Survey of DNN Development Resources

Course Administration (2019/12/3)

- Guest Speaker: 12/3 today
- Midterm: 12/10 (Will have class after Exam)
  - Scope: lecture materials until today
- HW6: Due: 12/17
- Paper Presentation: choose your paper and put on the google sheet

https://www.dropbox.com/home/Public/DeepLearningIC_Paper

About Final Project

- A project for the Pynq board
  - Very Flexible, the most important part is to use PYNQ(You can find project ideas on pynq.io)
  - Higher grade if hardware acceleration is used
- 5-min presentataion on 1/7
- Project submission
  - Powerpoint slides
  - Source codes
  - 3-min video to demo your project

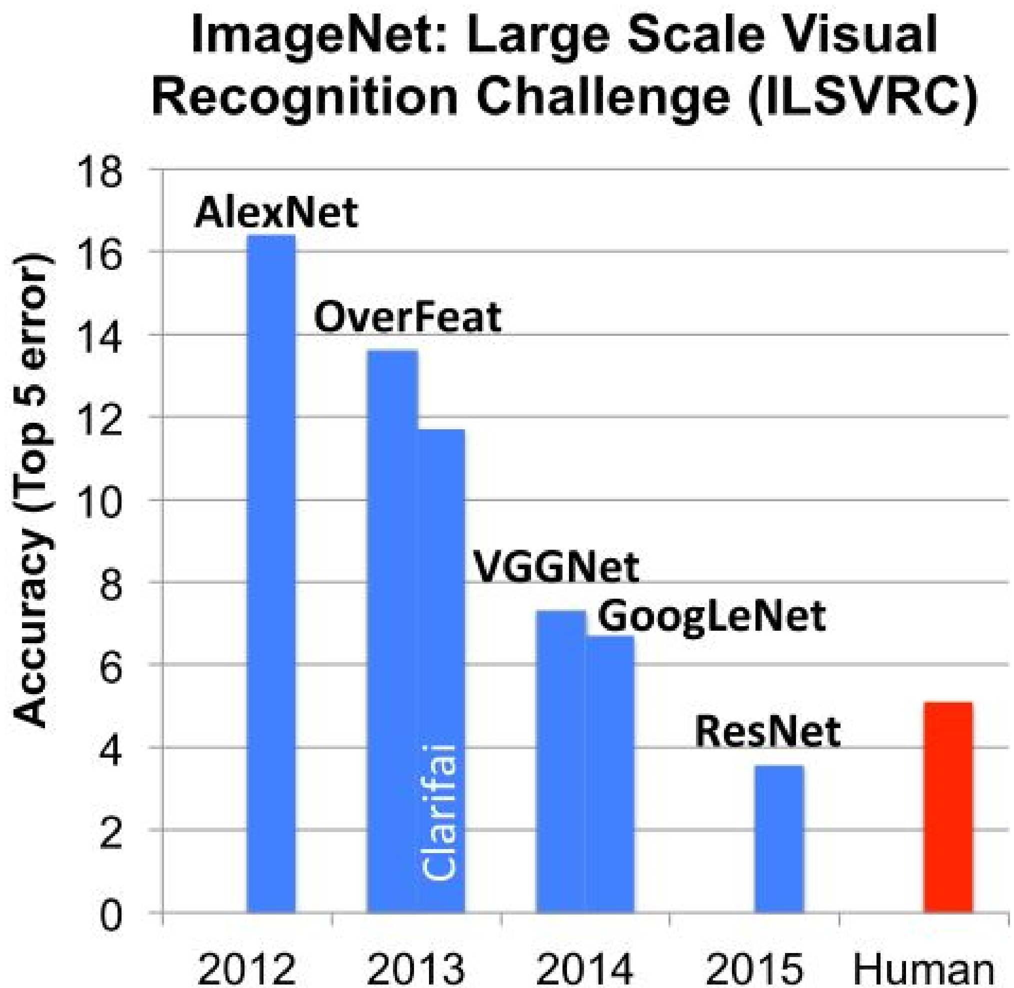
NVIDIA Hardware introduction and Deep Learning Research

李正匡 博士
(https://www.linkedin.com/in/cheng-kuang-lee-b97258157/?originalSubdomain=tw)

演講主題是:
簡介: AI 人工智慧的浪潮已經席捲全球，深度學習的數據運算需求呈現倍數增加，由於處理深度學習模型牽涉到複雜的矩陣運算，「效能」與「能耗」便成為企業亟需解決的兩項問題，為此全球 IC 設計企業致力研發更先進的 AI 晶片架構，並擺脫傳統追求製程改善的路徑，轉向系統級層面的解決方案，衍生出 GPU 、 FPGA 及 ASIC 等異質運算架構。
人工智慧架構從建構到應用可以分為兩層，分別是後端的「訓練（Training）」與前端的「推導（Inference）」。
訓練是指將複雜的圖形、影像或是語音數據輸入到深度學習模型，重複運算與修正以提高演算法準確度，最後產出可用的類神經網路軟體；後者是將已經訓練好的類神經網路軟體放入終端裝置，用以推導新的數據，實現生活中的人工智慧應用，例如自動駕駛、語音識別、圖像辨識及影像處理等等功能。
為了洞察未來 AI 晶片的產業趨勢，本次專題分享將從業界的角度探討 AI 晶片產業的發展現況。

Popular DNNs

- **LeNet (1998)**
- **AlexNet (2012)**
- **OverFeat (2013)**
- **VGGNet (2014)**
- **GoogleNet (2014)**
- **ResNet (2015)**



[O. Russakovsky et al., IJCV 2015]

