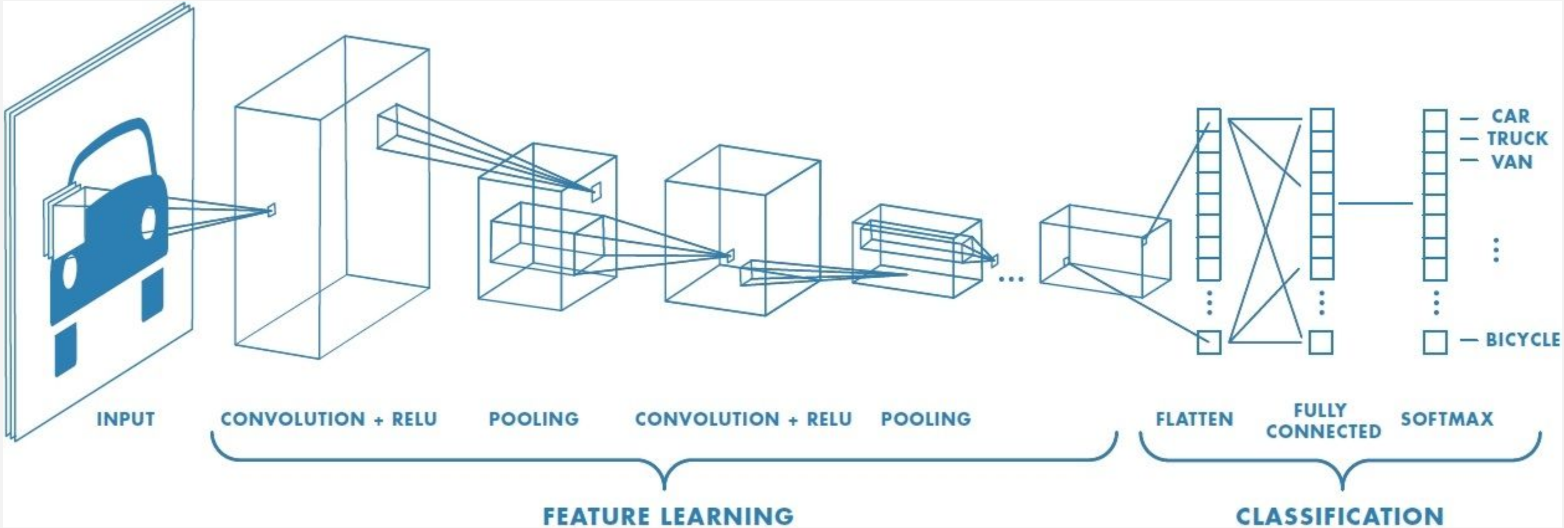
Cambrian Explosion:

Evolution of CNN Classifiers

Convolutional Neural Networks



Learn features that appear at different locations in image
Shared Weights for economy of learning

Hyperparameters

Activation function

Number of filters

Size of filters

Stride Size

Number of layers

Order of layers

Loss function

Learning rate

Pooling? Type?

How to choose?

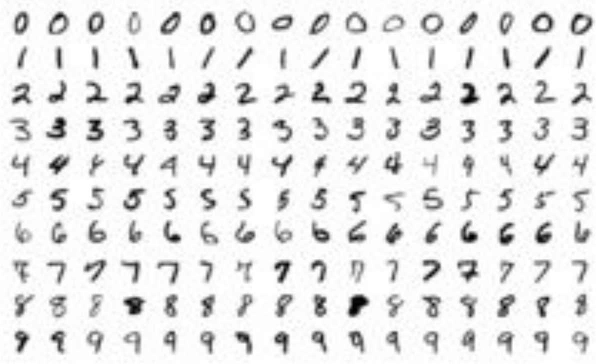
Metrics

- Validation accuracy
- Deployment accuracy in embedded applications (not always the same thing, as validation, see Teti et al, 2018)
- Speed of training
- Power consumption
- Adaptability to other learning tasks
- Interpretability

MNIST

- Created in ~1999
- 10 handwritten digits (0-9)
- 60,000 training examples
- 10,000 validation examples
- Derivatives:
  - Fashion Mnist - clothing
  - emnist - letters instead of numbers

