convolutional neural networks, swift and iOS 11

by brett koonce

august 10th, 2017

overview

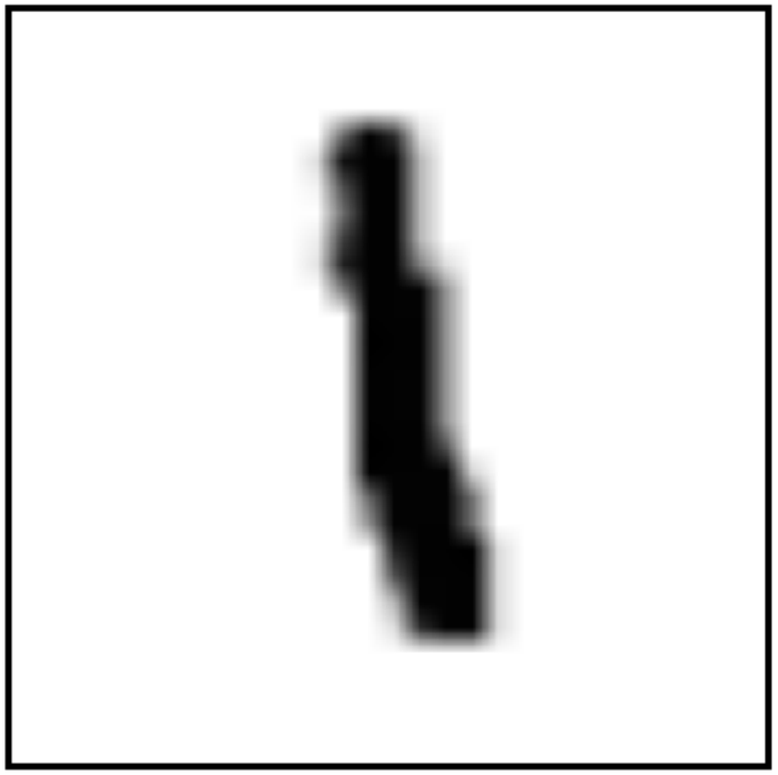
- goal: image recognition on mobile device
- machine learning, neural networks, demo: keras + tensorflow + coreml
- convolutional neural networks, different models, training/production improvements
- shiny things to play with

machine learning

ml concepts

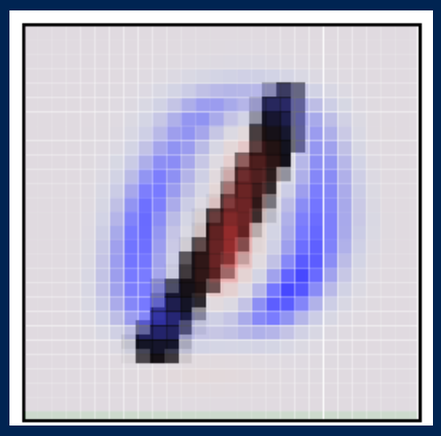
- an input (numbers, image, audio, video)
- known data (supervised learning)
- combine to produce function/black box
- train model, use on unknown data
- goals: quality, size, complexity