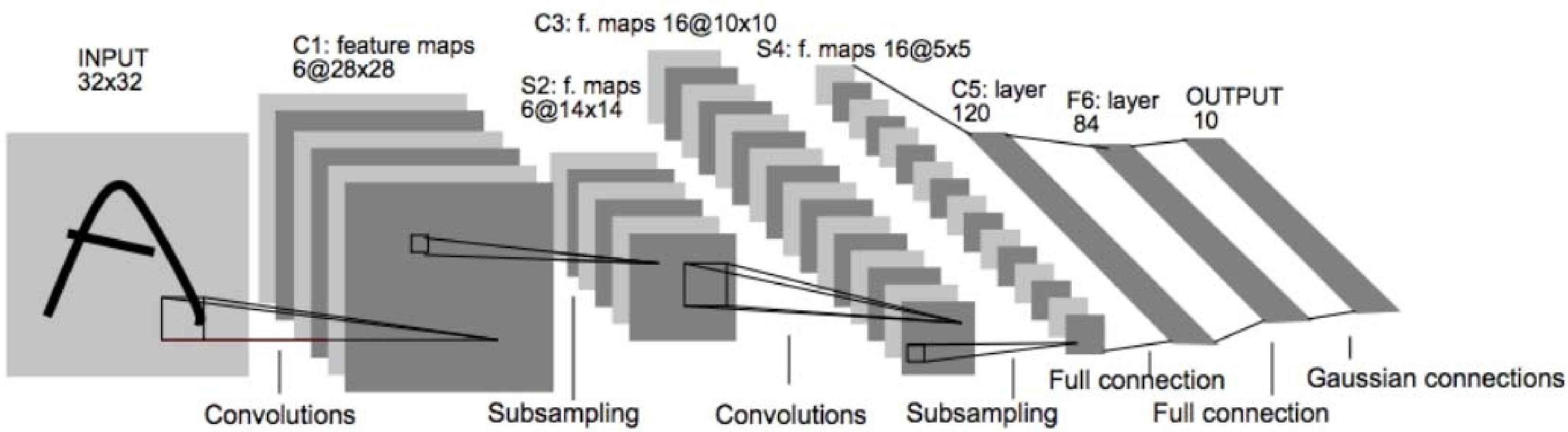
Metrics	LeNet 5	AlexNet	Overfeat fast	VGG 16	GoogLeNet v1	ResNet 50
Top-5 error†	n/a	16.4	14.2	7.4	6.7	5.3
Top-5 error (single crop)†	n/a	19.8	17.0	8.8	10.7	7.0
Input Size	28×28	227×227	231×231	224×224	224×224	224×224
# of CONV Layers	2	5	5	13	57	53
Depth in # of CONV Layers	2	5	5	13	21	49
Filter Sizes	5	3,5,11	3,5,11	3	1,3,5,7	1,3,7
# of Channels	1, 20	3-256	3-1024	3-512	3-832	3-2048
# of Filters	20, 50	96-384	96-1024	64-512	16-384	64-2048
Stride	1	1,4	1,4	1	1,2	1,2
Weights	2.6k	2.3M	16M	14.7M	6.0M	23.5M
MACs	283k	666M	2.67G	15.3G	1.43G	3.86G
# of FC Layers	2	3	3	3	1	1
Filter Sizes	1,4	1,6	1,6,12	1,7	1	1
# of Channels	50, 500	256-4096	1024-4096	512-4096	1024	2048
# of Filters	10, 500	1000-4096	1000-4096	1000-4096	1000	1000
Weights	58k	58.6M	130M	124M	1M	2M
MACs	58k	58.6M	130M	124M	1M	2M
Total Weights	60k	61M	146M	138M	7M	25.5M
Total MACs	341k	724M	2.8G	15.5G	1.43G	3.9G
Pretrained Model Website	[56]‡	[57, 58]	n/a	[57–59]	[57–59]	[57–59]

†Accuracy is Measured Based on Top-5 Error on ImageNet [14].
‡This Version of LeNet-5 has 431 000 Weights for the Filters and Requires 2.3 million MACs Per Image, and Uses ReLU Rather Than Sigmoid.

LeNet-5

CONV Layers: 2
Fully Connected Layers: 2
Weights: 60k
MACs: 341k
Sigmoid used for non-linearity
Using average pooling

Digit Classification!



six
5x5 filters

2x2
average
pooling

six
5x5 filters

2x2
average
pooling

[Y. Lecun et al, Proceedings of the IEEE, 1998]

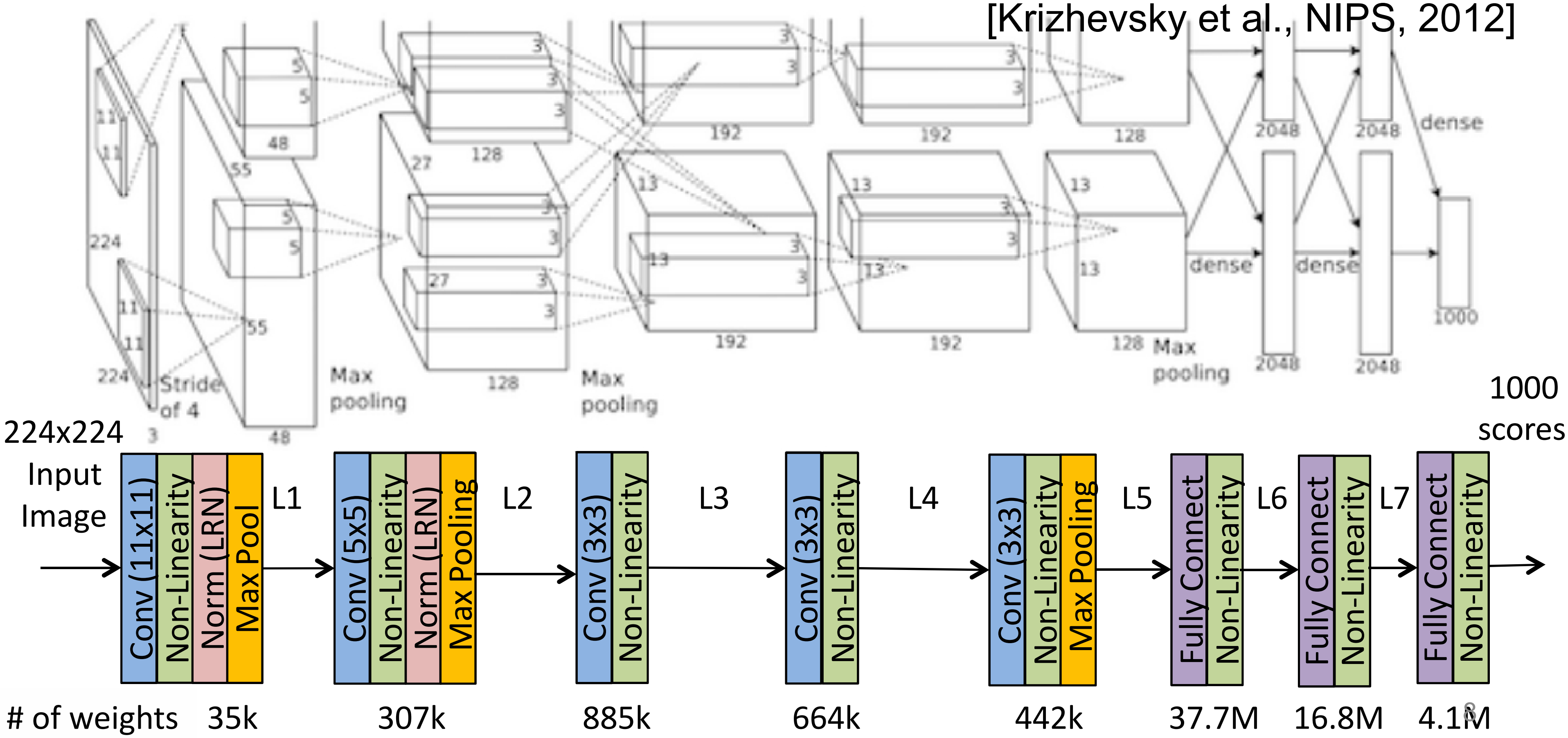
AlexNet

CONV Layers: 5
Fully Connected Layers: 3
Weights: 61M
MACs: 724M
ReLU used for non-linearity