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



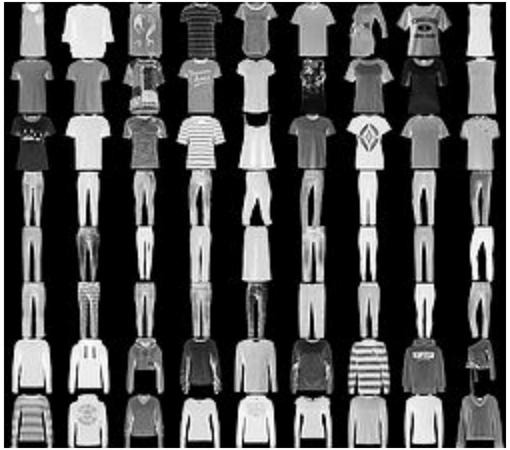
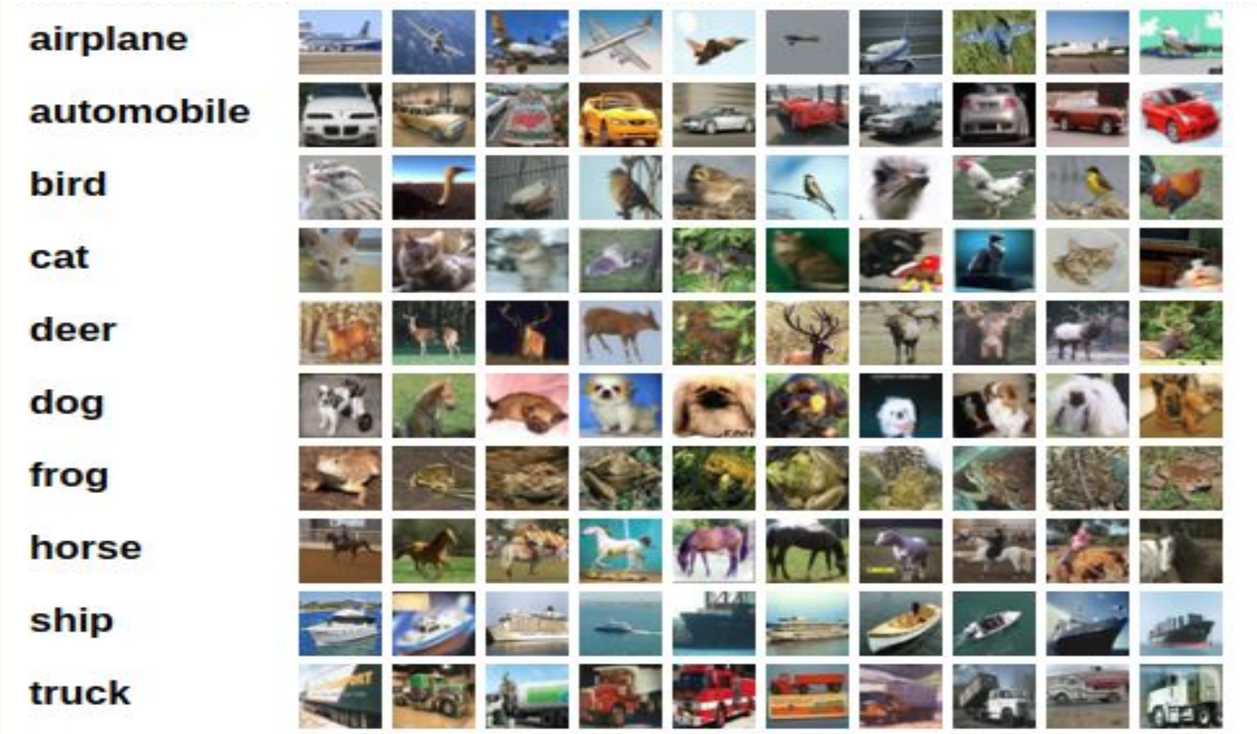


Image Classification Datasets - Cifar-10/100

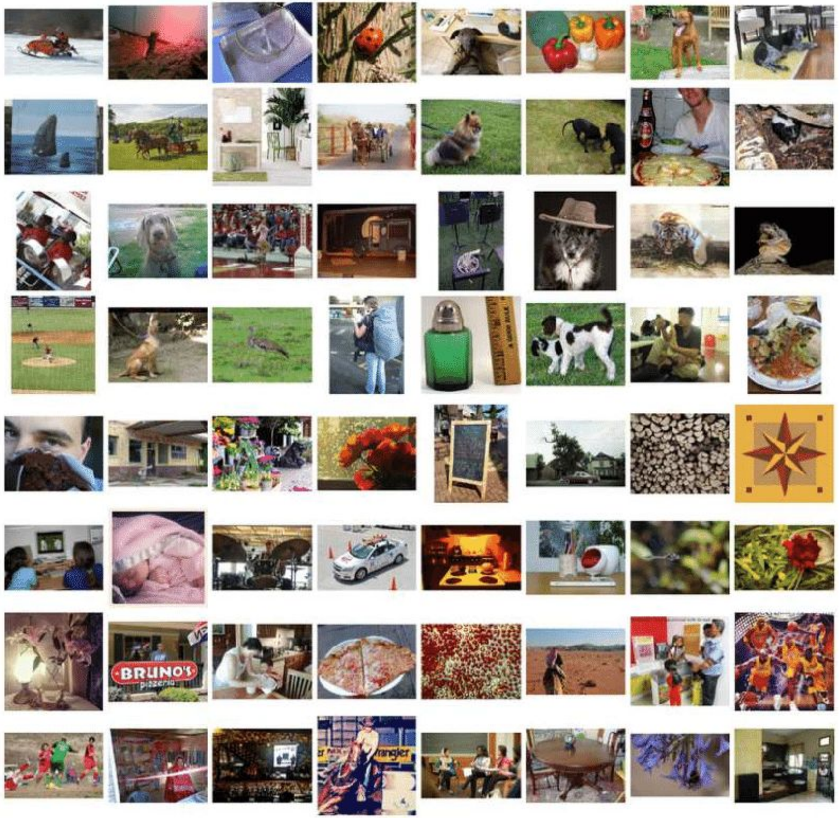
- 32x32 color images
- Cifar-10
  - 10 classes
  - 6000 images/class
  - 50000 training images
  - 10000 test images
- Cifar-100
  - 100 classes
  - 600 images/class
  - 50000 training images
  - 10000 test images