mnist: hello world



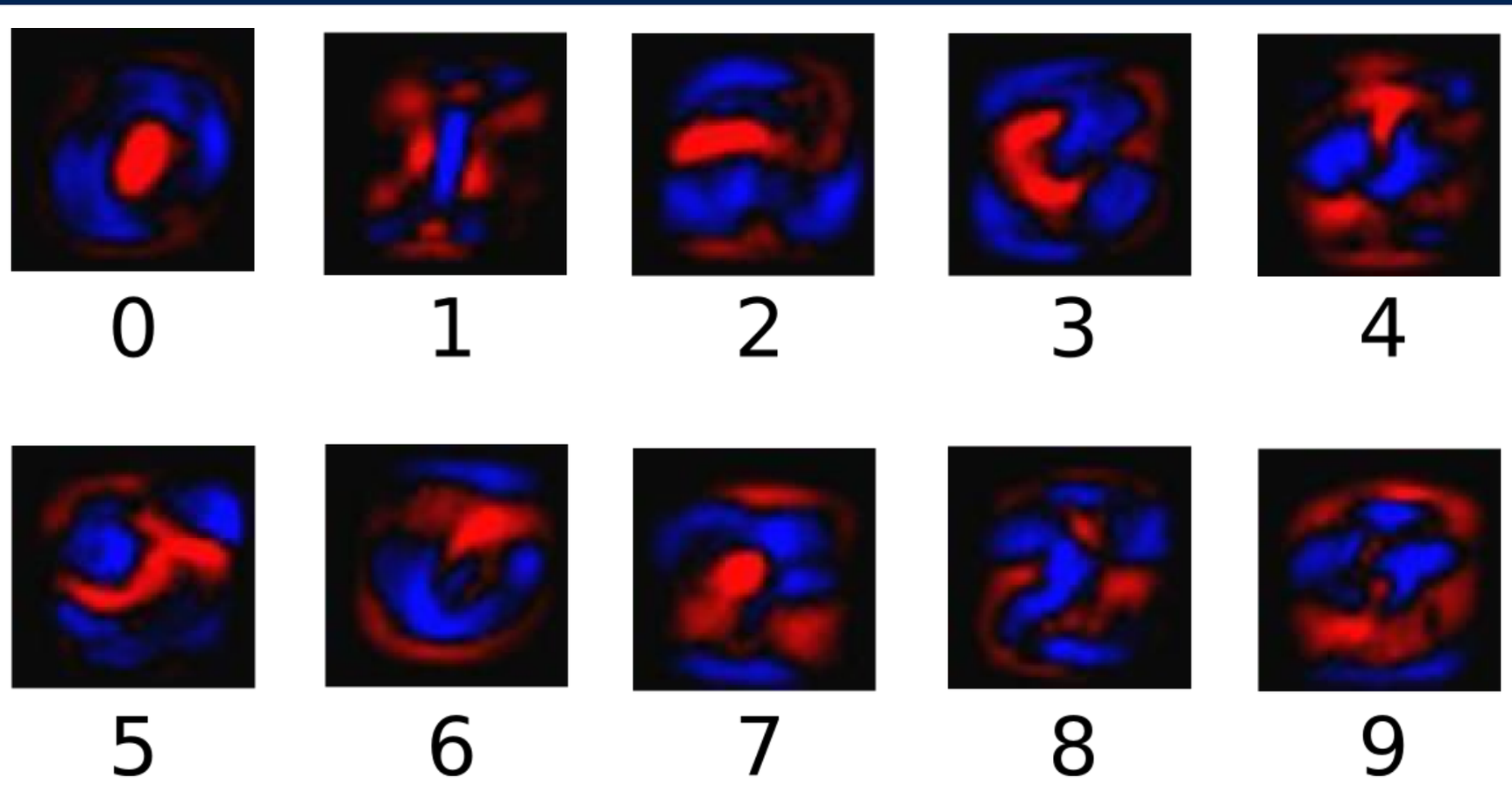
≃

0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	.6	.8	0	0	0	0	0	0
0	0	0	0	0	0	.7	1	0	0	0	0	0	0
0	0	0	0	0	0	.7	1	0	0	0	0	0	0
0	0	0	0	0	0	.5	1	.4	0	0	0	0	0
0	0	0	0	0	0	0	1	.4	0	0	0	0	0
0	0	0	0	0	0	0	1	.4	0	0	0	0	0
0	0	0	0	0	0	0	1	.7	0	0	0	0	0
0	0	0	0	0	0	0	1	1	0	0	0	0	0
0	0	0	0	0	0	0	.9	1	.1	0	0	0	0
0	0	0	0	0	0	0	.3	1	.1	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0



nn: activation layer

- [input] => [FC] => [FC] => [0-9]
- hidden layer



mnist demo

- goal: mnist recognition on device
- neural network: keras
- model training: tensorflow
- deploy model: coreml
- github.com/asparagui/keras_mnist_demo

