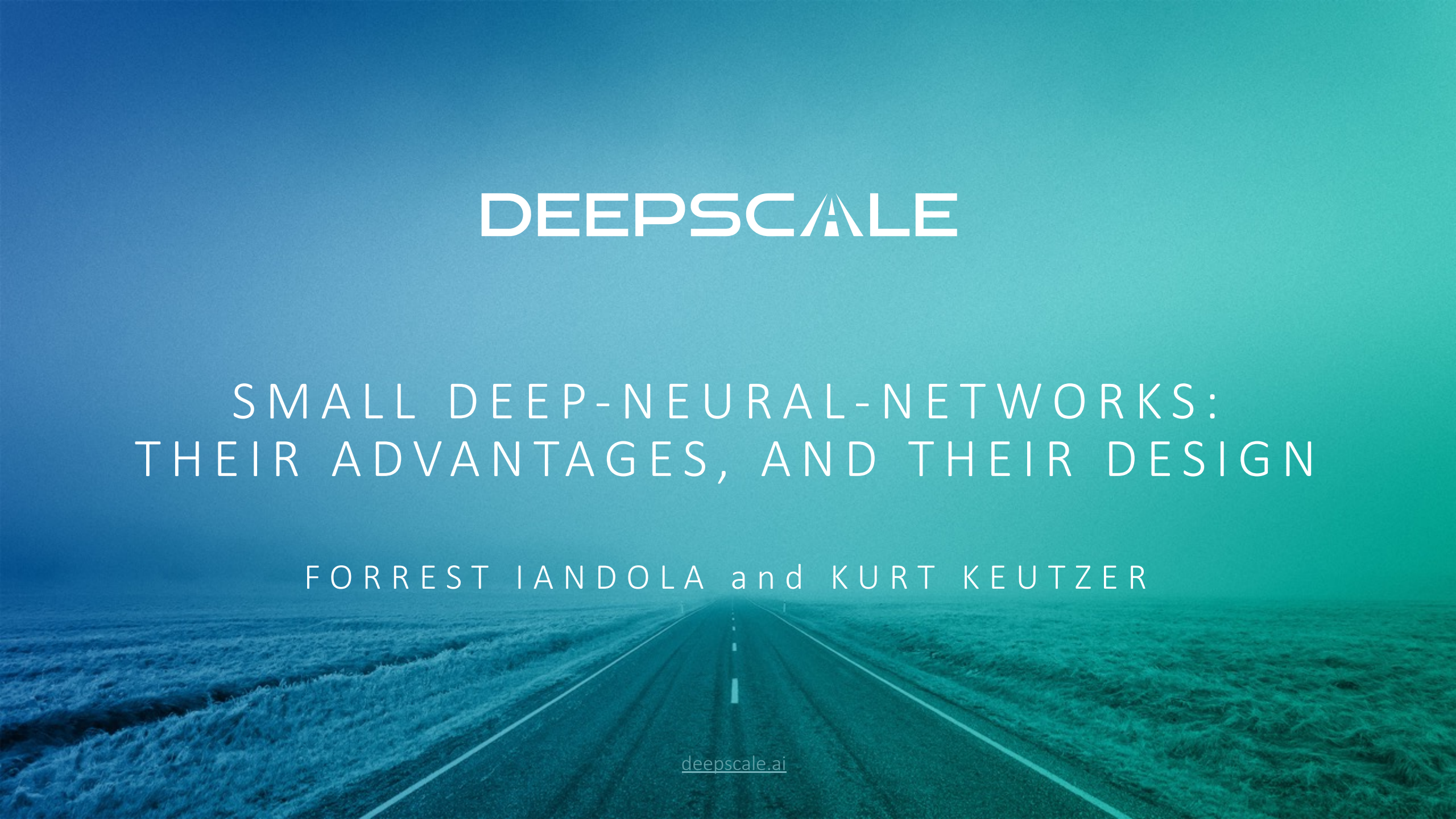


SMALL DEEP-NEURAL-NETWORKS: THEIR ADVANTAGES, AND THEIR DESIGN

FORREST IANDOLA and KURT KEUTZER

deepscale.ai

COMPUTER VISION FINALLY WORKS – NOW WHAT?

ADVANTAGES OF SMALL DEEP NEURAL NETWORKS

DESIGN OF SMALL DEEP NEURAL NETWORKS

OVERVIEW

similarly large accuracy improvements on tasks such as:

- semantic segmentation
- object detection
- 3D reconstruction
- …and so on

COMPUTER VISION FINALLY WORKS.

NOW WHAT?

DATACENTERS

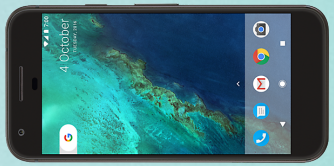
SOCIAL MEDIA ANALYSIS

WEB INDEXING

GOVERNMENT INTELLIGENCE

GADGETS

SELF-DRIVING CARS



SMARTPHONES

MQ-9 REAPER

DJI PHANTOM 4 DRONE

VENUES FOR COMMERCIAL COMPUTER VISION USAGE

DATACENTERS

RARELY SAFETY-CRITICAL

LOW-POWER IS NICE-TO-HAVE

REAL-TIME IS PREFERRABLE

GADGETS

USUALLY SAFETY-CRITICAL
(except smartphones)

LOW-POWER IS REQUIRED

REAL-TIME IS REQUIRED

THIS TALK IS ESPECIALLY TARGETED AT GADGETS

KEY REQUIREMENTS FOR COMMERCIAL COMPUTER VISION USAGE

DESIRABLE PROPERTIES:

- sufficiently high accuracy
- low computational complexity
- low energy usage
- small model size

WHAT'S THE "RIGHT" NEURAL NETWORK FOR USE IN A GADGET?