https://www.cs.toronto.edu/~kriz/cifar.html



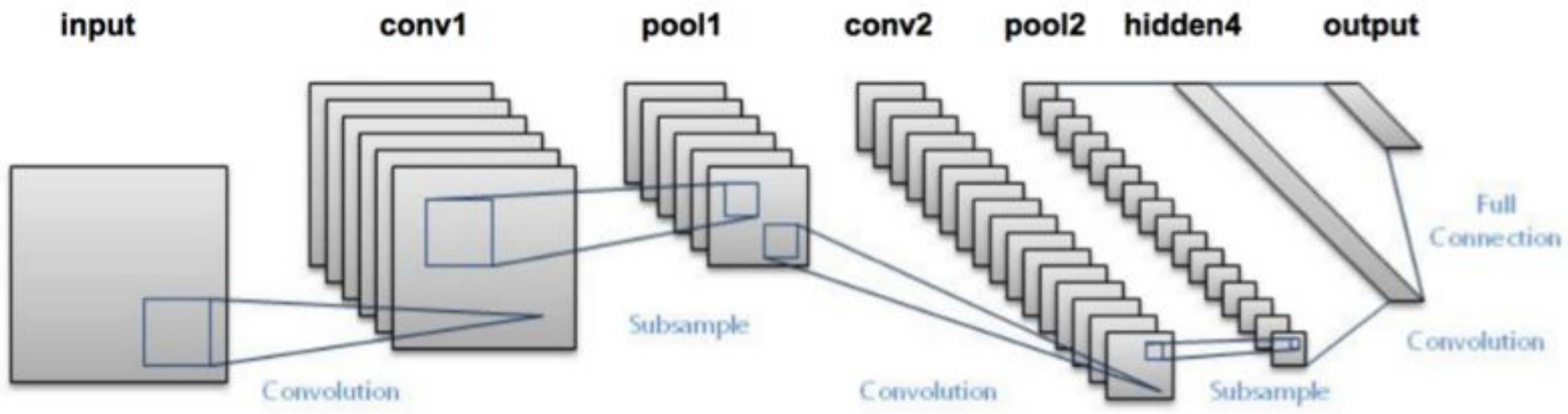
ImageNet

- Created in ~2010
- ImageNet Large Scale Visual Recognition Challenge (ILSVRC)
  - Image recognition competition held every year since 2010
- 1000 categories or classes
- ~ 13 million 224x224 color images
  - much larger than MNIST and Cifar images



LeNet-5 (Yann Lecun)

- First convolutional neural network (1998)
- 5 hidden layers - conv, pool, conv, pool, fc
- Trained on 32x32 grayscale images of handwritten digits
- Used in postal office, banks, etc.
- Main limitation was computing resources at that time



Convolutional Neural Networks

Convolutional Network Demo from 1993

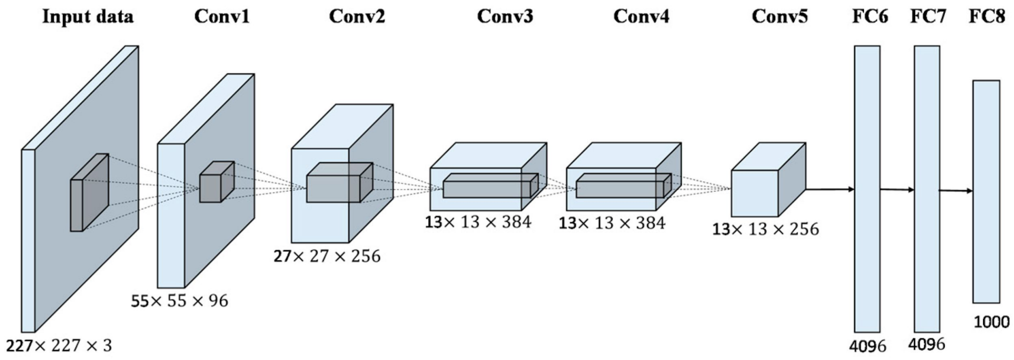
https://www.youtube.com/watch?v=FwFduRA_L6Q