convolutional neural networks

convolutions

- **convolution == matrix math == a[x] + b**

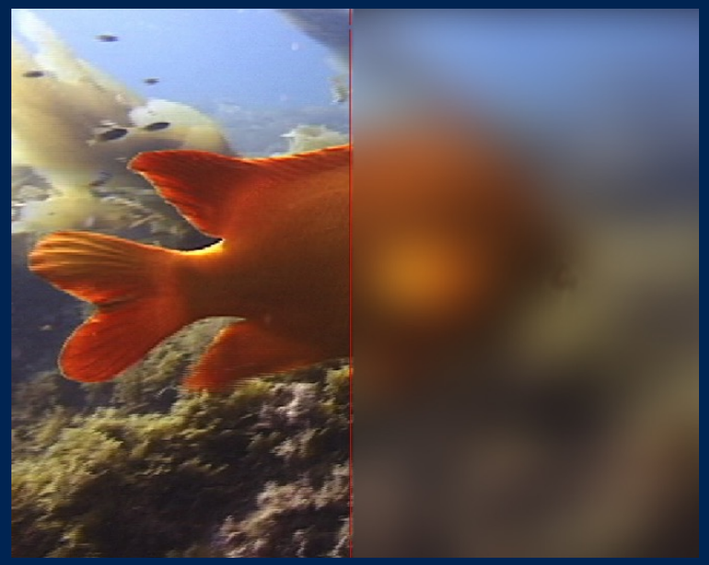
1/16	1/8	1/16
1/8	1/4	1/8
1/16	1/8	1/16

-1	0	1
-2	0	2
-1	0	1

Horizontal

-1	-2	-1
0	0	0
-1	-2	-1

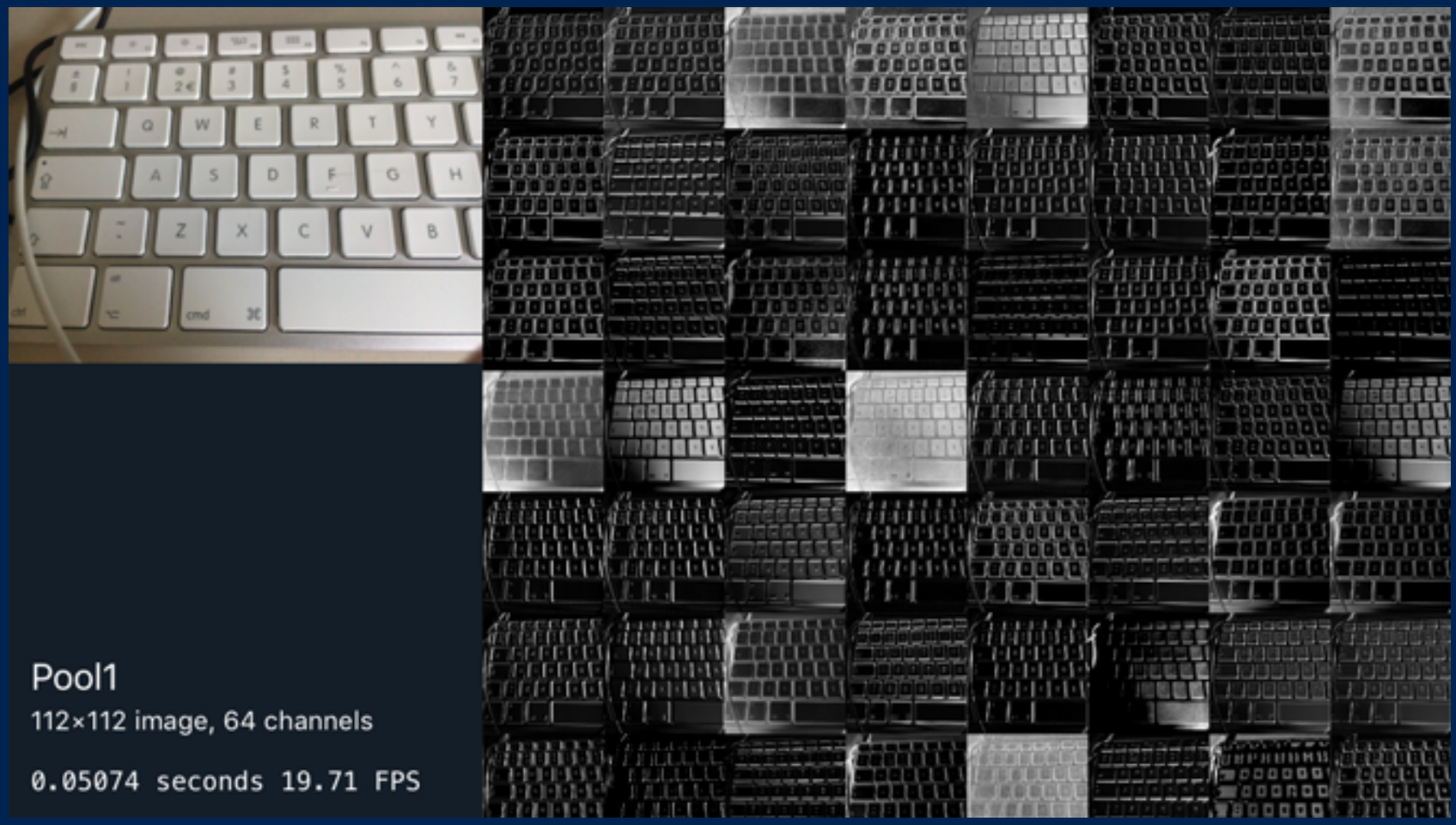
Vertical



conv: 3x3 striding

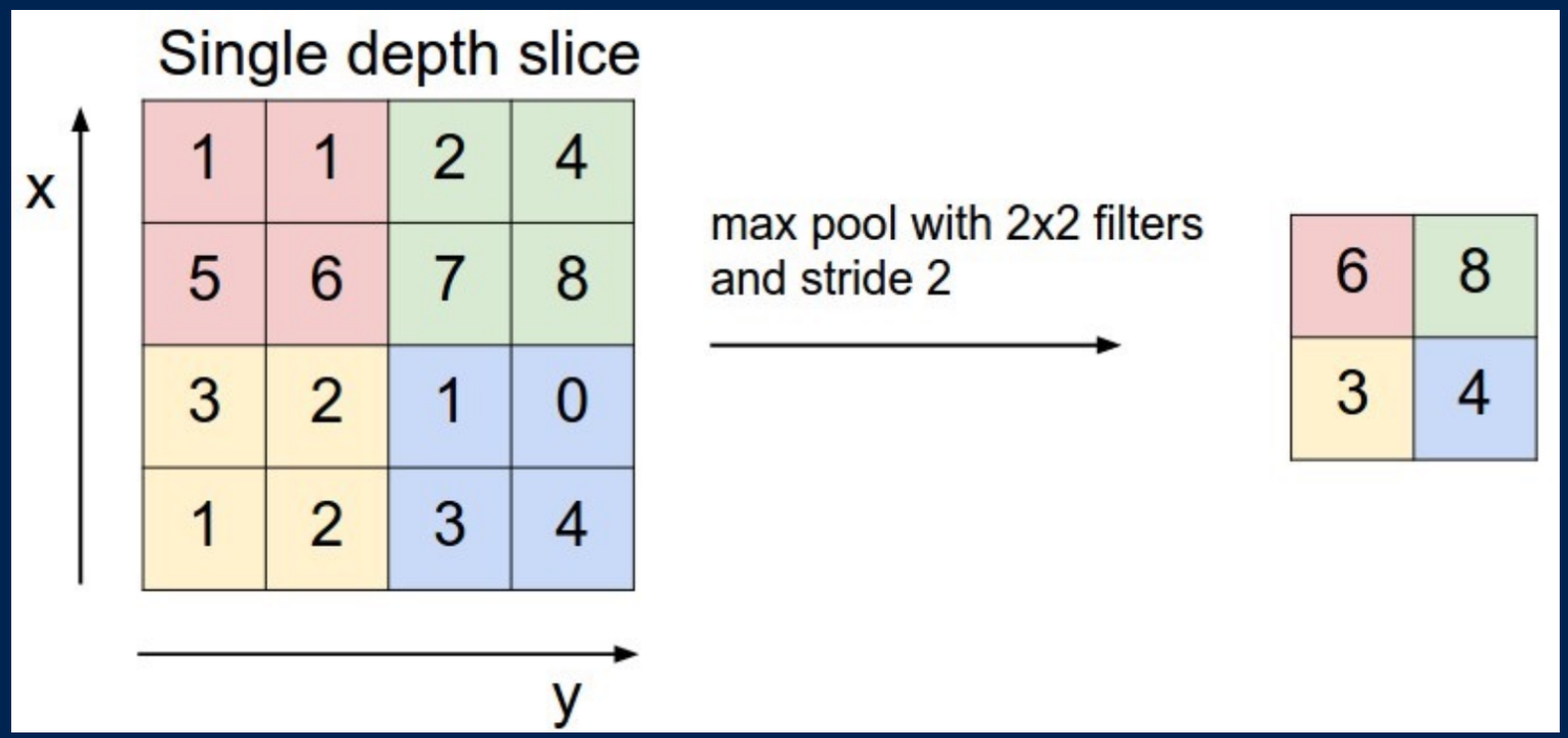
- break up input image into chunks
- cs231n.github.io

