Labwork 4 Report: Thread and Memory Model

Do Thanh Dat - M23.ICT.002

October 7, 2025

Objective

The objective of this labwork is to extend the grayscale image conversion from Labwork 3 using **2D CUDA thread blocks** with Numba. The goal is to analyze how different block sizes affect performance and identify the most efficient configuration.

Implementation

The grayscale conversion follows the equation:

$$Gray = \frac{R + G + B}{3}$$

Each CUDA thread computes the grayscale value of one pixel at coordinates (x, y), enabling massive parallelism.

```
@cuda.jit
def grayscale(src, dst):
    x, y = cuda.grid(2)
    if x < src.shape[0] and y < src.shape[1]:
        r = src[x, y, 0]
        g = src[x, y, 1]
        b = src[x, y, 2]
        dst[x, y] = (r + g + b) // 3
```

The kernel was launched using 2D thread blocks with varying sizes such as (8×8), (16×16), and (32×32) to evaluate performance.

Figure 1: Result of grayscale conversion using GPU CUDA (2D blocks).

Results and Discussion

The kernel was tested with three different block sizes: (8×8), (16×16), and (32×32). Each configuration was executed on the same image, and the total execution time was measured.

Threads per block	**Execution Time (s)**
(8, 8)	0.000706
(16, 16)	0.000241
(32, 32)	0.000247

Table 1: Execution time comparison for different block sizes.

From the results, the (16×16) configuration achieved the best performance. Smaller blocks such as (8×8) create more scheduling overhead, while very large blocks like (32×32) may not fully utilize GPU cores depending on image dimensions. Therefore, (16×16) provides the optimal balance between occupancy and overhead.

Conclusion

The 2D CUDA kernel successfully converted RGB images to grayscale with much better performance than the CPU version. By using 2D thread blocks, thread-to-pixel mapping became simpler and more efficient. Among the

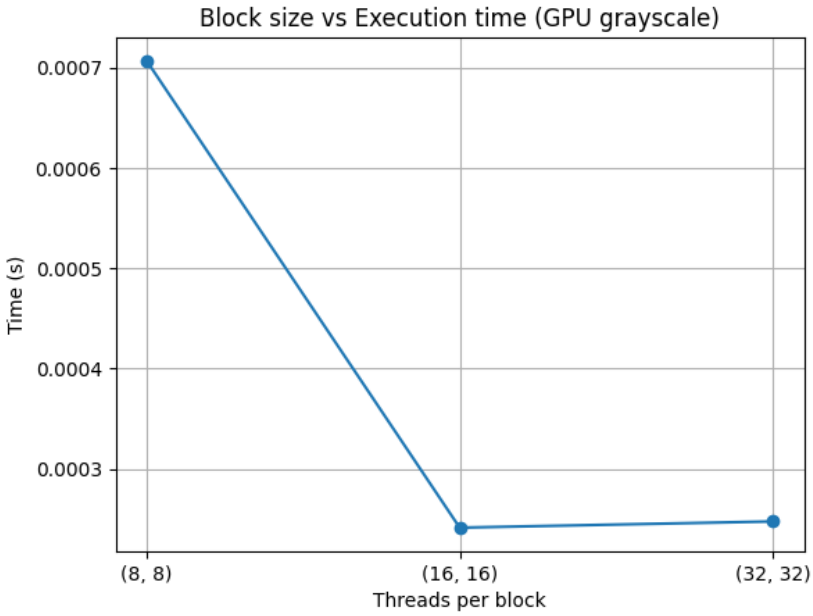

Figure 2: Execution time vs block size.

tested configurations, (16×16) delivered the lowest execution time, confirming that moderate block sizes often yield the best GPU performance for image processing tasks.