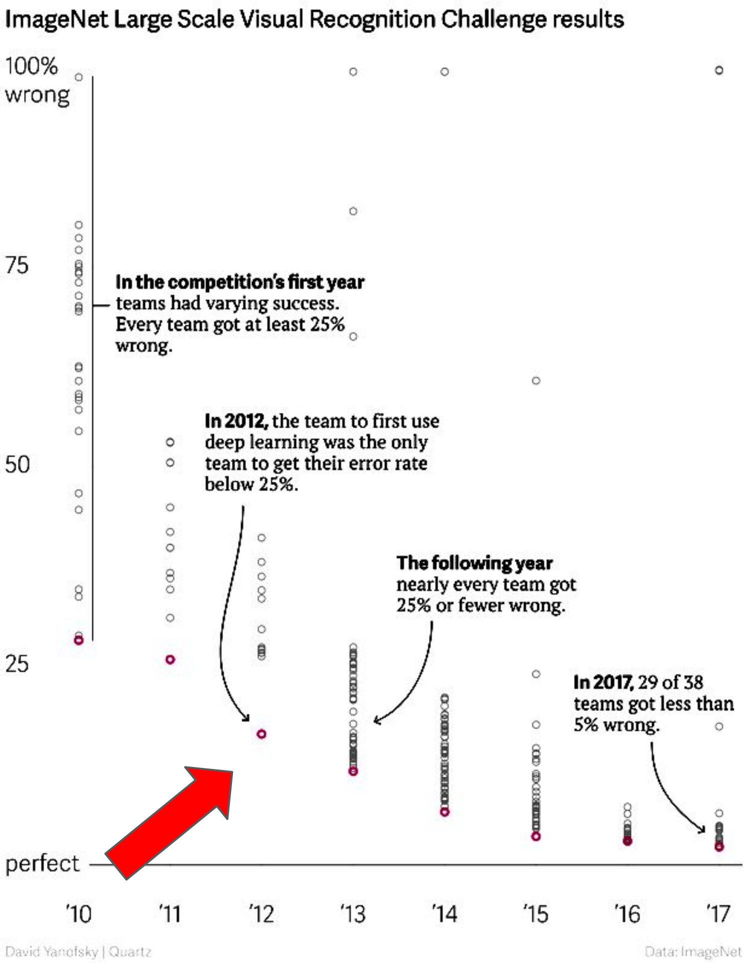
Semantic Segmentation with a Convolutional Network (33 categories)

https://www.youtube.com/watch?v=ZJMtDRbqH40

AlexNet

- First published in ~2012
  - cited over 25,000 times
- Received widespread attention from computer vision community
  - Did almost twice as good on ImageNet challenge as traditional computer vision methods
- ~62M weights!

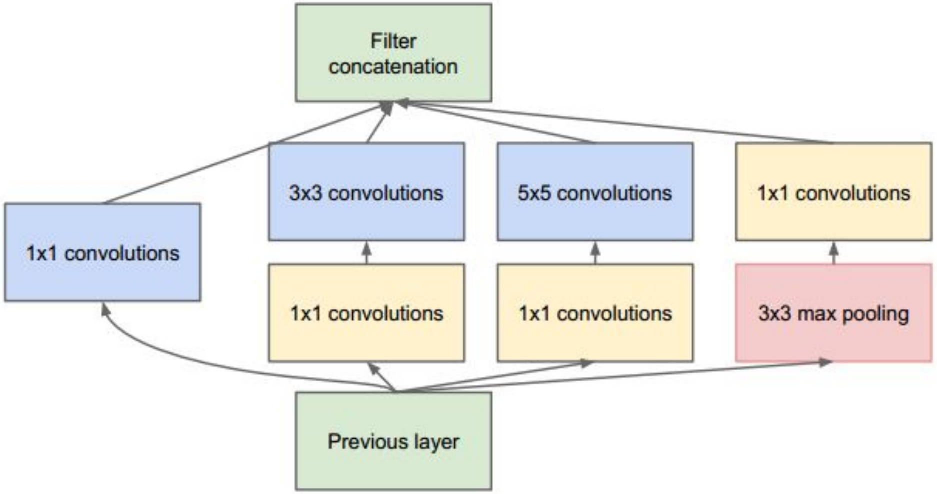




Inception (a.k.a. GoogleNet)

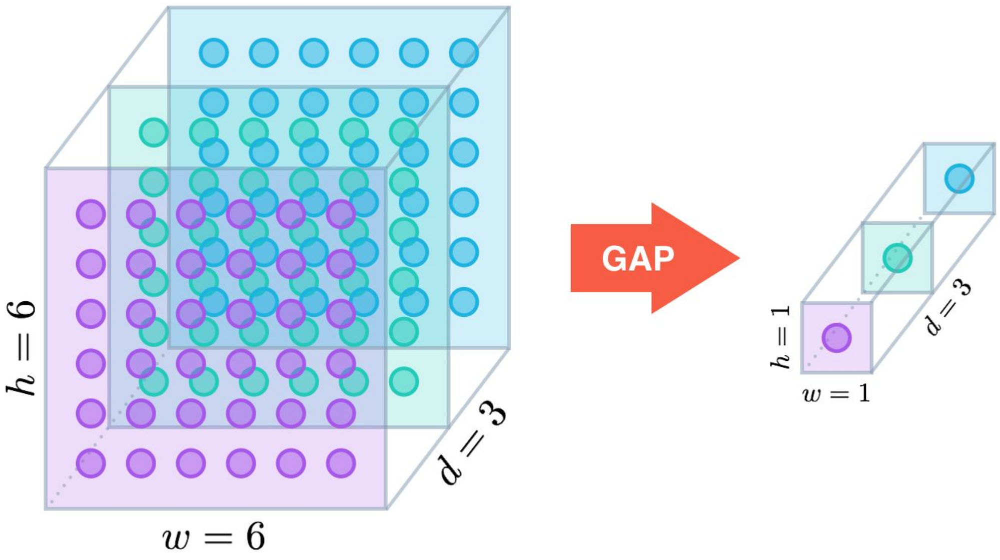
- Winner of 2014 ILSVRC 2014 challenge
- top-5 error rate of 6.67% (very close to human-level performance)
- Introduced Inception 'block' (right)
- Also global average pooling
  - Drastically reduced number of parameters (~5M)
- 22 layers - much deeper