- deepfish

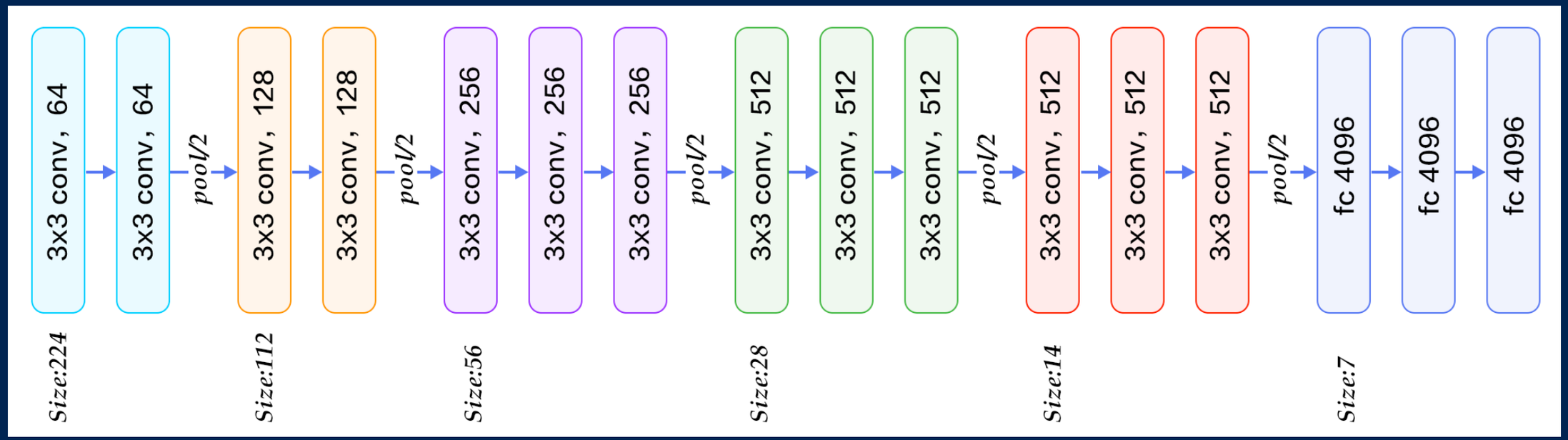


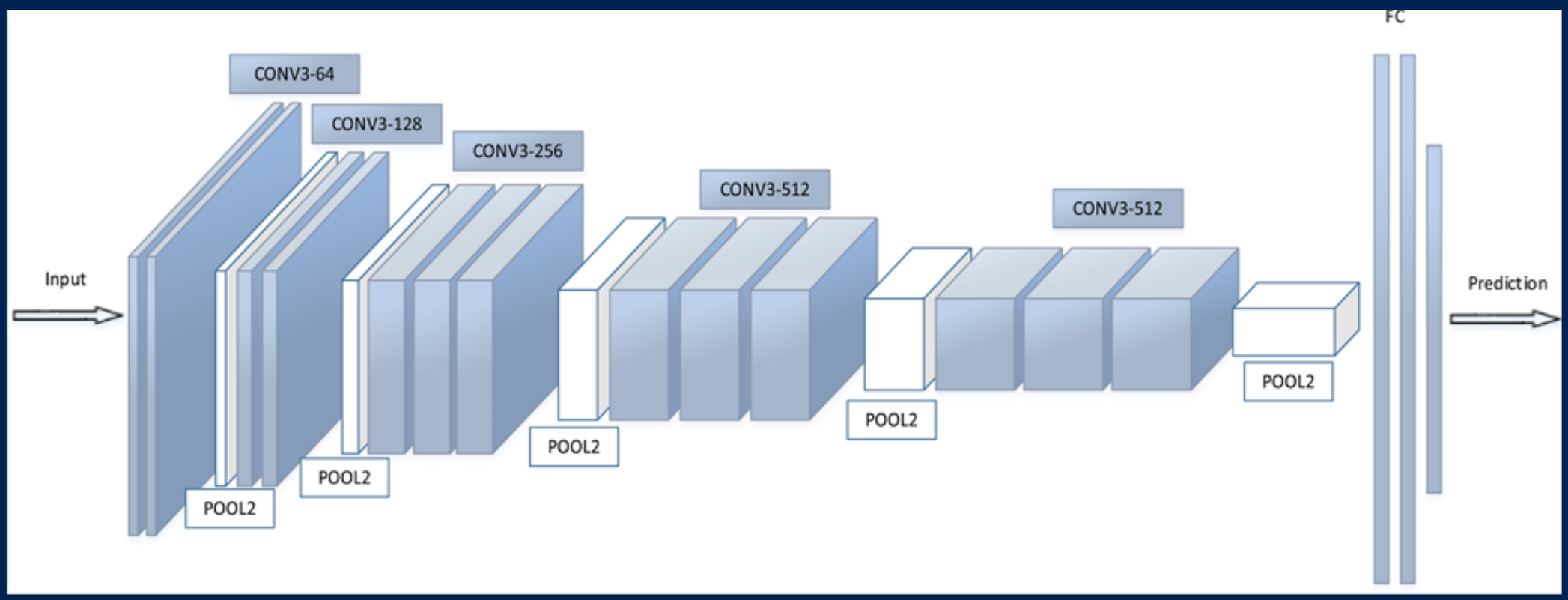
conv: maxpool

- striding produces a lot of samples
- next: reduce the data!



combined: vggnet





vgg demo

- state of art 2014
- github.com/hollance/forge
- demo running on phone (forge/metal)
- prior model + 512MB of weights
- works, but slow!