ILSCVR12 Winner

Uses Max pooling
Uses Local Response Normalization (LRN)

[Krizhevsky et al., NIPS, 2012]

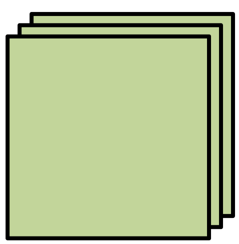


Large Sizes with Varying Shapes

AlexNet Convolutional Layer Configurations

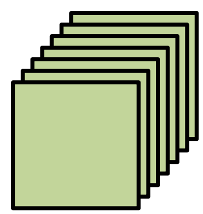
Layer	Filter Size (RxS)	# Filters (M)	# Channels (C)	Stride
1	11x11	96	3	4
2	5x5	256	48	1
3	3x3	384	256	1
4	3x3	384	192	1
5	3x3	256	192	1

Layer 1



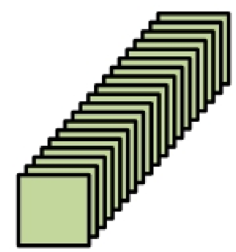
34k Params
105M MACs

Layer 2



307k Params
224M MACs

Layer 3



885k Params
150M MACs

[Krizhevsky et al., NIPS, 2012]

VGG-16 (Also, 19 layer version)

CONV Layers: 13, Fully Connected Layers: 3
Weights: 138M
MACs: 15.5G

More Layers →Deeper!

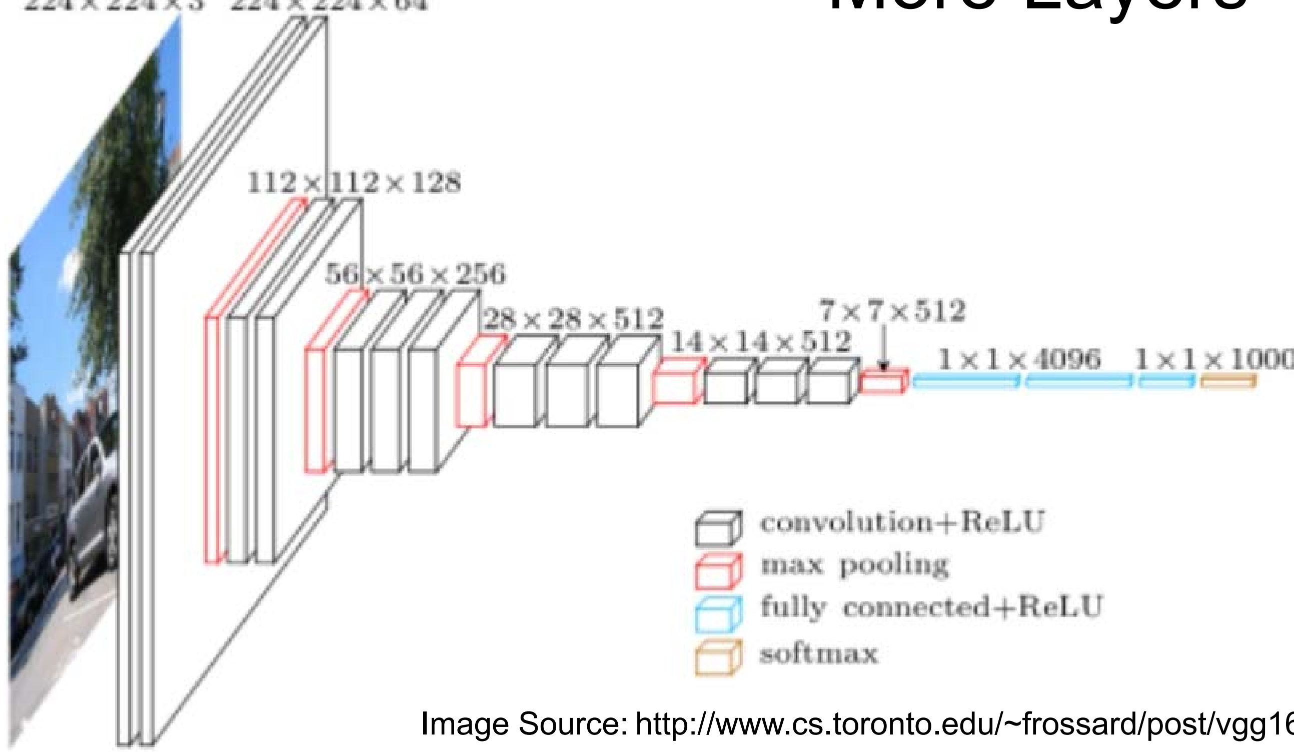


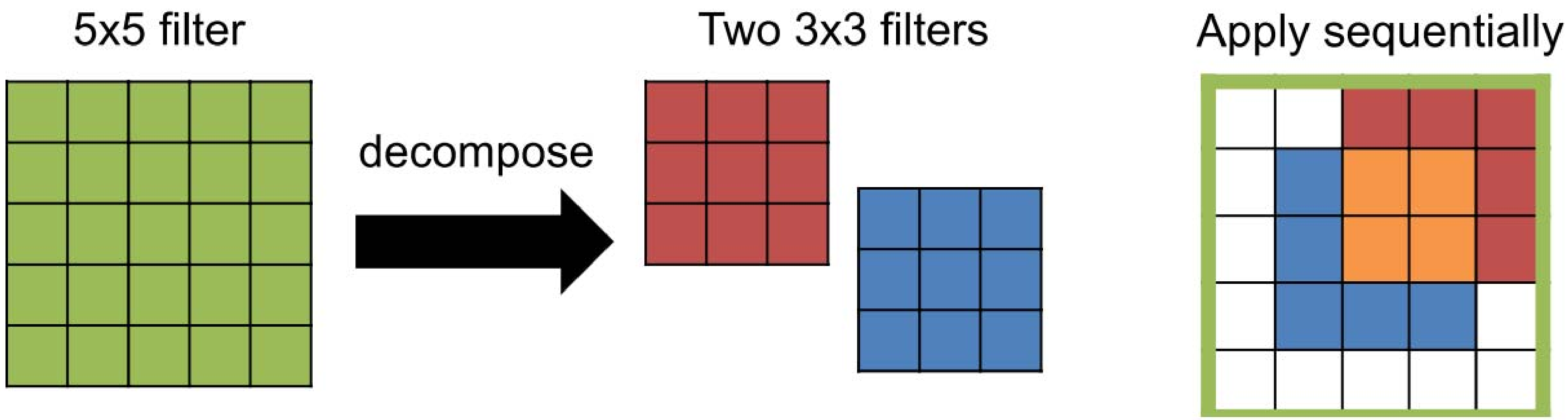
Image Source: http://www.cs.toronto.edu/~frossard/post/vgg16/