https://www.cs.unc.edu/~wliu/papers/GoogLeNet.pdf



Global Average Pooling

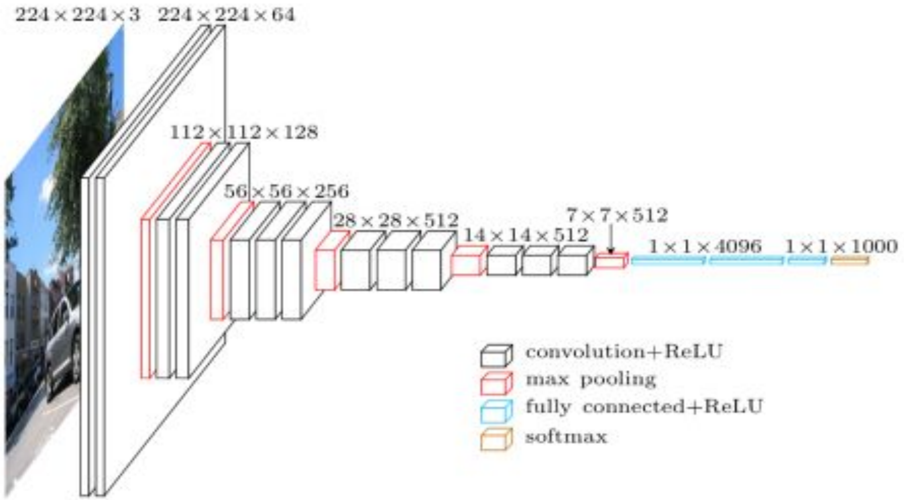
- Instead of flattening out last conv layer and having one or two fully-connected layers before output layer, take the average of each feature map and just have an output layer
- Saves on parameters



VGG

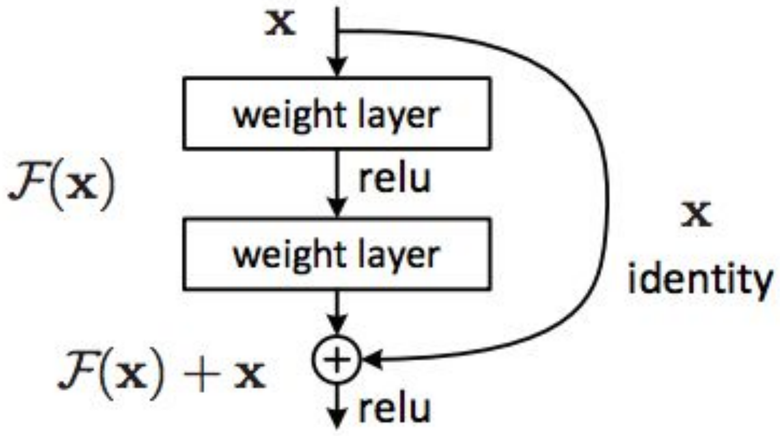
- ILSVRC 2014 runner-up
- Reduced number of hyper-parameters (which still requires choices by humans), but relatively
  - All convolutions have same size filters (3x3), which also reduces # of weights
  - Number of filters simply doubles at certain points
- 138M parameters!
- Generalizes well to other tasks

https://arxiv.org/pdf/1409.1556.pdf



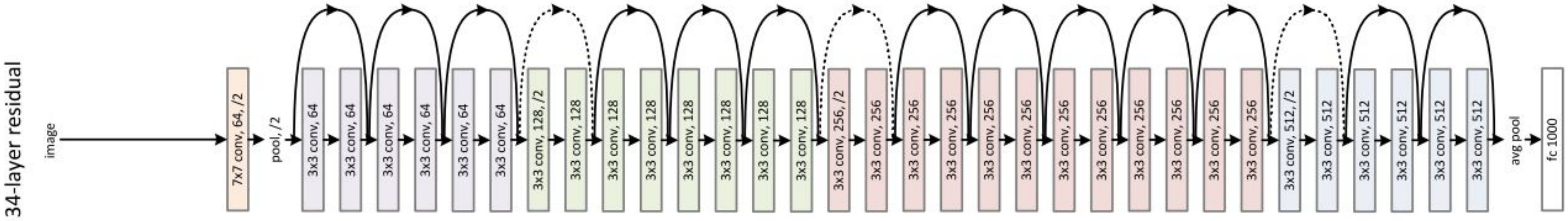
ResNet

- 1st place ILSVRC 2015 challenge - 3.57% error (better than humans)
- Built off the splitting/merging strategy of Inception nets as well as the simple block template of Inception and VGG
- Composed of ‘residual blocks’
  - increase representation power while not adding parameters
  - conv -> batch norm -> relu
- ResNet-34, ResNet-50, ResNet-101, …
- Despite being very deep, relatively few parameters (e.g. ~25M in ResNet-50).