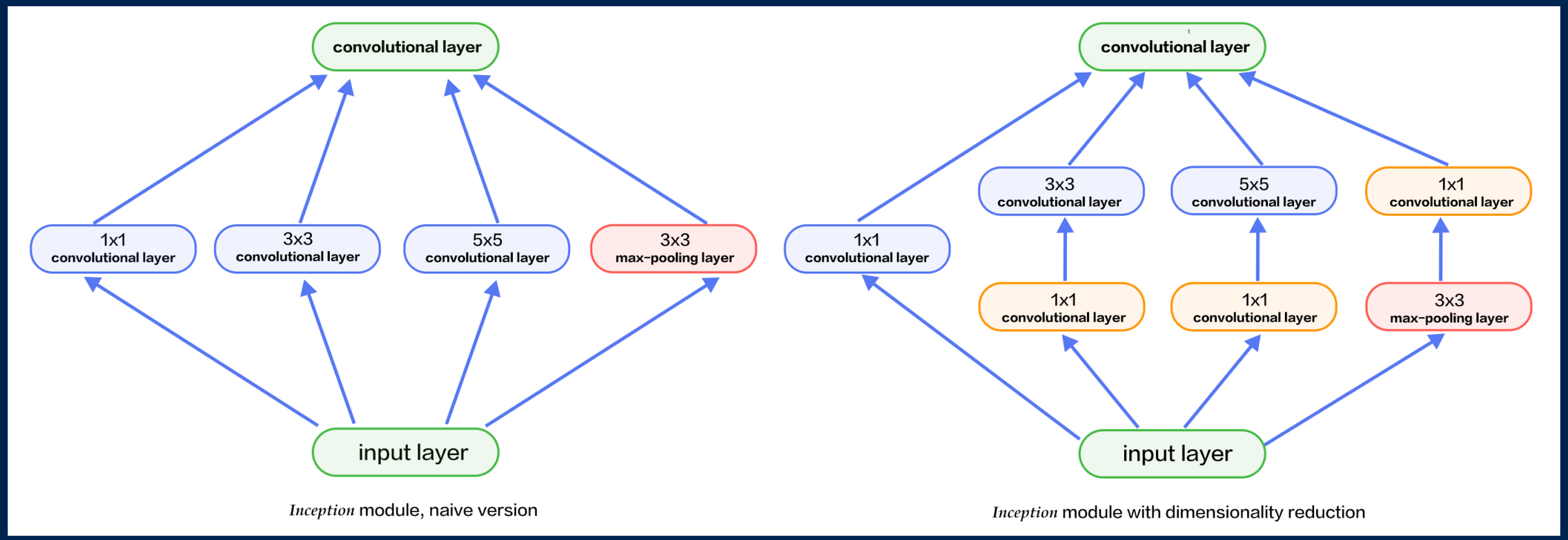
improvements

- different models (architecture)
- different training methods (speed)
- how is code actually executed? (hardware)
- what are our expectations?

