

# Parallel Image Processing

**Benjamin Barnes, Stephanie McCumsey, and  
Richard Northen**

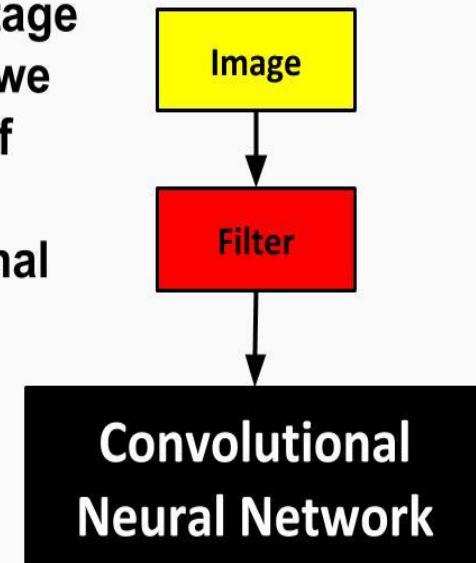
**Spring 2015**

# Problem Outline

- Training a CNN
  - Better performance with more training data
- Tasks applied to each image:
  - Pre-processing
- Our project: parallelize transformation of raw data
- Plenty more opportunities for parallelism

**Goal: take advantage of parallelism as we**

- a. Grow the set of images
- b. Create additional filters

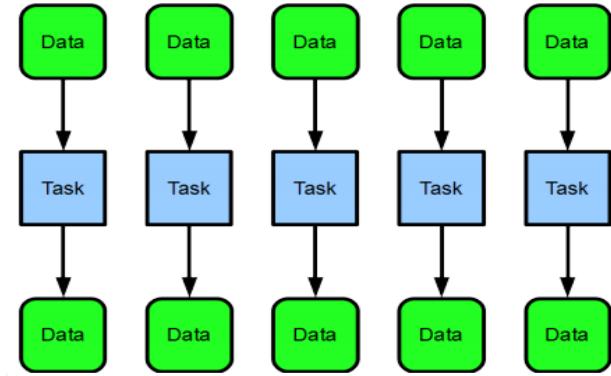


# Strategy for Parallelism

Data parallelism by processing the images in parallel

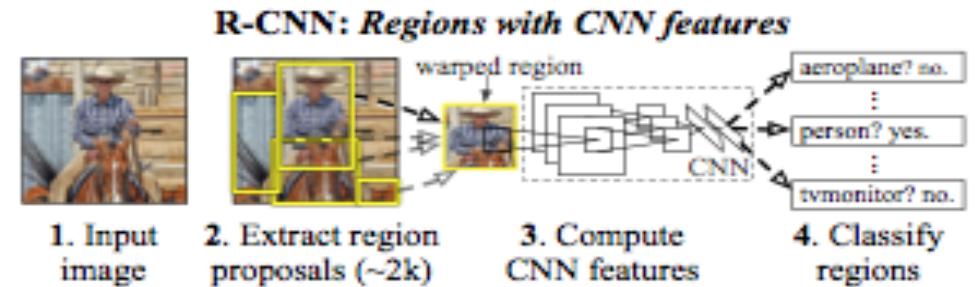
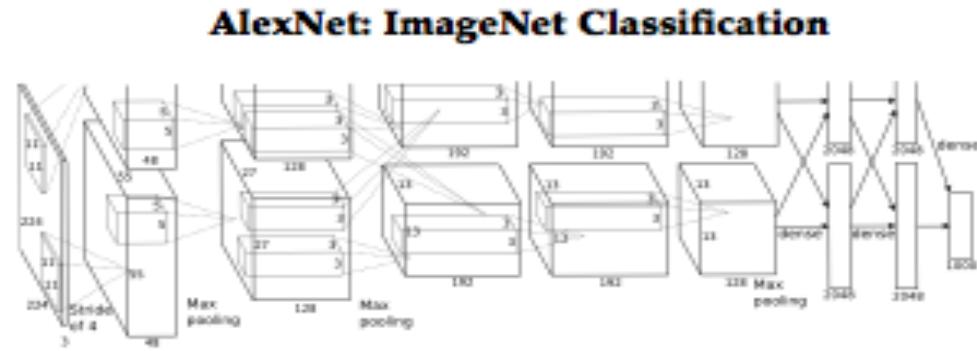
Task parallelism by way of the:

- **map** pattern to invert
- **map-reduce** pattern to overlay two images (add noise)
- **stencil** pattern to blur (image distortion)



