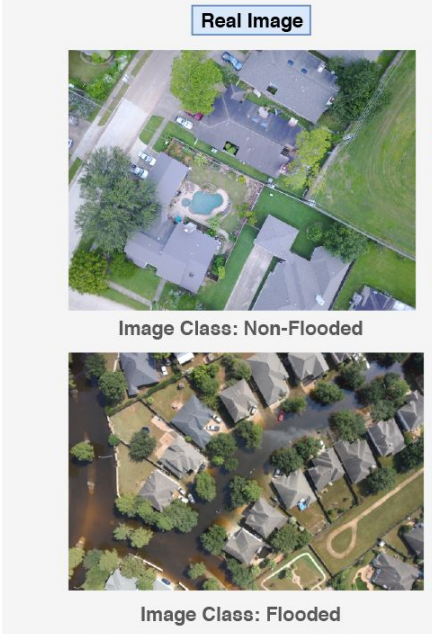
CS 205 Final Presentation: Large Scale Computer Vision for Flood Disaster Management

Michael Butler, Preston Ching, and M. Elaine Cunha

Project Review

- **Train** both supervised and semi-supervised Deep Image Classifier to detect images with flooding, using GPU
- **Deploy** model for fast, highly parallelizable classification, using AWS GPUs
- **Data** from Floodnet Challenge
    - 2,300 quadcopter images of land post-Hurricane Harvey
    - About **25%** of the training set is labeled (roughly 400 of 1,400 images)



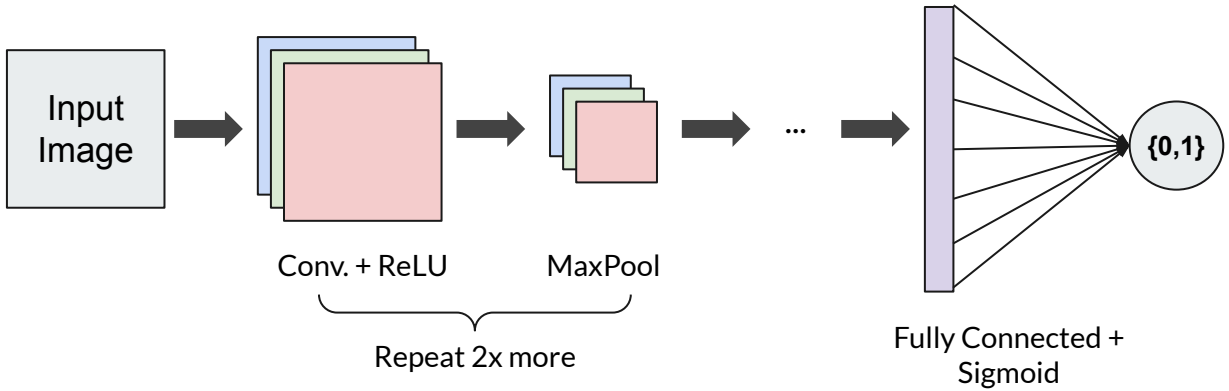
Programming Model & Infrastructure

- Trained Neural Net on a variety of different infrastructures
  - 32 CPUS
  - 32 CPUs + 1 GPUs
  - 32 CPUs + 2 GPUs
- Ubuntu 18.04 with AWS Deep Learning AMI (includes TensorFlow v1)
- G3.8xlarge (2 GPU Processors with 2048 cores, 32 CPU Processors)

Model Architecture

Model: Convolutional neural net for both supervised and semi-supervised classification

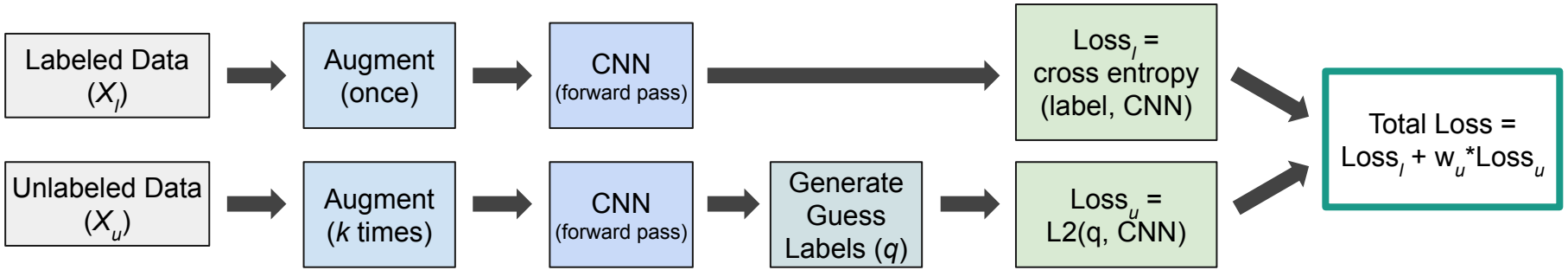
- 2 classes: “Flooded” and “Non-Flooded”