going deeper

inception node

- **parallel execution, 1x1 convolution**

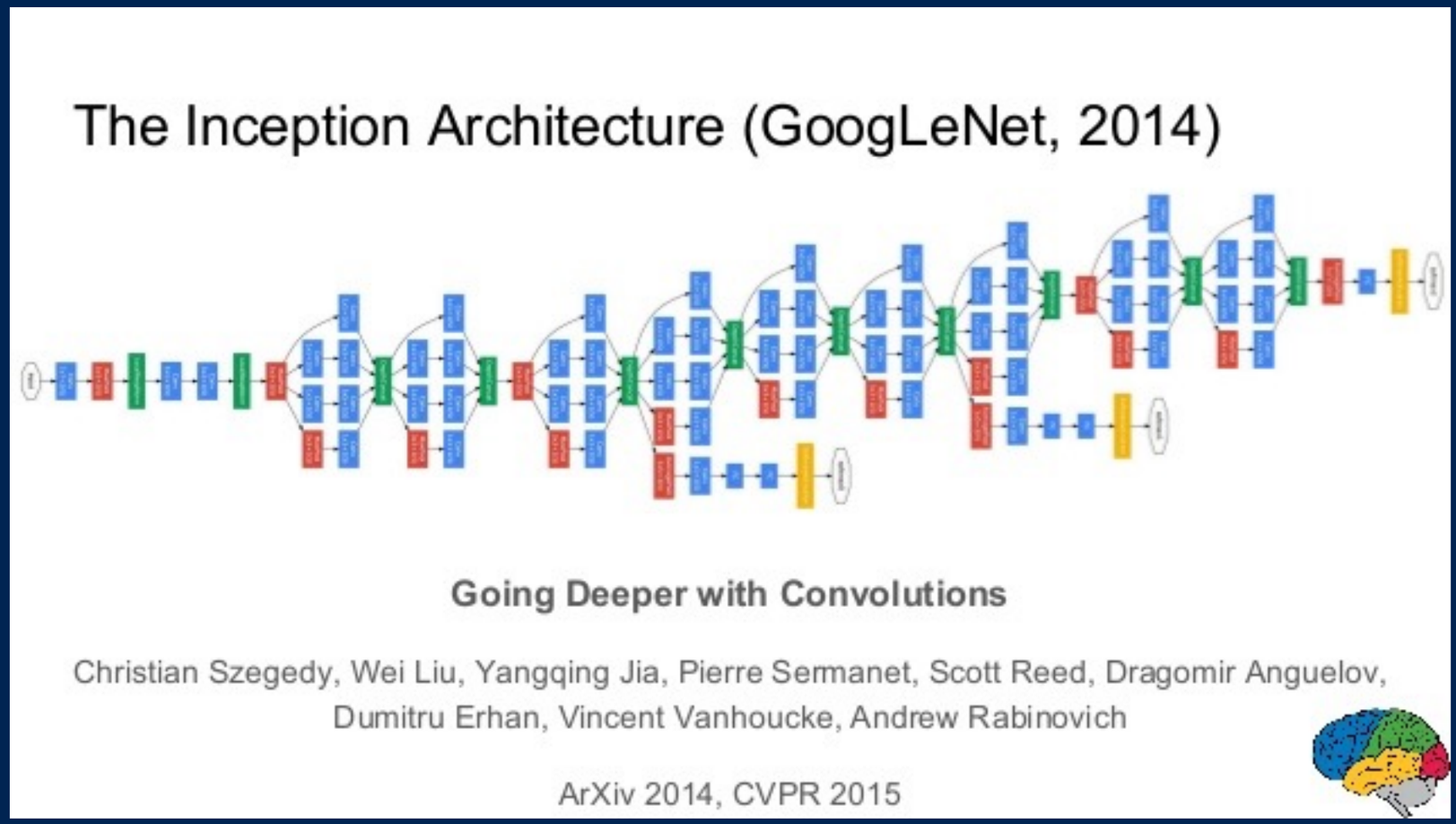


Inception module, naive version

Inception module with dimensionality reduction

- **iamaaditya.github.io/2016/03/one-by-one-convolution/**

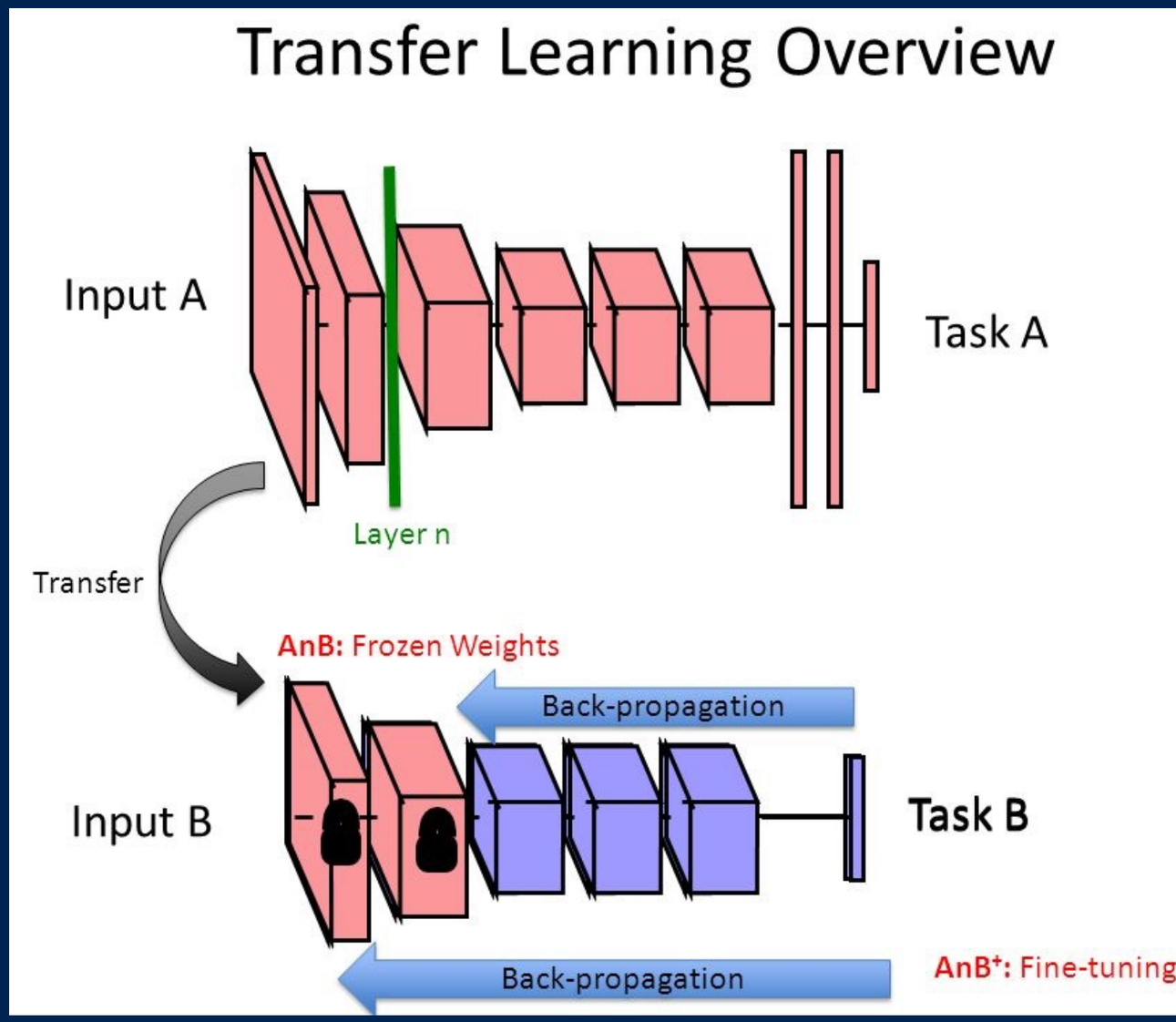
inception graph



- hacktilldawn.com/2016/09/25/inception-modules-explained-and-implemented/

model retraining

- **let's not rebuild our model from scratch!**
- **can reuse existing model**
- **re-run training on part of model with new data set**