ResNet Architecture

Skip connections denoted by arrows with lines going through them



Also uses global average pooling at the end. ResNet relatively efficient architecture

https://arxiv.org/pdf/1512.03385.pdf

ResNeXt

- 2nd place at ILSVRC 2016
- Builds off of residual blocks of ResNet
  - increased accuracy although number of parameters and complexity is the same
- Introduce a new dimension of conv layers - ‘cardinality’
  - number of different streams to take in a residual block (left)
- Recipe: divide the input into smaller inputs, do convolutions on those, add results together, then add that result to input

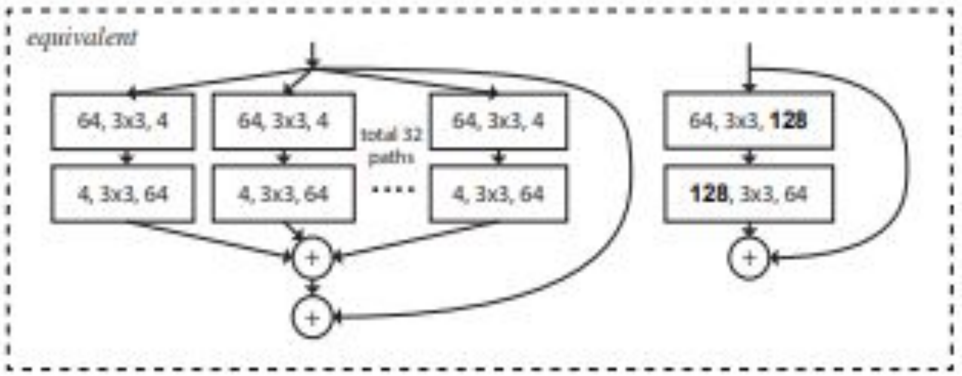


Figure 4. (**Left**): Aggregating transformations of depth = 2. (**Right**): An equivalent block, which is trivially wider.

https://arxiv.org/pdf/1611.05431.pdf

DenseNet

- Output feature maps of each layer are used as input to every subsequent layer
  - helps vanishing gradient problem
  - feature reuse/propagation
  - reduces number of weights
- Uses skip connections like ResNet, ResNeXt, etc.

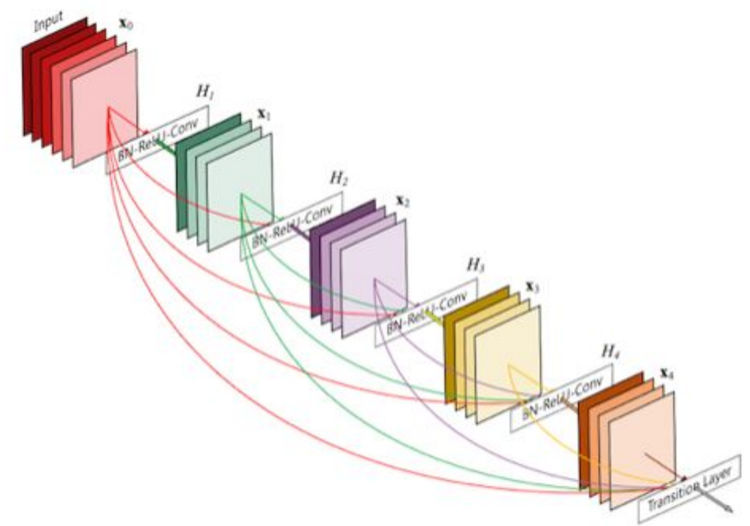
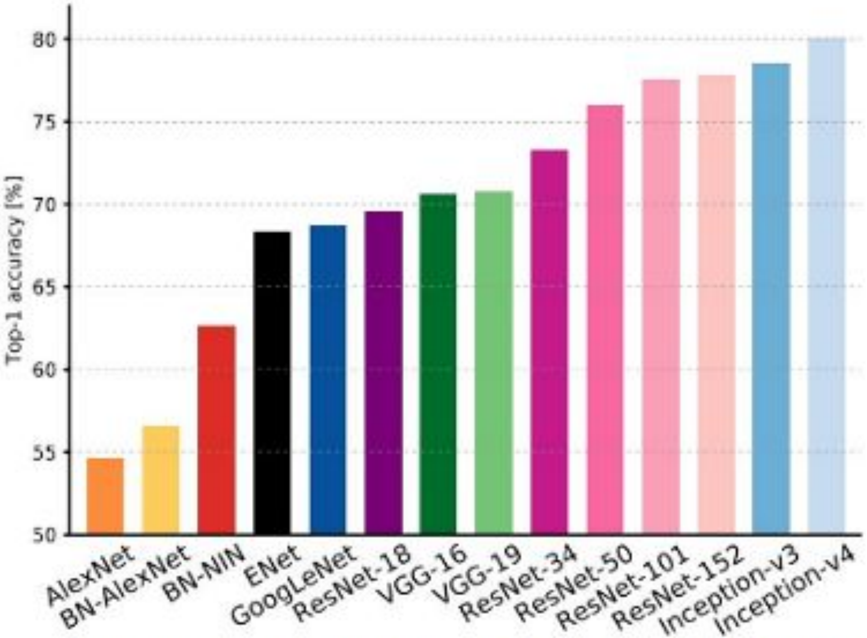
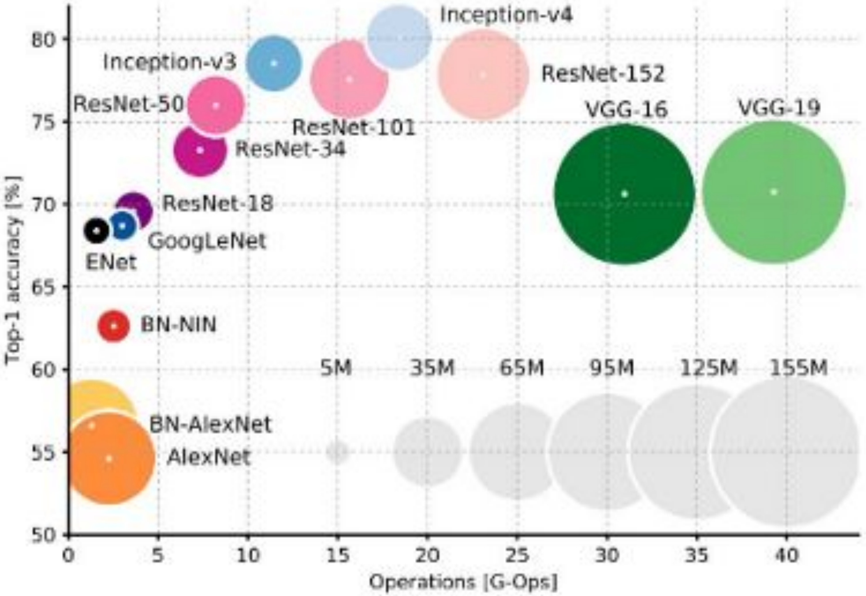