Model Training

Model: Convolutional neural net for semi-supervised classification

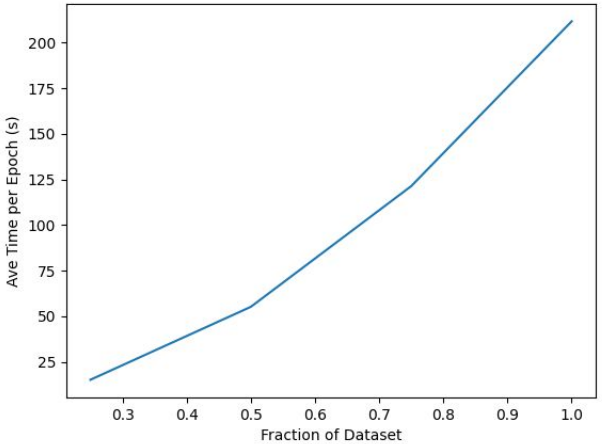
- For each batch/epoch:



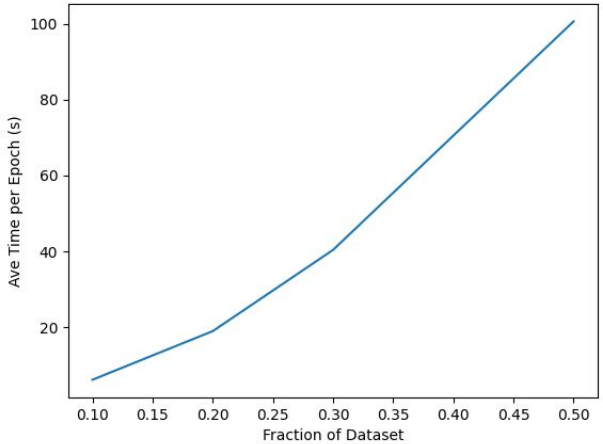
Performance Evaluation & Discussion

- Weak Scaling corroborated where computational times increase with larger training and testing datasets.

Fully-Supervised Model



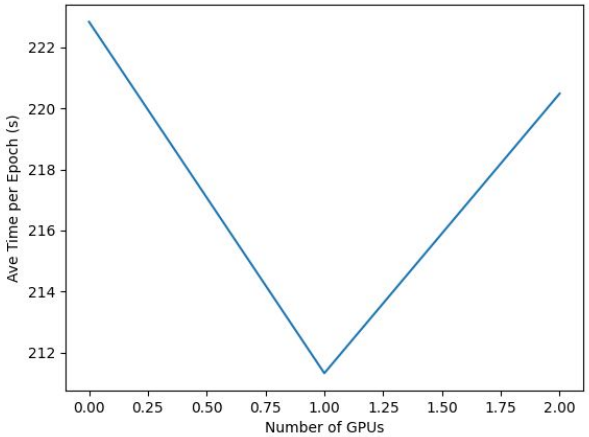
Semi-Supervised Model



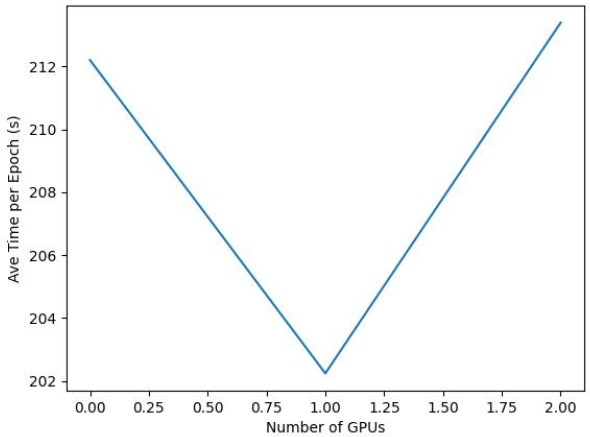
Performance Evaluation & Discussion

- Strong Scaling results are different from the theoretical expectations due to significant GPU-CPU communication overheads and averaging over a small number of epochs.

Fully-Supervised Model



Semi-Supervised Model



Conclusion & Future Work

- Successfully implemented a deep image classifier, both fully-supervised and semi-supervised models
    - Corroborated weak scaling while strong scaling could be extended to more epochs
- Leveraging Tensorboard for better insight into GPU analysis and potentially better optimizing the allocation of computations to multiple GPUs
- Explore different CNN architectures or parallelizing deployment of pre-trained models