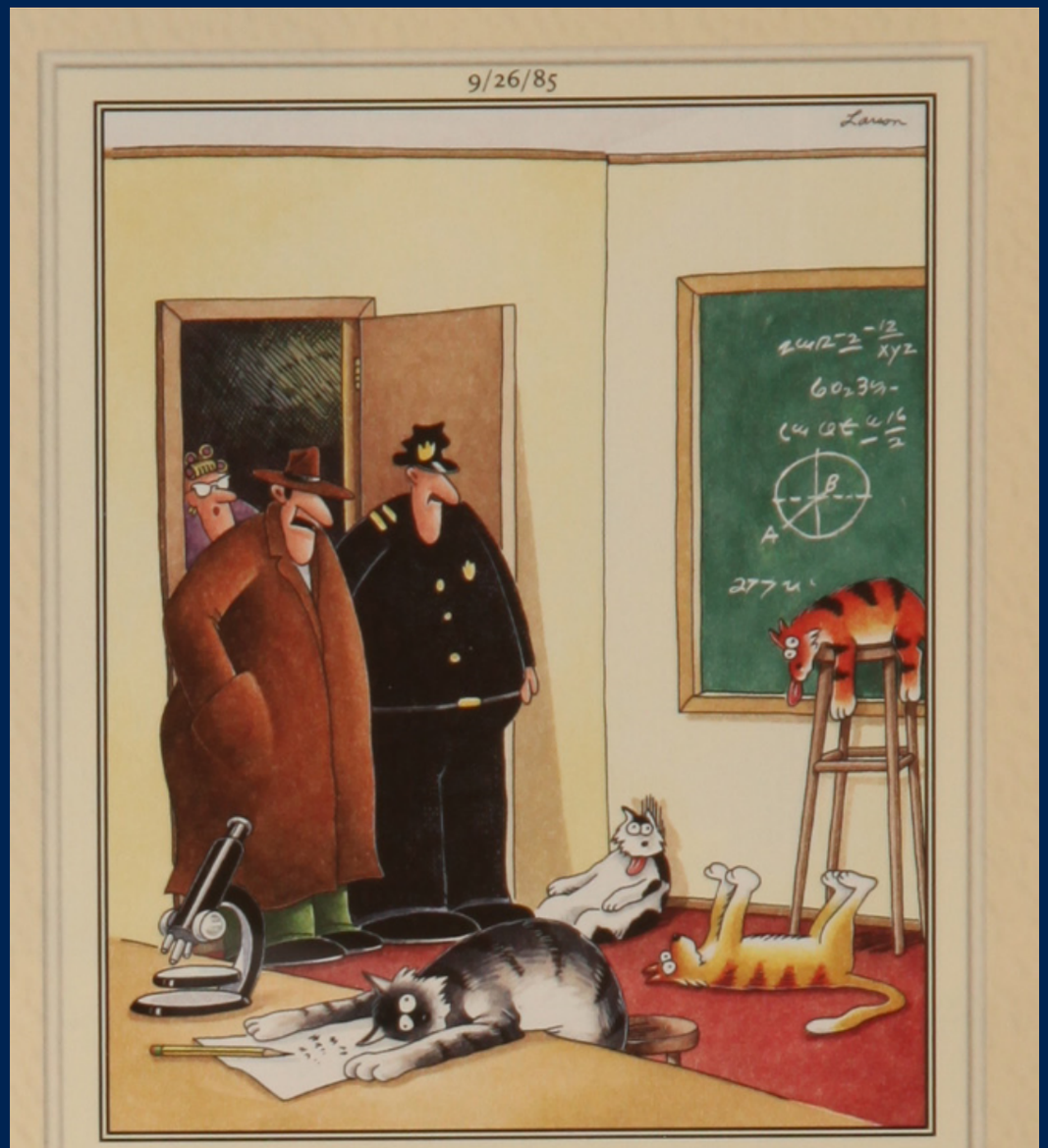
model optimization

- retrain an inception v3 graph model
- prune (remove extra nodes)
- reduce (combine nodes)
- quantize (double -> int)
- align (mmap result)

inception demo

- all of the above
- tensorflow for mobile poets (warden)
- live video => image recognition
- state of the art 2016
- tensorflow library, ios/android

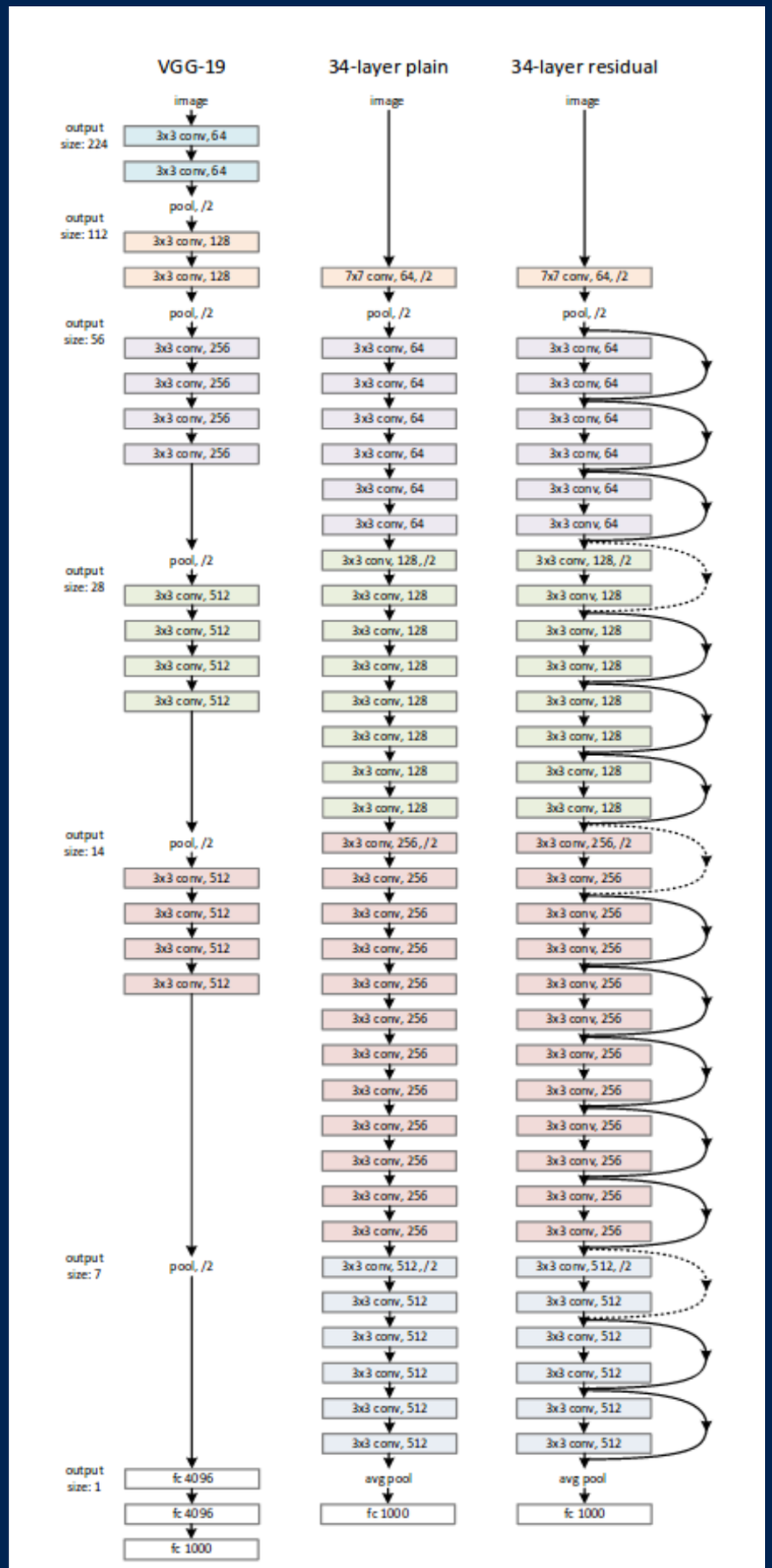
next steps



"Notice all the computations, theoretical scribblings, and lab equipment, Norm. ... Yes, curiosity killed these cats."

