Deep Image: Scaling Up Image Recognition

Ren Wu, Shengen Yan, Yi Shan, Qingqing Dang, Gang Sun

Presented by: Jake Varley

Deep Image

- custom built supercomputer (Minwa)
- parallel algorithms for Minwa
- data augmentation techniques
- training with multi-scale high res images

Minwa: The Super Computer

It is possible that other approaches will yield the same results with less demand on the computational side. The authors of this paper argue that with more human effort being applied, it is indeed possible to see such results. However human effort is precisely what we want to avoid.

Minwa

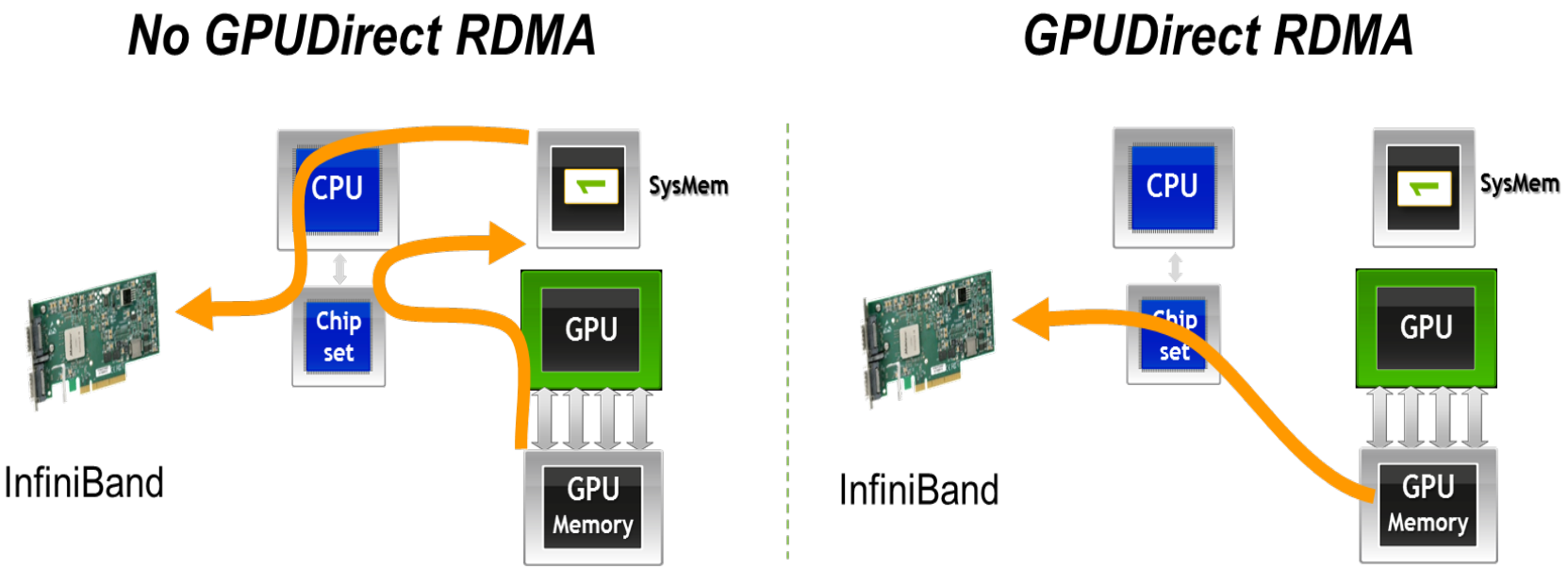
36 server nodes each with:

- 2 6-core Xeon E5-2620 processors
- 4 Nvidia Tesla K40m GPU’s
  - 12 Gb memory each
- 1 56GB/s FDR InfiniBand w/RDMA support

Remote Direct Memory Access

Direct memory access from the memory of one computer into that of another without involving either one’s operating system.