SMALL DNNs TRAIN FASTER ON DISTRIBUTED HARDWARE

SMALL DNNs ARE MORE DEPLOYABLE ON EMBEDDED PROCESSORS

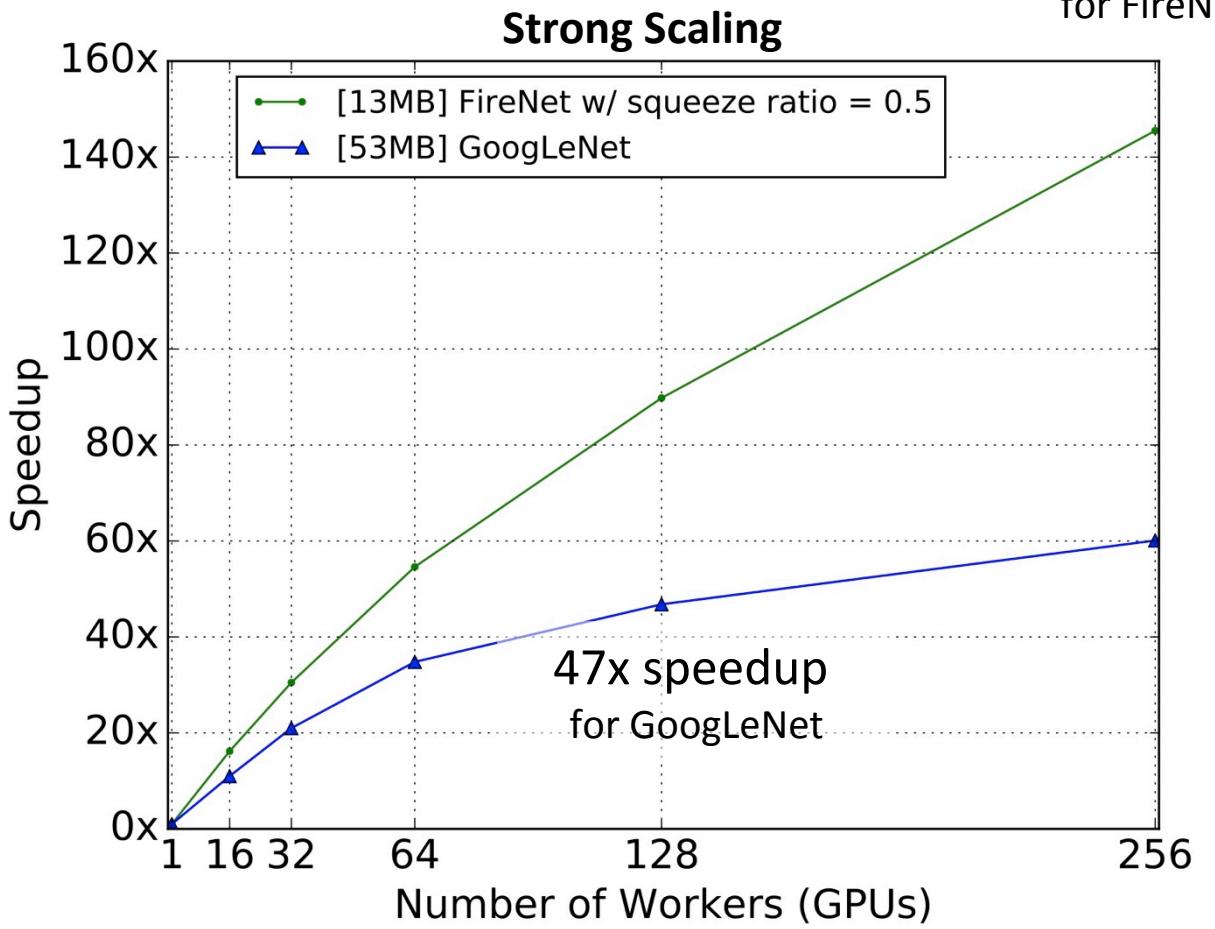
SMALL DNNs ARE EASILY UPDATABLE OVER-THE-AIR (OTA)

WHY *SMALL* DEEP-NEURAL-NETWORKS?

Small Models Have Big Advantages #1

- Fewer parameter weights means bigger opportunities for scaling training – 145X speedup on 256 GPUs

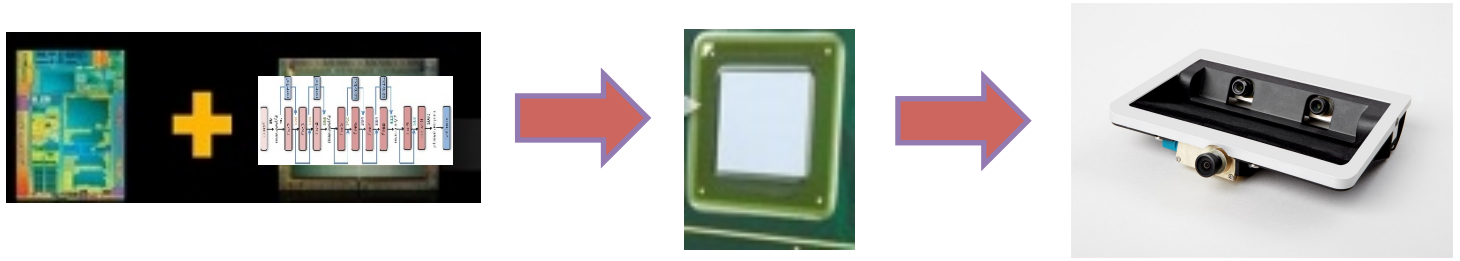
145x speedup
for FireNet



Using FireCaffe on the Titan cluster CVPR 2016

Small Models Have Big Advantages #2

- SqueezeNet's smaller number of weights enables complete on-chip integration of CNN model with weights – no need for off-chip memory
  - Dramatically reduces the energy for computing inference
  - Gives the potential for pushing the data processing (i.e. CNN model) up-close and personal to the data gathering (e.g. onboard cameras and other sensors)



Single-chip Integration of CNN Model

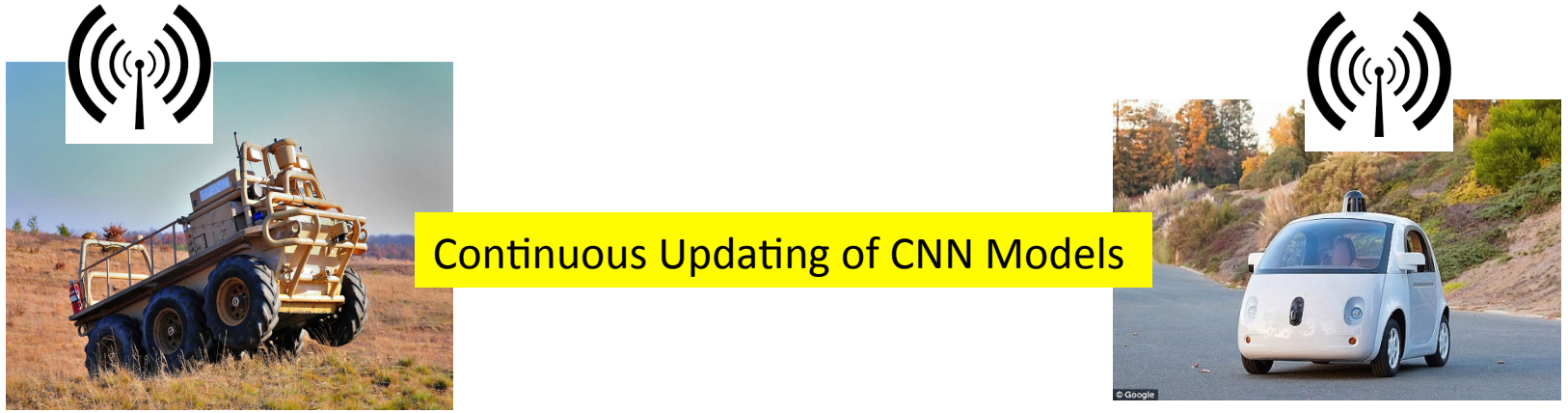
Closer integration with sensor (e.g. Camera)

Limited memory of embedded devices makes small models absolutely essential for many applications