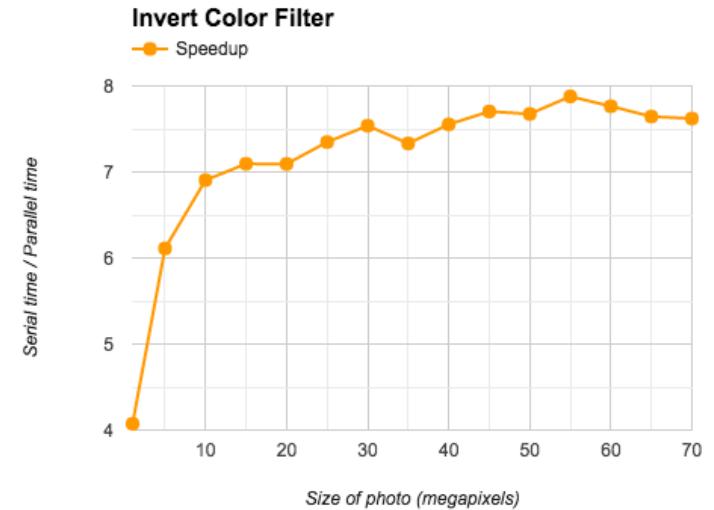
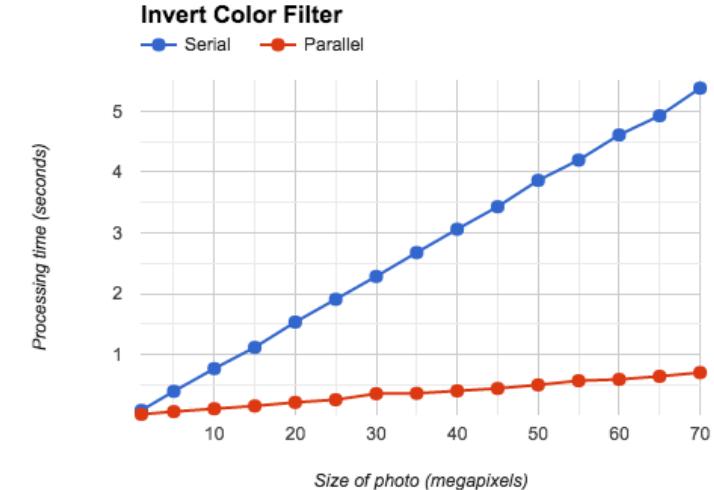
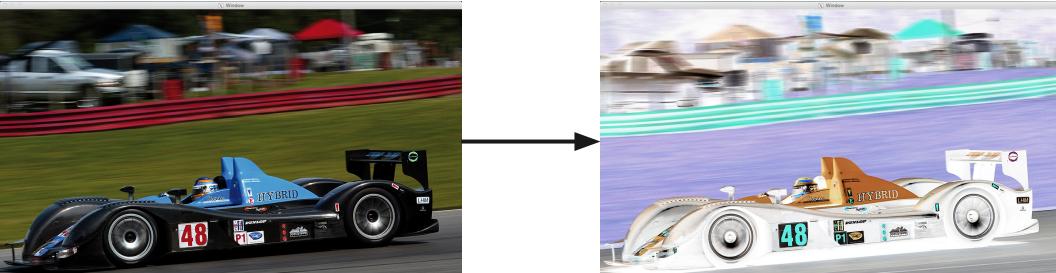
# Parallel Possibilities

- Krizhevsky et al trained a network with 1.2 million images
  - six days to train with two GPUs, 15.3% error rate reported in 2012
- Caffe: a framework for deep learning
  - design a convolutional neural network that uses raw images with minimal pre-processing
  - trains the network on GPU(s)



# Map

- Most basic image manipulation process
- Iterate over each pixel
  - Apply filter to pixel
    - Replace old pixel with modified pixel



# Map-Reduce

- Averaging RGB values of a single photo
- Overlaying multiple photos to see a potential match
  - Only works in very particular instances - matching letters

