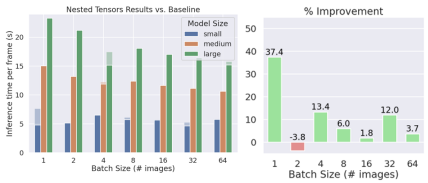
Post-Training Quantization

- Quantization to BF16: **Minor computation time improvements** across batch sizes.
- Memory savings outweighed by **lack of native CPU kernel support** for FP16/BF16 arithmetic on AMD Epyc 7313.
- Potential improvement on more **recent CPU architectures** – Intel Xeon 4th Gen, AMD MI8/MI25.



Nested Tensor operations

- Enables batching of prompt coordinates & labels: vector operations utilize faster CPU tensor kernels.
- Most evident improvement for **smaller batch sizes.**
- Larger batches (size >4): Batched **image encoding** improvements **override NestedTensor** benefits.



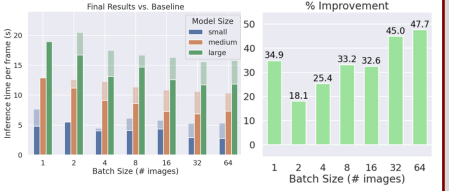
Results

- Efficiency improvement for all three model sizes:

Model Size (# params)	Runtime (seconds/frame)	
	Initial	Final
Small (91M)	5.26	2.89 (-45%)
Medium (308M)	10.61	6.86 (-35.3%)
Large (636M)	15.53	11.73 (-24.5%)

*Results with batch size 32, mean of median 50% of 10 runs on CPU.

- Most significant **computation savings for larger batch sizes**, due in large part to improved time complexity of **Flash Attention** vs. naïve SDPA:



- Enables **inference** for semantic image segmentation on edge devices **in under 3 seconds.**
- Existing SAM models can be **replaced with a size larger** (2x-3x parameter count) optimized SAM version, at **almost unchanged** computation cost.

Takeaways

1. State-of-the-art models are not out of reach for deployment on resource-constrained devices.
2. Optimization for edge devices entails specific hardware-aware computation considerations.
3. Significant performance improvements can be obtained with hardware-aware model designs.

References

1. R. Bommasani et. al. "On the opportunities and risks of foundation models," CoRR, vol. abs/2108.07258, 2021.
2. S. Nag et. al. "Vita: A vision transformer inference accelerator for edge applications," in 2023 IEEE International Symposium on Circuits and Systems (ISCAS), IEEE, May 2023.
3. A. Akhyar et. al., "Deep artificial intelligence applications for natural disaster management systems: A methodological review," Ecological Indicators, vol. 163, p. 112067, 2024.
4. A. Gupta et. al., "Deep learning-based aerial image segmentation with open data for disaster impact assessment," Neurocomputing, vol. 439, pp. 22–33, 2021.
5. A. Kirillov et. al., "Segment anything," 2023.
6. "Accelerating Generative AI with PyTorch: Segment Anything, Fast — pytorch.org." https://pytorch.org/blog/accelerating-generative-ai/.[Accessed 16-07-2024].
7. T. Dao, "Flashattention-2: Faster attention with better parallelism and work partitioning," 2023.
8. "Getting Started with Nested Tensors — pytorch.org." https://pytorch.org/tutorials/prototype/nestedtensor.html. [Accessed 16-07-2024]
9. C. Liu et. al., "Ptq4sam: Post-training quantization for segment anything," 2024.

Contact

Questions or feedback?
Email: saiganesh.3@osu.edu
GitHub: Sriram-Sai-Ganesh

Acknowledgements

The Research in Sustainable Energy (RISE) Summer Program offered by the *Ohio State EmPOWERment Program* – Convergent Graduate Training for a Sustainable Energy Future -- Gratefully acknowledges funding from the U.S. National Science Foundation National Research Traineeship Program (Grant 1922666), the Graduate School, Office of Diversity, Equity, and Inclusion, and Sustainability Institute.