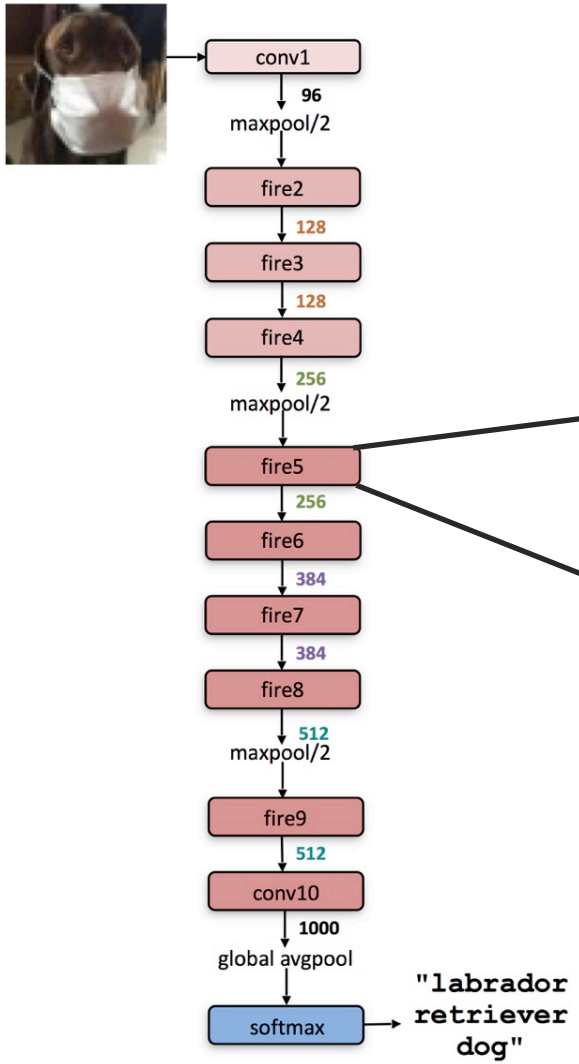
•

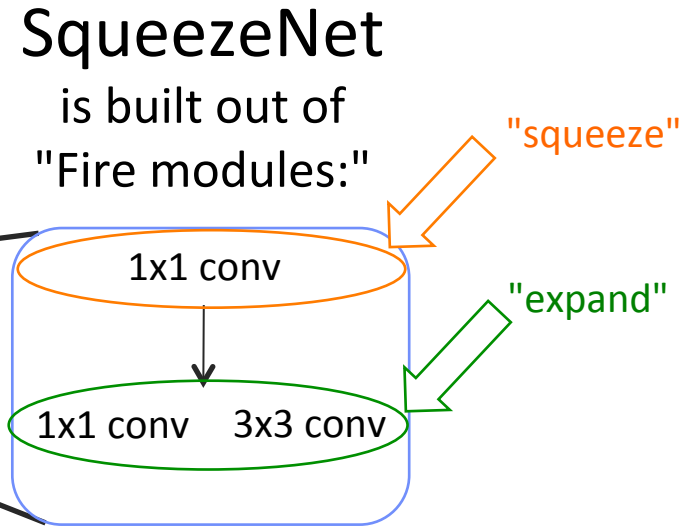
Small Models Have Big Advantages #3

- Small models enable continuous wireless updates of models
- Each time any sensor discovers a new image/situation that requires retraining, all models should be updated
- Data is uploaded to cloud and used for retraining
- But … how to update all the vehicles that are running the model?
- At 500KB downloading new model parameters is easy.



ADVANCES IN SMALL DNNs
IN THE PAST 18 MONTHS





[1] **F.N. Iandola**, S. Han, M. Moskewicz, K. Ashraf, W. Dally, K. Keutzer. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size. arXiv, 2016.

http://github.com/DeepScale/SqueezeNet

SqueezeNet

Compression Approach	DNN Architecture	Original Model Size	Compressed Model Size	Reduction in Model Size vs. AlexNet	Top-1 ImageNet Accuracy	Top-5 ImageNet Accuracy
None (baseline)	AlexNet [1]	240MB	240MB	1x	57.2%	80.3%
SVD [2]	AlexNet	240MB	48MB	5x	56.0%	79.4%
Network Pruning [3]	AlexNet	240MB	27MB	9x	57.2%	80.3%
Deep Compression [4]	AlexNet	240MB	6.9MB	35x	57.2%	80.3%

[1] A. Krizhevsky, I. Sutskever, G.E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. NIPS, 2012.

[2] E.L .Denton, W. Zaremba, J. Bruna, Y. LeCun, R. Fergus. Exploiting linear structure within convolutional networks for efficient evaluation. NIPS, 2014.

[3] S. Han, J. Pool, J. Tran, W. Dally. Learning both Weights and Connections for Efficient Neural Networks, NIPS, 2015.

[4] S. Han, H. Mao, W. Dally. Deep Compression…, arxiv:1510.00149, 2015.

[5] **F.N. Iandola**, **M. Moskewicz**, K. Ashraf, S. Han, W. Dally, **K. Keutzer**. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <1MB model size. arXiv, 2016.

SqueezeNet

Compression Approach	DNN Architecture	Original Model Size	Compressed Model Size	Reduction in Model Size vs. AlexNet	Top-1 ImageNet Accuracy	Top-5 ImageNet Accuracy
None (baseline)	AlexNet [1]	240MB	240MB	1x	57.2%	80.3%
SVD [2]	AlexNet	240MB	48MB	5x	56.0%	79.4%
Network Pruning [3]	AlexNet	240MB	27MB	9x	57.2%	80.3%
Deep Compression [4]	AlexNet	240MB	6.9MB	35x	57.2%	80.3%
None	**SqueezeNet [5] (ours)**	**4.8MB**	**4.8MB**	**50x**	57.5%	80.3%

[1] A. Krizhevsky, I. Sutskever, G.E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. NIPS, 2012.

[2] E.L .Denton, W. Zaremba, J. Bruna, Y. LeCun, R. Fergus. Exploiting linear structure within convolutional networks for efficient evaluation. NIPS, 2014.

[3] S. Han, J. Pool, J. Tran, W. Dally. Learning both Weights and Connections for Efficient Neural Networks, NIPS, 2015.

[4] S. Han, H. Mao, W. Dally. Deep Compression…, arxiv:1510.00149, 2015.

[5] **F.N. Iandola**, **M. Moskewicz**, K. Ashraf, S. Han, W. Dally, **K. Keutzer**. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <1MB model size. arXiv, 2016.

SqueezeNet

Compression Approach	DNN Architecture	Original Model Size	Compressed Model Size	Reduction in Model Size vs. AlexNet	Top-1 ImageNet Accuracy	Top-5 ImageNet Accuracy
None (baseline)	AlexNet [1]	240MB	240MB	1x	57.2%	80.3%
SVD [2]	AlexNet	240MB	48MB	5x	56.0%	79.4%
Network Pruning [3]	AlexNet	240MB	27MB	9x	57.2%	80.3%
Deep Compression [4]	AlexNet	240MB	6.9MB	35x	57.2%	80.3%
None	**SqueezeNet [5] (ours)**	**4.8MB**	**4.8MB**	**50x**	57.5%	80.3%
Deep Compression [4]	**SqueezeNet [5] (ours)**	**4.8MB**	**0.47MB**	**510x**	57.5%	80.3%

[1] A. Krizhevsky, I. Sutskever, G.E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. NIPS, 2012.