[Simonyan et al., arXiv 2014, ICLR 2015]

VGG-16 (Also, 19 layer version)

Filters size is 3x3 only

Stack 2 3x3 filters for a 5x5 receptive files



[Simonyan et al., arXiv 2014, ICLR 2015]

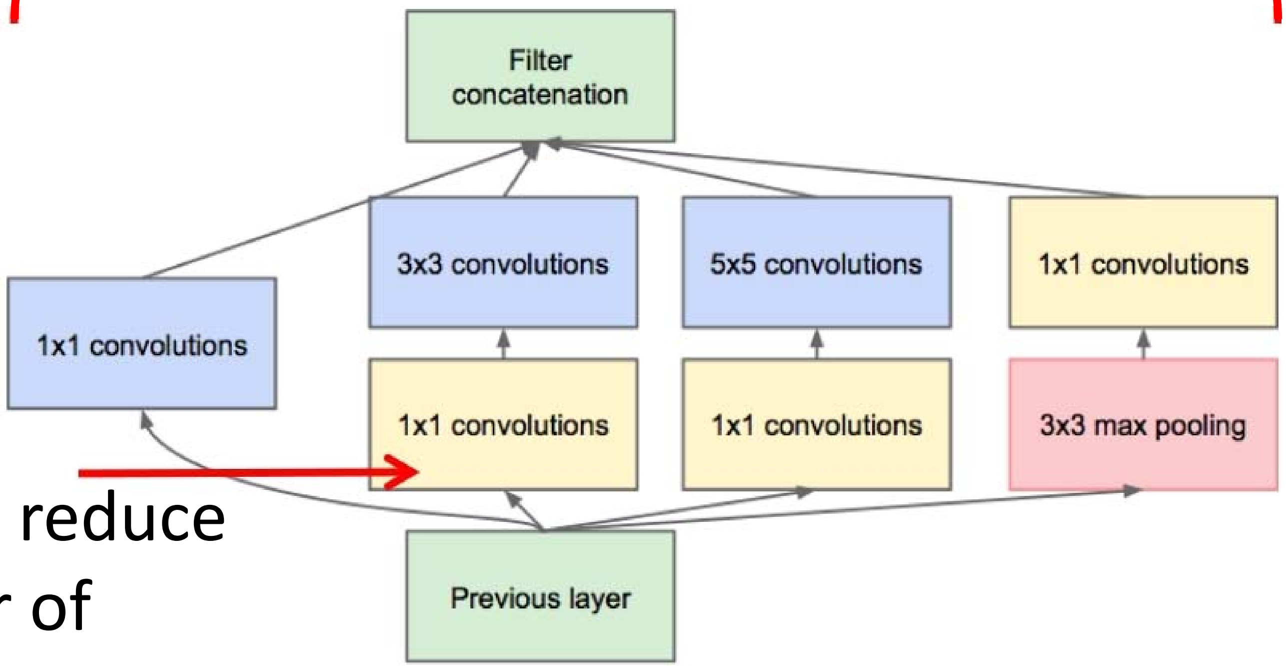
GoogLeNet (v1)

CONV Layers: 21 (depth), 57 (total)
Fully Connected Layers: 1
Weights: 7.0M
MACs: 1.43G

Also, v2, v3 and v4
ILSVRC14 Winner

parallel filters of different size has the effect of processing image at different scales

Inception Module



1x1 conv to reduce the number of channels

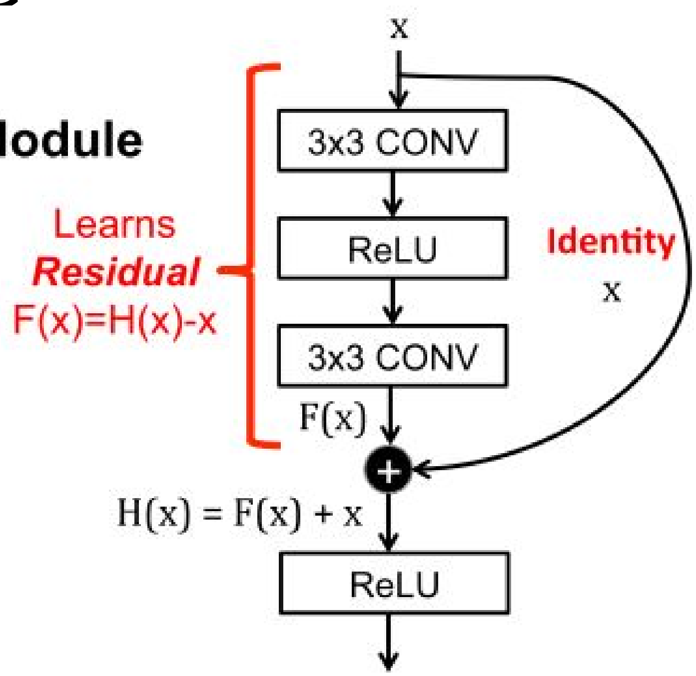
[Szegedy et al., arXiv 2014, CVPR 2015]

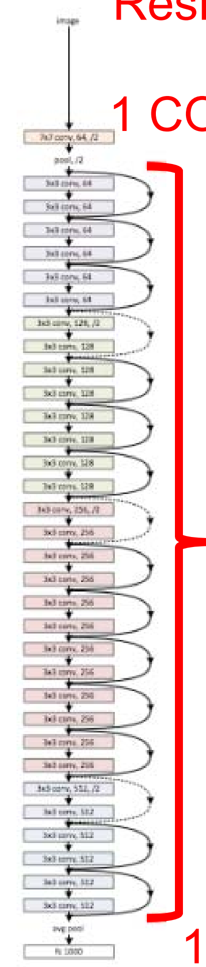
ResNet-50

CONV Layers: 49
Fully Connected Layers: 1
Weights: 25.5M
MACs: 3.9G

Also, 34,**152** and 1202 layer versions
ILSVRC15 Winner

Short Cut Module





Helps address the vanishing gradient challenge for training very deep networks

[He et al., arXiv 2015, CVPR 2016]