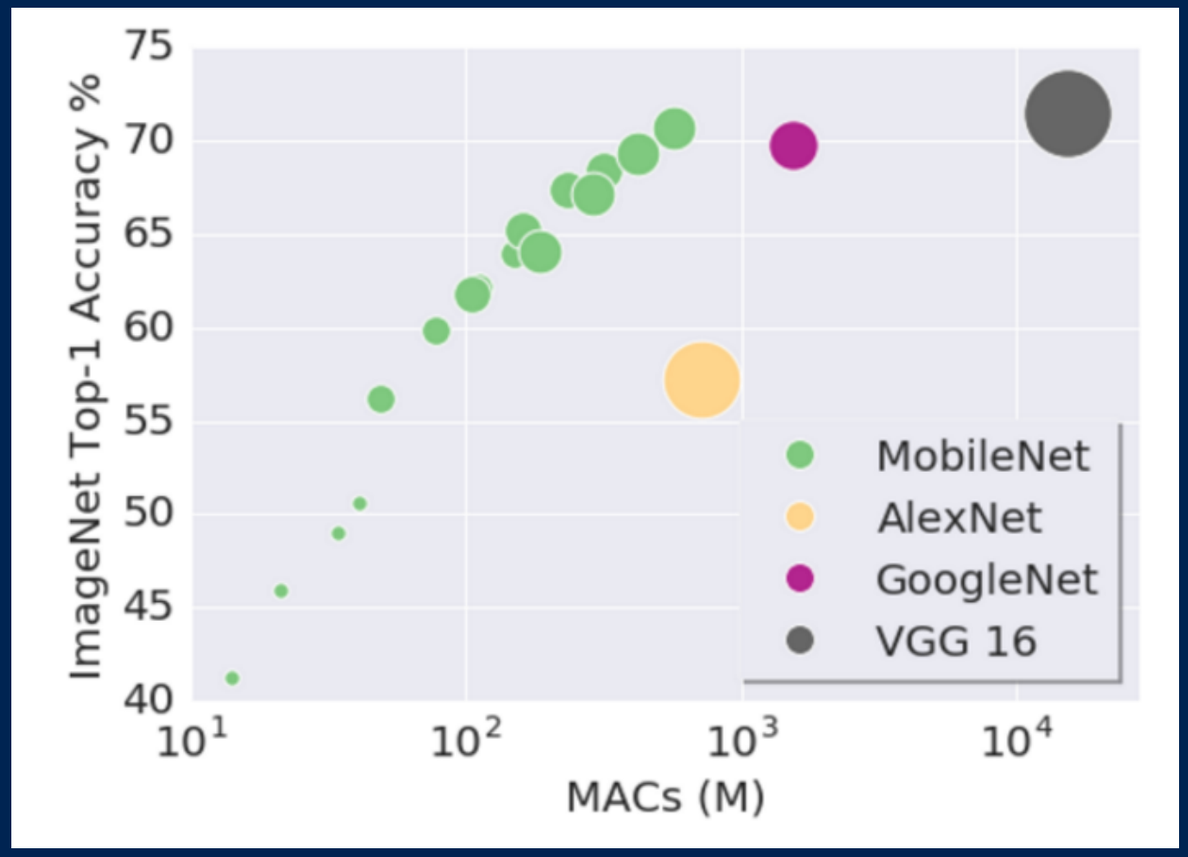
resnet

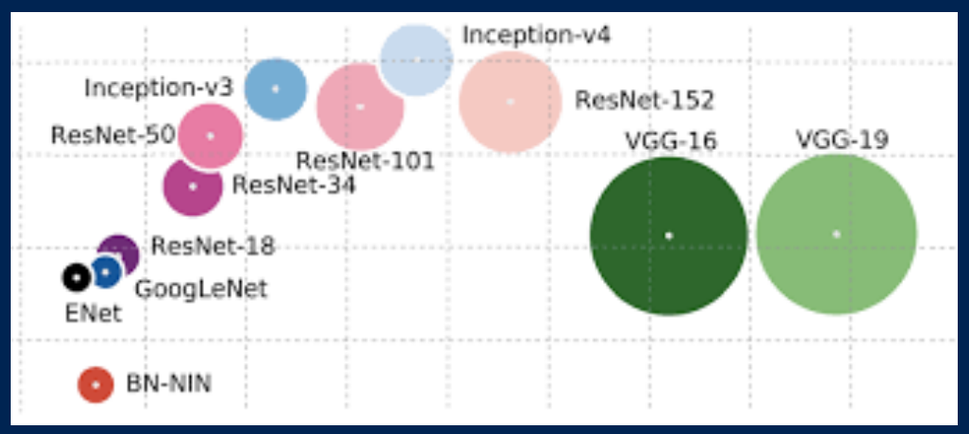
- residual networks
- skip layers
- even deeper training
- demo



mobilenets

- depthwise separable convolutions
- announced april, paper, demo
- modify inception retrain script
  - --architecture mobilenet_1.0_224





other models

- object detection:
  - YOLO (+demo)
  - SSD, SLAM, R-CNN
- see also: random forests, svm
- caffe model zoo!

thanks for coming!

questions

- me: brettkoonce.com
- apps: quarkworks.net
- tensorflow for mobile poets
- github.com/hollance/forge
- lab: cell.missouri.edu