WebGPU Image Super Resolution Milestone 1

Paul (San) Jewell
Andrew Zhu

Project Overview

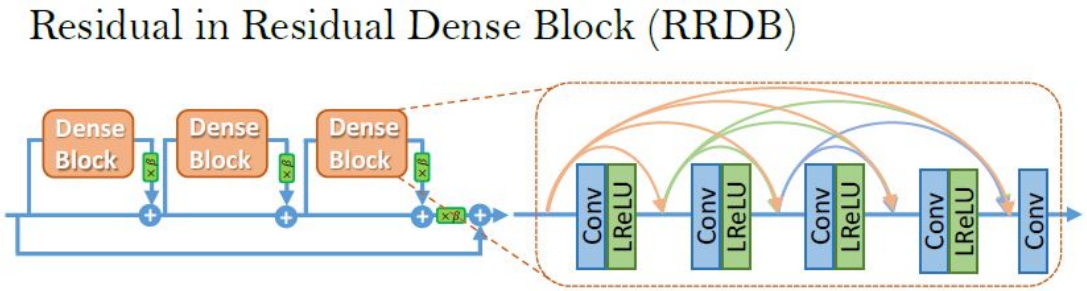
- A WebGPU based image super resolution program
- Input is fed to a neural network to generate the output
- Essentially a neural network inference engine

Progress

1. Identified SOTA in single image super resolution
2. Fully understood neural network architecture
3. Implemented CPU version of pytorch conv2d in numpy
4. Reference model with results from intermediate layers
5. Understood WebGPU compute shader
6. Implemented model weight export from python and import in javascript -- binary files with incremental execution capability
7. Implemented cache of model layers with javascript LocalStorage API

Super Resolution Neural Network Architecture

1. First Convolution Layer (Channel 3 -> 64)
2. 23 x Residual in Residual Dense Block



3. Second Convolution + Residual Layer
4. 2 x Super Resolution Layer (width x 2, height x 2)
5. 2 x Super Resolution Layer (width x 2, height x 2)
6. Last Convolution Layers (Channel 64 -> 3)

Challenges

- Figuring out Conv2d at a low level
- Getting webGPU samples to work
- Problems with using some super resolution models

Goals for Next Milestone

- Baseline implementation of super resolution neural network on WebGPU
- Unit test 2d convolution compute shader
- Unit test residual connection compute shader
- Unit test up sampling compute shader
- Identify potential areas for optimization

References

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