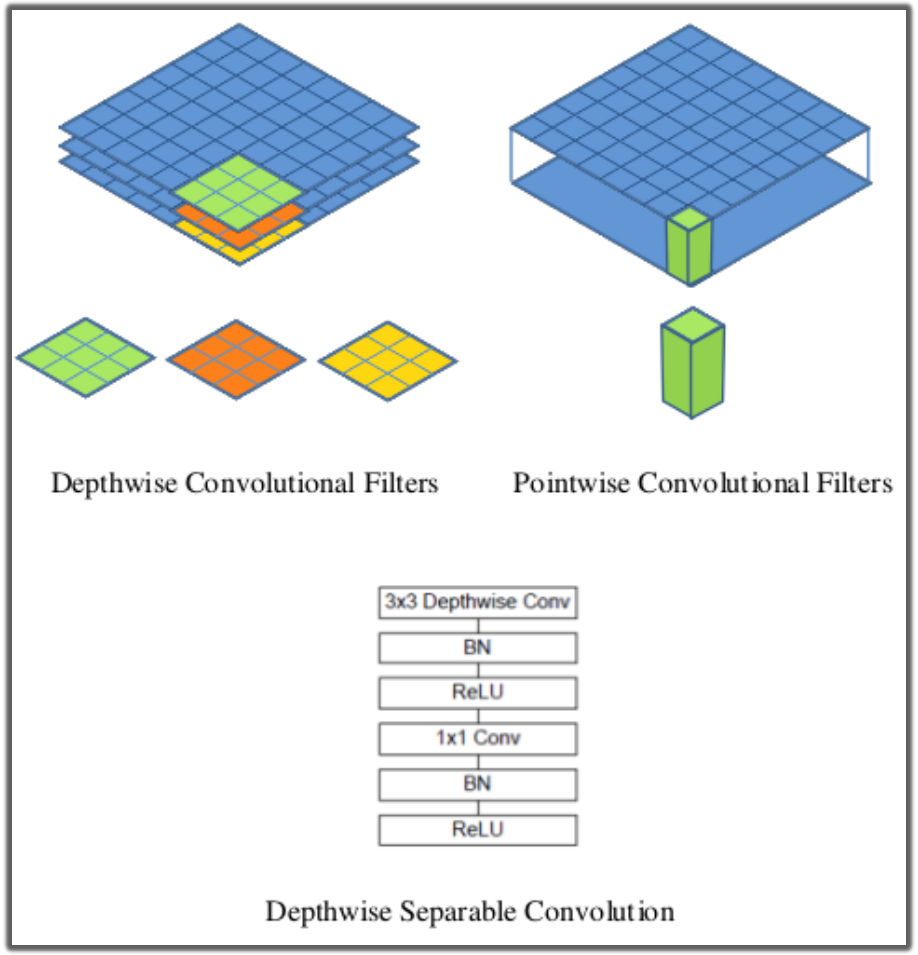
[2] E.L .Denton, W. Zaremba, J. Bruna, Y. LeCun, R. Fergus. Exploiting linear structure within convolutional networks for efficient evaluation. NIPS, 2014.

[3] S. Han, J. Pool, J. Tran, W. Dally. Learning both Weights and Connections for Efficient Neural Networks, NIPS, 2015.

[4] S. Han, H. Mao, W. Dally. Deep Compression…, arxiv:1510.00149, 2015.

[5] **F.N. Iandola**, **M. Moskewicz**, K. Ashraf, S. Han, W. Dally, **K. Keutzer**. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <1MB model size. arXiv, 2016.

SqueezeNet



slide credit: https://github.com/Zehaos/MobileNet

MobileNets

Howard, et al. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. arXiv, 2017.

roughly the size of SqueezeNet and more accurate

Table 4. Depthwise Separable vs Full Convolution MobileNet

Model	ImageNet Accuracy	Million Mult-Adds	Million Parameters
Conv MobileNet	71.7%	4866	29.3
MobileNet	70.6%	569	4.2

Table 5. Narrow vs Shallow MobileNet

Model	ImageNet Accuracy	Million Mult-Adds	Million Parameters
0.75 MobileNet	68.4%	325	2.6
Shallow MobileNet	65.3%	307	2.9

Table 6. MobileNet Width Multiplier

Width Multiplier	ImageNet Accuracy	Million Mult-Adds	Million Parameters
1.0 MobileNet-224	70.6%	569	4.2
0.75 MobileNet-224	68.4%	325	2.6
0.5 MobileNet-224	63.7%	149	1.3
0.25 MobileNet-224	50.6%	41	0.5

Table 7. MobileNet Resolution

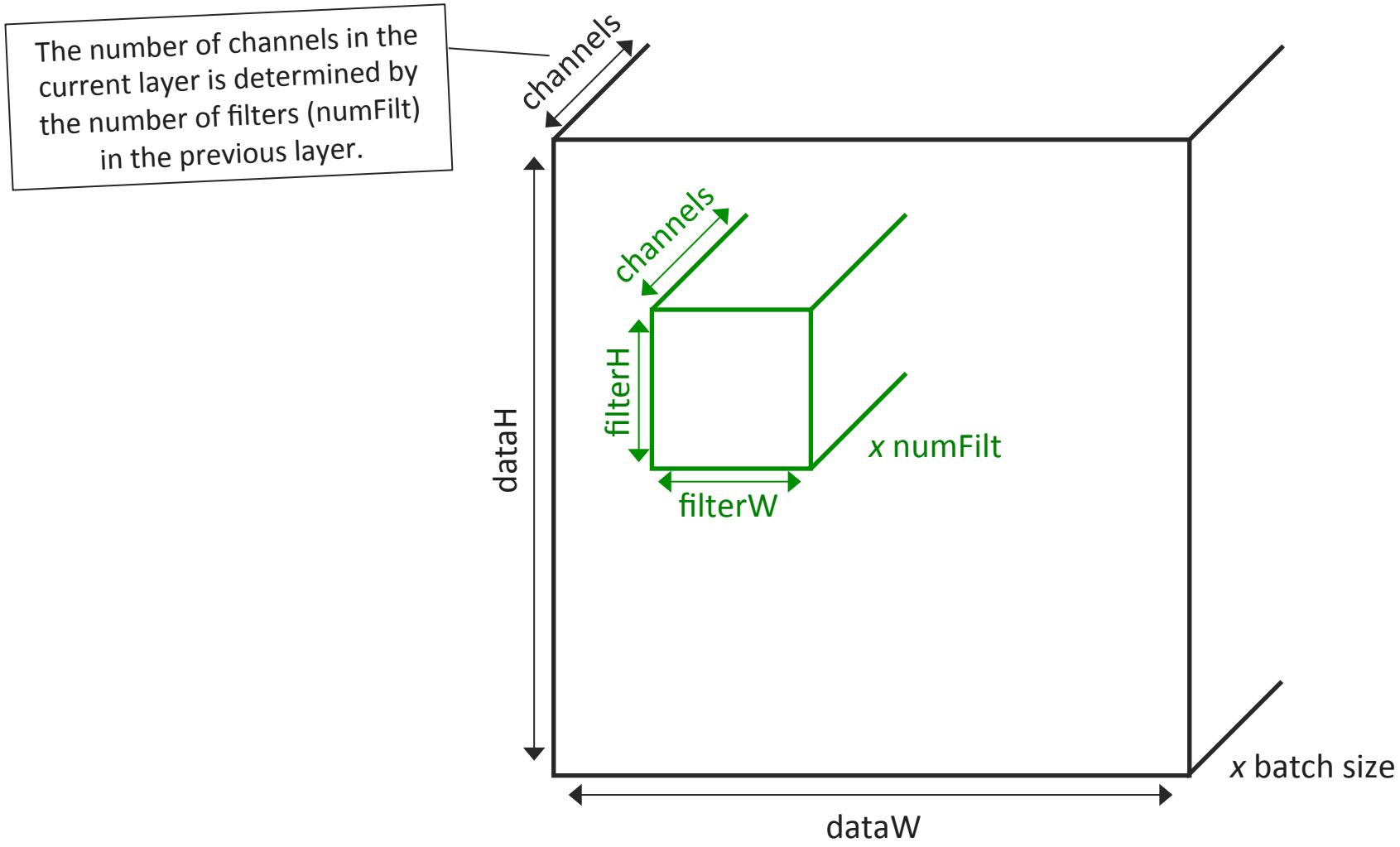
Resolution	ImageNet Accuracy	Million Mult-Adds	Million Parameters
1.0 MobileNet-224	70.6%	569	4.2
1.0 MobileNet-192	69.1%	418	4.2
1.0 MobileNet-160	67.2%	290	4.2
1.0 MobileNet-128	64.4%	186	4.2