Remote Direct Memory Access



Minwa in total:

- 6.9TB host memory
- 1.7TB device memory
- 0.6PFlops theoretical single precision peak performance. PetaFlop = 10^15

Parallelism

- Data Parallelism: distributing the data across multiple processors

- Model Parallelism: distribute the model across multiple processors

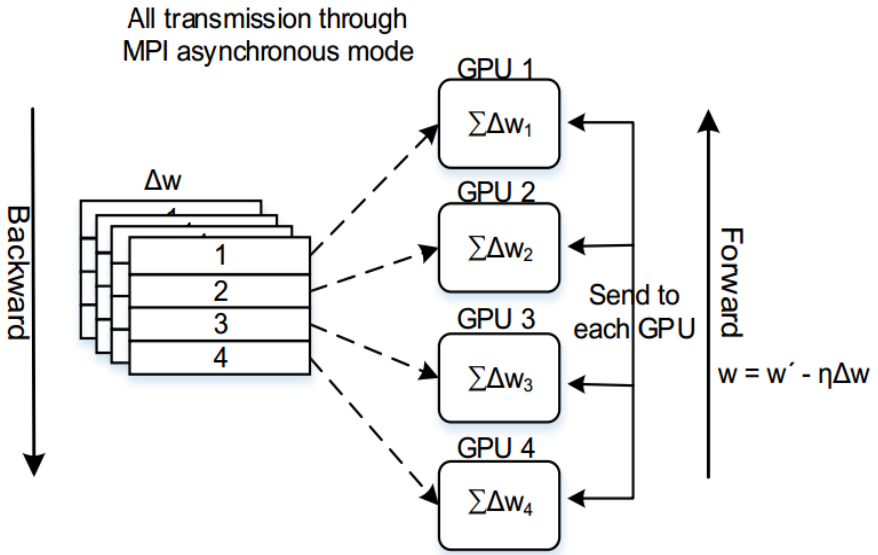
Data Parallelism

-Each GPU responsible for 1/Nth of a mini-batch and all GPUs work together on same mini-batch

-All GPUs compute gradients based on local training data and a local copy of weights. They then exchange gradients and update the local copy of weights.

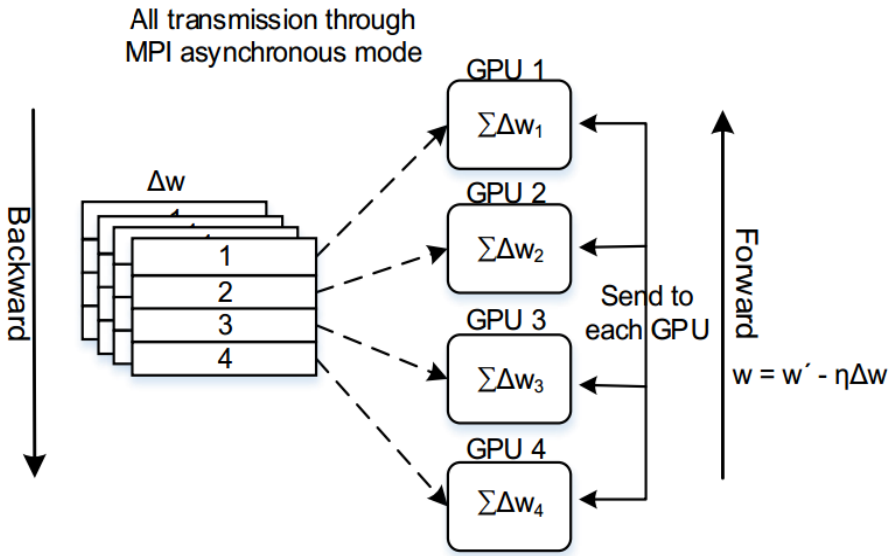
Butterfly Synchronization

GPU k receives the k-th layer's partial gradients from all other GPUs, accumulates them and broadcasts the result