Revolution of Depth

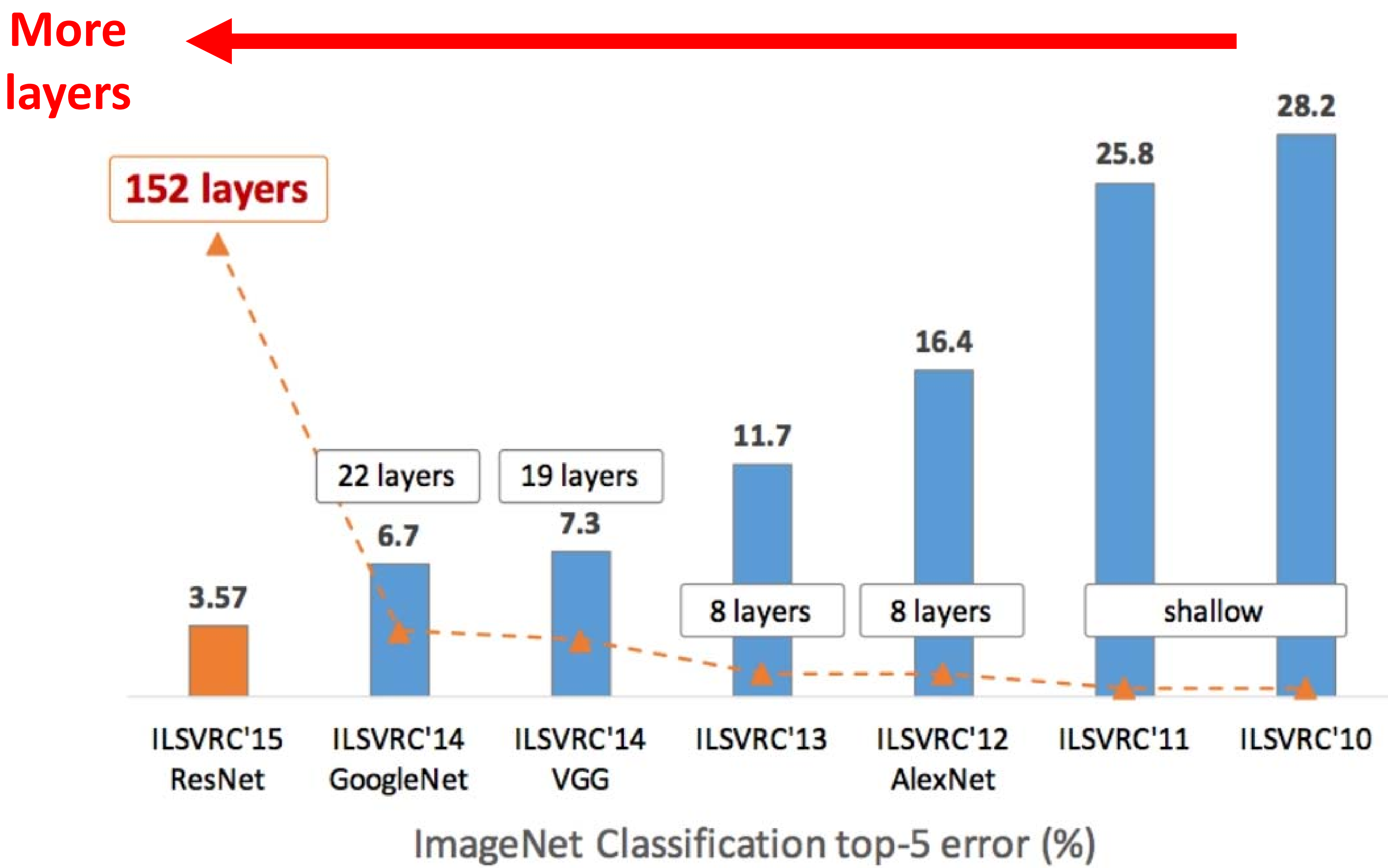


Image Source: http://icml.cc/2016/tutorials/icml2016_tutorial_deep_residual_networks_kaiminghe.pdf

Revolution of Depth

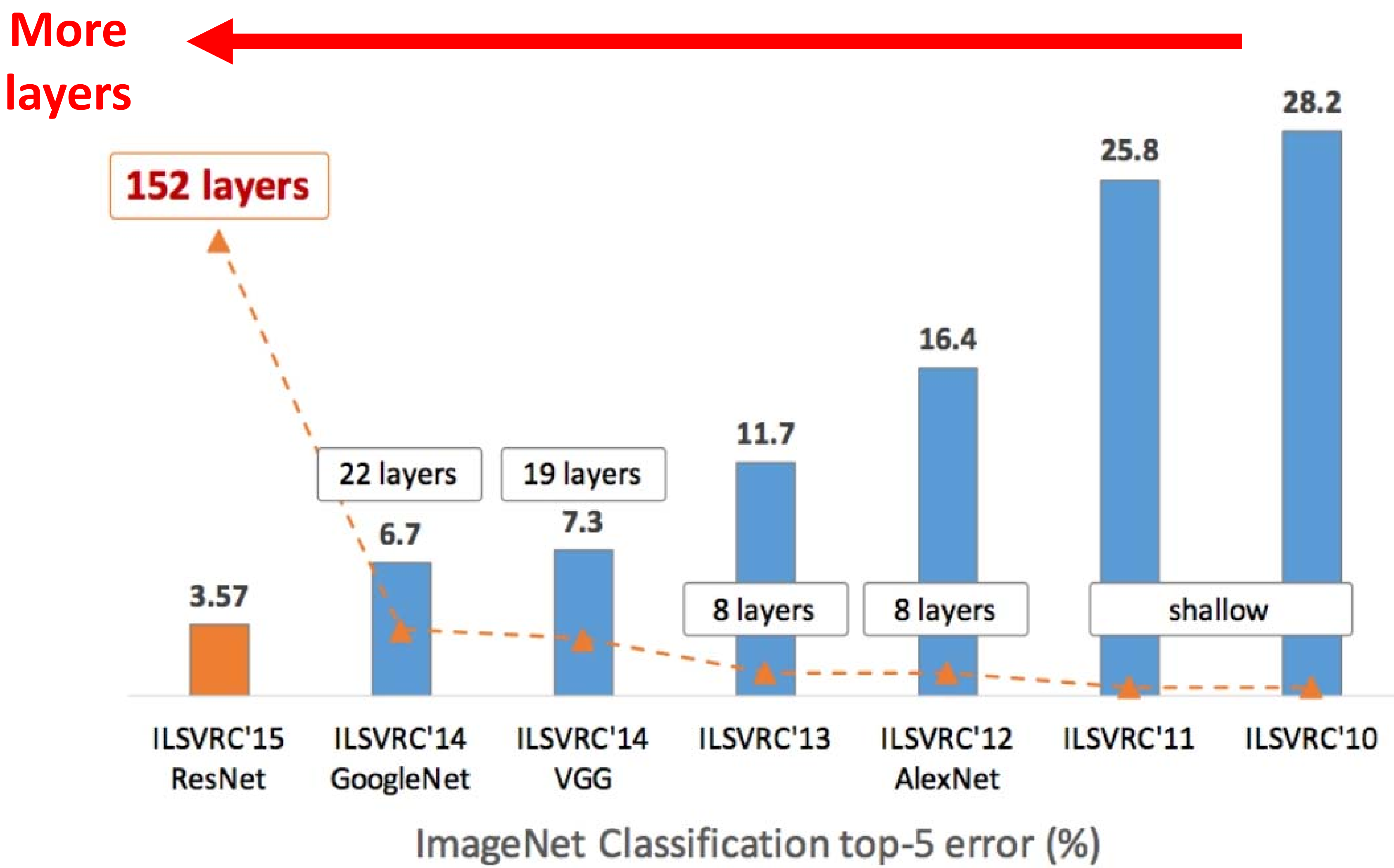


Image Source: http://icml.cc/2016/tutorials/icml2016_tutorial_deep_residual_networks_kaiminghe.pdf