The MobileNets paper reports a whole family of DNN architectures that the authors discovered while exploring the design space

MobileNets

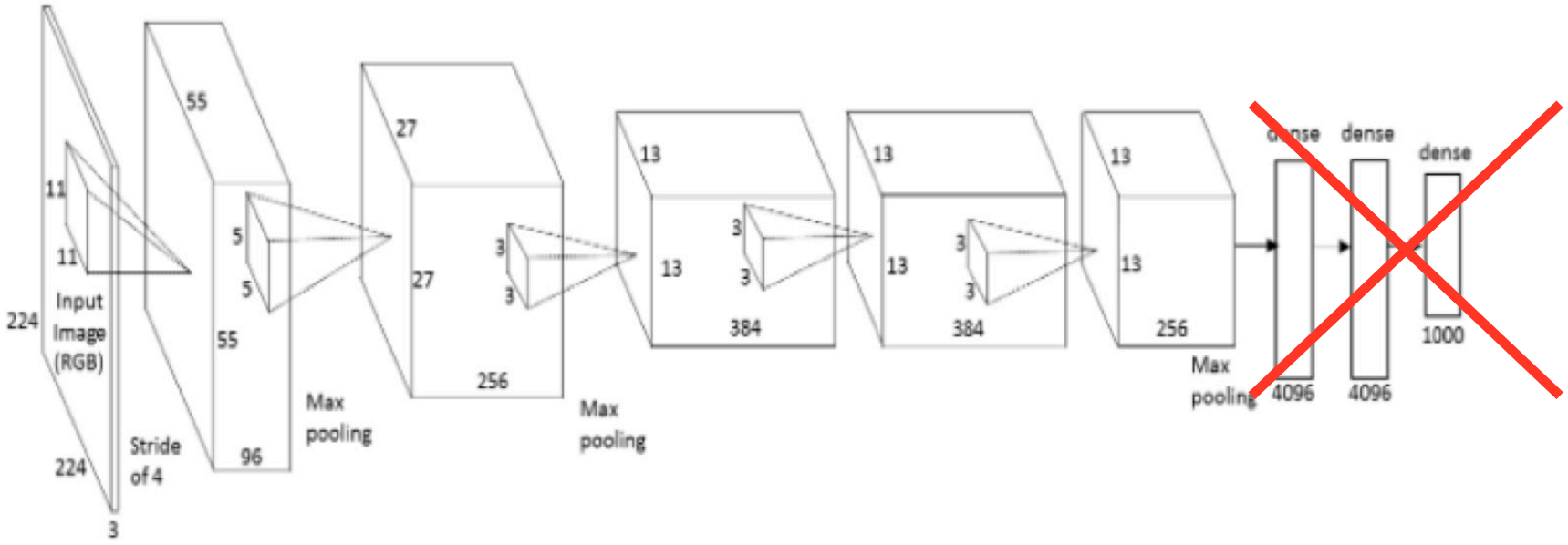
Howard, et al. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. arXiv, 2017.

OK. How do you create a small NN?



Anatomy of a convolution layer

IMPORTANT TO KNOW: MULTIPLE CHANNELS AND MULTIPLE FILTERS

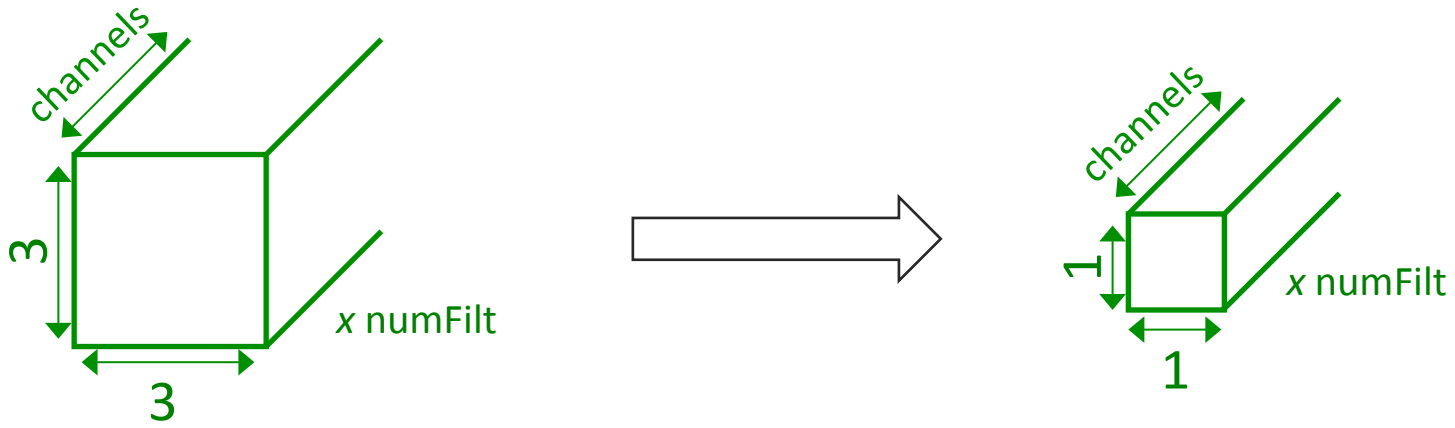


In AlexNet and VGG, the majority of the parameters are in the FC layers.

The FC7 layer in AlexNet has 4096 input channels and 4096 filters → 67MB of params

The mere presence of fully-connected layers is not the culprit for the high model size; the problem is that some FC layers of VGG and AlexNet have a huge number of channels and filters.