Lazy Update

Don’t synchronize until corresponding weight parameters are needed

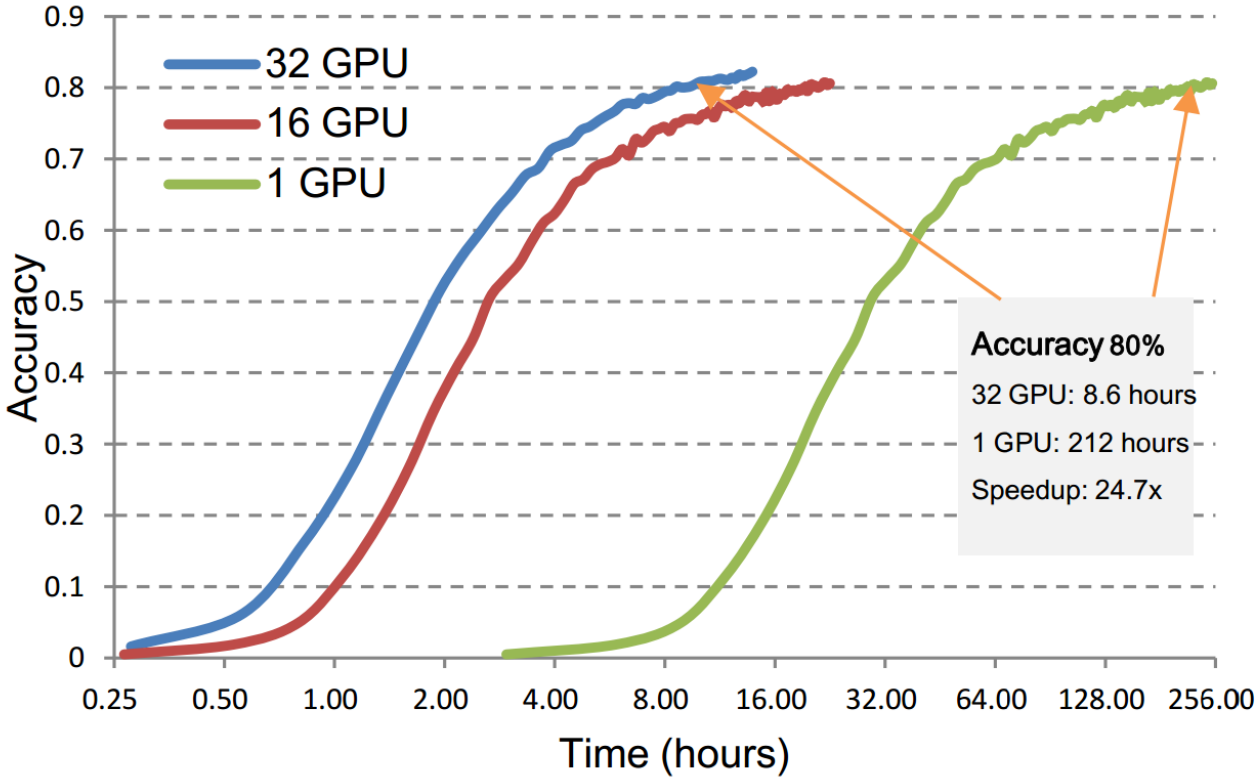


Model Parallelism

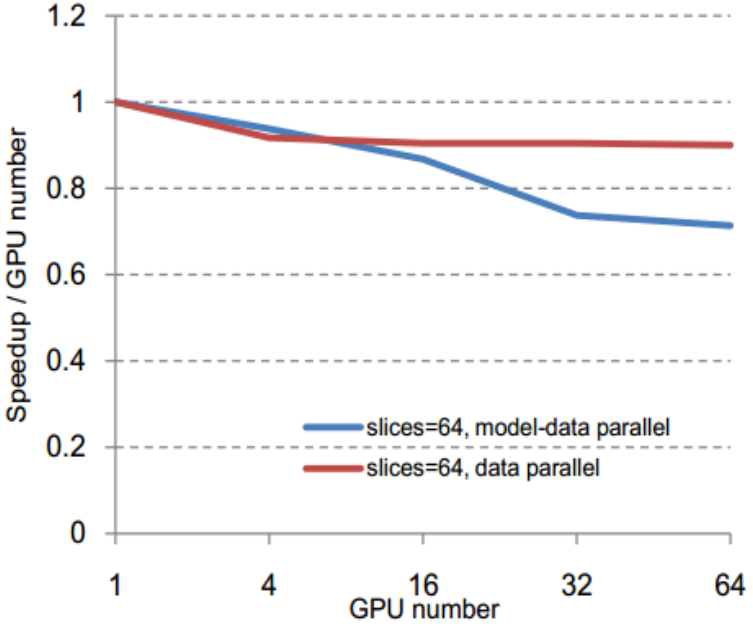
- Data Parallelism in convolutional layers