Summary of Popular DNNs

Metrics	LeNet-5	AlexNet	VGG-16	GoogLeNet (v1)	ResNet-50
Top-5 error	n/a	16.4	7.4	6.7	5.3
Input Size	28x28	227x227	224x224	224x224	224x224
# of CONV Layers	**2**	**5**	**16**	**21 (depth)**	**49**
Filter Sizes	5	3, 5,11	3	1, 3 , 5, 7	1, 3, 7
# of Channels	1, 6	3 - 256	3 - 512	3 - 1024	3 - 2048
# of Filters	6, 16	96 - 384	64 - 512	64 - 384	64 - 2048
Stride	1	1, 4	1	1, 2	1, 2
# of Weights	2.6k	2.3M	14.7M	6.0M	23.5M
# of MACs	283k	666M	15.3G	1.43G	3.86G
# of FC layers	**2**	**3**	**3**	**1**	**1**
# of Weights	58k	58.6M	124M	1M	2M
# of MACs	58k	58.6M	124M	1M	2M
Total Weights	**60k**	**61M**	**138M**	**7M**	**25.5M**
Total MACs	**341k**	**724M**	**15.5G**	**1.43G**	**3.9G**

CONV Layers increasingly important!

Summary of Popular DNNs

- **AlexNet**
  - **First CNN Winner of ILSVRC**
- **VGG-16**
  - **Goes Deeper (16+ layers)**
  - **Uses only 3x3 filters**
- **GoogLeNet (v1)**
  - **Reduces weights with Inception and only one FC layer**
  - **Batch Normalization**
- **ResNet**
  - **Goes Deeper (24+ layers)**
  - **Shortcut connections**

Frameworks

Caffe *

Berkeley / BVLC
(C, C++, Python, MATLAB)

TensorFlow *

Google (C++, Python)

mxnet

Chainer

PyTorch

Facebook / NYU
(C, C++, Lua)

Also, CNTK, etc.
More at: https://developer.nvidia.com/deep-learning-frameworks

** Lightweight mobile versions (Caffe2go, TensorFlow Mobile)*

Keras

```
from __future__ import print_function
import keras
from keras.datasets import mnist
from keras.models import Sequential
from keras.layers import Dense, Dropout, Flatten
from keras.layers import Conv2D, MaxPooling2D
from keras import backend as K

batch_size = 128
num_classes = 10
epochs = 12

# input image dimensions
img_rows, img_cols = 28, 28

# the data, split between train and test sets
(x_train, y_train), (x_test, y_test) = mnist.load_data()

if K.image_data_format() == 'channels_first':
   x_train = x_train.reshape(x_train.shape[0], 1, img_rows, img_cols)
   x_test = x_test.reshape(x_test.shape[0], 1, img_rows, img_cols)
   input_shape = (1, img_rows, img_cols)
else:
   x_train = x_train.reshape(x_train.shape[0], img_rows, img_cols, 1)
   x_test = x_test.reshape(x_test.shape[0], img_rows, img_cols, 1)
   input_shape = (img_rows, img_cols, 1)

x_train = x_train.astype('float32')
x_test = x_test.astype('float32')
x_train /= 255
```

https://keras.io/examples/mnist_cnn/

Benefits of Frameworks

- Rapid development
- Sharing models
- Workload profiling
- Network hardware co-design

Image Classification Datasets

- **Image Classification/Recognition**
  - **Given an entire image →Select 1 of N classes**

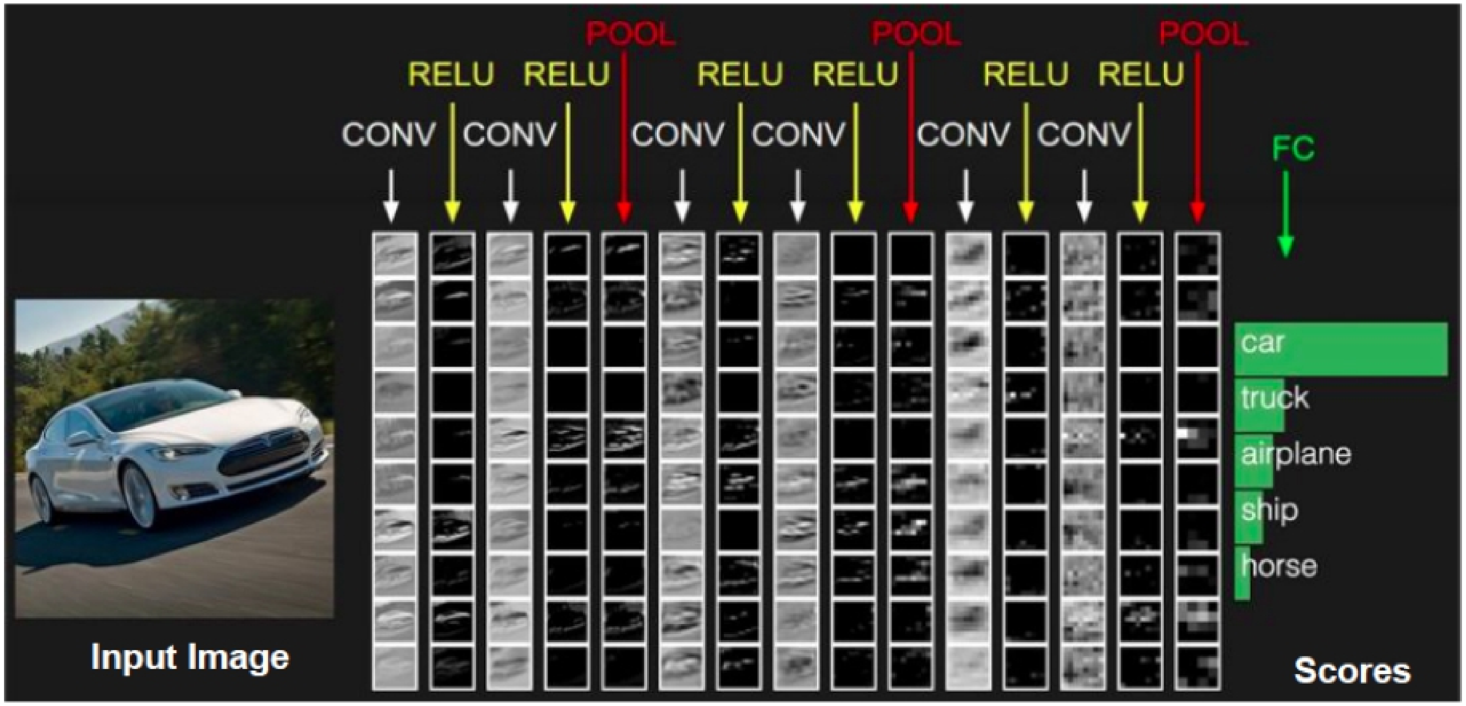


Image Source: Stanford cs231n

Datasets affect difficulty of task

MNIST

Digit Classification
28x28 pixels (B&W)
10 Classes
60,000 Training
10,000 Testing