1. Replace Fully-Connected Layers with Convolutions



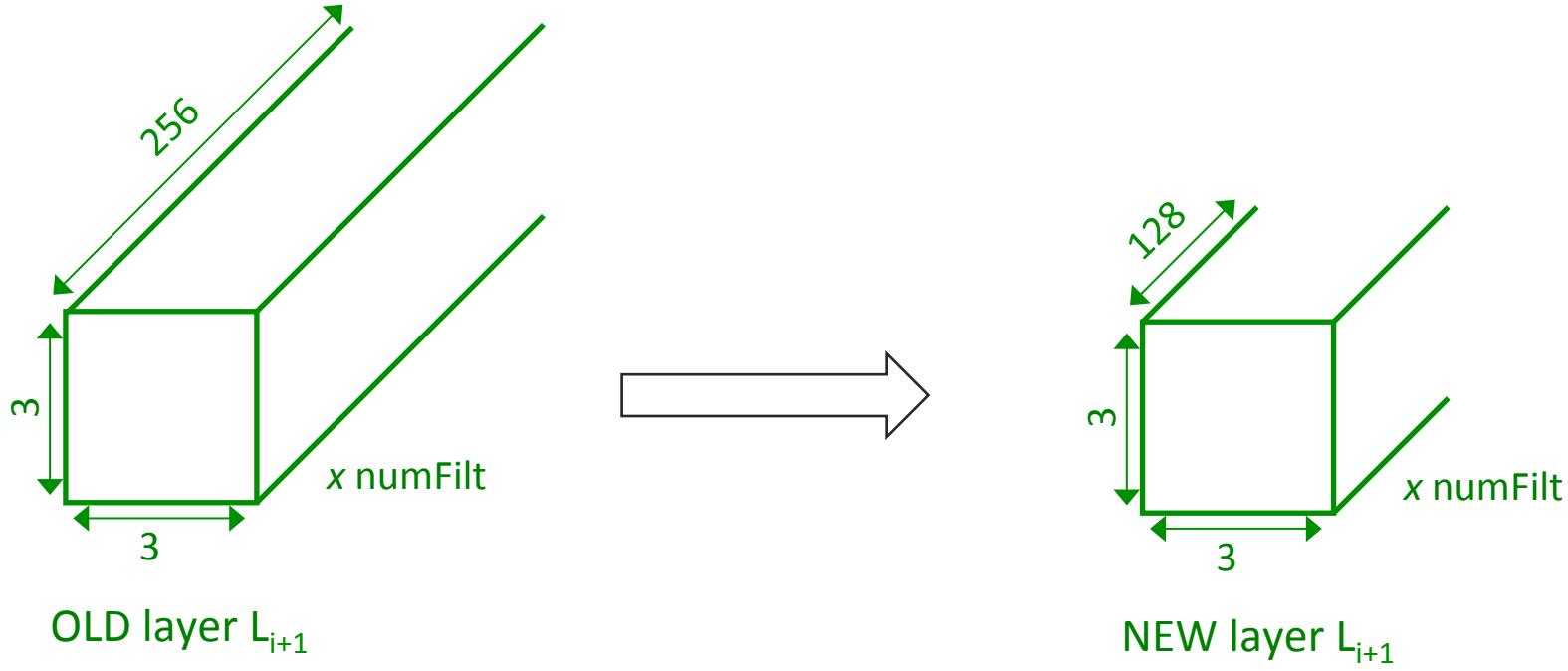
While 1x1 filters cannot see outside of a 1-pixel radius, they retain the ability to combine and reorganize information across channels.

In our design space exploration that led up to SqueezeNet, we found that we could replace half the 3x3 filters with 1x1's without diminishing accuracy

A "saturation point" is when adding more parameters doesn't improve accuracy.

2. Kernel Reduction

REDUCING THE HEIGHT AND WIDTH OF FILTERS



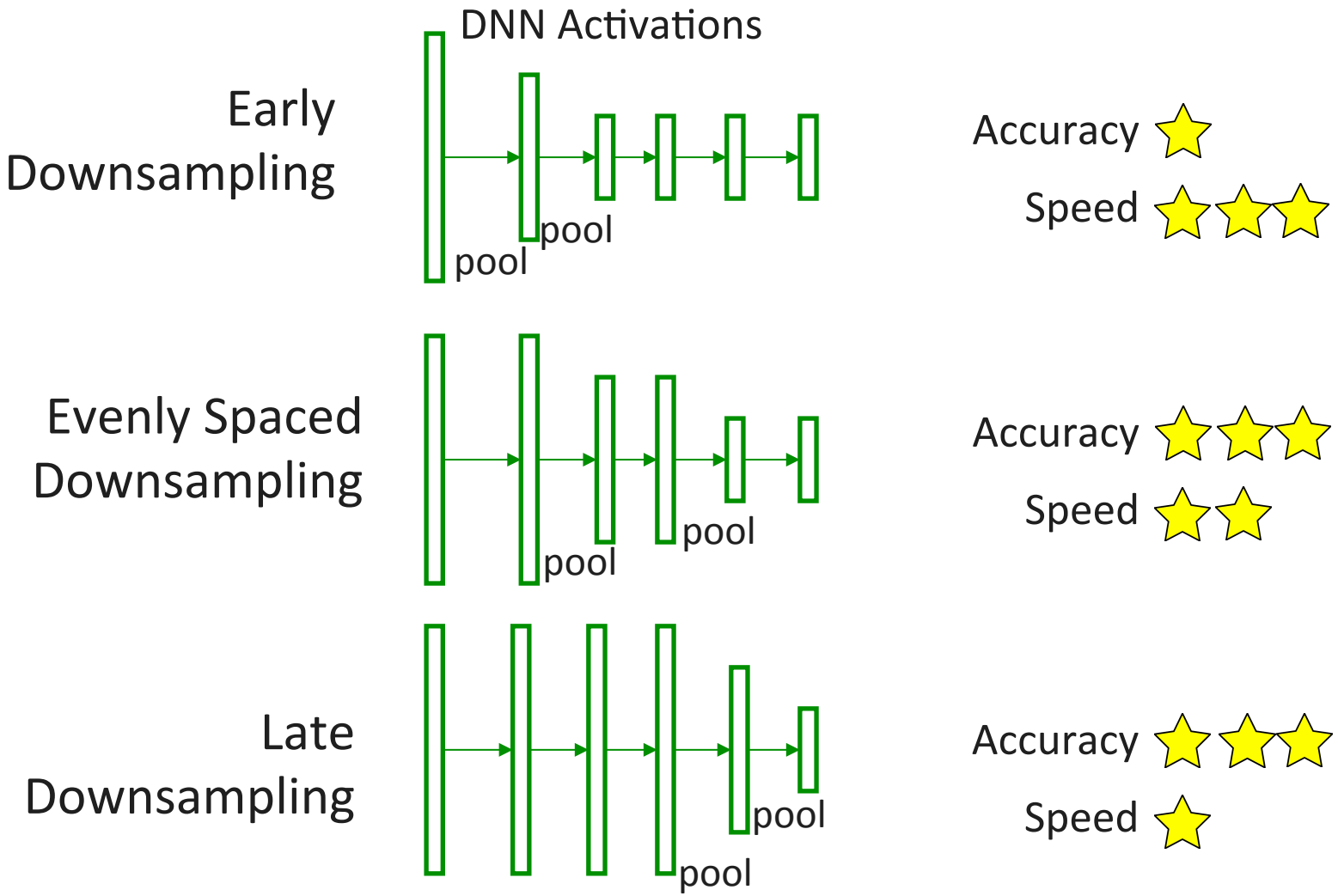
If we halve the number of filters in layer L_i