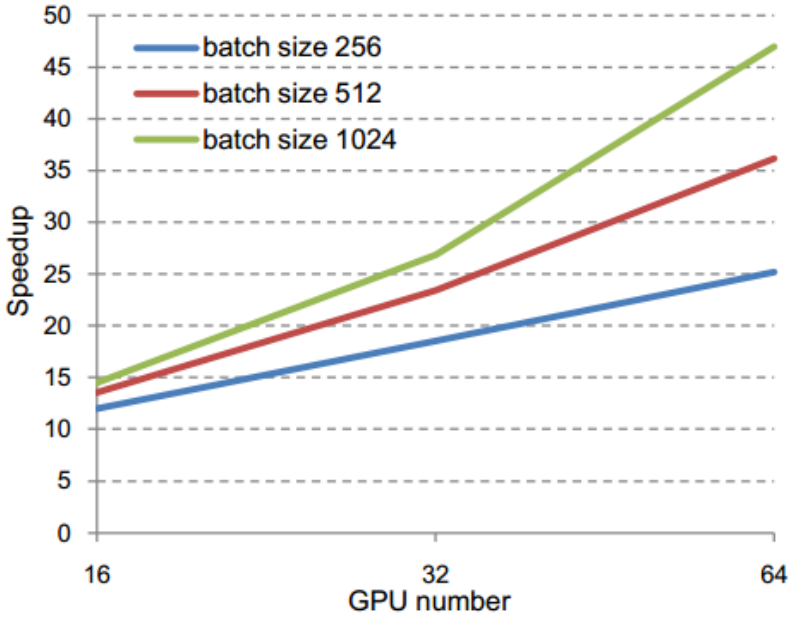
- Split fully connected layers across multiple GPUs

Scaling Efficiency



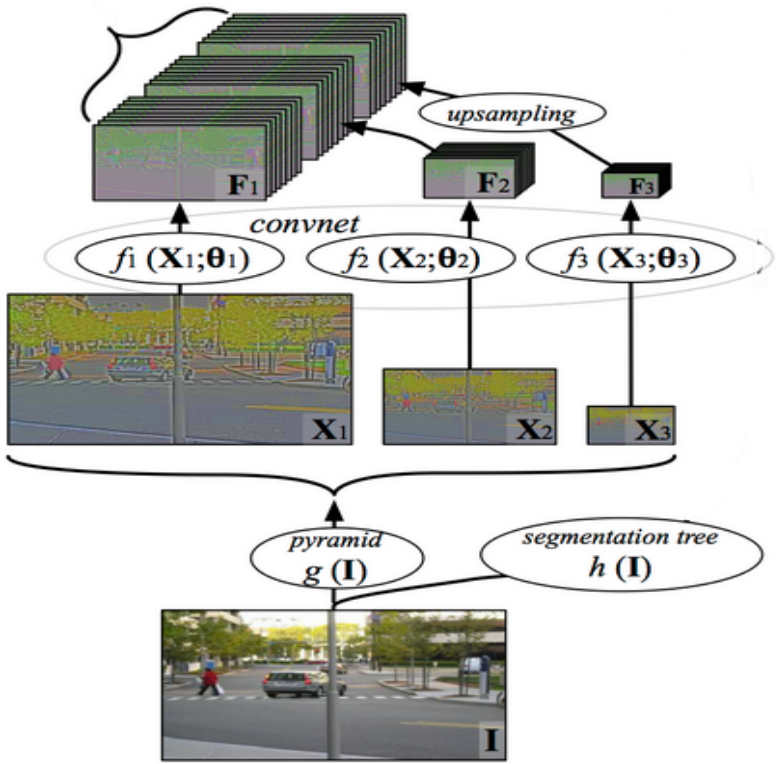
Scaling Efficiency





Data Augmentation

Previous Multi-Scale Approaches



Farabet et al. 2013

Multi-scale Training

- train several models at different resolutions

- combined by averaging softmax class posteriors

Image Resolution

- 224x224 vs 512x512

Advantage of High Res Input

Original image

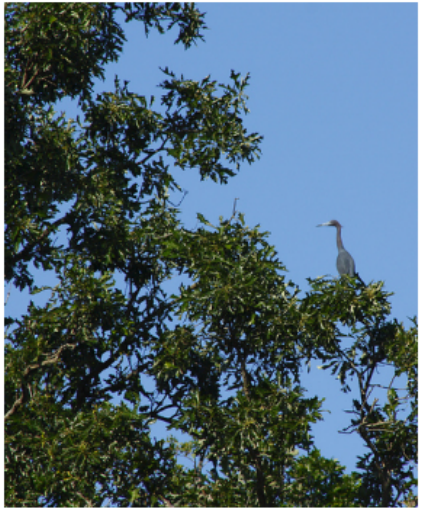
Low-resolution model

Rank	Score	Class
1	0.2287	ant
2	0.0997	damselfly
3	0.057	nematode
4	0.0546	chainlink fence
5	0.0522	long-horned
6	0.0307	walking stick
7	0.0287	dragonfly
8	0.0267	tiger beetle
9	0.0225	doormat
10	0.0198	flute
11	0.0198	grey whale
12	0.0178	mantis
13	0.0171	lacewing
14	0.0161	radiator
15	0.0161	scabbard
16	0.0157	slide rule
17	0.0148	fly
18	0.0129	leafhopper
19	0.0101	cucumber
20	0.0094	velvet

