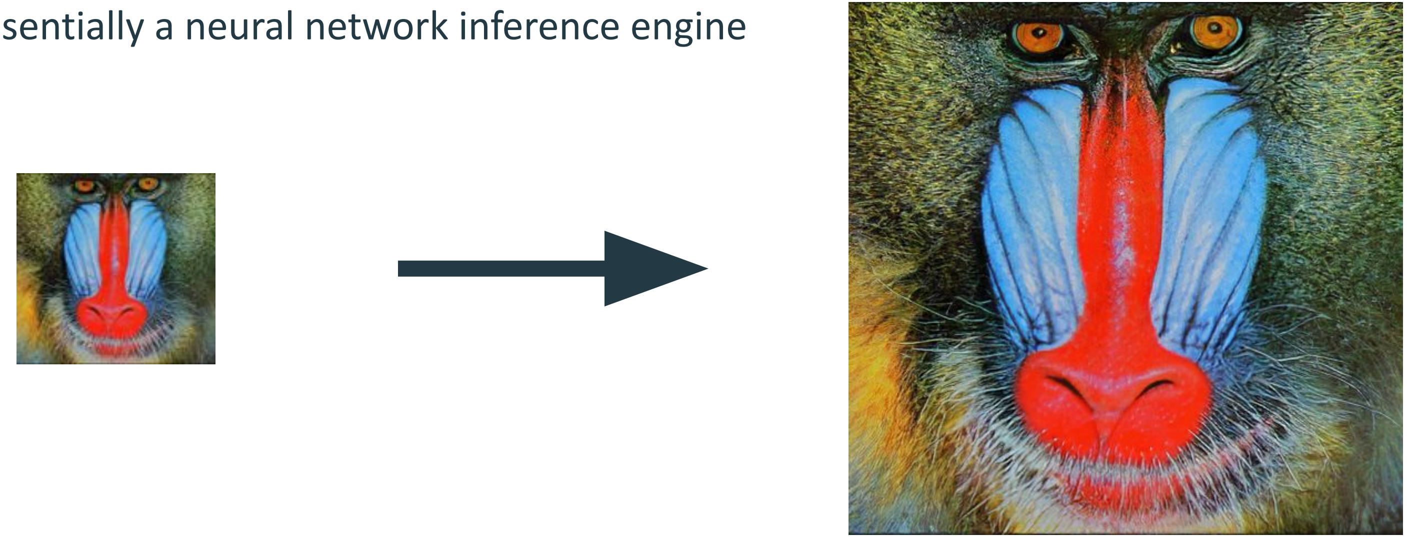
WebGPU Image Super Resolution Milestone 3

Paul (San) Jewell
Andrew Zhu

Project Overview

- A WebGPU based image super resolution program
- Input is fed to a neural network to generate the output
- Essentially a neural network inference engine



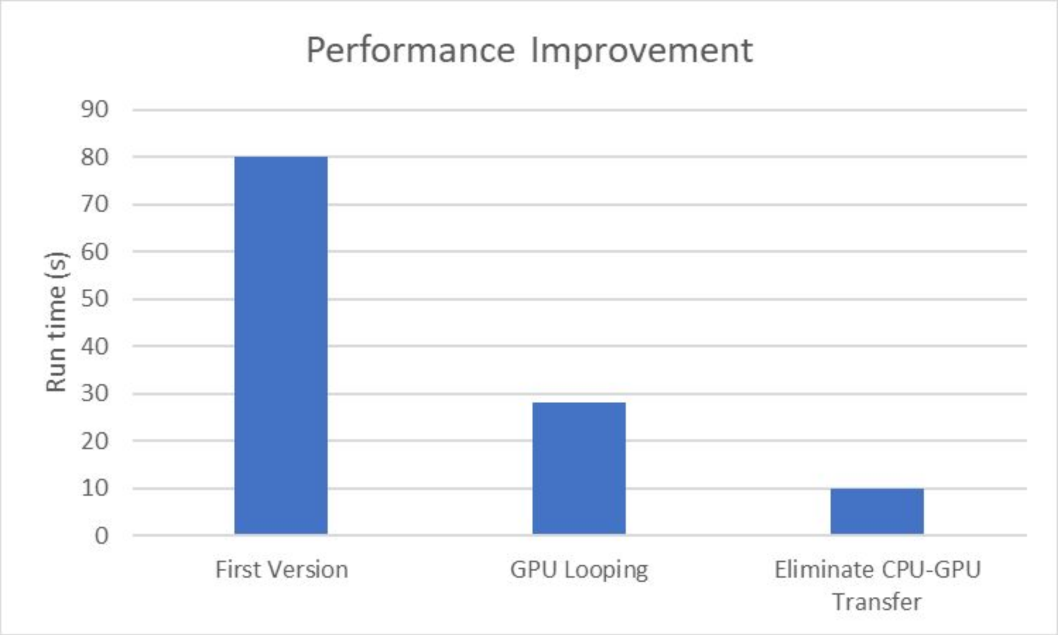
DEMO

https://sona1111.github.io/webgpu-super-resolution/

General Progress

- Completed and partially optimized implementation
- Created prettier web page with Bootstrap, dynamic loading and demos, added to github-pages
- Added PSNR weights and framework for adding more retrained networks to demo page
- Started to look into Quantization

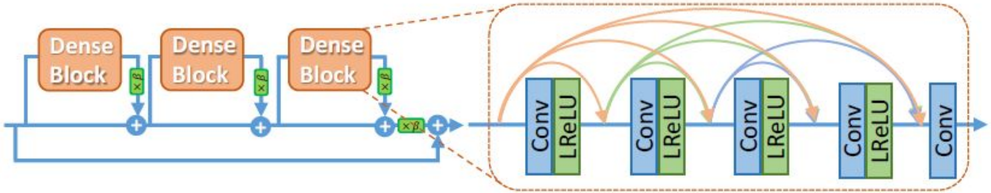
Performance Improvements



Super Resolution Neural Network Architecture

1. Input: RGB image (3 * height * width)
2. First Convolution Layer (Channel 3 -> 64)
3. 23 x Residual in Residual Dense Block

Residual in Residual Dense Block (RRDB)



4. Second Convolution + Residual Layer
5. 2 x Super Resolution Layer (width x 2, height x 2)
6. 2 x Super Resolution Layer (width x 2, height x 2)
7. Last Convolution Layers (Channel 64 -> 3)
8. Output: RGB image (3 * (4 * height) * (4 * width))

Challenges

- Lots of tedious pointer arithmetic.
- The usual struggles of CSS :3
- Crashing with medium size images due to limits on single buffer size - rewrite to use dynamic number of buffers to store data for one ‘meta buffer’ seems to be a maintainability nightmare
- Usual random bugs in Linux Vulkan version by chrome team (sorry >.<)
- No Float16 in Javascript? :o

Goals for Next Milestone

- Even faster implementation
  - Matrix multiply 2D convolution
  - Int8 quantization
  - Manual bins groups for all layers
  - Preload all small argument buffers, and change Storage->Uniform
- Writeup and performance analysis
- Retrain on a few other data sets and compare

References

- https://web.dev/gpu-compute/
- https://github.com/austinEng/webgpu-samples
- https://github.com/xinntao/ESRGAN