

Labwork 3: Hello, CUDA!

Pham Gia Phuc

October 2024

1 Subject

Make image RGB-to-gray converter using Numba CUDA

- Load an image from file (matplotlib's `imread`)
- Flatten image into 1D array of RGB (`reshape(pixelCount, 3)`)
- Implement grayscale using CPU (for range)
- Implement grayscale using GPU
- Save/show the image after each grayscale to validate the result visually
- Use `time.time()` to measure speedup

2 Implementation

This report is using CUDA kernel provided by Google Colaboratory.

- **Number of CUDA Devices Found:** 1
- **Device ID:** 0
- **Name:** Tesla T4
- **Compute Capability:** 7.5
- **PCI Device ID:** 4
- **PCI Bus ID:** 0
- **UUID:** GPU-af936f72-170a-716a-326e-6053e93d8f54
- **Watchdog:** Disabled
- **FP32/FP64 Performance Ratio:** 32
- **Multiprocessor count:** 40
- **Approximate core count:** 2560

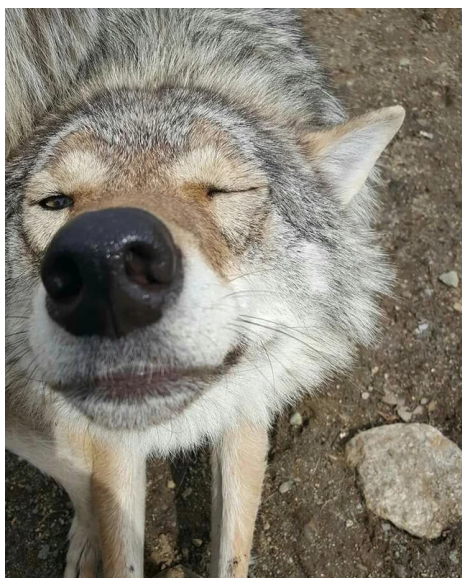


Figure 1: Sample image

- **Total memory size:** 14.75 GB
- **Environment:** Google Colab

2.1 Result

The implementation process makes use of the Sample image in Figure. 1.

CPU	GPU
0.043947	0.002064

Table 1: Processing time (seconds)

3 Conclusion

The GPU speeds up processing time by 21.29 times CPU's.