0 0 0 1 1 1 1 1 2

2 2 2 2 2 2 3 3 3

3 4 4 4 4 5 5 5 5

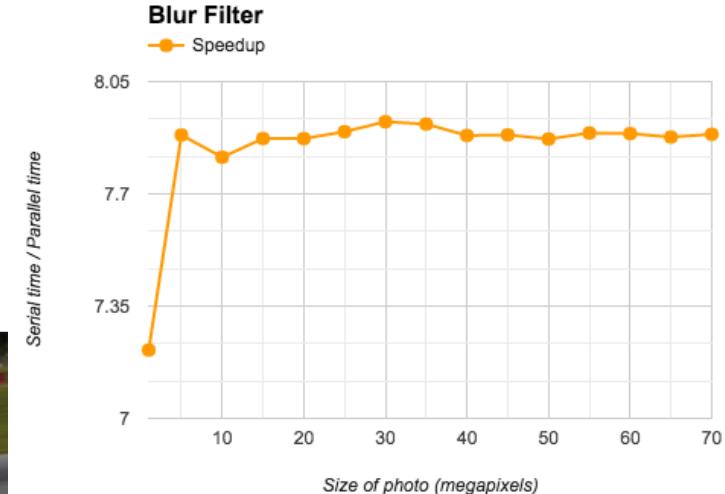
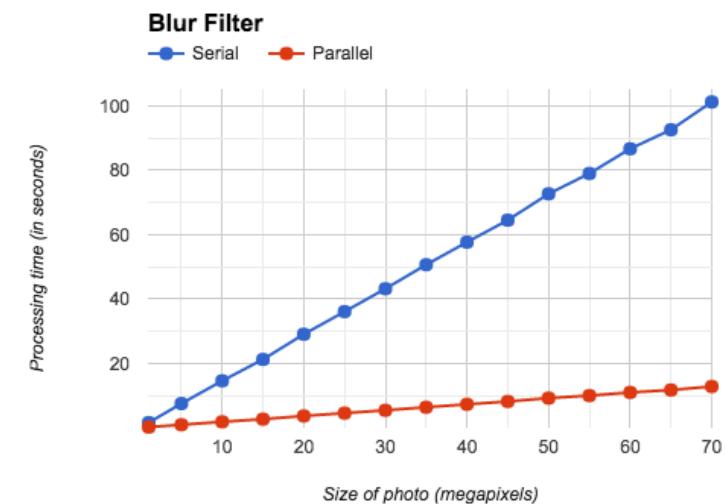
6 6 7 7 7 7 8 8 8

9 9 9 9 9 9 9 9



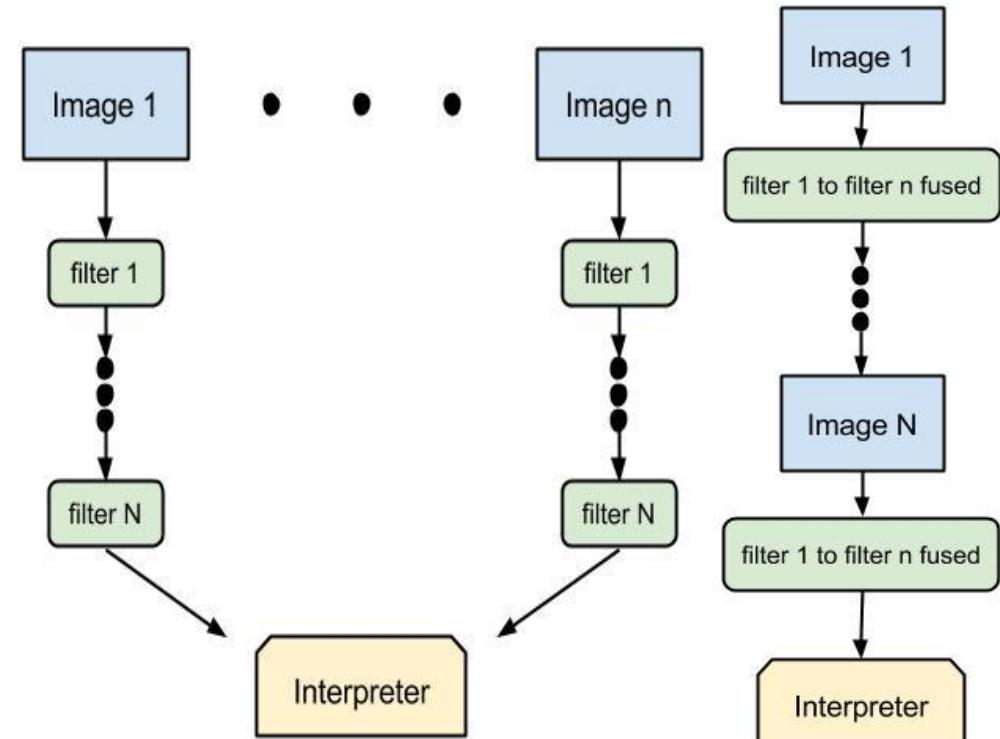
# Stencil (kernel)

- Blur
- Sobel kernel
  - Uses two nine point stencils to approximate derivatives



# Batch Processing & Pipeline

- No dependencies - many opportunities for parallelism
- While it depends on photo size, usually best utilized on a per-photo basis
  - Photos are typically standardized
- Processing 50 5MP photos
  - **Serial:** 402.308 seconds
  - **Left:** 55.1543 seconds
  - **Right:** 62.9466 seconds



# Summary

- Parallelism can be exposed easily, in a variety of ways, and to each facet of a machine learning pipeline
  - As shown by our data, we were limited by the WOPR hardware - using GPUs and CUDA would allow for a far greater speedup
  - Extend this project to enable learning with a neural net
  - Explore the performance of multiple frameworks that utilize the GPU(Caffe, Torch, Cuda-Convnet2)

# References

<http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf>

<http://yann.lecun.com/exdb/mnist/>

[http://www.cs.ucr.edu/~jtarango/cs122a\\_project.html](http://www.cs.ucr.edu/~jtarango/cs122a_project.html)