→ this halves the number of input channels in layer L_{i+1}

→ up to 4x reduction in number of parameters

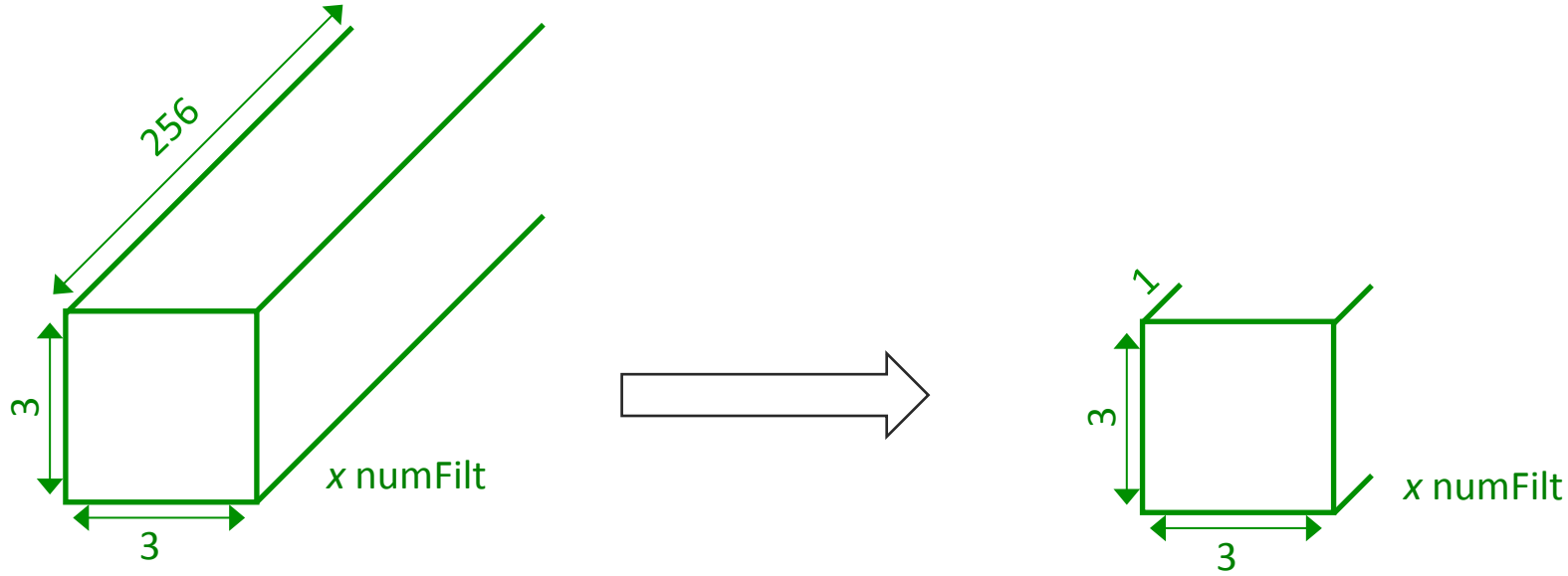
3. Channel Reduction

REDUCING THE NUMBER OF FILTERS AND CHANNELS



4. Evenly Spaced Downsampling

ACTIVATIONS DON'T CONTRIBUTE TO MODEL SIZE, BUT BIGGER ACTIVATIONS REQUIRE MORE COMPUTATION



Each 3x3 filter has 1 channel
Each filter gets applied to a different channel of the input

in 2017 papers such as: MobileNets, ResNeXt, "A Compact DNN"

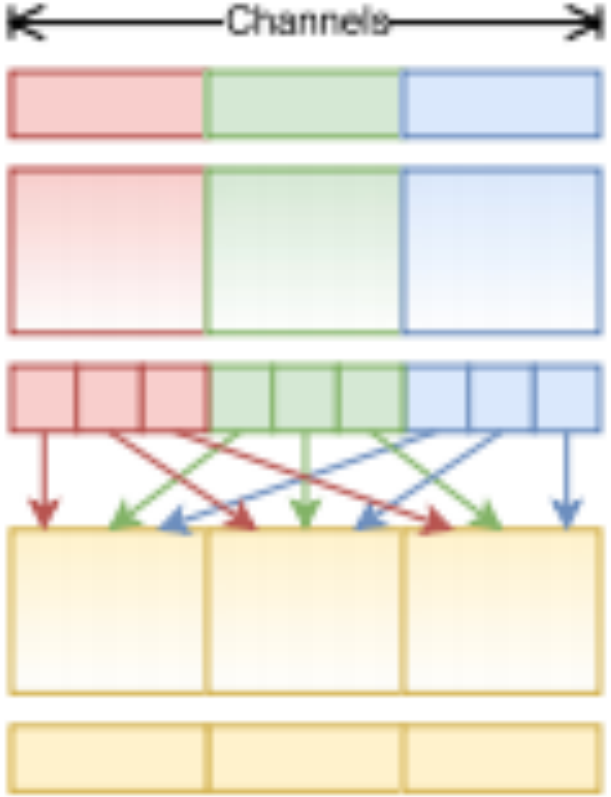
5. Depthwise Separable Convolutions

ALSO CALLED: "GROUP CONVOLUTIONS" or "CARDINALITY"

After applying the previous optimizations, we now have >90% of the parameters in 1x1 convolutions

Separable 1x1 convs would lead to multiple DNNs that don't communicate

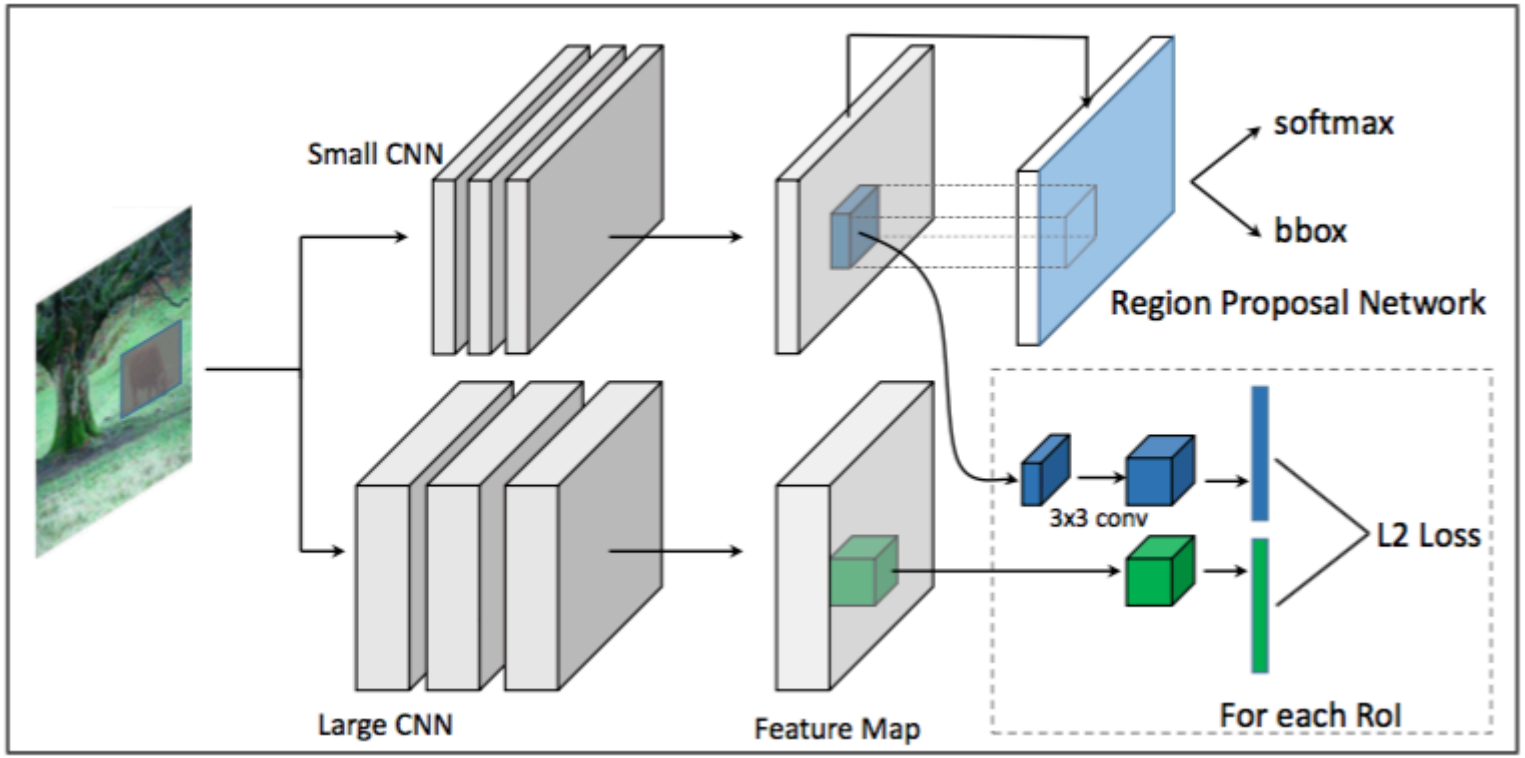
Recent approach (published last month): *shuffle* layer after separable 1x1 convs