LeNet in 1998
(0.95% error)

ICML 2013
(0.21% error)

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

ImageNet

Object Classification
~256x256 pixels (color)
1000 Classes
1.3M Training
100,000 Testing (50,000 Validation)

Image Source: http://karpathy.github.io/

http://www.image-net.org/challenges/LSVRC/

IMAGENET

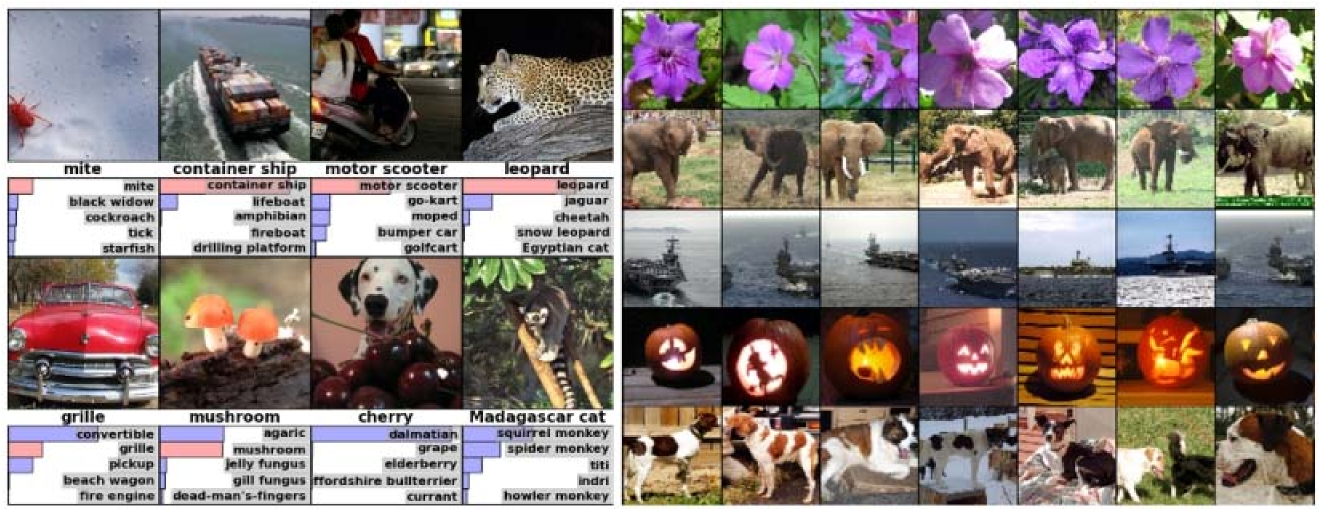
Fine grained Classes
(120 breeds)

Image Source: http://karpathy.github.io/

Image Source: Krizhevsky et al., NIPS 2012

Top-5 Error
Winner 2012
(16.42% error)

Winner 2016
(2.99% error)



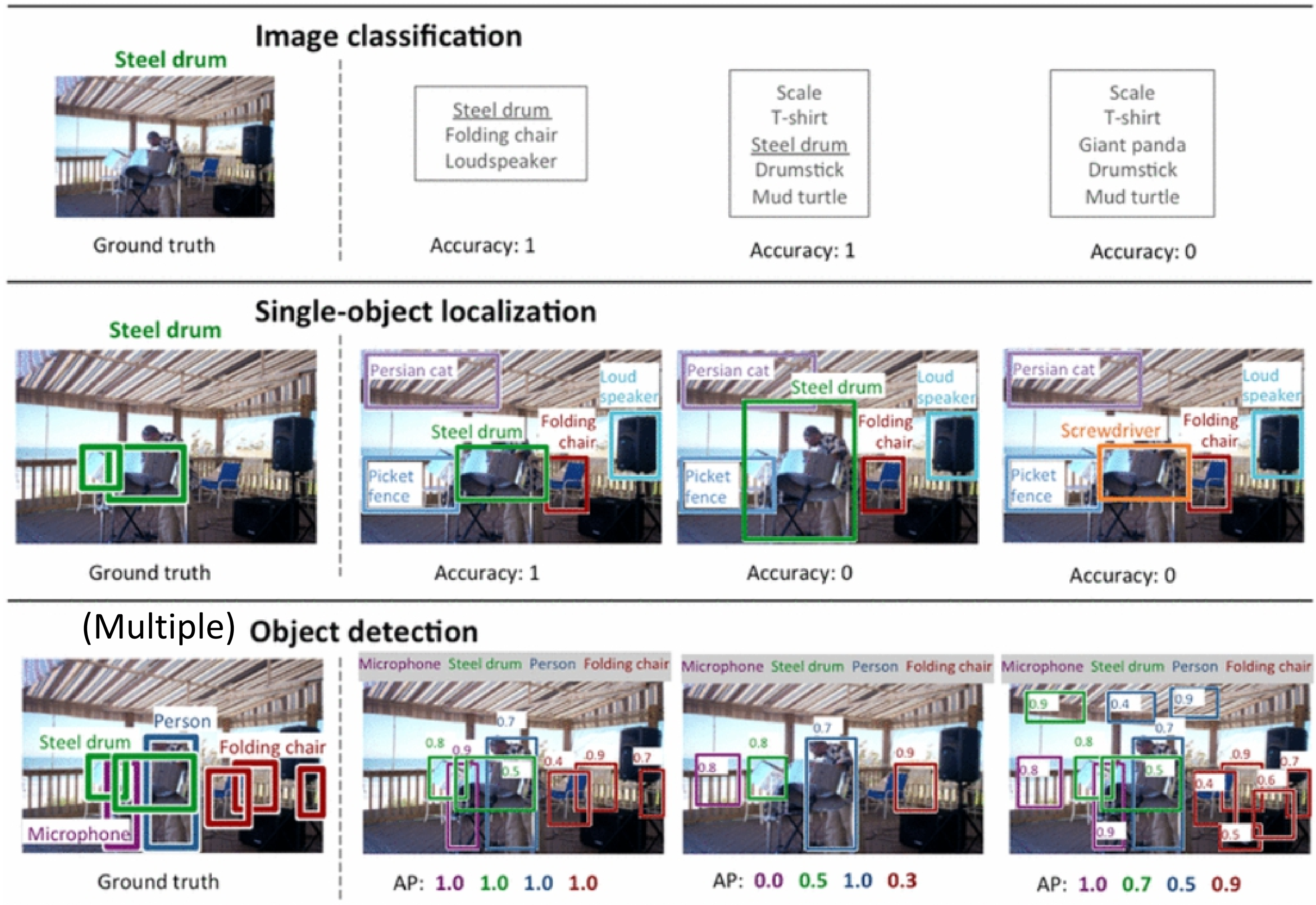
http://www.image-net.org/challenges/LSVRC/