High-resolution model

Rank	Score	Class
1	0.103	lacewing
2	0.074	dragonfly
3	0.074	damselfly
4	0.063	walking stic
5	0.039	long-horned
6	0.027	leafhopper
7	0.025	nail
8	0.023	grasshopper
9	0.019	ant
10	0.015	mantis
11	0.015	fly
12	0.013	hammer
13	0.012	American
14	0.012	gar
15	0.011	chainlink
16	0.011	padlock
17	0.011	tree frog
18	0.011	cicada
19	0.01	screwdriver
20	0.01	harvestman

Difficult for low resolution

			
Tricycle	bbb Washer	Backpack	Little blue heron

Complimentary Resolutions

Model	Error Rate
256 x 256	7.96%
512 x 512	7.42%
Average of both	6.97%

Architecture

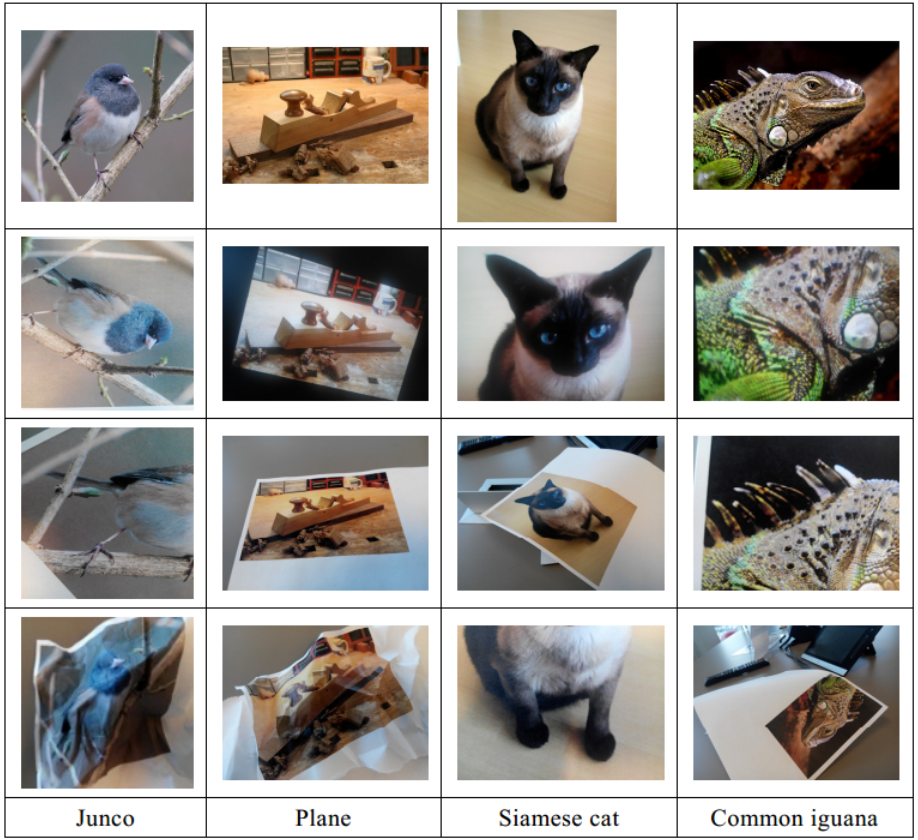
6 models combined with simple averaging

- trained for different scales

Single model:

Layers	Conv 1-2	Maxpool	Conv 3-4	Maxpool
# filters	64		128	
Conv 5-6-7		Maxpool	Conv 8-9-10	Maxpool
256			512	
Conv 11-12-13		Maxpool	FC 1-2	FC 2	Softmax
512			6144	1000	

Robust to Transformations



Summary

Everything was done as simply as possible on a supercomputer.