Convolution in CUDA

Sajina Pathak

March 30, 2025

Abstract

Objective to learn about Convolution in CUDA

1 Introduction

Floating-Point Computation Rate and Scaling

The floating-point computation rate (GFLOPS) of the GPU kernel can be calculated as:

$$\text{FLOPs} = 2 \times \text{FILTER\_SIZE}^2 \times \text{imageHeight} \times \text{imageWidth}$$

where the factor of 2 accounts for one multiplication and one addition per filter element.

Using the provided timings, the performance for different input sizes is summarized below:

Image Size	Total FLOPs	Kernel Time (s)	GFLOPS
100×100	0.5×10^6	0.000051	9.80
500×500	12.5×10^6	0.000087	143.7
1000×1000	50×10^6	0.000197	253.8
10000×10000	5×10^9	0.013658	365.9

Analysis: The computation rate improves with increasing input size. For small images, the performance is limited by kernel launch overhead and memory latency. As the image size grows, these overheads become negligible, and the GPU achieves significantly higher GFLOPS, approaching its computational capacity.

Conclusion: The kernel exhibits good scalability with input size, achieving up to 365.9 GFLOPS for large images. This trend is typical for compute-bound kernels as they better utilize the GPU's resources with larger workloads.

GPU Overhead Analysis

To determine the percentage of time spent as overhead, we consider:

- Device memory allocation time
- Host-to-Device (H2D) memory copy time
- Device-to-Host (D2H) memory copy time

The overhead percentage is calculated as:

$$\text{Overhead } \% = \frac{\text{Device Allocation} + \text{H2D Copy} + \text{D2H Copy}}{\text{Total Execution Time}} \times 100$$

Observation: As the input size increases, the overhead percentage decreases, because kernel execution becomes more dominant. For small inputs, the GPU is underutilized and overhead dominates, whereas for large inputs, data transfer and allocation are amortized over the large computation workload.

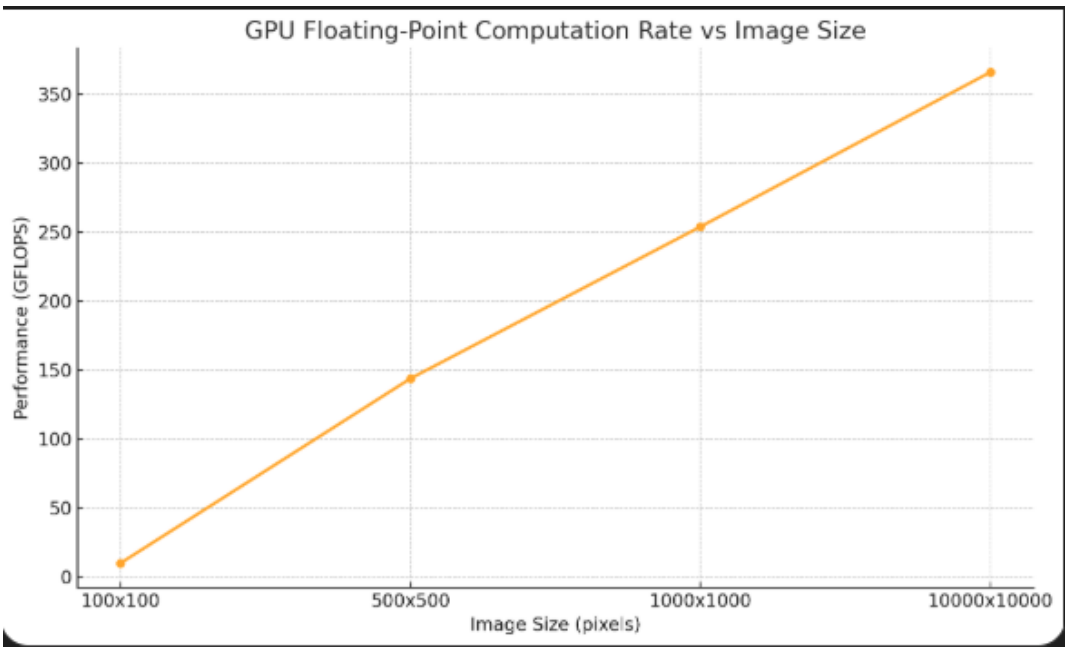

Figure 1: Plot showing how the GPU's floating-point performance improves as image size increases.

Image Size	Alloc (s)	H2D (s)	Kernel (s)	D2H (s)	Total (s)	Overhead %
100 × 100	0.1124	0.0002	0.00005	0.00005	0.1127	99.95%
500 × 500	0.1057	0.0005	0.00009	0.0009	0.1072	99.91%
1000 × 1000	0.1013	0.0012	0.0002	0.0030	0.1057	99.81%
10000 × 10000	0.1028	0.0839	0.0137	0.2783	0.4786	97.15%

Table 1: Overhead percentage for multiple input sizes

Resources Explored

- NVIDIA Developer Forum Discussion on Tiled vs Untiled 2D Convolution Performance
- Medium Blog: CUDA Programming – 2D Convolution
- Medium Blog: CUDA Programming – 1D Convolution