Image Classification Summary

	MNIST	IMAGENET
Year	1998	2012
Resolution	28x28	256x256
Classes	10	1000
Training	60k	1.3M
Testing	10k	100k
Accuracy	0.21% error (ICML 2013)	2.99% top-5 error (2016 winner)

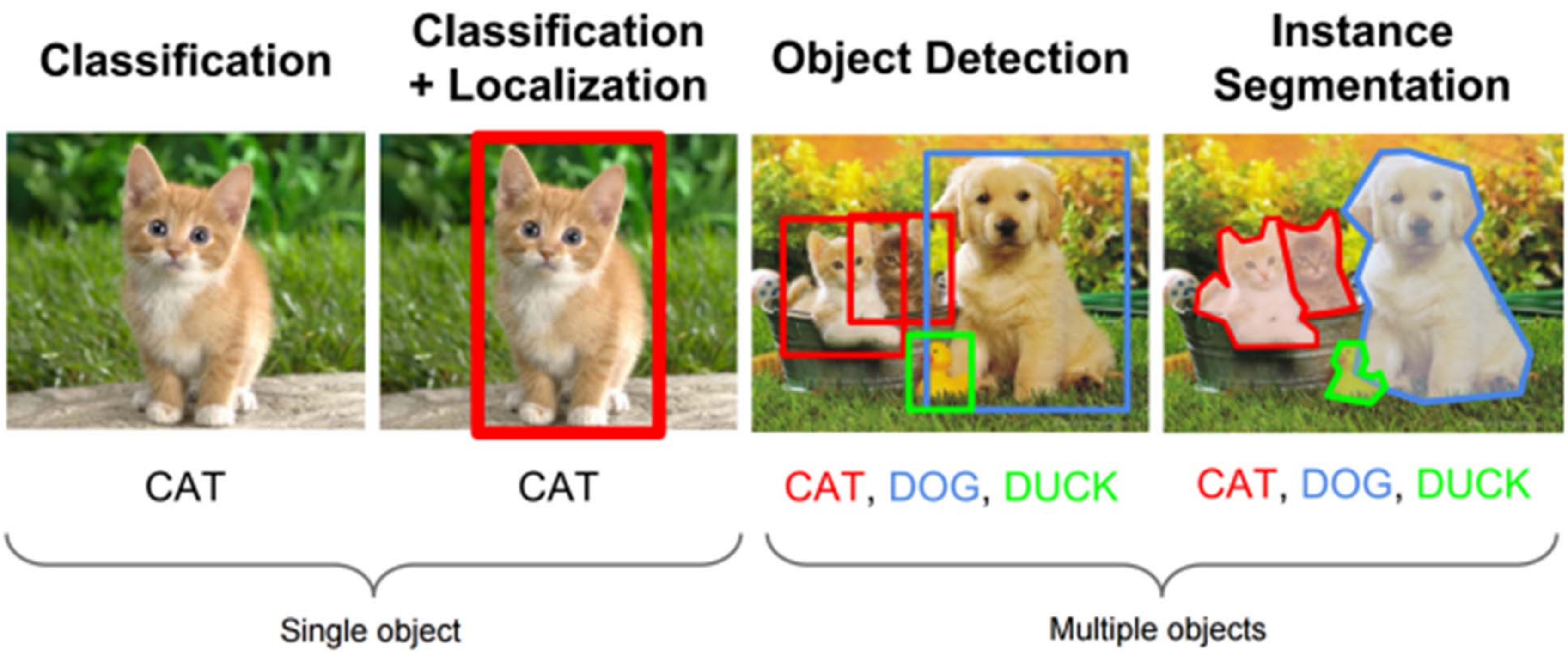
http://rodrigob.github.io/are_we_there_yet/build/classification_datasets_results.html

Next Tasks: Localization and Detection



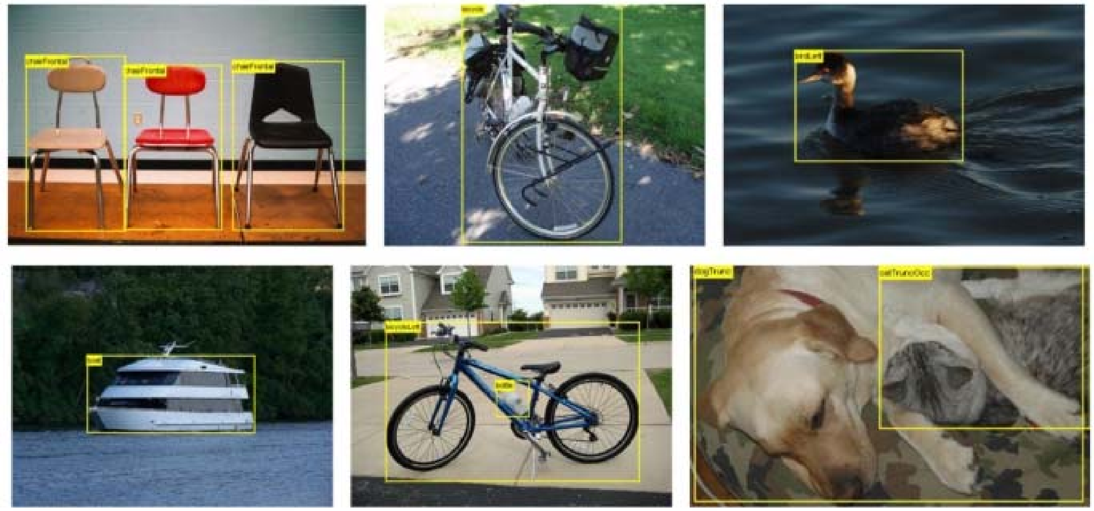
[Russakovsky et al., IJCV, 2015]

Object Localization and Detection



https://leonardoaraujosantos.gitbooks.io/artificial-inteligence/content/object_localization_and_detection.html

Others Popular Datasets



- **Pascal VOC**
  - **11k images**
  - **Object Detection**
  - **20 classes**
- **MS COCO**
  - **300k images**
  - **Detection, Segmentation**
  - **Recognition in context**

http://host.robots.ox.ac.uk/pascal/VOC/   http://mscoco.org/

Summary

- Popular DNN:
  - LeNet (1998), AlexNet (2012), VGGNet (2014), GoogleNet (2014), ResNet (2015)

- Development resources presented in this section enable us to evaluate hardware using the appropriate DNN model and dataset
  - Difficult tasks typically require larger models
  - Different datasets for different tasks
  - Number of datasets growing at a rapid pace

Practice Problem

- Explain popular DNNs
- **Explain Image Classification Datasets**

Backup Slides