Figure 1. A 5-layer dense block with a growth rate of $k = 4$. Each layer takes all preceding feature-maps as input.

https://arxiv.org/pdf/1608.06993.pdf

Size of each circle illustrates number of parameters.

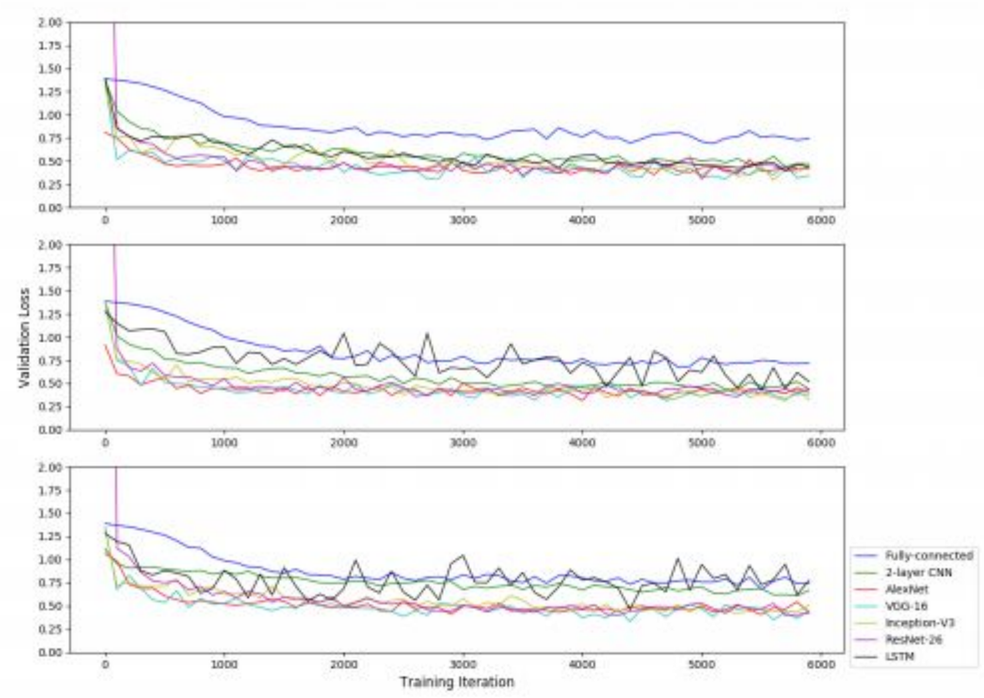




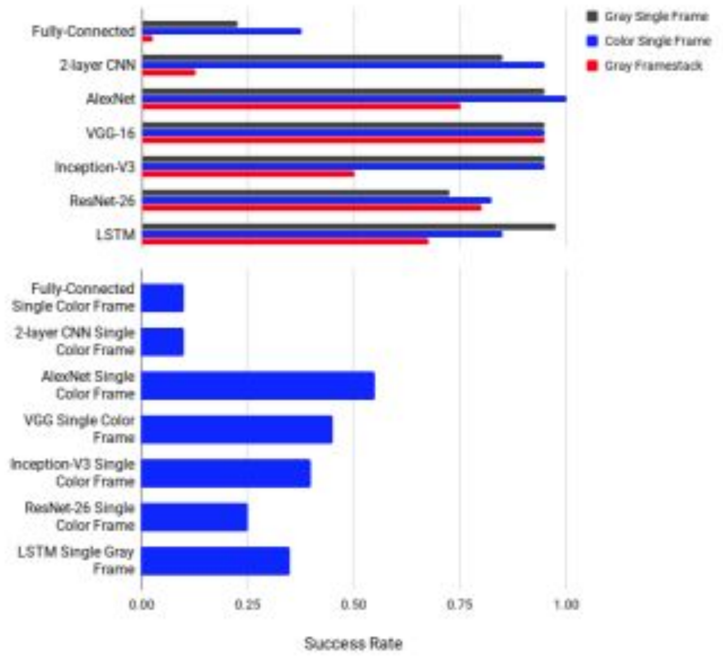
An Analysis of Deep Neural Network Models for Practical Applications, 2017.

http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture9.pdf