"shuffle" layer

6. Shuffle Operations

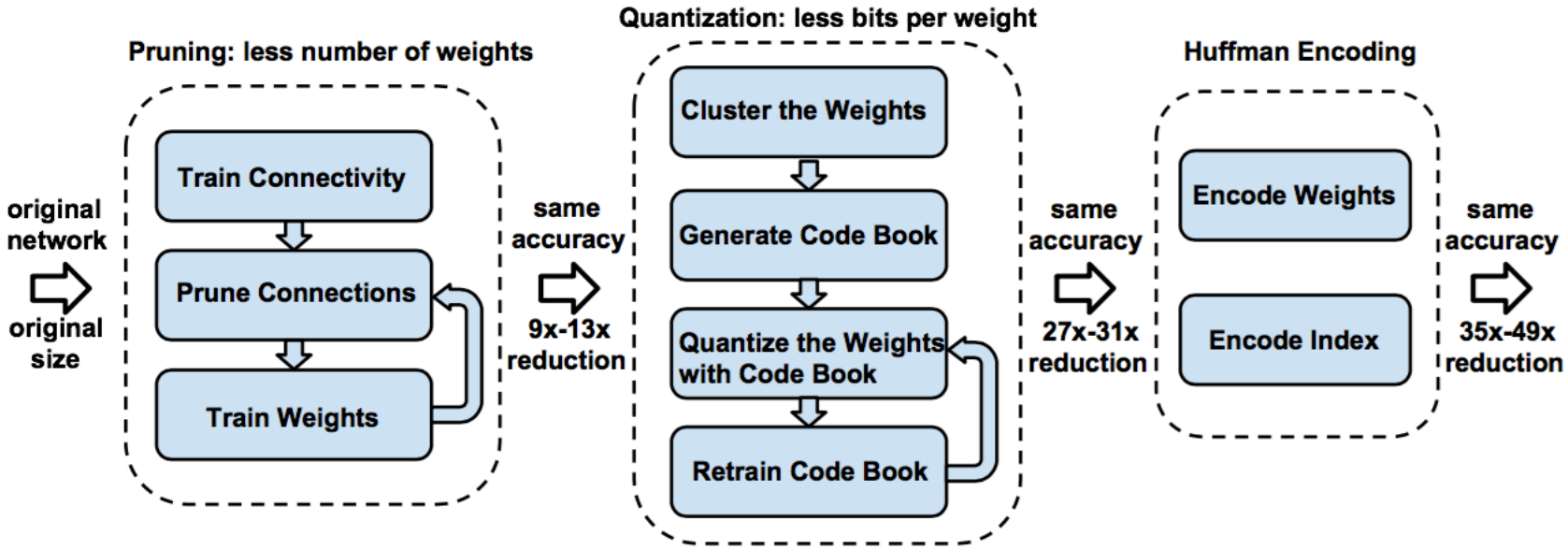
Zhang, et al. ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices. arXiv, 2017.



Model Distillation

Li, et al. Mimicking Very Efficient Network for Object Detection. CVPR, 2017.

7. Distillation & Compression



Deep Compression

Han, et al. Deep Compression: Compressing Deep Neural Networks... ICLR, 2016.

7. Distillation & Compression

Successful application of *small* DNNs to other computer vision tasks

https://github.com/lizeng614/SqueezeNet-Neural-Style-Pytorch

Style Transfer using SqueezeNet

Tim Anglade Follow

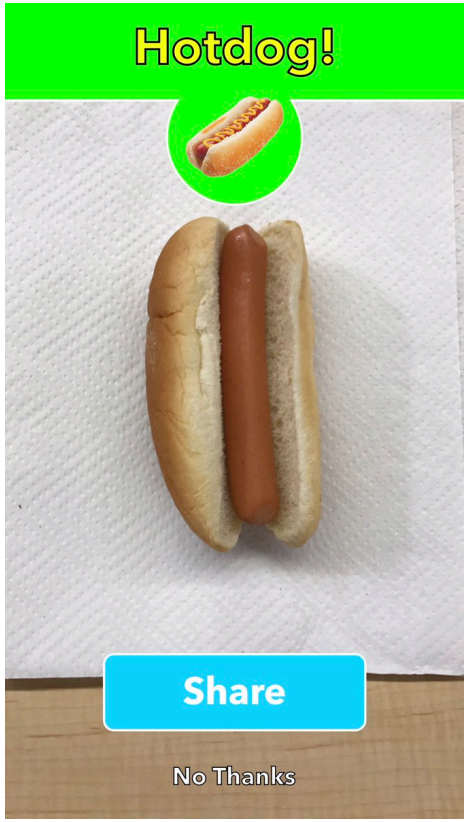
Startup guy working on the TV show Silicon Valley—timanglade@gmail.com

Jun 26 · 23 min read

How HBO's Silicon Valley built "Not Hotdog" with mobile TensorFlow, Keras & React Native

SqueezeNet powers Version 2 of the *Not Hotdog* app from the Silicon Valley TV show.

A variant of MobileNets powers Version 3.



Not Hotdog using SqueezeNet and MobileNets

Building Fast and Compact Convolutional Neural Networks for Offline Handwritten Chinese Character Recognition

Xuefeng Xiao[a], Lianwen Jin[a,*], Yafeng Yang[a], Weixin Yang[a], Jun Sun[b], Tianhai Chang[a]

[a]*School of Electronic and Information Engineering, South China University of Technology, Guangzhou, China*
[b]*Fujitsu Research & Development Center Co. Ltd., Beijing, China*

Chinese Character Recognition:
from 23MB to 2.3MB
(10x savings)

Chinese Character Recognition with a 2.3MB DNN

ENET: A DEEP NEURAL NETWORK ARCHITECTURE FOR REAL-TIME SEMANTIC SEGMENTATION

Adam Paszke
Faculty of Mathematics, Informatics and Mechanics
University of Warsaw, Poland
a.paszke@students.mimuw.edu.pl

Abhishek Chaurasia, Sangpil Kim & Eugenio Culurciello
Electrical and Computer Engineering
Purdue University, USA
aabhish, sangpilkim, euge@purdue.edu

Semantic Segmentation:
from 117MB to 0.7MB
(167x savings)

Semantic Segmentation with a 0.7MB DNN

Big Brave Future Through Small NNs

How small can we get? Could algorithmic information theory hold the answer?

- Can we design a DNN that achieves state-of-the-art ImageNet results (3% error) and fits into a 2 Watt embedded processor's cache?
- How much accuracy can we achieve in a 10KB DNN? (remember, compressed-SqueezeNet was 500KB)

How well do the techniques from this talk apply in other domains (e.g. audio, text, etc)

Open Problems

15+ startups are working on DNN-specific processors that promise 10-100X improvements in computation-per-Watt

Meanwhile, memory is improving by less than 2X per year

Unless we fit the DNNs on-chip, almost all the power will be used for memory

So, the importance of small DNNs will increase in the future

Future Hardware Will Demand Small Models

Guess what?
We're hiring (too)