‘Deployment Gap’ in a self-driving task (Teti et al., 2018)

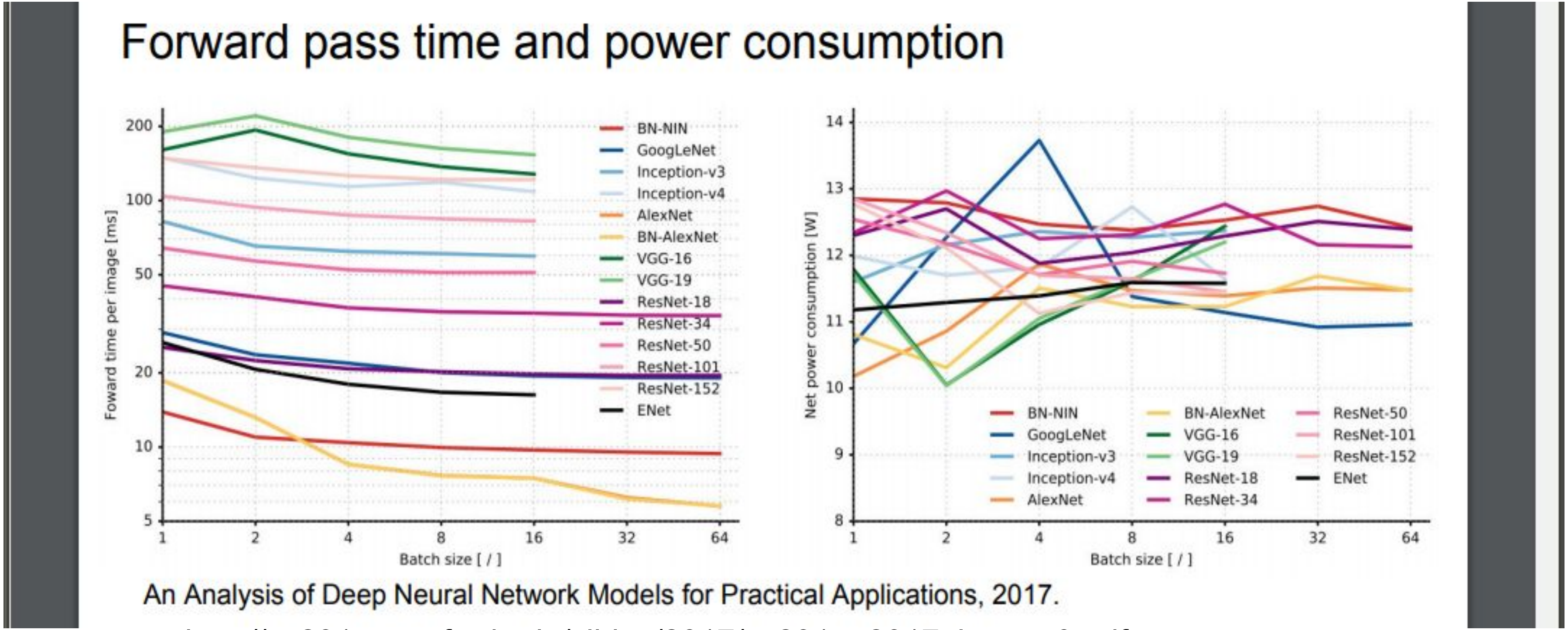


Validation loss across models



Self-driving performance across models

Recent Push Toward Mobile, Efficient Networks



http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture9.pdf

Many More…

- SqueezeNet
- MobileNet
- Xception
- NASNet
- ZFNet
- Network-in-Network
- Highway Networks
- Extreme Learning Machines
- FractalNet
- Yours?